Contact Information
Corporate Headquarters:
Palo Alto Networks
3000 Tannery Way
Santa Clara, CA 95054
www.paloaltonetworks.com/company/contact-support

About the Documentation
- For the most recent version of this guide or for access to related documentation, visit the Technical Documentation portal www.paloaltonetworks.com/documentation.
- To search for a specific topic, go to our search page www.paloaltonetworks.com/documentation/document-search.html.
- Have feedback or questions for us? Leave a comment on any page in the portal, or write to us at documentation@paloaltonetworks.com.

Copyright
Palo Alto Networks, Inc.
www.paloaltonetworks.com

© 2018-2020 Palo Alto Networks, Inc. Palo Alto Networks is a registered trademark of Palo Alto Networks. A list of our trademarks can be found at www.paloaltonetworks.com/company/trademarks.html. All other marks mentioned herein may be trademarks of their respective companies.

Last Revised
June 21, 2020
# Table of Contents

**Cortex XDR<sup>™</sup> Overview** ........................................................................................................... 7  
Cortex XDR Architecture .......................................................................................................................... 9  
Cortex XDR Concepts ............................................................................................................................. 11  
XDR ............................................................................................................................................... 11  
Sensors .............................................................................................................................................. 11  
Log Stitching .................................................................................................................................... 11  
Causality Analysis Engine ............................................................................................................. 12  
Causality Chain ............................................................................................................................... 12  
Causality Group Owner (CGO) ......................................................................................................... 12  
Cortex XDR Licenses .......................................................................................................................... 13  
Cortex XDR License Types .............................................................................................................. 13  
License Allocation ........................................................................................................................... 14  
License Expiration ............................................................................................................................ 15  
License Monitoring ........................................................................................................................... 15  
Migrate Your Cortex XDR License ................................................................................................. 16  

**Get Started with Cortex XDR Pro** ............................................................................................... 21  
Set up Cortex XDR Pro Overview ..................................................................................................... 23  
Plan Your Cortex XDR Deployment ................................................................................................. 25  
Manage Roles ..................................................................................................................................... 26  
  Predefined User Roles for Cortex XDR .......................................................................................... 28  
Activate your Network Devices ........................................................................................................ 34  
Activate Cortex XDR ....................................................................................................................... 36  
Set Up Directory Sync ...................................................................................................................... 39  
  Pairing Directory Sync ................................................................................................................ 39  
Allocate Log Storage for Cortex XDR ............................................................................................. 41  
Set up Endpoint Protection ............................................................................................................. 42  
  Plan Your Agent Deployment .................................................................................................... 43  
  Enable Access to Cortex XDR .................................................................................................. 44  
  Proxy Communication ............................................................................................................... 47  
Set up Network Analysis .................................................................................................................. 48  
  Ingest Logs and Alerts from External Sources ......................................................................... 49  
Set Up Pathfinder ............................................................................................................................... 56  
Configure XDR ................................................................................................................................... 60  
  Integrate External Threat Intelligence Services .................................................................... 61  
Set up Outbound Integration ............................................................................................................ 63  
Use the Cortex XDR Interface .......................................................................................................... 64  
  Manage Tables .......................................................................................................................... 65  

**Endpoint Security** ......................................................................................................................... 71  
Endpoint Security Concepts .............................................................................................................. 73  
  About Cortex XDR Endpoint Protection ............................................................................... 73  
File Analysis and Protection Flow .................................................................................................... 76  
Endpoint Protection Capabilities ...................................................................................................... 79  
Endpoint Protection Modules ........................................................................................................... 82  
Manage Cortex XDR Agents ............................................................................................................. 88  
  Create an Agent Installation Package ....................................................................................... 88  
  Set an Application Proxy for Cortex XDR Agents .................................................................. 89
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upgrade Cortex XDR Agents</td>
<td>90</td>
</tr>
<tr>
<td>Delete Cortex XDR Agents</td>
<td>92</td>
</tr>
<tr>
<td>Uninstall the Cortex XDR Agent</td>
<td>93</td>
</tr>
<tr>
<td>Define Endpoint Groups</td>
<td>94</td>
</tr>
<tr>
<td>About Content Updates</td>
<td>97</td>
</tr>
<tr>
<td>Endpoint Security Profiles</td>
<td>98</td>
</tr>
<tr>
<td>Add a New Malware Security Profile</td>
<td>99</td>
</tr>
<tr>
<td>Add a New Exploit Security Profile</td>
<td>103</td>
</tr>
<tr>
<td>Add a New Restrictions Security Profile</td>
<td>109</td>
</tr>
<tr>
<td>Manage Security Profiles</td>
<td>110</td>
</tr>
<tr>
<td>Customizable Agent Settings</td>
<td>113</td>
</tr>
<tr>
<td>Add a New Agent Settings Profile</td>
<td>116</td>
</tr>
<tr>
<td>Global Agent Configurations</td>
<td>120</td>
</tr>
<tr>
<td>Endpoint Data Collected by Cortex XDR</td>
<td>120</td>
</tr>
<tr>
<td>Apply Security Profiles to Endpoints</td>
<td>128</td>
</tr>
<tr>
<td>Exceptions Security Profiles</td>
<td>130</td>
</tr>
<tr>
<td>Add a New Exceptions Security Profile</td>
<td>131</td>
</tr>
<tr>
<td>Add a Global Endpoint Policy Exception</td>
<td>132</td>
</tr>
<tr>
<td>Hardened Endpoint Security</td>
<td>137</td>
</tr>
<tr>
<td>Device Control</td>
<td>138</td>
</tr>
<tr>
<td>Host Firewall</td>
<td>142</td>
</tr>
<tr>
<td>Disk Encryption Using BitLocker</td>
<td>145</td>
</tr>
<tr>
<td>Application Inventory</td>
<td>149</td>
</tr>
<tr>
<td>Vulnerability Assessment</td>
<td>151</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Investigation and Response</td>
<td>155</td>
</tr>
<tr>
<td>Cortex XDR Indicators</td>
<td>157</td>
</tr>
<tr>
<td>Working with BIOC's</td>
<td>157</td>
</tr>
<tr>
<td>Working with IOCs</td>
<td>162</td>
</tr>
<tr>
<td>Manage Existing Indicators</td>
<td>166</td>
</tr>
<tr>
<td>Search Queries</td>
<td>169</td>
</tr>
<tr>
<td>Cortex XDR Query Builder</td>
<td>169</td>
</tr>
<tr>
<td>Cortex XDR Query Center</td>
<td>190</td>
</tr>
<tr>
<td>Quick Launcher</td>
<td>194</td>
</tr>
<tr>
<td>Cortex XDR Scheduled Queries</td>
<td>194</td>
</tr>
<tr>
<td>Research a Known Threat</td>
<td>197</td>
</tr>
<tr>
<td>Investigate Incidents</td>
<td>198</td>
</tr>
<tr>
<td>External Integrations</td>
<td>200</td>
</tr>
<tr>
<td>Create an Incident Starring Configuration</td>
<td>202</td>
</tr>
<tr>
<td>Investigate Artifacts and Assets</td>
<td>204</td>
</tr>
<tr>
<td>Investigate an IP Address</td>
<td>204</td>
</tr>
<tr>
<td>Hash View</td>
<td>206</td>
</tr>
<tr>
<td>Investigate Alerts</td>
<td>210</td>
</tr>
<tr>
<td>Cortex XDR Alerts</td>
<td>210</td>
</tr>
<tr>
<td>Triage Alerts</td>
<td>216</td>
</tr>
<tr>
<td>Manage Alerts</td>
<td>216</td>
</tr>
<tr>
<td>Alert Exclusions</td>
<td>218</td>
</tr>
<tr>
<td>Causality View</td>
<td>221</td>
</tr>
<tr>
<td>Network Causality View</td>
<td>223</td>
</tr>
<tr>
<td>Timeline View</td>
<td>226</td>
</tr>
<tr>
<td>Analytics Alert View</td>
<td>227</td>
</tr>
<tr>
<td>Investigate Endpoints</td>
<td>230</td>
</tr>
<tr>
<td>Action Center</td>
<td>230</td>
</tr>
<tr>
<td>View Details About an Endpoint</td>
<td>233</td>
</tr>
</tbody>
</table>
# Table of Contents

## Broker VM

- Broker VM Overview................................................................. 269  
- Set up Broker VM............................................................................... 270  
  - Configure the Broker VM.......................................................... 270  
  - Activate the Agent Proxy........................................................... 279  
  - Activate the Syslog Collector...................................................... 279  
  - Activate the Windows Event Collector......................................... 280  
- Manage Your Broker VMs............................................................... 294  
  - View Broker VM Details............................................................. 294  
  - Edit Your Broker VM Configuration.......................................... 296  
  - Collect Broker VM Logs............................................................. 297  
  - Reboot a Broker VM................................................................. 297  
  - Upgrade a Broker VM............................................................... 297  
  - Remove a Broker VM............................................................... 298  
- Broker VM Notifications............................................................... 299

## Analytics

- Analytics Concepts............................................................................. 303  
- Analytics Engine............................................................................. 303  
- Coverage of the MITRE Attack Tactics........................................... 303  
- Analytics Detectors......................................................................... 305  
- Analytics Management....................................................................... 307  
  - Analytics Status........................................................................... 307  
  - Analytics Configuration Settings................................................ 314  
  - Analytics Management............................................................... 322

## Monitoring

- Cortex XDR Dashboard...................................................................... 327  
- Dashboard Widgets.......................................................................... 327  
- Predefined Dashboards.................................................................... 330  
- Build a Custom Dashboard.............................................................. 333  
- Manage Dashboards......................................................................... 335  
- Run or Schedule Reports............................................................... 335  
- Monitor Cortex XDR Incidents......................................................... 337  
- Monitor Administrative Activity....................................................... 339  
  - Forward Your Management Audit Log........................................ 340  
  - Monitor Agent Activity................................................................. 341
Log Forwarding............................................................................................................ 345
  Log Forwarding Data Types.......................................................................................... 347
  Integrate Slack for Outbound Notifications................................................................. 348
  Integrate a Syslog Receiver.......................................................................................... 350
  Configure Notification Forwarding.............................................................................. 352
  Cortex XDR Log Notification Formats............................................................................ 354
    Alert Notification Format......................................................................................... 354
    Agent Audit Log Notification Format....................................................................... 357
    Management Audit Log Notification Format............................................................. 358
    Cortex XDR Log Format for IOC and BIOC Alerts....................................................... 359
    Cortex XDR Analytics Log Format............................................................................. 367
    Cortex XDR Log Formats......................................................................................... 372

Managed Security...................................................................................................... 401
  About Managed Security.............................................................................................. 403
  Cortex XDR Managed Security Access Requirements.................................................. 404
  Set up Managed Threat Hunting.................................................................................. 405
  Pair a Parent Tenant with Child Tenant....................................................................... 406
    Pairing a Parent and Child Tenant............................................................................ 406
    Unpairing a Parent and Child Tenant....................................................................... 407
  Manage a Child Tenant............................................................................................... 408
    Track your Tenant Management............................................................................... 408
    View Child Tenant Data............................................................................................ 409
    Create and Allocate Configurations......................................................................... 410
    Create a Security Managed Action........................................................................... 411
Cortex XDR™ Overview

The Cortex XDR™ app offers you complete visibility over network traffic, user behavior, and endpoint activity. It simplifies threat investigation by correlating logs from your sensors to reveal threat causalities and timelines. This enables you to easily identify the root cause of every alert. The app also allows you to perform immediate response actions. Finally, to stop future attacks, you can pro-actively define IOCs and BIOC to detect and respond to malicious activity.

> Cortex XDR Architecture
> Cortex XDR Concepts
> Cortex XDR Licenses
Cortex XDR Architecture

Cortex XDR consumes data from the Cortex Data Lake and can correlate and stitch together logs across your different log sensors to derive event causality and timelines. A Cortex XDR deployment which uses the full set of sensors can include the following components:

- **Cortex XDR**—The Cortex XDR app provides complete visibility into all your data in the Cortex Data Lake. The app provides a single interface from which you can investigate and triage alerts, take remediation actions, and define policies to detect the malicious activity in the future.
- **Cortex Data Lake**—A cloud-based logging infrastructure that allows you to centralize the collection and storage of logs from your log data sources.
- **Cortex XDR Pro per TB**:
  - **Analytics engine**—The Cortex XDR analytics engine is a security service that utilizes network data to automatically detect and report on post-intrusion threats. The analytics engine does this by identifying good (normal) behavior on your network, so that it can notice bad (anomalous) behavior.
  - **Palo Alto Networks next-generation firewalls**—On-premise or virtual firewalls that enforce network security policies in your campus, branch offices, and cloud data centers.
  - **Palo Alto Networks Prisma Access and GlobalProtect**—If you extend your firewall security policy to mobile users and remote networks using Prisma Access or GlobalProtect, you can also forward related traffic logs to Cortex Data Lake. The analytics engine can then analyze those logs and raise alerts on anomalous behavior.
  - **External firewalls and alerts**—Cortex XDR can ingest traffic logs from external firewall vendors—such as Check Point—and use the analytics engine to analyze those logs and raise alerts on anomalous behavior. For additional context in your incidents, you can also send alerts from external alert sources.
- **Cortex XDR Pro per Endpoint**:
  - **Analytics engine**—The Cortex XDR analytics can also consume endpoint data to automatically detect and report on post-intrusion threats. The analytics engine can use endpoint data to raise alerts for abnormal network behavior (for example port scan activity).
  - **Cortex XDR agents**—Protects your endpoints from known and unknown malware and malicious behavior and techniques. Cortex XDR agents perform its own analysis locally on the endpoint but also consumes WildFire threat intelligence. The Cortex XDR agent reports all endpoint activity to the Cortex Data Lake for analysis by Cortex XDR apps.
• **External alert sources**—To add additional context to your incidents, you can send Cortex XDR alerts from external sources using the Cortex XDR API.
Cortex XDR Concepts

- XDR
- Sensors
- Log Stitching
- Causality Analysis Engine
- Causality Chain
- Causality Group Owner (CGO)
- Analytics Concepts

XDR

With Endpoint Detection and Response (EDR), enterprises rely on endpoint data as a means to trigger cybersecurity incidents. As cybercriminals and their tactics have become more sophisticated, the time to identify and contain breaches has only increased. XDR goes beyond the traditional EDR approach of using only endpoint data to identify and respond to threats by applying machine learning across all your enterprise, network, cloud, and endpoint data. This approach enables you to quickly find and stop targeted attacks and insider abuse and remediate compromised endpoints.

Sensors

Cortex XDR™ uses your existing Palo Alto Networks products as sensors to collect logs and telemetry data. The sensors that are available to you depend on your Cortex XDR license type.

With a Cortex XDR Pro per TB license, a sensor can be any of the following:
- Virtual (VM-Series) or physical firewalls—Identifies known threats in your network and cloud data center environments
- Prisma Access or GlobalProtect—Identifies known threats in your mobile user and remote network traffic
- External firewall vendors—You can forward traffic logs from any external vendor for analysis by the Cortex XDR analytics engine

With a Cortex XDR Pro per Endpoint license, a sensor can be any of the following:
- Cortex XDR agents—Identifies threats on your Windows, Mac, Linux, and Android endpoints and halts any malicious behavior or files

While more sensors increases the amount of data Cortex XDR can analyze, you only need to deploy one type of sensor to begin detecting and stopping threats with Cortex XDR.

Log Stitching

To provide a complete and comprehensive picture of the events and activity surrounding an event, Cortex XDR™ correlates together firewall network logs, endpoint raw data, and cloud data across your detection sensors. The act of correlating logs from different sources is referred to as log stitching and helps you identify the source and destination of security processes and connections made over the network.

Log stitching allows you to:
- Run investigation queries based on stitched network and endpoint logs
- Create granular BIOC rules over logs from Palo Alto Networks Next-Generation Firewalls and raw endpoint data
- Investigate correlated network and endpoint events in the Network Causality View
Log stitching streamlines detection and reduces response time by eliminating the need for manual analysis across different data sensors. Stitching data across the firewalls and endpoints allows you to obtain data from different sensors in a unified view, each sensor adding another layer of visibility. For example, when a connection is seen through the firewall and the endpoint, the endpoint can provide information on the processes involved and on the chain of execution while the firewall can provide information on the amount of data transferred over the connection and the different app ids involved.

Causality Analysis Engine

The Causality Analysis Engine correlates activity from all detection sensors to establish causality chains that identify the root cause of every alert. The Causality Analysis Engine also identifies a complete forensic timeline of events that helps you to determine the scope and damage of an attack, and provide immediate response. The Causality Analysis Engine determines the most relevant artifacts in each alert and aggregates all alerts related to an event into an incident.

Causality Chain

When a malicious file, behavior, or technique is detected, Cortex XDR™ correlates available data across your detection sensors to display the sequence of activity that led to the alert. This sequence of events is called the causality chain. The causality chain is built from processes, events, insights, and alerts associated with the activity. During alert investigation you should review the entire causality chain to fully understand why the alert occurred.

Causality Group Owner (CGO)

The Causality Group Owner (CGO) is the process in the causality chain that the Causality Analysis Engine identified as being responsible for or causing the activities that led to the alert.
Cortex XDR Licenses

- Cortex XDR License Types
- License Allocation
- License Expiration
- License Monitoring
- Migrate Your Cortex XDR License

Cortex XDR License Types

The following table describes the different license types that are available for Cortex XDR. You can use either Cortex XDR Prevent or a Cortex XDR Pro license. You can also use both Pro licenses for the most coverage. If you do not know which license type you have, see License Monitoring.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Cortex XDR Prevent</th>
<th>Cortex XDR Pro per Endpoint</th>
<th>Cortex XDR Pro per TB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log storage</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Minimum of 200 endpoints</td>
<td>• Minimum of 200 endpoints</td>
<td>Minimum 5TB log storage</td>
</tr>
<tr>
<td></td>
<td>• 30 day log retention</td>
<td>• 30 day log retention</td>
<td></td>
</tr>
</tbody>
</table>

**Endpoint Prevention Features**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Cortex XDR Prevent</th>
<th>Cortex XDR Pro per Endpoint</th>
<th>Cortex XDR Pro per TB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Endpoint management</td>
<td>✓</td>
<td>✓</td>
<td>—</td>
</tr>
<tr>
<td>Device control</td>
<td>✓</td>
<td>✓</td>
<td>—</td>
</tr>
<tr>
<td>Host firewall</td>
<td>✓</td>
<td>✓</td>
<td>—</td>
</tr>
<tr>
<td>Disk encryption</td>
<td>✓</td>
<td>✓</td>
<td>—</td>
</tr>
<tr>
<td>Vulnerability assessment and application inventory</td>
<td>—</td>
<td>✓</td>
<td>—</td>
</tr>
</tbody>
</table>

**Response Actions**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Cortex XDR Prevent</th>
<th>Cortex XDR Pro per Endpoint</th>
<th>Cortex XDR Pro per TB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Live Terminal</td>
<td>✓</td>
<td>✓</td>
<td>—</td>
</tr>
<tr>
<td>Endpoint isolation</td>
<td>✓</td>
<td>✓</td>
<td>—</td>
</tr>
<tr>
<td>Feature</td>
<td>Cortex XDR Prevent</td>
<td>Cortex XDR Pro per Endpoint</td>
<td>Cortex XDR Pro per TB</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>--------------------</td>
<td>-----------------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>Script execution</td>
<td>—</td>
<td>✓</td>
<td>—</td>
</tr>
<tr>
<td>External dynamic list (EDL)</td>
<td>—</td>
<td>—</td>
<td>✓</td>
</tr>
<tr>
<td>Analysis</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analytics</td>
<td>—</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Alert and Log Ingestion</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cortex XDR agent alerts</td>
<td>✓</td>
<td>✓</td>
<td>—</td>
</tr>
<tr>
<td>EDR data</td>
<td>—</td>
<td>✓</td>
<td>—</td>
</tr>
<tr>
<td>Other alerts (from Palo Alto Networks and third-party sources)</td>
<td>—</td>
<td>✓ (API)</td>
<td>✓</td>
</tr>
<tr>
<td>Other logs (from Palo Alto Networks and third-party sources)</td>
<td>—</td>
<td>—</td>
<td>✓</td>
</tr>
<tr>
<td>Integrations</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Threat intelligence (AutoFocus, VirusTotal)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Outbound integration and notification forwarding (Slack, Syslog)</td>
<td>✓ + agent audit logs</td>
<td>✓ + agent audit logs</td>
<td>✓</td>
</tr>
<tr>
<td>MSSP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSSP (requires additional MSSP license)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Managed Threat Hunting (requires an additional Managed Threat Hunting License)</td>
<td>—</td>
<td>✓ + a minimum of 500 endpoints</td>
<td>—</td>
</tr>
</tbody>
</table>

License Allocation

With Cortex XDR Prevent and Cortex XDR Pro per Endpoint licenses, Cortex XDR manages licensing for all endpoints in your organization. Each time you install a new Cortex XDR agent on an endpoint, the Cortex
XDR agent registers with Cortex XDR to obtain a license. In the case of non-persistent VDI, the Cortex XDR agent registers with Cortex XDR as soon as the user logs in to the endpoint.

Cortex XDR issues licenses until you exhaust the number of license seats available. Cortex XDR also enforces a license cleanup policy to automatically return unused licenses to the pool of available licenses. The time at which a license returns to the license pool depends on the type of endpoint:

<table>
<thead>
<tr>
<th>Endpoint Type</th>
<th>License Return</th>
<th>Agent Removal from Cortex XDR console</th>
<th>Agent Removal from Cortex XDR Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard and mobile devices</td>
<td>After 30 days</td>
<td>After 180 days</td>
<td>After 180 days</td>
</tr>
<tr>
<td><em>(Non-Persistent)</em> VDI and Temporary Session</td>
<td>Immediately after log-off for VDI, otherwise after 90 minutes</td>
<td>After 6 hours</td>
<td>After 7 days</td>
</tr>
</tbody>
</table>

The agent cannot be restored after this period of time.

If after a license is revoked the agent connects to Cortex XDR, reconnection will succeed as long as the agent has not been deleted.

*It can take up to an hour for Cortex XDR to display revived endpoints.*

Cortex XDR will notify you if you exceed the number of available licenses.

### License Expiration

After your Cortex XDR license expires, Cortex XDR allows access to your tenant for an additional grace period of 48 hours. After the 48-hour grace period, Cortex XDR disables access to the Cortex XDR app until you renew the license.

For the first 30 days of your expired license, Cortex XDR continues to protect your endpoints and/or network and retains data in the Cortex Data Lake according to your Cortex Data Lake data retention policy and licensing. After 30 days the tenant is decommissioned.

### License Monitoring

From the > Cortex XDR License dialog, you can view the license type associated with your Cortex XDR instance.
For each license you have, Cortex XDR displays a tile that has the expiration date of your license and additional details specific to your license type:

- **Cortex XDR Prevent**—Displays the total number of concurrent agents permitted by your license. You can also view a graph of the current license allocation (total and percentage).
- **Cortex XDR Pro per Endpoint**—Displays the total number of installed in addition to the number and percentage of agents that have EDR data collection enabled. Below the license card, you can also view the storage retention policy, total amount of storage allocated for endpoint XDR, and the actual data usage.
- **Cortex XDR Pro per TB**—Displays the amount of total storage included with your license and the amount of storage used.
- **Combination of Cortex XDR Pro per Endpoint and Cortex XDR Pro per TB**—Cortex XDR Pro per Endpoint displays the total number of installed agents, while Cortex XDR Pro per TB displays how many agents are enabled with endpoint data collection, allowing them to collect and send data to the server. Cortex XDR will send you a notification if you exceed the number of allowed agents.

To keep you informed of updates made to your license and avoid service disruptions, Cortex XDR now displays a notification of changes made to your license when you log in. If any actions are required from you,

**Migrate Your Cortex XDR License**

As part of the migration of Cortex XDR 1.0 to Cortex XDR 2.0, a new Cortex XDR licensing structure will go into effect. The new licensing structure allows you to better view and manage how your network data and endpoints are best utilized across your organization.

Cortex XDR 1.0 license was based on the amount of terabyte (TB) used for either:

- 1TB = 200 Pro per Endpoints (with EDR Collection)
  
  Or
  
  - 1TB = 1TB of network traffic analysis/third party data + 200 Prevent Endpoints (without EDR collection)

Cortex XDR 2.0 license structure is based on three **license types** you can purchase individually or as a combination. The endpoint licenses provide the number of endpoints, either Prevent or Pro, and the TB license the amount of TB used for network traffic analysis and collecting third-party data:

- **Cortex XDR Prevent License**—Number of Prevent Endpoints (without EDR collection)
- **Cortex XDR Pro per Endpoint License**—Number of Pro Endpoints (with EDR collection)
- **Cortex XDR Pro per TB License**—Amount of network data used for network traffic analysis and third-party data.

**License Conversion Method**
Converting Cortex XDR 1.0 license to Cortex XDR 2.0 license is calculated as follows:

- **Endpoints**
  - For each Cortex 1.0 license, 1 TB = 200 Pro per Endpoints (with EDR collection).
  
  The number of endpoints is converted based on the quota allocated in Hub > Cortex Data Lake > Cortex XDR > Endpoint XDR Data, previously Traps > Endpoint Data.

- **Network Data**
  - For each Cortex XDR 1.0 license, 1 TB = 1 TB of network data.

  *Since XDR 2.0 pro per TB license no longer includes Prevent endpoints, the license does not reflect them, however, you can keep using them until your renewal.*

After migration of Cortex 2.0, when navigating to > Cortex XDR License, the license displays the converted amounts of network data or its equivalent number of endpoints allocated to your license.

**Conversion Example**

**Cortex XDR 1.0 License**

- Cortex XDR 1.0 PAN-MGFR-XDR-1TB license - 100TB
- Hub > Cortex Data Lake > Traps > Endpoint Data - 10TB Endpoint Data.

**Post Migration Cortex XDR 2.0 License**

- Up to 20,000 Pro per Endpoints
- Up to 100TB for network traffic analysis and third-party data

**Converting Your License**

When your Cortex XDR app is migrated to Cortex XDR 2.0, we recommend you convert your Cortex XDR license to align with the new structure. To apply the new license structure, determine how the amount of network data and number of endpoints are distributed across your organization.

*After you convert your legacy license to Cortex XDR 2.0 license structure, your new network and endpoint allocation are applied immediately. You can edit the allocation at any time, however, after you convert to the new license structure you cannot revert to your legacy license.*

**STEP 1**

In Cortex XDR app, select > Cortex XDR License.
• (1) Network quota in TB and qualifying number of Pro per Endpoints
• (2,3) Number of agents installed and enabled to collect EDR data in your organization based on the quota allocated in Hub > Cortex Data Lake > Cortex XDR > Endpoint XDR Data, previously Traps > Endpoint Data.
• (4) Current number of days Cortex XDR retains your data.

**STEP 2** | Convert your Cortex XDR 1.0 license to Cortex XDR 2.0 license.

1. Select Convert License.
2. Use the Network Allocation slide bar to allocate your license between network and endpoints (1 network TB = 200 endpoints).

   If you allocate all of your license to network data then you disable endpoint capabilities (and vice versa).

3. Apply your new license allocations.

**STEP 3** | In your new Cortex XDR 2.0 license, review or Edit your license allocation:

- Number of Cortex XDR agents
- Amount of network TB
- Number of installed endpoints and endpoints enabled with EDR Data collection according to the number of agents allocated to your license, rather than the Cortex Data Lake distribution.
- Number of days remaining for Cortex XDR to retain your data.
Advanced endpoint protection, data collection and response

10,000 AGENTS

Active
Expires on Nov 24, 2020

Advanced network detection and response

50 TB

Active
Expires on Nov 24, 2020

Based on your license conversion. Last updated on Feb 19, 2020, 2:00:38 PM

Installed Agents

0% [1/11000]

Agents with Data Collection Enabled

0% [0/10000]

ENDPOINT XDR DATA RETENTION

109 days (Based on 99% of the data)
Get Started with Cortex XDR Pro

> Set up Cortex XDR Pro Overview
Set up Cortex XDR Pro Overview

Before you can use Cortex XDR for advanced detection and response, you must activate the Cortex XDR app and set up related apps and services. You must perform the setup activities as shown in the following image. Some steps are required only if you have the corresponding license type.

**STEP 1 | Plan Your Cortex XDR Deployment.**

As part of your planning, ensure that you or the person who is activating Cortex apps has the appropriate roles.

**STEP 2 | (Cortex XDR Pro per TB license only) Activate your Network Devices.**

**STEP 3 | Activate Cortex XDR and related apps and services.**

1. Locate the email that contains your activation information.
2. Activate Cortex XDR.
3. Activate Cortex Data Lake (if not using an existing instance).
4. *(Optional)* Create a Directory Sync Service instance
5. Review log storage.

**STEP 4 | (Cortex XDR Pro per Endpoint only) Set up Endpoint Protection.**

1. Plan your Cortex XDR agent deployment.
2. Create Cortex XDR agent installation packages
3. Define endpoint groups.
4. Deploy the Cortex XDR agent to your endpoints.
5. Configure your endpoint security policy.

**STEP 5 | (Cortex XDR Pro per TB license only) Set up Network Analysis.**

1. Perform any remaining setup of your network sensors.
2. Configure the internal networks that you want Cortex XDR to monitor.
3. Verify that Cortex XDR is receiving alerts.
4. Set up Pathfinder.
5. If you set up a Directory Sync Service instance, enable Cortex XDR to use it.

**STEP 6 | Configure XDR.**

1. *(Optional)* Integrate additional threat intelligence.
2. After 24 hours, enable Cortex XDR Analytics Analysis.
   1. Configure Network Coverage.
2. *(Recommended)* Set up Pathfinder to interrogate endpoints that do not have EDR or that do not have the Cortex XDR agent installed.
3. Define alert exclusions.
4. Prioritize incidents based on attributes by creating an incident starring policy.
5. Import or configure rules for known BIOC and IOCs.

**STEP 7 | (Optional) Set up Outbound Integration.**

- Integrate with Slack
- Integrate with a Syslog Server
- Integrate with Cortex XSOAR

**STEP 8 | (Optional) Set up Managed Security**

**STEP 9 | Use the Cortex XDR Interface.**
Plan Your Cortex XDR Deployment

Before you get started with Cortex XDR™, plan your deployment:

<table>
<thead>
<tr>
<th>Deployment Type</th>
<th>Deployment Considerations</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Cortex XDR tenants</td>
<td>□ Use the <a href="#">Cortex Data Lake Calculator</a> to determine the amount of log storage you need for your Cortex XDR deployment. Talk to your Partner or Sales Representative to determine whether you must purchase additional Cortex Data Lake storage.</td>
</tr>
<tr>
<td></td>
<td>□ Determine the region in which you want to host Cortex XDR and any associated services, such as Cortex Data Lake and Directory Sync Service:</td>
</tr>
<tr>
<td></td>
<td>• US—All Cortex XDR logs and data remain within the US boundary.</td>
</tr>
<tr>
<td></td>
<td>• UK—All Cortex XDR logs and data are stored within the UK boundary. However, Cortex XDR will continue to send files that require analysis by WildFire to the WildFire cloud for EU. If your compliance and privacy laws prohibit the sending of files outside the UK region, disable uploading of files to WildFire in your Malware Security Profiles.</td>
</tr>
<tr>
<td></td>
<td>• EU—All Cortex XDR logs and data remain within the Europe boundary.</td>
</tr>
<tr>
<td></td>
<td>□ (<a href="#">Cortex XDR Pro per Endpoint license only</a>) Calculate the bandwidth required to support the number of agents you plan to deploy. You need 1.2Mbps of bandwidth for every 1,000 agents. The bandwidth requirement scales linearly so, for example, to support 100,000 agents, you need to allocate 120Mbps of bandwidth.</td>
</tr>
<tr>
<td></td>
<td>□ Manage Roles to ensure you or the person who is activating Cortex apps has the appropriate permissions.</td>
</tr>
</tbody>
</table>

When you are ready to get started with a new tenant, [Activate Cortex XDR](#).
Manage Roles

Role-based access control (RBAC) enables you to use roles or specific permissions to assign access rights to administrative users. You can manage roles for all Cortex apps and services in the hub. By assigning roles, you enforce the separation of viewing access and initiating actions among functional or regional areas of your organization. The following options are available to help you manage access rights:

- Assign Predefined User Roles for Cortex XDR
- Create and save new roles based on the granular permission
- Edit role permissions (available for roles you create)
- Assign permissions to users without saving a role

Use roles to assign specific view and action access privileges to administrative user accounts. The way you configure administrative access depends on the security requirements of your organization. The built-in roles provide specific access rights that cannot be changed. The roles you create provide more granular access control.

When your organization purchases Cortex XDR, the Account Administrator can use the Palo Alto Networks hub to assign roles to other members that have accounts in the Customer Support Portal.

To activate Cortex XDR apps, you must be assigned either the Account Administrator or App Administrator role for Cortex XDR. If you are activating a new Cortex Data Lake instance you must also be assigned either administrative role for Cortex Data Lake.

After activation, Account Administrators can assign additional users roles to manage your apps. If the user only needs to manage a specific instance of an app, you can assign the Instance Administrator role.

To assign the roles, Account Administrators (or users that are assigned the App Administrator for the relevant app) can take the following steps:

**STEP 1 | If necessary, add a new Customer Support Portal user.**

To be eligible for role assignment in the hub, the user must have an account in the Customer Support Portal (https://support.paloaltonetworks.com/) and be assigned any of the following Customer Support Portal roles: Super User, Standard User, or Limited User. Skip this step if the user already has a Customer Support Portal account with an appropriate role.

**STEP 2 | Manage the level of access for a Cortex XDR user.**

1. Log in to the hub and select > Access Management.
2. Use the sidebar to filter users as needed or the search field to search for users.
3. Select one or more users and then Assign Roles.
4. In the Assign Roles page for each instance, select one of the following options:

- **Assign Permissions**—Create a new role or assign selected permissions.
- **Cortex XDR Predefined Role**—Select one of the predefined Cortex XDR role. Select Role Definitions to view a list of the Cortex predefined roles and the allocated views and actions.
- **No Role**—User is not assigned any view or action access to the Cortex XDR app.

5. *(Optional)* To create a new role:

   1. After you selected **Assign Permissions**, in the Assign Custom Permissions pop-up, select which **IN_APP VIEWS** and **IN_APP ACTIONS** permissions you want to grant.
   2. **Save As New Role** to create a new role that you can apply to other users, or **Save** to apply the selected permissions to the user without a defined role.

The new rule is displayed with User Created (UC) icon. Select the role to apply permissions to the user and then **Save**.
6. (Optional) To edit or clone a user created role:

2. In the Manage Roles Cortex XDR page, find your user created role and select Actions.
3. Edit Permissions, Clone, or Delete your role, as desired.

Predefined User Roles for Cortex XDR

Role-based access control (RBAC) enables you to use preconfigured roles to assign access rights to administrative users. You can manage roles for all Cortex apps and services in the hub. By assigning roles, you enforce the separation of access among functional or regional areas of your organization.

Each role extends specific privileges to users. The way you configure administrative access depends on the security requirements of your organization. Use roles to assign specific access privileges to administrative user accounts. The built-in roles provide specific access rights that cannot be changed. Use hub roles to provide full access to Cortex XDR with three levels: Account, App, or Instance. If you desire more granular access control, you can assign any of the Cortex XDR app roles.

The following table describes the Cortex XDR predefined roles and the view and action privileges associated with each.

Some features are license dependent. As a result users may not see a specific feature if the feature is not supported by the license type or if they do not have access based on their assigned role.

<table>
<thead>
<tr>
<th>Role</th>
<th>Description</th>
<th>View Privileges</th>
<th>Action Privileges</th>
</tr>
</thead>
<tbody>
<tr>
<td>App Administrator</td>
<td>The user has full access to the given apps, including all current and future app instances. App Administrator can assign roles for app instances, and can also activate app instances specific to that app.</td>
<td>• Endpoints&lt;br&gt;• Endpoint Profiles&lt;br&gt;• Global Exceptions&lt;br&gt;• Endpoint Policies&lt;br&gt;• Endpoint Management&lt;br&gt;• Endpoint Installations&lt;br&gt;• Device Control</td>
<td>• Investigation&lt;br&gt;• Rules&lt;br&gt;• Incidents&lt;br&gt;• Alerts&lt;br&gt;• Response&lt;br&gt;• Quarantine&lt;br&gt;• Request WildFire Verdict Change&lt;br&gt;• Blacklist</td>
</tr>
<tr>
<td>Role</td>
<td>Description</td>
<td>View Privileges</td>
<td>Action Privileges</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>Requires a Cortex XDR license.</td>
<td><strong>Investigation</strong>&lt;br&gt;- Rules&lt;br&gt;- Incidents&lt;br&gt;- Alerts&lt;br&gt;- Response&lt;br&gt;- Action Center&lt;br&gt;- Configurations&lt;br&gt;- Analytics Management&lt;br&gt;- Public API&lt;br&gt;- Auditing&lt;br&gt;- Alert Notifications&lt;br&gt;- Threat Intelligence&lt;br&gt;- On-demand Analytics&lt;br&gt;- External Alerts Mapping&lt;br&gt;- Broker Services</td>
<td><strong>Terminate Process</strong>&lt;br&gt;- Isolate&lt;br&gt;- Live Terminal&lt;br&gt;- EDL&lt;br&gt;- File Retrieval&lt;br&gt;- Endpoints&lt;br&gt;- Retrieve Endpoint Data&lt;br&gt;- Endpoint Scan&lt;br&gt;- Endpoint Profiles&lt;br&gt;- Global Exceptions&lt;br&gt;- Endpoint Policies&lt;br&gt;- Endpoint Management&lt;br&gt;- Endpoint Installations&lt;br&gt;- Device Control</td>
</tr>
<tr>
<td>Instance Administrator</td>
<td>The user has full access to the app instance. The Instance Administrator can make other users Instance Administrator for the app instance. If the app has predefined or custom roles, the Instance Administrator can assign those roles to other users.</td>
<td><strong>Endpoints</strong>&lt;br&gt;- Endpoint Profiles&lt;br&gt;- Global Exceptions&lt;br&gt;- Endpoint Policies&lt;br&gt;- Endpoint Management&lt;br&gt;- Endpoint Installations&lt;br&gt;- Device Control&lt;br&gt;- Investigation&lt;br&gt;- Rules&lt;br&gt;- Incidents&lt;br&gt;- Alerts&lt;br&gt;- Response&lt;br&gt;- Action Center&lt;br&gt;- Configurations&lt;br&gt;- Analytics Management&lt;br&gt;- Public API&lt;br&gt;- Auditing&lt;br&gt;- Alert Notifications&lt;br&gt;- Threat Intelligence&lt;br&gt;- On-demand Analytics&lt;br&gt;- External Alerts Mapping&lt;br&gt;- Broker Services</td>
<td><strong>Investigation</strong>&lt;br&gt;- Rules&lt;br&gt;- Incidents&lt;br&gt;- Alerts&lt;br&gt;- Response&lt;br&gt;- Quarantine&lt;br&gt;- Request WildFire Verdict Change&lt;br&gt;- Blacklist&lt;br&gt;- Terminate Process&lt;br&gt;- Isolate&lt;br&gt;- Live Terminal&lt;br&gt;- EDL&lt;br&gt;- File Retrieval&lt;br&gt;- Endpoints&lt;br&gt;- Retrieve Endpoint Data&lt;br&gt;- Endpoint Scan&lt;br&gt;- Endpoint Profiles&lt;br&gt;- Global Exceptions&lt;br&gt;- Endpoint Policies&lt;br&gt;- Endpoint Management&lt;br&gt;- Endpoint Installations&lt;br&gt;- Device Control</td>
</tr>
<tr>
<td>Role</td>
<td>Description</td>
<td>View Privileges</td>
<td>Action Privileges</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------</td>
<td>-------------------</td>
</tr>
</tbody>
</table>
| Viewer                      | Can view the majority of the features of the XDR app for this instance, but can take no actions. Requires a Cortex XDR license. | • Endpoints  
  • Endpoint Profiles  
  • Global Exceptions  
  • Endpoint Policies  
  • Endpoint Management  
  • Endpoint Installations  
  • Device Control  
  • Investigation  
  • Rules  
  • Incidents  
  • Alerts  
  • Response  
  • Action Center  
  • Configurations  
  • Auditing | —                              |
| Security Admin              | Can triage and investigate alerts and incidents, respond (excluding Live Terminal), and edit profiles and policies. Requires a Cortex XDR Prevent or Cortex XDR Pro per Endpoint license. | • Endpoints  
  • Endpoint Profiles  
  • Global Exceptions  
  • Endpoint Policies  
  • Endpoint Management  
  • Endpoint Installations  
  • Device Control  
  • Investigation  
  • Rules  
  • Incidents  
  • Alerts  
  • Response  
  • Probe  
  • EDL  
  • Endpoints  
  • Retrieve Endpoint Data  
  • Endpoint Scan  
  • Endpoint Profiles  
  • Endpoint Policies | • Investigation  
  • Rules  
  • Incidents  
  • Alerts  
  • Response  
  • Quarantine  
  • Request WildFire Verdict Change  
  • Blacklist  
  • Terminate Process  
  • Isolate  
  • EDL  
  • Endpoints  
  • Retrieve Endpoint Data  
  • Endpoint Scan  
  • Endpoint Profiles  
  • Endpoint Policies |
| Privileged Security Admin   | Can triage and investigate alerts and incident, respond, and edit profiles and policies. Requires a Cortex XDR Prevent or Cortex XDR Pro per Endpoint license. | • Endpoints  
  • Endpoint Profiles  
  • Global Exceptions  
  • Endpoint Policies  
  • Endpoint Management  
  • Endpoint Installations | • Investigation  
  • Rules  
  • Incidents  
  • Alerts  
  • Response  
  • Quarantine  
  • Request WildFire Verdict Change |
<table>
<thead>
<tr>
<th>Role</th>
<th>Description</th>
<th>View Privileges</th>
<th>Action Privileges</th>
</tr>
</thead>
<tbody>
<tr>
<td>IT Admin</td>
<td>Can manage and control endpoints and installations, configure brokers, view profiles, policies, and alerts. Requires a Cortex XDR Prevent or Cortex XDR Pro per Endpoint license.</td>
<td>• Endpoints&lt;br&gt;- Endpoint Profiles&lt;br&gt;- Global Exceptions&lt;br&gt;- Endpoint Policies&lt;br&gt;- Endpoint Management&lt;br&gt;- Endpoint Installations&lt;br&gt;- Device Control&lt;br&gt;- Investigation&lt;br&gt;- Incidents&lt;br&gt;- Alerts&lt;br&gt;- Response&lt;br&gt;- Action Center&lt;br&gt;- Configurations&lt;br&gt;- Broker Services</td>
<td>• Endpoints&lt;br&gt;- Retrieve Endpoint Data&lt;br&gt;- Global Exceptions&lt;br&gt;- Endpoint Management&lt;br&gt;- Endpoint Installations</td>
</tr>
<tr>
<td>Privileged IT Admin</td>
<td>Can manage and control endpoints and installations, configure brokers, create profiles and policies, view alerts, and initiate Live Terminal. Requires a Cortex XDR Prevent or Cortex XDR Pro per Endpoint license.</td>
<td>• Endpoints&lt;br&gt;- Endpoint Profiles&lt;br&gt;- Endpoint Policies&lt;br&gt;- Endpoint Management&lt;br&gt;- Endpoint Installations&lt;br&gt;- Device Control&lt;br&gt;- Investigation&lt;br&gt;- Incidents&lt;br&gt;- Alerts</td>
<td>• Investigation&lt;br&gt;- Incidents&lt;br&gt;- Alerts&lt;br&gt;- Response&lt;br&gt;- Request WildFire Verdict Change&lt;br&gt;- Live Terminal&lt;br&gt;- File Retrieval&lt;br&gt;- Endpoints&lt;br&gt;- Retrieve Endpoint Data</td>
</tr>
<tr>
<td>Role</td>
<td>Description</td>
<td>View Privileges</td>
<td>Action Privileges</td>
</tr>
<tr>
<td>----------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Deployment Admin</td>
<td>Can manage and control endpoints and installations, and configure brokers.</td>
<td>• Endpoints</td>
<td>• Endpoints</td>
</tr>
<tr>
<td></td>
<td>Requires a Cortex XDR Prevent or Cortex XDR Pro per Endpoint license.</td>
<td>• Global Exceptions</td>
<td>• Endpoint Management</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Endpoint Management</td>
<td>• Endpoint Installations</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Endpoint Installations</td>
<td>• Device Control</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Configurations</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Auditing</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Broker Services</td>
<td></td>
</tr>
<tr>
<td>Investigation Admin</td>
<td>Can view and triage alerts and incidents, configure rules, and view the</td>
<td>• Endpoints</td>
<td>• Investigation</td>
</tr>
<tr>
<td></td>
<td>profiles and policies and analytics management screens.</td>
<td>• Endpoint Profiles</td>
<td>• Rules</td>
</tr>
<tr>
<td></td>
<td>Requires a Cortex XDR license.</td>
<td>• Endpoint Policies</td>
<td>• Incidents</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Device Control</td>
<td>• Alerts</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Investigation</td>
<td>• Endpoints</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Rules</td>
<td>• Retrieve Endpoint Data</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Incidents</td>
<td>• Endpoint Scan</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Alerts</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Endpoints</td>
<td></td>
</tr>
<tr>
<td>Investigator</td>
<td>Can view and triage alerts and incidents.</td>
<td>• Investigation</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Requires a Cortex XDR license.</td>
<td>• Incidents</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Alerts</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Endpoints</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Retrieve Endpoint Data</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Endpoint Scan</td>
<td></td>
</tr>
<tr>
<td>Privileged</td>
<td>Can view and triage alerts, incidents and rules, profiles and policies and</td>
<td>• Endpoints</td>
<td>• Investigation</td>
</tr>
<tr>
<td>Investigator</td>
<td>analytics management screens.</td>
<td>• Endpoint Profiles</td>
<td>• Incidents</td>
</tr>
<tr>
<td></td>
<td>Requires a Cortex XDR Pro per Endpoint license.</td>
<td>• Endpoint Policies</td>
<td>• Alerts</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Device Control</td>
<td>• Endpoints</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Investigation</td>
<td>• Retrieve Endpoint Data</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Rules</td>
<td>• Endpoint Scan</td>
</tr>
</tbody>
</table>

© 2020 Palo Alto Networks, Inc.
<table>
<thead>
<tr>
<th>Role</th>
<th>Description</th>
<th>View Privileges</th>
<th>Action Privileges</th>
</tr>
</thead>
</table>
| Responder             | Can view and triage alerts, and access all response capabilities excluding Live Terminal. Requires a Cortex XDR Prevent or Cortex XDR Pro per Endpoint license.                                                 | • Investigation  
• Rules  
• Incidents  
• Alerts  
• Response  
• Action Center | • Response  
• Quarantine  
• Request WildFire Verdict Change  
• Blacklist  
• Terminate Process  
• Isolate  
• EDL  
• Endpoints  
• Retrieve Endpoint Data  
• Endpoint Scan |
| Privileged Responder  | Can view and triage alerts and incidents, access all response capabilities, and configure rules, policies, and profiles. Requires a Cortex XDR license.                                                     | • Endpoints  
• Endpoint Profiles  
• Endpoint Policies  
• Endpoint Management  
• Investigation  
• Rules  
• Incidents  
• Alerts  
• Response  
• Action Center  
• Configurations  
• Analytics Management | • Investigation  
• Rules  
• Incidents  
• Alerts  
• Response  
• Quarantine  
• Request WildFire Verdict Change  
• Blacklist  
• Terminate Process  
• Isolate  
• Live Terminal  
• EDL  
• File Retrieval  
• Endpoints  
• Retrieve Endpoint Data  
• Endpoint Scan  
• Device Control |
Activate your Network Devices

With a Cortex XDR Pro per TB license, if you use Palo Alto Networks firewalls as a traffic log source, you must activate your firewalls and Panorama. If you use Panorama to manage firewalls, you must activate your firewalls before you continue with activation of Cortex XDR. If you only use one firewall or use multiple firewalls but do not manage them using Panorama, you can activate your firewalls after you activate Cortex XDR.

STEP 1 | Register and activate your firewalls and Panorama.

STEP 2 | Onboard Panorama-Managed Firewalls to Cortex Data Lake.

STEP 3 | Upgrade firewalls and Panorama to the latest software and content releases.

   PAN-OS 8.0.6 is the minimum required software release version for Palo Alto Networks firewalls and Panorama. However, to enable Cortex XDR to leverage the Directory Sync Service and Enhanced Application Logs, upgrade firewalls and Panorama to PAN-OS 8.1.1 or later and to the latest content release:

   □ Get the latest application and threat content updates.
   □ Upgrade to PAN-OS 8.1.1.

STEP 4 | Ensure that firewalls have visibility into internal traffic and applications.

   It’s important that at least one firewall sending logs to the Cortex Data Lake is processing or has visibility into internal traffic and applications.

   If you have deployed only internet gateway firewalls, one option might be to configure a tap interface to give a firewall visibility into data center traffic even though the firewall is not in the traffic flow. Connect the tap mode interface to a data center switch SPAN or mirror port that provides the firewall with the mirrored traffic, and make sure that the firewall is enabled to log the traffic and send it to the Cortex Data Lake.

   Because data center firewalls already have visibility into internal network traffic, you don’t need to configure these firewalls in tap mode; however, contact Palo Alto Networks Professional Services for best practices to ensure that the Cortex Data Lake and Cortex XDR-required configuration updates do not affect data center firewall deployments.

STEP 5 | Configure firewalls to forward Cortex XDR-required logs to Cortex Data Lake.

   The Cortex Data Lake provides centralized, cloud-based log storage for firewalls, and Panorama provides an interface you can use to view the stored logs. The rich log data that firewalls forward to the Cortex Data Lake provides the Cortex XDR analytics engine the network visibility it requires to perform data analytics.

   To support Cortex XDR, firewalls must forward at least Traffic logs to the Cortex Data Lake. The complete set of log types that a firewall should forward to the Cortex Data Lake are:

   □ Traffic (required)
   □ URL Filtering
   □ User-ID
   □ Configuration
   □ Correlation
   □ HIP
   □ System Logs
   □ Enhanced application logs (PAN-OS 8.1.1 or later)
**Enhanced application logs** are designed to increase visibility into network activity for Palo Alto Networks Cloud Services apps, and Cortex XDR requires these logs to support certain features.

Follow the complete workflow to **configure Panorama-managed firewalls to forward logs to the Cortex Data Lake**.
Activate Cortex XDR

Use the hub (https://apps.paloaltonetworks.com) to activate Cortex XDR. This is a one-time task you'll need to perform when you first start using Cortex XDR. After you've activated the Cortex XDR app—and completed all the steps described in Set up Cortex XDR Pro Overview—you'll only need to repeat the activation if you want to add additional app instances.

To activate the Cortex XDR app, you must be assigned a required role and locate your activation email containing a link to begin activation in the hub. Activating Cortex XDR automatically includes activation of Cortex Data Lake.

**STEP 1 | Begin activation.**

1. Click the activation link you received in email to begin activation in the hub.
2. If you manage multiple company CSP accounts, make sure you select the specific account to which you want to allocate the Cortex XDR license to before proceeding with activation.

![The hub will associate activation of Cortex XDR and the included apps and services only with the selected account.]

3. From the Cortex XDR tile, select the serial number you want to activate.
   
   If there is only one serial number associated with your company account, you can click the tile to begin activation.

   ![If you have multiple serial numbers associated, click each one to activate.]

Provide details about the Cortex XDR app you’re activating.

Activate Cortex XDR (Serial Number: [redacted])

Please provide the following information to set up the app:

- **Company Account**—Identifies the company account under which you are activating Cortex XDR.
- **Name**—Give your Cortex XDR app instance an easily-recognizable name and optional **Description**.
  If you have more than one Cortex XDR instance, the hub displays the name in the instance list when you select the Cortex XDR tile. Choose a name that is 59 or fewer characters and is unique across your company account.
- **Subdomain**—Give your Cortex XDR instance an easy to recognize name. The hub displays the name you assign on the list of available instances for the Cortex XDR app. You can also access the Cortex XDR app directly using the full URL (https://<subdomain>.xdr.<region>.paloaltonetworks.com). If you are converting an existing Traps management service to Cortex XDR, this field is grayed out.
- **Cortex Data Lake**—Select the Cortex Data Lake instance that will provide the Cortex XDR apps with log data.
  If you activated with an auth code, provision a new Cortex Data Lake instance by selecting the link to **activate purchased licenses** and provide the separate Cortex Data Lake auth code you received in email.
  If you activated with the activation link, you can automatically provision a new Cortex Data Lake instance in the region you select or select an existing Cortex Data Lake and increase its size.

You can only select a Cortex Data Lake instance that is not allocated to another Cortex XDR instance. When you select a Cortex Data Lake instance, the hub provisions your Cortex XDR instance in the same region.

EULA
By clicking “Agree & Activate”, you accept the terms of the End User License Agreement.
• **Region**—Select a region in which you want to set up your Cortex Data Lake instance. If you selected an existing Cortex Data Lake instance, this field automatically displays the region in which your Cortex Data Lake instance is deployed and cannot be changed.

• **Directory Sync**—*(Optional)* Select the Directory Sync Service instance that will provide the Cortex XDR app with Active Directory data. If there is only one Directory Sync Service instance for the selected Cortex Data Lake region, the hub automatically selects it for pairing with the Cortex XDR app, however you can clear the default selection, if desired. If you do not currently have a Directory Sync Service activated and configured for your account, you can select the link to **create an instance** now, or you can add one at a later time.

**STEP 3** | Review the end user license agreement and **Agree & Activate**.

The hub displays the activation status as it activates and provisions your apps. It can take up to an hour to complete activation. After activation completes, the hub displays a summary that shows the details for your apps and services.

**STEP 4** | **Manage Apps** to view the current status of your apps.

When the app is available you will see a green check mark in the **STATUS** column. To return to the status page at a later time, return to the hub and select > **Manage Apps**.

**STEP 5** | When your app is available, log in to your Cortex XDR app to confirm that you can successfully access the Cortex XDR app interface.

**STEP 6** | **Allocate Log Storage for Cortex XDR**.

Review the storage allocation for your Cortex Data Lake and adjust the quota as needed. You must be an assigned an Instance Administrator or higher role to for Cortex Data Lake to manage logging storage.

**STEP 7** | **Assign roles to additional administrators, if needed**.

**STEP 8** | **Complete your configuration**.

If you have a Cortex XDR Pro per Endpoint license, continue to **Set up Endpoint Protection**. Otherwise proceed to **Set up Network Analysis**.
Set Up Directory Sync

Directory Sync is an optional service that enables you to leverage Active Directory user, group, and computer information in Cortex XDR apps to provide context when you investigate alerts. You can use Active Directory information in policy configuration and endpoint management.

After you finish the setup, Cortex XDR syncs with Directory Sync every 24 hours.

To set up Directory Sync:

**STEP 1** | Add and configure your Directory Sync instance.
See the Directory Sync Service Getting Started Guide for instructions.

**STEP 2** | Pair the Directory Sync to Cortex XDR apps.
Pairing can occur during Cortex XDR activation or after you activate Cortex XDR apps.

**STEP 3** | After you activate and pair Cortex XDR apps with Directory Sync, you must define which Active Directory domain the analytics engine should use.

Wait about ten minutes after you have paired Directory Sync before you do this.

1. From Cortex XDR, select > Analytics Management > Configuration > Directory Sync.

   If a pop up message appears in the upper-right indicating that Directory Sync is not configured, then you haven't waited long enough after pairing. Wait a few more minutes before trying again.

2. In the Domain drop-down, select the Active Directory domain that you want Cortex XDR to use.
3. Submit and you're done configuring the analytics engine to use Directory Sync.

Pairing Directory Sync

If you did not pair Directory Sync to your Cortex apps during Cortex XDR activation, you can later pair it with your Cortex XDR instance.

**STEP 1** | Log into the hub.

**STEP 2** | Click the gear > Manage Apps in the upper-right corner.

**STEP 3** | Locate the Directory Sync instance that you want to use with Cortex XDR. Make a note of the instance’s name, which appears in the left-most column.

   If you have more than one instance, make sure you choose the instance that is in the same region as the Cortex Data Lake instance you are using with your apps.
STEP 4 | Pair the Directory Sync instance with your Cortex XDR instance.

1. Scroll down until you find your Cortex XDR instance in the Cortex XDR section.
2. Click on its name in the left-most column.
3. In the resulting pop-up configuration screen, select the desired Directory Sync instance, and then click OK.
Allocate Log Storage for Cortex XDR

You receive Cortex Data Lake log storage based on the amount of storage associated with your Cortex XDR Licenses. Generally, this capacity is determined by factors such as the size of your network and number of endpoints in your deployment. To increase your capacity, contact your Palo Alto Network account representative.

When you activate Cortex XDR, Cortex Data Lake assigns a default storage allocation for your logs, EDR data, and alerts. While some Cortex apps receive a default allocation, with a Cortex XDR Pro per TB license, you must manually allocate storage for firewall logs. After you activate Cortex XDR, it is recommended to review and adjust your log storage allocation depending on your storage requirements.

Cortex Data Lake displays the current possible allocation but does not display the storage usage.

To allocate your log storage quota:

**STEP 1 | Sign In** to the Palo Alto Networks hub at https://apps.paloaltonetworks.com/.

**STEP 2 | Select your Cortex Data Lake instance.**

If you have multiple Cortex Data Lake instances, select the Cortex Data Lake tile and then select the Cortex Data Lake instance from the list of available instances associated with your account.

Cortex Data Lake displays the service status and your total logging storage capacity.

**STEP 3 | Select Configuration to define logging storage settings.**

Cortex Data Lake displays the total storage allocated for the apps and services associated with the Cortex Data Lake instance.

The Cortex Data Lake depicts your storage allocation graphically. As you adjust your storage allocation, the graphic updates to display the changes to your storage policy. The Cortex Data Lake storage policy specifies the distribution of your total storage allocated to each app or service and the minimum retention warning (not supported with Cortex XDR).

**STEP 4 | Allocate quota for Cortex XDR.**

1. If you purchased quota for firewall logs, allocate quota to the Firewall log type.
   
   To use the same Cortex Data Lake instance for both firewall logs and Cortex XDR logs, you must first associate Panorama with the Cortex Data Lake instance before you can allocate quota for firewall logs.

2. Review your storage allocation for Cortex XDR.
   
   By default, 80% of your available storage for Cortex XDR is assigned to logs and data, and 20% is assigned to alerts. It is recommended to review the status of your Cortex Data Lake instance after about two weeks of data collection and make adjustments as needed but to use the default allocations as a starting point.

**STEP 5 | Apply your changes.**
Set up Endpoint Protection

The Cortex XDR agent monitors endpoint activity and collects endpoint data that Cortex XDR uses to raise alerts. Before you can begin collecting endpoint data, you must deploy the Cortex XDR agent and configure endpoint policy. To use endpoint management functions in Cortex XDR you must be assigned an administrative role in the hub.

**STEP 1 | Verify the status of your Cortex XDR tenant.**
1. From the hub, click the gear icon next to your name.
2. In the Cortex area, review the **STATUS** for the tenant you just activated.
   
   When Cortex XDR tenant is available, the status changes to the green check mark.

**STEP 2 | Plan Your Agent Deployment.**

**STEP 3 | Enable Access to Cortex XDR.**

**STEP 4 | (Optional) Set up Broker VM communication.**

**STEP 5 | Install the Cortex XDR agent on your endpoints.**

Install the agent software directly on an endpoint or use a software deployment tool of your choice (such as JAMF or GPO) to distribute and install the software on multiple endpoints.

1. **Create an Agent Installation Package.**
2. **Install the Cortex XDR agent.**

   For instructions by operating system, see the Cortex XDR Agent Administrator's Guide or the Traps Agent Administrator's Guide if you use an earlier version.

**STEP 6 | Define Endpoint Groups to which you can apply endpoint security policy.**

**STEP 7 | Customize your Endpoint Security Profiles and assign them to your endpoints.**

Cortex XDR provides out-of-the-box exploit and malware protection. However, at minimum, you must enable **Data Collection** in an Agent Settings profile to leverage endpoint data in Cortex XDR apps. Data collection for Windows endpoints is available with Traps 6.0 and later releases and on endpoints running Windows 7 SP1 and later releases. Data collection on macOS and Linux endpoints are available with Traps 6.1 and later releases.

**STEP 8 | (Optional) Configure Device Control profiles to restrict file execution on USB-connected devices.**

**STEP 9 | Verify that the Cortex XDR agent can connect to your Cortex XDR instance.**

If successful, the Cortex XDR console displays a Connected status. You can view the status of all agents on the **Endpoints > Endpoint Management** of your Cortex XDR interface.

**STEP 10 | Configure the internal networks that you want Cortex XDR to monitor.**

1. Log in to your Cortex XDR app either using the direct link or from the Cortex XDR tile on the hub.
2. To view existing network segments, select the gear (⚙️) in the upper right corner and select **Analytics Management > Status > Analytics Network Coverage Status**. This page provides a table of the IP address ranges Cortex XDR Analytics monitors, which is pre-populated with the default IPv4 and IPv6 address spaces.
3. To add custom network segments, select **Configuration** and then **Network Segments Configuration**.
4. Add (➕) a new segment and enter the first and last IP address of the range to monitor.
5. Save (✔) the network segment. If the Configuration saved notification does not appear, save again.

**STEP 11** | If you also have a Cortex XDR Pro per TB license, proceed to **Set up Network Analysis**.
Otherwise, proceed to **Configure XDR**.

## Plan Your Agent Deployment

You typically deploy Cortex XDR agent software to endpoints across a network after an initial proof of concept (POC), which simulates your corporate production environment. During the POC or deployment stage, you analyze security events to determine which are triggered by malicious activity and which are due to legitimate processes behaving in a risky or incorrect manner. You also simulate the number and types of endpoints, the user profiles, and the types of applications that run on the endpoints in your organization and, according to these factors, you define, test, and adjust the security policy for your organization.

The goal of this multi-step process is to provide maximum protection to the organization without interfering with legitimate workflows.

After the successful completion of the initial POC, we recommend a multi-step implementation in the corporate production environment for the following reasons:

- The POC doesn't always reflect all the variables that exist in your production environment.
- There is a rare chance that the Cortex XDR agent will affect business applications, which can reveal vulnerabilities in the software as a prevented attack.
- During the POC, it is much easier to isolate issues that appear and provide a solution before full implementation in a large environment where issues could affect a large number of users.

A multi-step deployment approach ensures a smooth implementation and deployment of the Cortex XDR solution throughout your network. Use the following steps for better support and control over the added protection.

<table>
<thead>
<tr>
<th>Step</th>
<th>Duration</th>
<th>Plan</th>
</tr>
</thead>
<tbody>
<tr>
<td>0. Calculate the bandwidth required to support the number of agents you plan to deploy.</td>
<td>as needed</td>
<td>For every 100,000 agents, you will need to allocate 120Mbps of bandwidth. The bandwidth requirement scales linearly. For example, to support 300,000 agents, plan to allocate 360Mbps of bandwidth (three times the amount required for 100,000 agents).</td>
</tr>
<tr>
<td>1. Install Cortex XDR on endpoints.</td>
<td>1 week</td>
<td>Install the Cortex XDR agent on a small number of endpoints (3 to 10). Test normal behavior of the Cortex XDR agents (injection and policy) and confirm that there is no change in the user experience.</td>
</tr>
<tr>
<td>2. Expand the Cortex XDR deployment.</td>
<td>2 weeks</td>
<td>Gradually expand agent distribution to larger groups that have similar attributes (hardware, software, and users). At the end of two weeks you can have Cortex XDR deployed on up to 100 endpoints.</td>
</tr>
</tbody>
</table>
### Enable Access to Cortex XDR

After you receive your account details, enable and verify access to Cortex XDR.

Some of the IP addresses required for access are registered in the United States. As a result, some GeoIP databases do not correctly pinpoint the location in which IP addresses are used. In regard to customer data, Cortex Data Lake stores all data in your deployment region, either EU or US, regardless of the IP address registration and restricts data transmission through any infrastructure to that region.

Throughout this topic, `<xdr-tenant>` refers to the chosen subdomain of your Cortex XDR tenant and `<region>` is the region in which your Cortex Data Lake is deployed, either `us` or `eu`.

Some steps also require you to know your Cortex Data Lake tenant ID (`<cortex-data-lake-tenant-ID>`). To identify the ID of your Cortex Data Lake tenant:

<table>
<thead>
<tr>
<th>Cortex XDR management console</th>
<th>Traps management service</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Log in to the Cortex XDR management console.</td>
<td>1. Log in to the Cortex XDR management console.</td>
</tr>
<tr>
<td>2. From your username, select About.</td>
<td>2. From your username, select About.</td>
</tr>
</tbody>
</table>

**STEP 1 | (Optional)** If you are deploying the broker VM as a proxy between Cortex XDR and the Cortex XDR agents, start by enabling the communication between them.
STEP 2 | In your firewall configuration, enable access to Cortex XDR communication servers and storage buckets.

With Palo Alto Networks firewalls, we recommend that you use the following App-IDs to allow communication between Cortex XDR agents and Cortex XDR management console when you configure your security policy:

- **cortex-xdr**—Used for communication between the XDR console and the information stored in Google Cloud Platform. Requires PAN-OS Applications and Threats content update version 8279 or a later release.
- **traps-management-service**—Used for communication between Cortex XDR agents and the Cortex XDR management console. Requires PAN-OS Applications and Threats content update version 793 or a later release.

If you do not use Palo Alto Networks firewalls, ensure that you configure your firewall policy to enable communication with the FQDNs.

<table>
<thead>
<tr>
<th>FQDN</th>
<th>App-ID Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>distributions.traps.paloaltonetworks.com</td>
<td>traps-management-service</td>
</tr>
<tr>
<td>dc-&lt;xdr-tenant&gt;.traps.paloaltonetworks.com</td>
<td>traps-management-service</td>
</tr>
<tr>
<td>ch-&lt;xdr-tenant&gt;.traps.paloaltonetworks.com</td>
<td>traps-management-service</td>
</tr>
<tr>
<td>cc-&lt;xdr-tenant&gt;.traps.paloaltonetworks.com</td>
<td>traps-management-service</td>
</tr>
<tr>
<td>contentprod.traps.paloaltonetworks.com</td>
<td>traps-management-service</td>
</tr>
<tr>
<td>wss://lrc-&lt;region&gt;.paloaltonetworks.com</td>
<td>cortex-xdr</td>
</tr>
<tr>
<td>panw-xdr-installers-prod-us.storage.googleapis.com</td>
<td>cortex-xdr</td>
</tr>
</tbody>
</table>

*This storage bucket is used for all regions.*

<table>
<thead>
<tr>
<th>FQDN</th>
<th>App-ID Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>panw-xdr-payloads-prod-us.storage.googleapis.com</td>
<td>cortex-xdr</td>
</tr>
</tbody>
</table>

*This storage bucket is used for all regions.*

<table>
<thead>
<tr>
<th>FQDN</th>
<th>App-ID Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>global-content-profiles-policy.storage.googleapis.com</td>
<td>cortex-xdr</td>
</tr>
<tr>
<td>panw-xdr-evr-prod-&lt;region&gt;.storage.googleapis.com</td>
<td>cortex-xdr</td>
</tr>
</tbody>
</table>

STEP 3 | To establish secure communication (TLS) to Cortex XDR, the endpoints, and any other devices that initiate a TLS connection with Cortex, you must have the following certificates installed on the operating system:
STEP 4 | If you use SSL decryption, we recommend that you do not decrypt Cortex XDR services.

To exclude Cortex XDR services from decryption, add the following domains to your SSL Decryption Exclusion list where `<region>` is your deployment region, either `us` or `eu`:

- `*.traps.paloaltonetworks.com`
- `*.xdr.<region>.paloaltonetworks.com`
- `panw-xdr-evr-prod-<region>.storage.googleapis.com`
- `panw-xdr-installers-prod-us.storage.googleapis.com`
- `panw-xdr-payloads-prod-us.storage.googleapis.com`
- `global-content-profiles-policy.storage.googleapis.com`

In PAN-OS 8.0 and later releases, you can configure the list in Device > Certificate Management > SSL Decryption Exclusion.

STEP 5 | (Windows only) Enable access for Windows CRL checks.

(Endpoints running the following or later releases: Traps 6.0.3, Traps 6.1.1, and Cortex XDR 7.0) When the Cortex XDR agent examines portable executables (PEs) running on the endpoint as part of the enforced Malware Security Profile, the agent performs a certificate revocation (CRL) check. The CRL check ensures that the certificate used to sign a given PE is still considered valid by its Certificate Authority (CA), and has not been revoked. To validate the certificate, the Cortex XDR agent leverages Microsoft Windows APIs and triggers the operating system to fetch the specific Certificate Revocation List (CRL) from the internet. To complete the certificate revocation check, the endpoint needs HTTP access to a dynamic list of URLs, based on the PEs that are executed or scanned on the endpoint.

1. If a system-wide proxy is defined for the endpoint (statically or using a PAC file), Microsoft Windows downloads the CRL lists through the proxy.
2. If a specific proxy is defined for the Cortex XDR agent, and the endpoint has no access to the internet over HTTP, then Microsoft Windows will fail to download the CRL lists. As a result, the certificate revocation check will fail and the certificate will be considered valid by the agent, while creating a latency in executing PEs. If the Cortex XDR agent is running in an isolated environment that prohibits the successful completion of certificate revocation checks, the Palo Alto Networks Support team can provide a configuration file that will disable the revocation checks and avoid unnecessary latency in the execution time of PEs.

STEP 6 | (Windows only) Enable serverless peer-to-peer (P2) content updates.

By default, the Cortex XDR agent retrieves content updates from its peer Cortex XDR agents on the same subnet. To enable P2P, you must enable UDP and TCP over port 33221. You can change the port...
number or choose to download the content directly from the Cortex XDR server in the Agent settings profile.

STEP 7 | Verify that you can access your Cortex XDR tenant.

After you download and install the Cortex XDR agent software on your endpoints and configure your endpoint security policy, verify that the Cortex XDR agents can check in with Cortex XDR to receive the endpoint policy.

Proxy Communication

You can configure communication through proxy servers between the Cortex XDR server and the Cortex XDR agents running on Windows, Mac, and Linux endpoints. The Cortex XDR agent uses the proxy settings defined as part of the Internet & Network settings or WPAD protocol on the endpoint. You can also configure a list of proxy servers that your Cortex XDR agent will use to communicate with Cortex XDR server.

Cortex XDR supports the following types of proxy configurations:

- **System-wide proxy**—Use system-wide proxy to send all communication on the endpoint including to and from the Cortex XDR agent through a proxy server configured for the endpoint. Cortex XDR supports proxy communication for proxy settings defined explicitly on the endpoint, as well as proxy settings configured in a proxy auto-config (PAC) file.

- **Application-specific proxy**—(Available with Traps agent 5.0.9, Traps agent 6.1.2, and Cortex XDR agent 7.0 and later releases) Configure a Cortex XDR specific proxy that applies only to the Cortex XDR agent and does not enforce proxy communications with other apps or services on your endpoint. You can set up to five proxy servers either during the Cortex XDR agent installation process, or following agent installation, directly from the Cortex XDR management console.

  If the endpoints in your environment are not connected directly to the internet, you can deploy a Palo Alto Networks broker VM.

Application-specific proxy configurations take precedence over system-wide proxy configurations. The Cortex XDR agent retrieves the proxy list defined on the endpoint and tries to establish communication with the Cortex XDR server first through app-specific proxies. Then, if communication is unsuccessful, the agent tries to connect using the system-wide proxy, if defined. If none are defined, the Cortex XDR agent attempts communication with the Cortex XDR server directly.
Set up Network Analysis

With a Cortex XDR Pro per TB license you must set up your network sensors and define network coverage for your internal networks.

**STEP 1 | Set up your network sensors.**

1. If you use unmanaged Palo Alto Networks firewalls, and did not configure log-forwarding on your firewalls before activating Cortex XDR, **Start Sending Logs to Cortex Data Lake**.
2. (Optional) **Ingest Logs and Alerts from External Sources.**
   
   If you have external (non-Palo Alto Networks) network sensors, you can set up a syslog collector to receive alerts or logs from them. If you send external alerts, Cortex XDR can include any them in relevant incidents for a more complete picture of the activity involved. If you send logs and alerts from external sources such as Check Point firewalls, Cortex XDR can apply analytics analysis and raise analytics alerts on the external logs and include the external alerts in incidents for additional context.
3. (Optional) If you use Okta or Azure AD, you can **Ingest Authentication Logs and Data into authentication stories.** After you set up log collection, you can search for authentication data using the Query Builder.
4. (Optional) If you want to use Pathfinder to examine network hosts, servers, and workstations for malicious or risky software, **Set Up Pathfinder.** If you want to use Pathfinder to supplement the Cortex XDR agent or choose not to use Cortex XDR for endpoint protection, **Set Up Pathfinder.**

**STEP 2 | Configure the internal networks that you want Cortex XDR to monitor.**

1. Log in to your Cortex XDR app either using the direct link or from the Cortex XDR tile on the hub.
2. To view existing network segments, select the gear (⚙️) in the upper right corner and select **Analytics Management > Status > Analytics Network Coverage Status.** This page provides a table of the IP address ranges Cortex XDR Analytics monitors, which is pre-populated with the default IPv4 and IPv6 address spaces.
3. To add custom network segments, select **Configuration** and then **Analytics Network Coverage Status.**
4. Add (+) a new segment and enter the first and last IP address of the range to monitor.
5. Specify the **Assigned Pathfinder VM** to assign a Pathfinder VM to the network segment. If you do not want Pathfinder to scan a particular segment, then leave the field blank.
6. (Optional) If you want to further limit Pathfinder scans to specific devices, go to the **Pathfinder** page and then select **Per Asset Configuration.** Use these settings to override the default Pathfinder configuration on a per-asset basis.
7. Leave **Reserved for VPN** blank. See the following step for adding your GlobalProtect VPN IP address pool to the Cortex XDR app as a network segment to monitor.
8. Save (✔️) the network segment. If the Configuration saved notification does not appear, save again.

**STEP 3 | If you use GlobalProtect or Prisma Access, add the GlobalProtect VPN IP address pool for the VPN traffic that you want to monitor.**

1. To enable the Cortex XDR app to analyze your VPN traffic, add (+) a new segment and specify the first and last IP address of your GlobalProtect VPN IP address pool.
2. Leave the Pathfinder VM assignment blank for GlobalProtect VPN IP address pool network segments. The app creates virtual profiles of endpoints from VPN traffic from the username-associated traffic, and Pathfinder cannot scan those virtual profiles.
3. Identify this network segment as **Reserved for VPN.** GlobalProtect dynamically assigns IP addresses from the IP pool to the mobile endpoints that connect to your network. The Cortex XDR analytics engine creates virtual entity profiles for network segments that are reserved for VPN.
4. Save (✔) the network segment. If the Configuration saved notification does not appear, save again.

STEP 4 | After you have configured the analytics engine, wait about an hour, and then verify that Cortex XDR is receiving alerts on the various networks that the analytics engine is monitoring.

1. To view existing network segments, select > Analytics Management > Status and then select Analytics Network Coverage Status.
2. Select the report duration, or enter a custom date and time range, and click Generate.
3. Verify that the IP ranges match the network segments the firewall sees; the DNS % should be over 50. The DHCP % column should reflect the correct percentage for IP ranges that contain endpoints with dynamic IP addresses.
4. In a deployment with GlobalProtect or Prisma Access, verify that the app generates alerts on VPN traffic.

STEP 5 | If you want to use Pathfinder to interrogate endpoints for risky or malicious software, Set Up Pathfinder.

If you also use Cortex XDR Pro per Endpoint, you can use Pathfinder to supplement endpoint detection using the Cortex XDR agent.

STEP 6 | If you selected a Directory Sync Service instance during the Cortex XDR activation process, configure Cortex XDR to use it.

Ingest Logs and Alerts from External Sources

To provide you with a more complete and detailed picture of the activity involved in an incident, you can ingest logs and alerts from external sources in Cortex XDR.

Cortex XDR supports log and alert ingestion from the following external sources:
### Vendor | Device Type | Operating System Version | Format Requirements
---|---|---|---
Check Point | FW1/VPN1 | • R77.30  
• R80.10  
• R80.20  
• R80.30  
• R80.40 | CEF format using Log Exporter
Fortinet | Fortigate | 6.2.1 and above | timestamp must be in nanoseconds
Cisco | ASA | — | • Syslog in Cisco-ASA format  
• Must include timestamps  
• Only supports messages: 302013, 302014, 302015, 302016

To ingest alerts from external sources, you can use the Cortex XDR ingestion API or set up a syslog collector applet on a broker VM inside your network.

The applet functions as a syslog collector that can receive logs and alerts from your external sources. To enable Cortex XDR to display your alerts, you must also map your alert fields to the Cortex XDR field format. However, Cortex XDR automatically maps fields in alerts and logs from Check Point firewalls (no additional mapping is required).

- Set Up a Windows Event Collector
- Ingest External Alerts
- Ingest Logs from Check Point Firewalls
- Ingest Logs from Cisco ASA Firewalls
- Ingest Logs from Fortinet Fortigate Firewalls
- Ingest Authentication Logs and Data

### Ingest External Alerts

For a more complete and detailed picture of the activity involved in an incident, Cortex XDR can ingest alerts from any external source. Cortex XDR stitches the external alerts together with relevant endpoint data and displays alerts from external sources in relevant incidents and alerts tables. You can also see external alerts and related artifacts and assets in Causality views.

To ingest alerts from an external source, you configure your alert source to forward alerts (in CEF format) to the syslog collector. You can also ingest alerts from external sources using the Cortex XDR API.

After Cortex XDR begins receiving external alerts, you must map the following required fields to the Cortex XDR format:

- Timestamp
- Severity
- Source IP address
- Source port
- Destination IP address
• Destination port

*If you send pre-parsed alerts using the Cortex XDR API, additional mapping is not required.*

Storage of external alerts is determined by your Cortex Data Lake data retention policy.

To ingest external alerts:

**STEP 1 |** Send alerts from an external source to Cortex XDR.

There are two ways to send alerts:

- **Cortex XDR API**—Use the `insert_cef_alerts` API to send the raw CEF syslog alerts or use the `insert_parsed_alerts` API to convert the CEF syslog alerts to the Cortex XDR format before sending them to Cortex XDR. If you use the API to send logs, you do not need to perform the additional mapping step in Cortex XDR.
- **Activate Syslog collector**— Activate the syslog collector and then configure the alert source to forward alerts in CEF format to the syslog collector. Then configure an alert mapping rule as follows.

**STEP 2 |** In Cortex XDR, select > Settings > External Alerts.

**STEP 3 |** Right-click the Vendor Product for your alerts and select Filter and Map.

**STEP 4 |** Use the filters at the top of the table to narrow the results to only the alerts you want to map. Cortex XDR displays a limited sample of results during the mapping rule creation. As you define your filters, Cortex XDR applies the filter to the limited sample but does not apply the filters across all alerts. As a result, you might not see any results from the alert sample during the rule creation.

**STEP 5 |** Click Next to begin a new mapping rule.

1. On the left, define a Name and optional Description to identify your mapping rule.
2. Map each required Cortex XDR field to a field in your alert source.
If needed, use the field converter (flammatory) to translate the source field to the Cortex XDR syntax. For example, if you use a different severity system, you need to use the converter to map your severities fields to the Cortex XDR risks of High, Medium, and Low.

You can also use regex to convert the fields to extract the data to facilitate matching with the Cortex XDR format. For example, say you need to map the port but your source field contains both IP address and port (192.168.1.200:8080). To extract everything after the : , use the following regex:

```
^[^:]*$
```

For additional context when you are investigating an incident, you can also map additional optional fields to fields in your alert source.

STEP 6 | Submit your alert filter and mapping rule when finished.

*Ingest Logs from Check Point Firewalls*

If you use Check Point firewalls, you can still take advantage of Cortex XDR investigation and detection capabilities by forwarding your Check Point firewall logs to Cortex XDR. By forwarding firewall logs, Cortex XDR can examine your network traffic to detect anomalous behavior. Cortex XDR can use Check Point firewall logs as the sole data source, but can also use Check Point firewall logs in conjunction with Palo Alto Networks firewall logs. For additional endpoint context, you can also use Traps to collect and alert on endpoint data.

As an estimate for initial sizing, note that the average Check Point log size is roughly 700 bytes. For proper sizing calculations, test the log sizes and log rates produced by your Check Point firewalls.
As soon as Cortex XDR starts to receive logs, the app can begin analyzing and raising Analytics alerts. Cortex XDR stores Analytics alerts according to your Cortex Data Lake storage retention policy but does not store the Check Point firewall logs. As a result, you cannot query or apply IOC and BIOC rule matching to Check Point firewall logs.

To integrate your logs, you first need to set up an applet in a broker VM within your network to act as a syslog collector. You then configure firewall policy to log all traffic and set up the Log Exporter on your Check Point Log Server to forward logs to the syslog collector in a CEF format.

**STEP 1 | Activate the Syslog Collector.**

**STEP 2 | Configure the Check Point firewall to forward syslog events in CEF format to the syslog collector.**

Configure your firewall policy to log all traffic and set up the Log Exporter to forward logs to the syslog collector. By logging all traffic, you enable Cortex XDR to detect anomalous behavior from Check Point firewall logs. For more information on setting up Log Exporter, see the Check Point documentation.

**Ingest Logs from Cisco ASA Firewalls**

If you use Cisco ASA firewalls, you can still take advantage of Cortex XDR investigation and detection capabilities by forwarding your firewall logs to Cortex XDR. This enables Cortex XDR to examine your network traffic to detect anomalous behavior. Cortex XDR can use Cisco ASA firewall logs as the sole data source, but can also use Cisco ASA firewall logs in conjunction with Palo Alto Networks firewall logs. For additional endpoint context, you can also use Cortex XDR to collect and alert on endpoint data.

As an estimate for initial sizing, note that the average Cisco ASA log size is roughly 180 bytes. For proper sizing calculations, test the log sizes and log rates produced by your Cisco ASA firewalls.

As soon as Cortex XDR starts to receive logs, the app can begin analyzing and raising Analytics alerts. Cortex XDR stores Analytics alerts according to your Cortex Data Lake storage retention policy but does not store the Cisco ASA firewall logs. As a result, you cannot query or apply IOC and BIOC rule matching to Cisco ASA firewall logs.

To integrate your logs, you first need to set up an applet in a broker VM within your network to act as a syslog collector. You then configure forwarding on your log devices to send logs to the syslog collector.

**STEP 1 | Activate the Syslog Collector.**

**STEP 2 | Configure the Cisco ASA firewall or the log device forwarding logs from it to log to the syslog collector.**

Configure your firewall policy to log all traffic and forward the traffic logs to the syslog collector. By logging all traffic, you enable Cortex XDR to detect anomalous behavior from Cisco ASA firewall logs. For more information on setting up Log Forwarding on Cisco ASA firewalls, see the Cisco ASA Series documentation.

**Ingest Logs from Fortinet Fortigate Firewalls**

If you use Fortinet Fortigate firewalls, you can still take advantage of Cortex XDR investigation and detection capabilities by forwarding your firewall logs to Cortex XDR. This enables Cortex XDR to examine your network traffic to detect anomalous behavior. Cortex XDR can use Fortinet Fortigate firewall logs as the sole data source, but can also use Fortinet Fortigate firewall logs in conjunction with Palo Alto Networks firewall logs. For additional endpoint context, you can also use Cortex XDR to collect and alert on endpoint data.
As an estimate for initial sizing, note that the average Fortinet Fortigate log size is roughly 1,070 bytes. For proper sizing calculations, test the log sizes and log rates produced by your Fortinet Fortigate firewalls.

As soon as Cortex XDR starts to receive logs, the app can begin analyzing and raising Analytics alerts. Cortex XDR stores Analytics alerts according to your Cortex Data Lake storage retention policy but does not store the Fortinet Fortigate firewall logs. As a result, you cannot query or apply IOC and BIOC rule matching to Fortinet Fortigate firewall logs.

To integrate your logs, you first need to set up an applet in a broker VM within your network to act as a syslog collector. You then configure forwarding on your log devices to send logs to the syslog collector.

**STEP 1 | Activate the Syslog Collector.**

**STEP 2 | Configure the log device that receives Fortinet Fortigate firewall logs to forward syslog events to the syslog collector.**

Configure your firewall policy to log all traffic and forward the traffic logs to the syslog collector. By logging all traffic, you enable Cortex XDR to detect anomalous behavior from Fortinet Fortigate firewall logs. For more information on setting up Log Forwarding on Fortinet Fortigate firewalls, see the Fortinet FortiOS documentation.

**Ingest Authentication Logs and Data**

*Ingesting Authentication Logs and Data requires a Cortex XDR Pro per TB license.*

Cortex XDR can ingest authentication logs and data from both Okta and Azure Active Directory (AD) into authentication stories. An authentication story unites logs and data regardless of the information source (for example, from an on-premise KDC or from a cloud-based authentication service) into a uniform schema. To search authentication stories, you can use the Query Builder or Native Search.

To receive authentication logs from Okta or Azure AD, you must first configure the SaaS Log Collection settings in Cortex XDR. After you set up log collection, Cortex XDR immediately begins receiving new authentication logs and data from the source.

To ingest logs from Azure AD, you must have a Premium 1 or Premium 2 license and create an app for Cortex XDR with the following API permissions from the Microsoft Azure Console: AuditLog.ReadAll and Directory.ReadAll. For more information on Microsoft Azure, see the following instructions on the Microsoft documentation portal:

- Register an app: https://docs.microsoft.com/en-us/azure/active-directory/develop/quickstart-register-app
- Add API permissions for Directory.Read.All and AuditLog.Read.All with type Application: https://docs.microsoft.com/en-us/azure/active-directory/develop/quickstart-configure-app-access-web-apis#add-permissions-to-access-web-apis
- Create an application secret: https://docs.microsoft.com/en-us/azure/active-directory/develop/howto-create-service-principal-portal#create-a-new-application-secret

**STEP 1 | Select ➔ Settings ➔ SaaS Log Collection.**

**STEP 2 | Set up the authentication service you want to integrate with Cortex XDR.**

Okta:
1. Enter the domain name of your Okta service.
2. Enter the token Cortex XDR can use to authenticate with Okta.
3. Test the connection settings.
4. If successful, Enable Okta log collection.

**Azure AD:**
1. Enter the **Tenant Domain** of your Microsoft Azure AD tenant.
2. Enter the **Application Client ID** and **Secret** for your Microsoft Azure AD service.
   
   You can obtain these values from the Microsoft Azure Console. These values enable Cortex XDR to authenticate with your Azure AD service.
3. Test the connection settings.
4. If successful, Enable Azure AD log collection.

**STEP 3 |** Create an Authentication Query or Native Search to search for specific authentication logs or data.
Set Up Pathfinder

Pathfinder™ is a highly recommended, but optional, component that Cortex XDR™ uses to examine network hosts, servers, and workstations for malicious or risky software. When paired with Pathfinder, Cortex XDR supports all the Analytics alerts described in the Cortex XDR Analytics Alert Reference.

To enable Pathfinder to investigate your network endpoints, you must install one or more Pathfinder virtual machines (VMs) on your network. The Pathfinder VMs use Remote Procedure Calls (RPCs) to examine endpoints, so that you don’t need to locally install kernel drivers or other software agents on each host. The steps to set up Pathfinder include deploying the Pathfinder VM and pairing Pathfinder with the app.

Before you set up Pathfinder, enable communication to the Broker VM URLs and the following:

<table>
<thead>
<tr>
<th>FQDN, Protocol, and Port</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>• pathfinder.xdr.us.paloaltonetworks.com • pathfinder.xdr.eu.paloaltonetworks.com</td>
<td>HTTPS over TCP port 443 Pathfinder server depending on the region of your deployment.</td>
</tr>
<tr>
<td>• lic.lc.prod.us.cs.paloaltonetworks.com • lic2.lc.prod.us.cs.paloaltonetworks.com • api3.lcaas.beta.ap.paloaltonetworks.com</td>
<td>HTTPS over TCP port 444 For passing Cortex XDR Analytics ID and receiving FQDN.</td>
</tr>
<tr>
<td>• wildfire.paloaltonetworks.com • eu.wildfire.paloaltonetworks.com • jp.wildfire.paloaltonetworks.com</td>
<td>HTTPS over TCP port 443 WildFire URL is depending on the region where Cortex XDR Analytics is provisioned.</td>
</tr>
</tbody>
</table>

**STEP 1** | Start by confirming that you have everything you need to set up the Pathfinder VM.

Prerequisites that are specific to a Cortex XDR deployment with Pathfinder include:

- Hardware to support Pathfinder virtual machine (2 core, 8 GB RAM, 128 GB disk).
- VMware ESXi or Hyper-V.
- An internal DNS server.
- Pathfinder requires Local Administrator permissions for all endpoints. For more information, see this Microsoft procedure.
- Pathfinder requires the following ports to be open for communication with the devices it scans:
  - RPC Endpoint Mapper (port 135)
  - NetBIOS over TCP/IP Name Services (port 137)
  - NetBIOS over TCP/IP Session Services (port 139)
  - SMB over TCP/IP (port 445)
  - Pathfinder requires port 443 to be open to the internet to communicate with the Cortex XDR app.
  - Pathfinder requires port 444 to be open to the internet. It uses the FQDNs on port 444 to perform query and validity checks as part of the process to pair with the Cortex XDR app.
  - All devices that Pathfinder scans must provide the following services:
    - WMI Service
• Eventlog Service
• PowerShell

STEP 2 | Download the latest Pathfinder software from the Palo Alto Networks Software Updates page and deploy the Pathfinder VMware or Hyper-V virtual machine (VM).

If you deploy a Hyper-V VM, follow the wizard, and specify generation 1. For both virtual machines, make sure the VM has at least 8 GBs of startup memory. Do not use dynamic memory and select the same subnet connection as Pathfinder.

STEP 3 | Ensure the Pathfinder VM clock is correctly synchronized.

Clock synchronization is required both to pair Pathfinder with Cortex XDR, and for proper network-to-process association (N2PA).

By default, the Pathfinder VM performs time synchronization using pre-configured NTP servers: rolex.usg.edu, ntp2.netwrx1.com, and 0.north-america.pool.ntp.org. You can configure your NTP servers to change the default settings, or to enable a VMware ESX host to perform time synchronization.

STEP 4 | Open the Pathfinder VM Console.

STEP 5 | Configure Pathfinder to use an internal DNS server.

1. Select netconfig.
2. Choose the network card that the Pathfinder VM is using to connect to your network.
3. Choose either DHCP or Static. If you choose DHCP then your DHCP server must be configured to use an internal DNS server. In this case, accept the configuration change and you’re done.

If you choose Static, then in the Networking panel enter the IP Address, Netmask, Default gateway, and DNS server you want Pathfinder to use. Then, select OK to save your changes.

STEP 6 | Return to the top-level Pathfinder VM console screen, and select the pair menu. Record the Pathfinder VM ID (you will use this in the next step).

STEP 7 | Generate the token you will use to pair the Pathfinder VM with the Cortex XDR app.
1. In Cortex XDR, click > Analytics Management > Configuration.
2. Select the Pathfinder VMs page.
3. Enter the Pathfinder VM ID that was displayed in the Pathfinder VM console in the UUID field.
4. Click Generate Pairing Token.
5. Record the Pairing Token and the tenant ID to use them in the next step.

STEP 8 | Connect the Pathfinder VM to the Cortex XDR app.

1. Open the Pathfinder VM console and select the pair menu.

2. Give the Pathfinder VM a descriptive name: Pathfinder VM Name.
3. Enter the Pairing Token and the tenant ID that you recorded from the Cortex XDR app in the last step.
4. Click OK.

The pairing might take a few moments, after which you’ll be prompted to continue the pairing process in the Cortex XDR app.

STEP 9 | Authorize the Cortex XDR app and Pathfinder VM pairing.

1. In the Cortex XDR app, click > Analytics Management > Configuration.
2. Select the Pathfinder VMs page.
3. Verify that the Pathfinder VM that you just connected to the Cortex XDR app is displayed, and Authorize the Pathfinder VM and Cortex XDR app pairing.
4. Wait until the Pathfinder VM status displays Connected.

STEP 10 | Configure the credentials for Pathfinder to use to authenticate to the devices it examines.
To configure these credentials locally on the Pathfinder VM, navigate to the Pathfinder Console menu, and select **Credentials**. Otherwise, you can configure them in the Cortex XDR app:

1. In the Cortex XDR app, click **Analytics Management > Configuration**.
2. Select **Pathfinder**.
3. Select **Default Configuration** and enter the login credentials that Pathfinder should use to access your Microsoft Windows endpoints for interrogation. You can also use this page to indicate whether you want Pathfinder to automatically scan workstations and/or servers, and whether you want N2PA enabled.

**STEP 11** | Enable the Pathfinder VM to scan your network devices or limit the Pathfinder VM to scan certain network ranges or specific devices.

It's recommended to assign the Pathfinder VM to scan all network ranges; however, you can assign different Pathfinder VMs to scan different network ranges or in certain environments, like a lab environment, you can choose to limit Pathfinder scans to certain devices.

1. In the Cortex XDR app, click **Analytics Management > Configuration**.
2. Select **Network Segments** and, if you haven't done so already, configure the IP ranges (network assets) that Cortex XDR monitors.
3. On the Network Segments page, use the final column (**Assigned Pathfinder VM**) for each table row to assign a Pathfinder VM to the network segment. If you do not want Pathfinder to scan a particular segment, then do not identify a Pathfinder VM for that segment's table row.
4. *(Optional)* If you want to further limit Pathfinder scans to specific devices, go to the **Pathfinder** page and then select **Per Asset Configuration**. Use these settings to override the default Pathfinder configuration on a per-asset basis.
Configure XDR

Before you can begin using Cortex XDR, you must set up your alert sensors. The more sensors that you integrate with Cortex XDR, the more context you have when a threat is detected. You can also set up Cortex XDR to raise Analytics alerts on network or endpoint data (or both) depending or your Cortex XDR Pro licenses.

The following workflow highlights the tasks that you must perform (in order) to configure Cortex XDR.

**STEP 1 | Integrate External Threat Intelligence Services.**

Integrating external threat intelligence services enables you to view feeds from sources such as AutoFocus and VirusTotal in the context of your incident investigation.

**STEP 2 | After you activate Cortex XDR apps and services, wait 24 hours and then configure the Cortex XDR analytics.**

1. Specify the internal networks that you want Cortex XDR to monitor.
   1. Log in to your Cortex XDR app either using the direct link or from the Cortex XDR tile on the hub.
   2. To view existing network segments, select the gear (⚙️) in the upper right corner and select **Analytics Management > Status > Analytics Network Coverage Status.** This page provides a table of the IP address ranges Cortex XDR Analytics monitors, which is pre-populated with the default IPv4 and IPv6 address spaces.
   3. To add custom network segments, select **Configuration** and then **Network Segments Configuration.**
   4. Add (+) a new segment and enter the first and last IP address of the range to monitor.
   5. Specify the **Assigned Pathfinder VM** to assign a Pathfinder VM to the network segment. If you do not want Pathfinder to scan a particular segment, then leave the field blank.
   6. (Optional) If you want to further limit Pathfinder scans to specific devices, go to the **Pathfinder** page and then select **Per Asset Configuration.** Use these settings to override the default Pathfinder configuration on a per-asset basis.
   7. Leave **Reserved for VPN** blank.
   8. Save (✔️) the network segment. If the Configuration saved notification does not appear, save again.

2. (Recommended) If you want to use Pathfinder to supplement the Cortex XDR agent, **Set Up Pathfinder.**

3. Activate Cortex XDR - Analytics.

By default, Cortex XDR - Analytics is disabled. Activating Cortex XDR - Analytics enables the Cortex XDR analytics engine to analyze your endpoint data to develop a baseline and raise Analytics and Analytics BIOC alerts when anomalies and malicious behaviors are detected. To create a baseline, Cortex XDR requires a minimum set of data. To satisfy the requirement you must have either EDR logs from a minimum of 30 endpoints or 675MB of network traffic logs from your Palo Alto Networks firewalls in the last 24 hours.

1. In Cortex XDR, select the gear (⚙️) in the upper right corner and then select **Settings > Cortex XDR - Analytics.**

   The **Enable** option will be grayed out if you do not have the required data set.
2. When available, **Enable** Cortex XDR - Analytics. The analytics engine will immediately begin analyzing your Cortex data for anomalies.

**STEP 3 | Add an Alert Exclusion Policy.**

**STEP 4 | Create an Incident Starring Configuration.**

**STEP 5 | (Optional)** Palo Alto Networks also automatically delivers behavioral indicators of compromise (BIOCs) rules defined by the Palo Alto Networks threat research team to all Cortex XDR tenants, but you can also import any additional indicators as rules, as needed.

To alert on specific BIOCs, **Create a BIOC Rule.** To immediately being alerting on known malicious indicators of compromise (IOCs)—such as known malicious IP addresses—**Create an IOC Rule.**

**Integrate External Threat Intelligence Services**

To aid you with threat investigation, Cortex XDR displays the WildFire-issued verdict for each Key Artifact in an incident. To provide additional verification sources, you can integrate an external threat intelligence service with Cortex XDR. The threat intelligence services the app supports are:

- **AutoFocus™**—AutoFocus groups conditions and indicators related to a threat with a tag. Tags can be user-defined or come from threat-research team publications and are divided into classes, such as exploit, malware family, and malicious behavior. When you add the service, the relevant tags display in the incident details page under **Key Artifacts.** Without an AutoFocus license key, you can still pivot from Cortex XDR to the service to initiate a query for the artifact. See the **AutoFocus Administrator’s Guide** for more information on AutoFocus tags.

- **VirusTotal**—VirusTotal provides aggregated results from over 70 antivirus scanners, domain blacklist services, and user contributions. The VirusTotal score is represented as a fraction, where, for example, a score of 34/52 means out of 52 queried services, 34 services determined the artifact to be malicious. When you add the service, the relevant VirusTotal score displays in the incident details page under **Key Artifacts.** Without a VirusTotal license key, you can still pivot from Cortex XDR to the service to initiate a query for the artifact.

- **WildFire®**—WildFire detects known and unknown threats, such as malware. The WildFire verdict contains detailed insights into the behavior of identified threats. The WildFire verdict displays next to relevant **Key Artifacts** in the incidents details page, the causality view, and within the Live Terminal view of processes.

  *WildFire provides verdicts and analysis reports to Cortex XDR users without requiring a license key. Using WildFire for next-generation firewalls or other use-cases continues to require an active license.*
Before you can view external threat intelligence in Cortex XDR incidents, you must obtain the license key for the service and add it to the Cortex XDR Configuration. After you integrate any services, you will see the verdict or verdict score when you Investigate Incidents.

To integrate an external threat intelligence service:

STEP 1 | Get your the API License Key for the service.
- Get your AutoFocus API key.
- Get your VirusTotal API key.

STEP 2 | Enter the license key in the Cortex XDR app.

Select the gear in the menu bar, then Settings > Threat Intelligence and then enter the license key.

STEP 3 | Test your license key.

Select Test. If there is an issue, an error message provides more details.

STEP 4 | Verify the service integration in an incident.

After adding the license key, you should see the additional verdict information from the service included in the Key Artifacts of an incident. You can right-click the service, such as VirusTotal (VT) or AutoFocus (AF), to see the entire verdict. See Investigate Incidents for more information on where these services are used within the Cortex XDR app.
Set up Outbound Integration

With Cortex XDR, you can set up any of the following optional outbound integrations:

- **Integrate Slack for Outbound Notifications**
- **Integrate a Syslog Receiver**
- **Integrate with Cortex XSOAR**—Send alerts to Cortex XSOAR for automated and coordinated threat response. From Cortex XSOAR, you define, adjust, and test playbooks that respond to Cortex XDR alerts. You can also manage your incidents in Cortex XSOAR with any changes automatically synced to Cortex XDR. For more information, see the in-app documentation in Cortex XSOAR.
- **Integrate with external receivers such as ticketing systems**—To manage incidents from the application of your choice, you can use the Cortex XDR API Reference to send alerts and alert details to an external receiver. After you generate your API key and set up the API to query Cortex XDR, external apps can receive incident updates, request additional data about incidents, and make changes such as to set the status and change the severity, or assign an owner. To get started, see the Cortex XDR API Reference.
Use the Cortex XDR Interface

Cortex XDR provides an easy-to-use interface that you can access from the hub. By default, Cortex XDR displays the Incident Management Dashboard when you log in. If desired, you can change the default dashboard or Build a Custom Dashboard that displays when you log in.

Depending on your license and assigned role, you can explore and the following areas in the app.

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reporting</td>
<td>From this menu, you can manage your dashboards and run reports.</td>
</tr>
<tr>
<td>Investigation</td>
<td>From this menu you can investigate a lead or hunt for threats. You can access the <strong>Query Builder</strong> to search logs from your Palo Alto Networks sensors, or the <strong>Query Center</strong> to view the status of all queries, and <strong>Scheduled Queries</strong> to view the status and modify the frequency of reoccurring queries. You can also view all incidents, prioritize incidents, and set alert exceptions.</td>
</tr>
<tr>
<td>Response</td>
<td>From this menu, you can respond to identified threats and take action. With a Cortex XDR Prevent or Cortex XDR Pro per Endpoint license, you can view the Action Center where you can initiate investigation and response actions such as isolating an endpoint or initiating a live terminal session to investigate processes and files locally. From this menu, you can also add malicious domains and IP addresses to an external dynamic list (EDL) enforceable on your Palo Alto Networks firewall. EDL management requires a Cortex XDR Pro per TB license.</td>
</tr>
<tr>
<td>Endpoints</td>
<td>With a Cortex XDR Prevent or Cortex XDR Pro per Endpoint license, you can manage your endpoints and endpoint security policy from this menu.</td>
</tr>
<tr>
<td>Security</td>
<td>From this menu, you can configure additional add-on security services such as Device Control. Device Control requires a Cortex XDR Prevent or Cortex XDR Pro per Endpoint license.</td>
</tr>
<tr>
<td>Rules</td>
<td>With a Cortex XDR Pro per TB license, you can define indicators of known threats to enable Cortex XDR to raise alerts when detected. As you investigate and research threats and uncover specific indicators and behaviors associated with</td>
</tr>
</tbody>
</table>
Interface | Description
--- | ---
|  | a threat, you can create rules to detect and alert you when the behavior occurs.

Settings and management | From the gear icon, you can view a log of actions initiated by Cortex XDR analysts, configure Cortex XDR settings to integrate with other apps and services, and manage settings for the analytics engine.

Notifications | View Cortex XDR notifications such as when a query completes.

User | User who is logged into the Cortex XDR app and additional information about the app.

The following topics describe additional management actions you can perform on page results:

- Filter Page Results
- Save and Share Filters
- Show or Hide Results
- Manage Columns and Rows

Manage Tables

Most pages in Cortex XDR present data in table format and provide controls to help you manage and filter the results. If additional views or actions are available for a specific value, you can pivot (right-click) from the value in the table. For example, you can view the incident details, or pivot to the Causality View for an alert or you can pivot to the results for a query.

On most pages, you can also refresh (⟳) the content on the page.

To manage tables in the app:

- Filter Page Results
- Export Results to File
- Save and Share Filters
- Show or Hide Results
- Manage Columns and Rows

Filter Page Results

To reduce the number of results, you can filter by any heading and value. When you apply a filter, Cortex XDR displays the filter criteria above the results table. You can also filter individual columns for specific values using the icon to the right of the column heading.

Some fields also support additional operators such as =, !=, Contains, not Contains, *, !*.

There are three ways you can filter results:

- By column using the filter next to a field heading
- By building a filter query for one or more fields using the filter builder
- By pivoting from the contents of a cell (show or hide rows containing)
Filters are persistent. When you navigate away from the page and return, any filter you added remain active.

To build a filter using one or more fields:

**STEP 1** | From a Cortex XDR page, select **Filter**.

Cortex XDR adds the filter criteria above the top of the table. For example, on the filter page:

**STEP 2** | For each field you want to filter:

1. Select or search the field.
2. Select the operator by which to match the criteria.
   
   In most cases this will be = to include results that match the value you specify, or != to exclude results that match the value.
3. Enter a value to complete the filter criteria.

   CMD fields have a 128 character limit. Shorten longer query strings to 127 characters and add an asterisk (*).

   Alternatively, you can select **Include empty values** to create a filter that excludes or includes results when the field has an empty values.

**STEP 3** | To add additional filters, click +AND (within the filter brackets) to display results that must match all specified criteria, or +OR to display results that match any of the criteria.

**STEP 4** | Click out of the filter area into the results table to see the results.

**STEP 5** | Next steps:

- If at any time you want to remove the filter, click the X next to it. To remove all filters, click the trash icon.
- **Save and Share Filters**.

**Export Results to File**

If needed, you can export the page results for most pages in Cortex XDR to a tab separated values (TSV) file.

**STEP 1** | (Optional) **Filter Page Results** to reduce the number of results for export.

**STEP 2** | To the left of the refresh icon (○), **Export to file**.

Cortex XDR exports any results matching your applied filters in TSV format.
Save and Share Filters

You can save and share filters across your organization.

- **Save a filter:**
  
  Saved filters are listed on the Filters tab for the table layout and filter manager menu.
  
  1. Save ( ) the active filter.
  2. Enter a name to identify the filter.
  3. Choose whether to Share this filter or whether to keep it private for your own use only.

- **Share a filter:**
  
  You can share a filter across your organization.
  
  1. Select the table layout and filter menu indicated by the three vertical dots, then select Filters.
  2. Select the filter to share and click the share icon.
  3. If needed, you can later unshare ( ) or delete ( ) a filter.

  Unsharing a filter will turn a public filter private. Deleting a shared filter will remove it for all users.

Show or Hide Results

As an alternative to building a filter query from scratch or using the column filters, you can pivot from rows and specific values to define the match criteria to fine tune the results in the table. You can also pivot on empty values to show only results with empty values or only results that do not have empty values in the column from which you pivot.

*CMD fields are limited to 128 characters. If you pivot on a CMD field with a truncated value, the app shows or hides all results that match the first 128 characters.*

The show or hide action is a temporary means of filtering the results: If you navigate away from the page and later return, any results you previously hid will appear again.

This option is available for fields which have a finite list of options.

To hide or show only results that match a specific field value:

**STEP 1 |** Right-click the matching field value by which you want to hide or show.

**STEP 2 |** Select the desired action:
• Hide rows with `<field value>`
• Show rows with `<field value>`
• Hide empty rows
• Show empty rows

**Manage Columns and Rows**

From Cortex XDR pages, you can manage how you want to view the results table and what information you want XDR app to display.

• Adjust the row height
• Adjust the column width
• Add or remove fields in the table

Any adjustments you make to the columns or rows persist when you navigate away from and later return to the page.

• Adjust the row height:
  1. On the Cortex XDR page select the menu indicated by three vertical dots to the right of the Filter button.
  2. Select the desired **ROW VIEW** option.

Cortex XDR updates the table to present the results in the desired row height view ranging from short to tall.

• Adjust the column width:
  1. On the Cortex XDR page, select the menu indicated by three vertical dots to the right of the Filter button.
  2. Select the desired column width option from the **COLUMN MANAGER**.
Cortex XDR updates the table to present the results in the desired view: narrow, fixed width, or scaled to the column heading.

- **Add or remove fields in the table:**
  1. On an Cortex XDR page, select the menu indicated by three vertical dots to the right of the **Filter** button.
  2. Below the column manager, search for a column by name, or select the fields you want to add or clear any fields you want to hide.
    
    Cortex XDR adds or removes the fields to the table as you select or clear the fields.
  3. If desired, drag and drop the fields to change the order in which they appear in the table.
Endpoint Security

Endpoint security features require a Cortex XDR Pro - Endpoint license.

> Endpoint Security Concepts
> Manage Cortex XDR Agents
> Define Endpoint Groups
> About Content Updates
> Endpoint Security Profiles
> Customizable Agent Settings
> Apply Security Profiles to Endpoints
> Exceptions Security Profiles
> Hardened Endpoint Security
Endpoint Security Concepts

- About Cortex XDR Endpoint Protection
- File Analysis and Protection Flow
- Endpoint Protection Capabilities
- Endpoint Protection Modules

About Cortex XDR Endpoint Protection

Cyberattacks are attacks performed on networks or endpoints to inflict damage, steal information, or achieve other goals that involve taking control of computer systems that do not belong to the attackers. These adversaries perpetrate cyberattacks either by causing a user to unintentionally run a malicious executable file, known as malware, or by exploiting a weakness in a legitimate executable file to run malicious code behind the scenes without the knowledge of the user.

One way to prevent these attacks is to identify executable files, dynamic-link libraries (DLLs), and other pieces of code to determine if they are malicious and, if so, to prevent them from executing by testing each potentially dangerous code module against a list of specific, known threat signatures. The weakness of this method is that it is time-consuming for signature-based antivirus (AV) solutions to identify newly created threats that are known only to the attacker (also known as zero-day attacks or exploits) and add them to the lists of known threats, which leaves endpoints vulnerable until signatures are updated.

Cortex XDR takes a more efficient and effective approach to preventing attacks that eliminates the need for traditional AV. Rather than try to keep up with the ever-growing list of known threats, Cortex XDR sets up a series of roadblocks—traps, if you will—that prevent the attacks at their initial entry points—the point where legitimate executable files are about to unknowingly allow malicious access to the system.

Cortex XDR provides a multi-method protection solution with exploit protection modules that target software vulnerabilities in processes that open non-executable files and malware protection modules that examine executable files, DLLs, and macros for malicious signatures and behavior. Using this multi-method approach, the Cortex XDR solution can prevent all types of attacks, whether they are known or unknown threats.
Exploit Protection Overview

An exploit is a sequence of commands that takes advantage of a bug or vulnerability in a software application or process. Attackers use these exploits to access and use a system to their advantage. To gain control of a system, the attacker must exploit a chain of vulnerabilities in the system. Blocking any attempt to exploit a vulnerability in the chain will block the entire exploitation attempt.

To combat an attack in which an attacker takes advantage of a software exploit or vulnerability, Cortex XDR employs exploit protection modules (EPMs). Each EPM targets a specific type of exploit attack in the attack chain. Some capabilities that Cortex XDR EPMs provide are reconnaissance prevention, memory corruption prevention, code execution prevention, and kernel protection.

Malware Protection Overview

Malicious files, known as malware, are often disguised as or embedded in non-malicious files. These files can attempt to gain control, gather sensitive information, or disrupt the normal operations of the system. Cortex XDR prevents malware by employing the Malware Prevention Engine. This approach combines several layers of protection to prevent both known and unknown malware that has not been seen before from causing harm to your endpoints. The mitigation techniques that the Malware Prevention Engine employs vary by the endpoint type:

- **Malware Protection for Windows**
- **Malware Protection for Mac**
- **Malware Protection for Linux**
- **Malware Protection for Android**

**Malware Protection for Windows**

- **WildFire integration**—Enables automatic detection of known malware and analysis of unknown malware using WildFire threat intelligence.
- **Local static analysis**—Enables Cortex XDR to use machine learning to analyze unknown files and issue a verdict. Cortex XDR uses the verdict returned by the local analysis module until it receives a verdict from Cortex XDR.
- **DLL file protection**—Enables Cortex XDR to block known and unknown DLLs on Windows endpoints.
- **Office file protection**—Enables Cortex XDR to block known and unknown macros when run from Microsoft Office files on Windows endpoints.
- **Behavioral threat protection** *(Windows 7 SP1 and later versions)*—Enables continuous monitoring of endpoint activity to identify and analyze chains of events—known as *causality chains*. This enables Cortex XDR to detect malicious activity that could otherwise appear legitimate if inspected as individual events. Behavioral threat protection requires Traps agent 6.0 or a later release.
- **Evaluation of trusted signers**—Permits unknown files that are signed by highly trusted signers to run on the endpoint.
- **Malware protection modules**—Targets behaviors—such as those associated with ransomware—and enables you to block the creation of child processes.
- **Policy-based restrictions**—Enables you to block files from executing from within specific local folders, network folders, or external media locations.
- **Periodic and automated scanning**—Enables you to block dormant malware that has not yet tried to execute on endpoints.

**Malware Protection for Mac**

- **WildFire integration**—Enables automatic detection of known malware and analysis of unknown malware using WildFire threat intelligence.
- **Local static analysis**—Enables Cortex XDR to use machine learning to analyze unknown files and issue a verdict. The Cortex XDR agent uses the verdict returned by the local analysis module until it receives the WildFire verdict from Cortex XDR.
- **Behavioral threat protection**—Enables continuous monitoring of endpoint activity to identify and analyze chains of events—known as *causality chains*. This enables the Cortex XDR agent to detect malicious activity that could otherwise appear legitimate if inspected as individual events. Behavioral threat protection requires Traps agent 6.1 or a later release.
- **Mach-O file protection**—Enables you to block known malicious and unknown mach-o files on Mac endpoints.
- **Evaluation of trusted signers**—Permits unknown files that are signed by trusted signers to run on the endpoint.
- **Periodic and automated scanning**—Enables you to block dormant malware that has not yet tried to execute on endpoints. Scanning requires Cortex XDR agent 7.1 or a later release.

**Malware Protection for Linux**

- **WildFire integration**—Enables automatic detection of known malware and analysis of unknown malware using WildFire threat intelligence. WildFire integration requires Traps agent 6.0 or a later release.
- **Local static analysis**—Enables the Cortex XDR agent to use machine learning to analyze unknown files and issue a verdict. The Cortex XDR agent uses the verdict returned by the local analysis module until it receives the WildFire verdict from Cortex XDR. Local analysis requires Traps agent 6.0 or a later release.
- **Behavioral threat protection**—Enables continuous monitoring of endpoint activity to identify and analyze chains of events—known as *causality chains*. This enables Cortex XDR to detect malicious activity that could otherwise appear legitimate if inspected as individual events. Behavioral threat protection requires Traps agent 6.1 or a later release.
- **ELF file protection**—Enables you to block known malicious and unknown ELF files executed on a host server or within a container on a Cortex XDR-protected endpoint. Cortex XDR automatically suspends the file execution until a WildFire or local analysis verdict is obtained. ELF file protection requires Traps agent 6.0 or a later release.
- **Malware protection modules**—Targets the execution behavior of a file—such as those associated with reverse shell protection.
Malware Protection for Android

- **WildFire integration**—Enables automatic detection of known malware and grayware, and analysis of unknown APK files using WildFire threat intelligence.
- **APK files examination**—Analyze and prevent malicious APK files from running.
- **Evaluation of trusted signers**—Permits unknown files that are signed by trusted signers to run on the Android device.

File Analysis and Protection Flow

The Cortex XDR agent utilizes advanced multi-method protection and prevention techniques to protect your endpoints from both known and unknown malware and software exploits.

Exploit Protection for Protected Processes

In a typical attack scenario, an attacker attempts to gain control of a system by first corrupting or bypassing memory allocation or handlers. Using memory-corruption techniques, such as buffer overflows and heap corruption, a hacker can trigger a bug in software or exploit a vulnerability in a process. The attacker must then manipulate a program to run code provided or specified by the attacker while evading detection. If the attacker gains access to the operating system, the attacker can then upload malware, such as Trojan horses (programs that contain malicious executable files), or can otherwise use the system to their advantage. The Cortex XDR agent prevents such exploit attempts by employing roadblocks—or traps—at each stage of an exploitation attempt.

When a user opens a non-executable file, such as a PDF or Word document, and the process that opened the file is protected, the Cortex XDR agent seamlessly injects code into the software. This occurs at the earliest possible stage before any files belonging to the process are loaded into memory. The Cortex XDR agent then activates one or more protection modules inside the protected process. Each protection module targets a specific exploitation technique and is designed to prevent attacks on program vulnerabilities based on memory corruption or logic flaws.

In addition to automatically protecting processes from such attacks, the Cortex XDR agent reports any security events to Cortex XDR and performs additional actions as defined in the endpoint security policy. Common actions that the Cortex XDR agent performs include collecting forensic data and notifying the user about the event.

The default endpoint security policy protects the most vulnerable and most commonly used applications but you can also add other third-party and proprietary applications to the list of protected processes.
Malware Protection

The Cortex XDR agent provides malware protection in a series of four evaluation phases:

**Phase 1: Evaluation of Child Process Protection Policy**

When a user attempts to run an executable, the operating system attempts to run the executable as a process. If the process tries to launch any child processes, the Cortex XDR agent first evaluates the child process protection policy. If the parent process is a known targeted process that attempts to launch a restricted child process, the Cortex XDR agent blocks the child processes from running and reports the security event to Cortex XDR. For example, if a user tries to open a Microsoft Word document (using the winword.exe process) and that document has a macro that tries to run a blocked child process (such as WScript), the Cortex XDR agent blocks the child process and reports the event to Cortex XDR. If the parent process does not try to launch any child processes or tries to launch a child process that is not restricted, the Cortex XDR agent next moves to **Phase 2: Evaluation of the Restriction Policy**.

**Phase 2: Evaluation of the Restriction Policy**

When a user or machine attempts to open an executable file, the Cortex XDR agent first evaluates the child process protection policy as described in **Phase 1: Evaluation of Child Process Protection Policy**. The Cortex XDR agent next verifies that the executable file does not violate any restriction rules. For example, you might have a restriction rule that blocks executable files launched from network locations. If a restriction rule applies to an executable file, the Cortex XDR agent blocks the file from executing and reports the security event to Cortex XDR and, depending on the configuration of each restriction rule, the Cortex XDR agent can also notify the user about the prevention event.

If no restriction rules apply to an executable file, the Cortex XDR agent next moves to **Phase 3: Evaluation of Hash Verdicts**.

**Phase 3: Hash Verdict Determination**

The Cortex XDR agent calculates a unique hash using the SHA-256 algorithm for every file that attempts to run on the endpoint. Depending on the features that you enable, the Cortex XDR agent performs additional analysis to determine whether an unknown file is malicious or benign. The Cortex XDR agent can also submit unknown files to Cortex XDR for in-depth analysis by WildFire.

To determine a verdict for a file, the Cortex XDR agent evaluates the file in the following order:
1. **Hash exception**—A hash exception enables you to override the verdict for a specific file without affecting the settings in your Malware Security profile. The hash exception policy is evaluated first and takes precedence over all other methods to determine the hash verdict.

   For example, you may want to configure a hash exception for any of the following situations:
   
   - You want to block a file that has a benign verdict.
   - You want to allow a file that has a malware verdict to run. In general, we recommend that you only override the verdict for malware after you use available threat intelligence resources—such as WildFire and AutoFocus—to determine that the file is not malicious.
   - You want to specify a verdict for a file that has not yet received an official WildFire verdict.

   After you configure a hash exception, Cortex XDR distributes it at the next heartbeat communication with any endpoints that have previously opened the file.

   When a file launches on the endpoint, the Cortex XDR agent first evaluates any relevant hash exception for the file. The hash exception specifies whether to treat the file as malware. If the file is assigned a benign verdict, the Cortex XDR agent permits it to open.

   If a hash exception is not configured for the file, the Cortex XDR agent next evaluates the verdict to determine the likelihood of malware. The Cortex XDR agent uses a multi-step evaluation process in the following order to determine the verdict: Highly trusted signers, WildFire verdict, and then Local analysis.

2. **Highly trusted signers (Windows and Mac)**—The Cortex XDR agent distinguishes highly trusted signers such as Microsoft from other known signers. To keep parity with the signers defined in WildFire, Palo Alto Networks regularly reviews the list of highly trusted and known signers and delivers any changes with content updates. The list of highly trusted signers also includes signers that you whitelist from Cortex XDR. When an unknown file attempts to run, the Cortex XDR agent applies the following evaluation criteria: Files signed by highly trusted signers are permitted to run regardless of the WildFire verdict. When a file is not signed by a highly trusted signer, the Cortex XDR agent next evaluates the WildFire verdict. For Windows endpoints, evaluation of other known signers takes place if WildFire evaluation returns an unknown verdict for the file.

3. **WildFire verdict**—If a file is not signed by a highly trusted signer on Windows and Mac endpoints, the Cortex XDR agent performs a hash verdict lookup to determine if a verdict already exists in its local cache.

   If the executable file has a malware verdict, the Cortex XDR agent reports the security event to the Cortex XDR and, depending on the configured behavior for malicious files, the Cortex XDR agent then does one of the following:
   
   - Blocks the malicious executable file
   - Blocks and quarantines the malicious executable file
   - Notifies the user about the file but still allows the file to execute
   - Logs the issue without notifying the user and allows the file to execute.

   If the verdict is benign, the Cortex XDR agent moves on to the next stage of evaluation (see Phase 4: Evaluation of Malware Protection Policy).

   If the hash does not exist in the local cache or has an unknown verdict, the Cortex XDR agent next evaluates whether the file is signed by a known signer.

4. **Local analysis**—When an unknown executable, DLL, or macro attempts to run on a Windows or Mac endpoint, the Cortex XDR agent uses local analysis to determine if it is likely to be malware. On Windows endpoints, if the file is signed by a known signer, the Cortex XDR agent permits the file to run and does not perform additional analysis. For files on Mac endpoints and files that are not signed by a known signer on Windows endpoints, the Cortex XDR agent performs local analysis to determine whether the file is malware. Local analysis uses a statistical model that was developed with machine learning on WildFire threat intelligence. The model enables the Cortex XDR agent to examine hundreds of characteristics for a file and issue a local verdict (benign or malicious) while the endpoint is offline or
Cortex XDR is unreachable. The Cortex XDR agent can rely on the local analysis verdict until it receives an official WildFire verdict or hash exception.

Local analysis is enabled by default in a Malware Security profile. Because local analysis always returns a verdict for an unknown file, if you enable the Cortex XDR agent to Block files with unknown verdict, the agent only blocks unknown files if a local analysis error occurs or local analysis is disabled. To change the default settings (not recommended), see Add a New Malware Security Profile.

**Phase 4: Evaluation of Malware Security Policy**

If the prior evaluation phases do not identify a file as malware, the Cortex XDR agent observes the behavior of the file and applies additional malware protection rules. If a file exhibits malicious behavior, such as encryption-based activity common with ransomware, the Cortex XDR agent blocks the file and reports the security event to the Cortex XDR.

If no malicious behavior is detected, the Cortex XDR agent permits the file (process) to continue running but continues to monitor the behavior for the lifetime of the process.

**Endpoint Protection Capabilities**

Each security profile provides a tailored list of protection capabilities that you can configure for the platform you select. The following table describes the protection capabilities you can customize in a security profile. The table also indicates which platforms support the protection capability (a dash (—) indicates the capability is not supported).

<table>
<thead>
<tr>
<th>Protection Capability</th>
<th>Windows</th>
<th>Mac</th>
<th>Linux</th>
<th>Android</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Exploit Security Profiles</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Browser Exploits Protection</strong></td>
<td>✓</td>
<td>✓</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Browsers can be subject to exploitation attempts from malicious web pages and exploit kits that are embedded in compromised websites. By enabling this capability, the Cortex XDR agent automatically protects browsers from common exploitation attempts.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Logical Exploits Protection</strong></td>
<td>✓</td>
<td>✓</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Attackers can use existing mechanisms in the operating system—such as DLL-loading processes or built in system processes—to execute malicious code. By enabling this capability, the Cortex XDR agent automatically protects endpoints from attacks that try to leverage common operating system mechanisms for malicious purposes.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Known Vulnerable Processes Protection</strong></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>—</td>
</tr>
<tr>
<td>Common applications in the operating system, such as PDF readers, Office applications, and even processes that are a part of the operating system itself can contain bugs and vulnerabilities</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
that an attacker can exploit. By enabling this capability, the Cortex XDR agent protects these processes from attacks which try to exploit known process vulnerabilities.

<table>
<thead>
<tr>
<th>Protection Capability</th>
<th>Windows</th>
<th>Mac</th>
<th>Linux</th>
<th>Android</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exploit Protection for Additional Processes</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>−</td>
</tr>
<tr>
<td>To extend protection to third-party processes that are not protected by the default policy from exploitation attempts, you can add additional processes to this capability.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Operating System Exploit Protection

Attackers commonly leverage the operating system itself to accomplish a malicious action. By enabling this capability, the Cortex XDR agent protects operating system mechanisms such as privilege escalation and prevents them from being used for malicious purposes.

### Malware Security Profiles

#### Behavioral Threat Protection
Prevents sophisticated attacks that leverage built-in OS executables and common administration utilities by continuously monitoring endpoint activity for malicious causality chains.

#### Ransomware Protection
Targets encryption based activity associated with ransomware to analyze and halt ransomware before any data loss occurs.

#### Prevent Malicious Child Process Execution
Prevents script-based attacks used to deliver malware by blocking known targeted processes from launching child processes commonly used to bypass traditional security approaches.

#### Portable Executables and DLLs Examination

© 2020 Palo Alto Networks, Inc.
<table>
<thead>
<tr>
<th>Protection Capability</th>
<th>Windows</th>
<th>Mac</th>
<th>Linux</th>
<th>Android</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analyze and prevent malicious executable and DLL files from running.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ELF Files Examination</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analyze and prevent malicious ELF files from running.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Office Files Examination</strong></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analyze and prevent malicious macros embedded in Microsoft Office files from running.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Mach-O Files Examination</strong></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analyze and prevent malicious mach-o files from running.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>APK Files Examination</strong></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Analyze and prevent malicious APK files from running.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Restrictions Security Profiles**

<table>
<thead>
<tr>
<th>Security Profiles</th>
<th>Windows</th>
<th>Mac</th>
<th>Linux</th>
<th>Android</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Execution Paths</strong></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Many attack scenarios are based on writing malicious executable files to certain folders such as the local temp or download folder and then running them. Use this capability to restrict the locations from which executable files can run.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Network Locations</strong></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>To prevent attack scenarios that are based on writing malicious files to remote folders, you can restrict access to all network locations except for those that you explicitly trust.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Removable Media</strong></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>To prevent malicious code from gaining access to endpoints using external media such as a removable drive, you can restrict the executable files, that users can launch from external drives attached to the endpoints in your network.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Optical Drive</strong></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>To prevent malicious code from gaining access to endpoints using optical disc</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
drives (CD, DVD, and Blu-ray), you can restrict the executable files, that users can launch from optical disc drives connected to the endpoints in your network.

Endpoint Protection Modules

Each security profile applies multiple security modules to protect your endpoints from a wide range of attack techniques. While the settings for each module are not configurable, the Cortex XDR agent activates a specific protection module depending on the type of attack, the configuration of your security policy, and the operating system of the endpoint. When a security event occurs, the Cortex XDR agent logs details about the event including the security module employed by the Cortex XDR agent to detect and prevent the attack based on the technique. To help you understand the nature of the attack, the alert identifies the protection module the Cortex XDR agent employed.

The following table lists the modules and the platforms on which they are supported. A dash (—) indicates the module is not supported.

<table>
<thead>
<tr>
<th>Module</th>
<th>Windows</th>
<th>Mac</th>
<th>Linux</th>
<th>Android</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anti-Ransomware</td>
<td>✔️</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Targets encryption-based activity associated with ransomware and has the ability to analyze and halt ransomware activity before any data loss occurs.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>APC Protection</td>
<td>✔️</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Prevents attacks that change the execution order of a process by redirecting an asynchronous procedure call (APC) to point to the malicious shellcode.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Behavioral Threat</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>—</td>
</tr>
<tr>
<td>Prevents sophisticated attacks that leverage built-in OS executables and common administration utilities by continuously monitoring endpoint activity for malicious causality chains.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Brute Force Protection</td>
<td>—</td>
<td>—</td>
<td>✔️</td>
<td>—</td>
</tr>
<tr>
<td>Prevents attackers from hijacking the process control</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Module</td>
<td>Windows</td>
<td>Mac</td>
<td>Linux</td>
<td>Android</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>---------</td>
<td>-----</td>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>Child Process Protection</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prevents script-based attacks that are used to deliver malware, such as ransomware, by blocking known targeted processes from launching child processes that are commonly used to bypass traditional security approaches.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPL Protection</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Protects against vulnerabilities related to the display routine for Windows Control Panel Library (CPL) shortcut images, which can be used as a malware infection vector.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Execution Prevention (DEP)</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prevents areas of memory defined to contain only data from running executable code.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DLL Hijacking</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prevents DLL-hijacking attacks where the attacker attempts to load dynamic-link libraries on Windows operating systems from unsecure locations to gain control of a process.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DLL Security</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prevents access to crucial DLL metadata from untrusted code locations.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dylib Hijacking</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prevents Dylib-hijacking attacks where the attacker attempts to load dynamic libraries on Mac operating systems from unsecure locations to gain control of a process.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Module</td>
<td>Windows</td>
<td>Mac</td>
<td>Linux</td>
<td>Android</td>
</tr>
<tr>
<td>------------------------------</td>
<td>---------</td>
<td>-----</td>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>Exploit Kit Fingerprint</strong></td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Protects against the fingerprinting technique used by browser exploit kits to identify information—such as the OS or applications which run on an endpoint—that attackers can leverage when launching an attack to evade protection capabilities.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Font Protection</strong></td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Prevents improper font handling, a common target of exploits.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Gatekeeper Enhancement</strong></td>
<td>-</td>
<td>✓</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Enhances the macOS gatekeeper functionality that allows apps to run based on their digital signature. This module provides an additional layer of protection by extending gatekeeper functionality to child processes so you can enforce the signature level of your choice.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Hash Exception</strong></td>
<td>✓</td>
<td>✓</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Halts execution of files that an administrator identified as malware regardless of the WildFire verdict.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Hot Patch Protection</strong></td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Prevents the use of system functions to bypass DEP and address space layout randomization (ASLR).</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Java Deserialization</strong></td>
<td>-</td>
<td>-</td>
<td>✓</td>
<td>-</td>
</tr>
<tr>
<td>Blocks attempts to execute malicious code during the Java objects deserialization process on Java-based servers.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>JIT</strong></td>
<td>✓</td>
<td>✓</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Prevents an attacker from bypassing the operating</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Module</td>
<td>Windows</td>
<td>Mac</td>
<td>Linux</td>
<td>Android</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------</td>
<td>-----</td>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>system's memory mitigations using just-in-time (JIT) compilation engines.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Local Analysis</strong></td>
<td>✓</td>
<td>✓</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Examines hundreds of characteristics of an unknown executable file, DLL, or macro to determine if it is likely to be malware. The local analysis module uses a statistical model that was developed using machine learning on WildFire threat intelligence.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Local Privilege Escalation Protection</strong></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>—</td>
</tr>
<tr>
<td>Prevents attackers from performing malicious activities that require privileges that are higher than those assigned to the attacked or malicious process.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Null Dereference</strong></td>
<td>✓</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Prevents malicious code from mapping to address zero in the memory space, making null dereference vulnerabilities unexploitable.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Restricted Execution - Local Path</strong></td>
<td>✓</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Prevents unauthorized execution from a local path.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Restricted Execution - Network Location</strong></td>
<td>✓</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Prevents unauthorized execution from a network path.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Restricted Execution - Removable Media</strong></td>
<td>✓</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Prevents unauthorized execution from removable media.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Reverse Shell Protection</strong></td>
<td>—</td>
<td>—</td>
<td>✓</td>
<td>—</td>
</tr>
<tr>
<td>Module</td>
<td>Windows</td>
<td>Mac</td>
<td>Linux</td>
<td>Android</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>---------</td>
<td>-----</td>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>Blocks malicious activity where an attacker redirects standard input and output streams to network sockets.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ROP</strong></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>—</td>
</tr>
<tr>
<td>Protects against the use of return-oriented programming (ROP) by protecting APIs used in ROP chains.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>SEH</strong></td>
<td>✓</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Prevents hijacking of the structured exception handler (SEH), a commonly exploited control structure that can contain multiple SEH blocks that form a linked list chain, which contains a sequence of function records.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Shellcode Protection</strong></td>
<td>—</td>
<td>—</td>
<td>✓</td>
<td>—</td>
</tr>
<tr>
<td>Reserves and protects certain areas of memory commonly used to house payloads using heap spray techniques.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ShellLink</strong></td>
<td>✓</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Prevents shell-link logical vulnerabilities.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>SO Hijacking Protection</strong></td>
<td>—</td>
<td>—</td>
<td>✓</td>
<td>—</td>
</tr>
<tr>
<td>Prevents dynamic loading of libraries from unsecure locations to gain control of a process.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>SysExit</strong></td>
<td>✓</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Prevents using system calls to bypass other protection capabilities.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>UASLR</strong></td>
<td>✓</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Improves or altogether implements ASLR (address space layout randomization) with greater entropy,</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Module

<table>
<thead>
<tr>
<th>Module</th>
<th>Windows</th>
<th>Mac</th>
<th>Linux</th>
<th>Android</th>
</tr>
</thead>
<tbody>
<tr>
<td>WildFire</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Leverages WildFire for threat intelligence to determine whether a file is malware. In the case of unknown files, Cortex XDR can forward samples to WildFire for in-depth analysis.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WildFire Post-Detection (Malware and Grayware)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Identifies a file that was previously allowed to run on an endpoint that is now determined to be malware. Post-detection events provide notifications for each endpoint on which the file executed.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Manage Cortex XDR Agents

- Create an Agent Installation Package
- Set an Application Proxy for Cortex XDR Agents
- Upgrade Cortex XDR Agents
- Delete Cortex XDR Agents
- Uninstall the Cortex XDR Agent

Create an Agent Installation Package

To install the Cortex XDR agent on the endpoint for the first time, you must first create an agent installation package. After you create and download an installation package, you can then install it directly on an endpoint or you can use a software deployment tool of your choice to distribute the software to multiple endpoints. To install the Cortex XDR agent, you must use a valid installation package that exists in your Cortex XDR management console. If you delete an installation package, any agents installed from this package are not able to register to Cortex XDR.

After you install the Cortex XDR agent for the first time, you can upgrade individual or batches of agents remotely from the Cortex XDR management console.

To create a new installation package:

**STEP 1** | From Cortex XDR, select **Endpoints > Endpoint Management > Agent Installations**.

<table>
<thead>
<tr>
<th>Creation Time</th>
<th>Status</th>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Platform</th>
<th>Agent Version</th>
<th>CR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oct 27th 2019</td>
<td>Completed</td>
<td>MayTest</td>
<td>Standalone Installer</td>
<td>Windows</td>
<td>6.2.0.23021</td>
<td>Ma</td>
<td></td>
</tr>
<tr>
<td>Oct 27th 2019</td>
<td>Completed</td>
<td>Test</td>
<td>Standalone Installer</td>
<td>Windows</td>
<td>6.2.0.23021</td>
<td>G8</td>
<td></td>
</tr>
<tr>
<td>Oct 17th 2019</td>
<td>In Progress</td>
<td>Test</td>
<td>Standalone Installer</td>
<td>Windows</td>
<td>6.2.0.23021</td>
<td>Ma</td>
<td></td>
</tr>
<tr>
<td>Oct 17th 2019</td>
<td>Completed</td>
<td>Test</td>
<td>Standalone Installer</td>
<td>Windows</td>
<td>6.2.0.23021</td>
<td>M8</td>
<td></td>
</tr>
</tbody>
</table>

**STEP 2** | **Create** a new installation package.

**STEP 3** | Enter a unique **Name** and an optional **Description** to identify the installation package.
The package Name must be no more than 32 characters but can contain letters, numbers, or spaces.

STEP 4 | Select the Package Type.
- **Standalone Installers**—Use for fresh installations and to Upgrade Cortex XDR Agents on a registered endpoint that is connected to Cortex XDR.
- **(Windows, macOS, and Linux only) Upgrade from ESM**—Use this package to upgrade Traps agents which connect to the on-premise Traps Endpoint Security Manager to Cortex XDR.

STEP 5 | Select the Platform for which you want to create the installation package.

STEP 6 | *(Windows, macOS, and Linux only)* Select the Agent Version for the package.

STEP 7 | Create the installation package.
Cortex XDR prepares your installation package and makes it available on the Agent Installations page.

STEP 8 | Download your installation package.
When the status of the package shows Completed, right-click the agent version, and click Download.
- For Windows endpoints, select between the architecture type.
- For Android endpoints, Cortex XDR creates a tenant-specific download link which you can distribute to Android endpoints. When a newer agent version is available, Cortex XDR identifies older package versions as [Outdated].
- For Linux endpoints, you can download .rpm or .deb installers (according to the endpoint Linux distribution), and deploy the installers on the endpoints using the Linux package manager. Alternatively, you can download a Shell installer and deploy it manually on the endpoint.

> **When you upgrade a Cortex XDR agent version without package manager, Cortex XDR will upgrade the installation process to package manager by default, according to the endpoint Linux distribution.**

STEP 9 | Next steps:
As needed, you can return to the Agent Installations page to manage your agent installation packages. To manage a specific package, right click the agent version, and select the desired action:
- **Edit** the package name or description.
- **Delete** the installation package. Deleting an installation package does not uninstall the Cortex XDR agent software from any endpoints. However, if you install the Cortex XDR agent from a package after you delete it, Cortex XDR denies the registration request leaving the agent in an unprotected state. If this is undesirable, you can instead hide the installation package from the main view of the Agent Installations page. Hiding a package can be useful to filter earlier or less relevant versions from the main view.
- **Copy text to clipboard** to copy the text from a specific field in the row of an installation package.
- **Hide** installation packages. Using the Hide option provides a quick method to filter out results based on a specific value in the table. You can also use the filters at the top of the page to build a filter from scratch. To create a persistent filter, save it.

Set an Application Proxy for Cortex XDR Agents

> **This capability is supported on endpoints with Traps agent 5.0.9 (Windows only) or Cortex XDR agent 7.0 and later releases.**
In environments where agents communicate with the Cortex XDR server through a wide-system proxy, you can now set an application-specific proxy for the Traps and Cortex XDR agent without affecting the communication of other applications on the endpoint. You can set the proxy in one of three ways: during the agent installation or after installation using Cytool on the endpoint or from Endpoints Management in Cortex XDR as described in this topic. You can assign up to five different proxy servers per agent. The proxy server the agent uses is selected randomly and with equal probability. If the communication between the agent and the Cortex XDR server through the app-specific proxies fails, the agent resumes communication through the system-wide proxy defined on the endpoint. If that fails as well, the agent resumes communication with Cortex XDR directly.

**STEP 1** | From Cortex XDR, select Endpoints > Endpoint Management > Endpoint Administration.

**STEP 2** | If needed, filter the list of endpoints.

**STEP 3** | Set an agent proxy.

1. Select the row of the endpoint for which you want to set a proxy.
2. Right-click the endpoint and select Endpoint Control > Set Endpoint Proxy.
3. For each proxy, enter the IP address and port number. You can assign up to five different proxies per agent.
4. Set when you’re done.

5. If necessary, you can later Disable Endpoint Proxy from the right-click menu.
   When you disable the proxy configuration, all proxies associated with that agent are removed. The agent resumes communication with the Cortex XDR server through the wide-system proxy if defined, otherwise if a wide-system is not defined the agent resumes communicating directly with the Cortex XDR server. If neither a wide-system proxy nor direct communication exist and you disable the proxy, the agent will disconnect from Cortex XDR.

### Upgrade Cortex XDR Agents

After you install the Cortex XDR agent and the agent registers with Cortex XDR, you can upgrade the Cortex XDR agent software using a method supported by the endpoint platform:

- **Android**—Upgrade the app directly from the Google Play Store or push the app to your endpoints from an endpoint management system such as AirWatch.
- **Windows, Mac, or Linux**—Create new installation packages and push the Cortex XDR agent package to up to 5,000 endpoints from Cortex XDR.

**You cannot upgrade VDI endpoints.**

Upgrades are supported using actions which you can initiate from the Action Center or from Endpoint Administration as described in this workflow.
**STEP 1 | Create an Agent Installation Package** for each operating system version for which you want to upgrade the Cortex XDR agent.

Note the installation package names.

**STEP 2 | Select Endpoints > Endpoint Management.**

If needed, filter the list of endpoints. To reduce the number of results, use the endpoint name search and filters Filters at the top of the page.

**STEP 3 | Select the endpoints you want to upgrade.**

You can also select endpoints running different operating systems to upgrade the agents at the same time.

**STEP 4 | Right-click your selection and Upgrade agent version.**

For each platform, select the name of the installation package you want to push to the selected endpoints.

Starting in the Cortex XDR agent 7.1 release, you can install the Cortex XDR agent on Linux endpoints using package manager. When you upgrade an agent on a Linux endpoint that is not using package manager, Cortex XDR upgrades the installation process by default according to the endpoint Linux distribution. Alternatively, if you do not want to use the package manager, clear the option Upgrade to installation by package manager.
STEP 5 | Upgrade.

Cortex XDR distributes the installation package to the selected endpoints at the next heartbeat communication with the agent. To monitor the status of the upgrades, go to Response > Action Center. From the Action Center you can also view additional information about the upgrade (right-click the action and select Additional data) or cancel the upgrade (right-click the action and select Cancel Agent Upgrade).

During the upgrade process, the endpoint operating system might request for a reboot. However, you do not have to perform the reboot for the Cortex XDR agent upgrade process to complete successfully.

Delete Cortex XDR Agents

From Cortex XDR, you can delete a Cortex XDR agent from one or more Windows, Mac, or Linux endpoints that have disconnected from the Cortex XDR management console. Deleting an endpoint triggers the following lifespan flow:

- Standard agents are deleted after 180 days of inactivity.
- VDI and TS agents are deleted after 6 hours of inactivity.

To reinstate an endpoint, you have to uninstall and reinstall the endpoint.

After an endpoint is deleted, data associated with the deleted endpoint is displayed in the Action Center tables and in the Causality View with an Endpoint Name - N/A (Endpoint Deleted). Alerts that already include the endpoint data at the time of the alert creation are not affected.
The following workflow describes how to delete the Cortex XDR agent from one or more Windows, Mac, or Linux endpoints.

**STEP 1** | Select **Endpoints > Endpoint Management > Endpoint Administration**.

**STEP 2** | Right-click the endpoint you want to remove.

You can also select multiple endpoints if you want to perform a bulk delete.

**STEP 3** | Select **Endpoint Control > Delete Endpoint**.

### Uninstall the Cortex XDR Agent

From Cortex XDR, you can uninstall the Cortex XDR agent from one or more Windows, Mac, or Linux endpoints at any time. You can uninstall the Cortex XDR agent from an unlimited number of endpoints in a single bulk action. To uninstall the Cortex XDR app for Android, you must do so from the Android endpoint.

The following workflow describes how to uninstall the Cortex XDR agent from one or more Windows, Mac, or Linux endpoints.

**STEP 1** | Log in to Cortex XDR.

Go to **Response > Action Center > + New Action**.

**STEP 2** | Select **Agent Uninstall**.

**STEP 3** | Click **Next**.

**STEP 4** | Select the target endpoints (up to 100) for which you want to uninstall the Cortex XDR agent.

*If needed, Filter the list of endpoints by attribute or group name.*

**STEP 5** | Click **Next**.

**STEP 6** | Review the action summary and click **Done** when finished.

**STEP 7** | To track the status of the uninstallation, return to the **Action Center**.
Define Endpoint Groups

To easily apply policy rules to specific endpoints, you can define an endpoint group. There are two methods you can use to define an endpoint group:

- Create a dynamic group by allowing Cortex XDR to populate your endpoint group dynamically using endpoint characteristics such as a partial hostname or alias; full or partial domain or workgroup name; IP address, range or subnet; installation type (VDI, temporary session, or standard endpoint); agent version; endpoint type (workstation, server, mobile); or operating system version.
- Create a static group by selecting a list of specific endpoints.

After you define an endpoint group, you can then use it to target policy and actions to specific recipients. The Endpoint Groups page displays all endpoint groups along with the number of endpoints and policy rules linked to the endpoint group.

To define an endpoint static or dynamic group:

**STEP 1** | From Cortex XDR, select **Endpoints > Endpoint Management > Endpoint Groups > +Add Group**.

**STEP 2** | Select either Create New to create an endpoint group from scratch or Upload From File to populate a static endpoint group from a file containing IP addresses, hostnames, or aliases.

**STEP 3** | Enter a **Group Name** and optional **Description** to identify the endpoint group. The name you assign to the group will be visible when you assign endpoint security profiles to endpoints.

**STEP 4** | Determine the endpoint properties for creating an endpoint group:

- **Dynamic**—Use the filters to define the criteria you want to use to dynamically populate an endpoint group. Dynamic groups support multiple criteria selections and can use **AND** or **OR** operators. For endpoint names and aliases, and domains and workgroups, you can use **•** to match any string of characters. As you apply filters, Cortex XDR displays any registered endpoint matches to help you validate your filter criteria.
Cortex XDR supports only IPv4 addresses.

- **Static**—Select specific registered endpoints that you want to include in the endpoint group. Use the filters, as needed, to reduce the number of results.

  When you create a static endpoint group from a file, the IP address, hostname, or alias of the endpoint must match an existing agent that has registered with Cortex XDR. You can select up to 250 endpoints.

  *When you disconnect the Directory Sync Service (DSS) in your Cortex XDR deployment, it might affect existing endpoint groups and policy rules based on Active Directory properties.*
STEP 5 | Create the endpoint group.

After you save your endpoint group, it is ready for use to assign security profiles to endpoints and in other places where you can use endpoint groups.

STEP 6 | Manage an endpoint group, as needed.

At any time, you can return to the Endpoint Groups page to view and manage your endpoint groups. To manage a group, right-click the group and select the desired action:

- **Edit**—View the endpoints that match the group definition, and optionally refine the membership criteria using filters.
- **Delete** the endpoint group.
- **Save as new**—Duplicate the endpoint group and save it as a new group.
- **Export group**—Export the list of endpoints that match the endpoint group criteria to a tab separated values (TSV) file.
About Content Updates

To increase security coverage and quickly resolve any issues in policy, Palo Alto Networks can seamlessly deliver software packages for Cortex XDR called content updates. Content updates can contain changes or updates to any of the following:

Starting with the Cortex XDR 7.1 agent release, Cortex XDR delivers to the agent the content update in parts and not as a single file, allowing the agent to retrieve only the updates and additions it needs.

- Default security policy including exploit, malware, restriction, and agent settings profiles
- Default compatibility rules per module
- Protected processes
- Local analysis logic
- Trusted signers
- Blacklisted processes by signers
- Behavioral threat protection rules
- Ransomware module logic including Windows network folders susceptible to ransomware attacks
- Windows Event Logs
- Python scripts provided by Palo Alto Networks
- Python modules supported in script execution

When a new update is available, Cortex XDR notifies the Cortex XDR agent. The Cortex XDR agent then randomly chooses a time within a six-hour window during which it will retrieve the content update from Cortex XDR. By staggering the distribution of content updates, Cortex XDR reduces the bandwidth load and prevents bandwidth saturation due to the high volume and size of the content updates across many endpoints. You can view the distribution of endpoints by content update version from the Cortex XDR Dashboard.

To adjust content update distribution for your environment, you can configure the following optional settings:

- Content distribution bandwidth as part of the Cortex XDR global agent configurations.
- Content download source, as part of the Cortex XDR agent setting profile.
### Endpoint Security Profiles

Cortex XDR provides default security profiles that you can use out of the box to immediately begin protecting your endpoints from threats. While security rules enable you to block or allow files to run on your endpoints, security profiles help you customize and reuse settings across different groups of endpoints. When the Cortex XDR agent detects behavior that matches a rule defined in your security policy, the Cortex XDR agent applies the security profile that is attached to the rule for further inspection.

<table>
<thead>
<tr>
<th>Profile Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exploit Profiles</td>
<td>Exploit profiles block attempts to exploit system flaws in browsers, and in the operating system. For example, Exploit profiles help protect against exploit kits, illegal code execution, and other attempts to exploit process and system vulnerabilities. Exploit profiles are supported for Windows, Mac, and Linux platforms.</td>
</tr>
<tr>
<td></td>
<td><a href="#">Add a New Exploit Security Profile.</a></td>
</tr>
<tr>
<td>Malware Profiles</td>
<td>Malware profiles protect against the execution of malware including trojans, viruses, worms, and grayware. Malware profiles serve two main purposes: to define how to treat behavior common with malware, such as ransomware or script-based attacks, and to define how to treat known malware and unknown files. Malware profiles are supported for all platforms.</td>
</tr>
<tr>
<td></td>
<td><a href="#">Add a New Malware Security Profile.</a></td>
</tr>
<tr>
<td>Restrictions Profiles</td>
<td>Restrictions profiles limit where executables can run on an endpoint. For example, you can restrict files from running from specific local folders or from removable media. Restrictions profiles are supported only for Windows platforms.</td>
</tr>
<tr>
<td></td>
<td><a href="#">Add a New Restrictions Security Profile.</a></td>
</tr>
<tr>
<td>Agent Settings Profiles</td>
<td>Agent Settings profiles enable you to customize settings that apply to the Cortex XDR agent (such as the disk space quota for log retention). For Mac and Windows platforms, you can also customize user interface options for the Cortex XDR console, such as accessibility and notifications.</td>
</tr>
<tr>
<td></td>
<td><a href="#">Add a New Agent Settings Profile.</a></td>
</tr>
<tr>
<td>Exceptions Profiles</td>
<td>Exceptions Security Profiles override the security policy to allow a process or file to run on an endpoint, to disable a specific BTP rule, to allow a known digital signer, and to import exceptions from the Cortex XDR support team. Exceptions</td>
</tr>
</tbody>
</table>

© 2020 Palo Alto Networks, Inc.
Add a New Exploit Security Profile

Exploit security profiles allow you to configure the action the Cortex XDR agent takes when attempts to exploit software vulnerabilities or flaws occur. To protect against specific exploit techniques, you can customize exploit protection capabilities in each Exploit security profile.

By default, the Cortex XDR agent will receive the default profile that contains a pre-defined configuration for each exploit capability supported by the platform. To fine-tune your Exploit security policy, you can override the configuration of each capability to block the exploit behavior, allow the behavior but report it, or disable the module.

To define an Exploit security profile:

**STEP 1 | Add a new profile.**
1. From Cortex XDR, select **Endpoints > Policy Management > Profiles > + New Profile**.
2. Select the platform to which the profile applies and **Exploit** as the profile type.
3. Click **Next**.

**STEP 2 | Define the basic settings.**
1. Enter a unique **Profile Name** to identify the profile. The name can contain only letters, numbers, or spaces, and must be no more than 30 characters. The name you choose will be visible from the list of profiles when you configure a policy rule.
2. To provide additional context for the purpose or business reason that explains why you are creating the profile, enter a profile **Description**. For example, you might include an incident identification number or a link to a help desk ticket.

**STEP 3 | Configure the action to take when the Cortex XDR agent detects an attempt to exploit each type of software flaw.**

For details on the different exploit protection capabilities, see **Endpoint Protection Capabilities**.

- **Block**—Block the exploit attack.
- **Report**—Allow the exploit activity but report it to Cortex XDR.
- **Disabled**—Disable the module and do not analyze or report exploit attempts.
- **Default**—Use the default configuration to determine the action to take. Cortex XDR displays the current default configuration for each capability in parenthesis. For example, **Default (Block)**.

To view which processes are protected by each capability, see **Processes Protected by Exploit Security Policy**.

For **Logical Exploits Protection**, you can also configure a blacklist for the DLL Hijacking module. The blacklist enables you to block specific DLLs when run by a protected process. The DLL folder or file must include the complete path. To complete the path, you can use environment variables or the asterisk (*) as a wildcard to match any string of characters (for example, */windows32/*).

For **Exploit Protection for Additional Processes**, you also add one or more additional processes.
In Exploit Security profiles, if you change the action mode for processes, you must restart the protected processes for the following security modules to take effect on the process and its forked processes: Brute Force Protection, Java Deserialization, ROP, and SO Hijacking.

STEP 4 | Save the changes to your profile.

STEP 5 | Apply Security Profiles to Endpoints.

You can do this in two ways: You can Create a new policy rule using this profile from the right-click menu or you can launch the new policy wizard from Policy Rules.

Processes Protected by Exploit Security Policy

By default, your exploit security profile protects endpoints from attack techniques that target specific processes. Each exploit protection capability protects a different set of processes that Palo Alto Networks researchers determine are susceptible to attack. The following tables display the processes that are protected by each exploit protection capability for each operating system.

### Windows Processes Protected by Exploit Security Policy

#### Browser Exploits Protection

<table>
<thead>
<tr>
<th>Processes Protected by Exploit Security Policy</th>
</tr>
</thead>
<tbody>
<tr>
<td>• [updated version of Adobe Flash Player for Firefox installed on endpoint]</td>
</tr>
<tr>
<td>• browser_broker.exe</td>
</tr>
<tr>
<td>• chrome.exe</td>
</tr>
<tr>
<td>• firefox.exe</td>
</tr>
<tr>
<td>• flashutil Activex.exe</td>
</tr>
<tr>
<td>• iexplore.exe</td>
</tr>
<tr>
<td>• microsoftedge.exe</td>
</tr>
<tr>
<td>• microsoftedgecp.exe</td>
</tr>
<tr>
<td>• opera_plugin_wrapper.exe</td>
</tr>
<tr>
<td>• microsoftedgecp.exe</td>
</tr>
<tr>
<td>• opera.exe</td>
</tr>
<tr>
<td>• plugin-container.exe</td>
</tr>
<tr>
<td>• safari.exe</td>
</tr>
<tr>
<td>• webkit2webprocess.exe</td>
</tr>
</tbody>
</table>

#### Logical Exploits Protection

<table>
<thead>
<tr>
<th>Logical Exploits Protection</th>
</tr>
</thead>
<tbody>
<tr>
<td>• cliconfg.exe</td>
</tr>
<tr>
<td>• dism.exe</td>
</tr>
<tr>
<td>• dllhost.exe</td>
</tr>
<tr>
<td>• excel.exe</td>
</tr>
<tr>
<td>• migwiz.exe</td>
</tr>
<tr>
<td>• mmc.exe</td>
</tr>
<tr>
<td>• powerpnt.exe</td>
</tr>
<tr>
<td>• sysprep.exe</td>
</tr>
<tr>
<td>• winword.exe</td>
</tr>
</tbody>
</table>

#### Known Vulnerable Processes Protection

<table>
<thead>
<tr>
<th>Known Vulnerable Processes Protection</th>
</tr>
</thead>
<tbody>
<tr>
<td>• 7z.exe</td>
</tr>
<tr>
<td>• 7zfm.exe</td>
</tr>
<tr>
<td>• 7zg.exe</td>
</tr>
<tr>
<td>• acrobat.exe</td>
</tr>
<tr>
<td>• acrord32.exe</td>
</tr>
<tr>
<td>• acrord32info.exe</td>
</tr>
<tr>
<td>• allplayer.exe</td>
</tr>
<tr>
<td>• applemobiledeviceservice.exe</td>
</tr>
<tr>
<td>• apwebmgr.exe</td>
</tr>
<tr>
<td>• armsvc.exe</td>
</tr>
<tr>
<td>• blazehdtv.exe</td>
</tr>
<tr>
<td>• bsplayer.exe</td>
</tr>
<tr>
<td>• cmd.exe</td>
</tr>
<tr>
<td>• ipodservice.exe</td>
</tr>
<tr>
<td>• itunes.exe</td>
</tr>
<tr>
<td>• ituneshelper.exe</td>
</tr>
<tr>
<td>• journal.exe</td>
</tr>
<tr>
<td>• jqs.exe</td>
</tr>
<tr>
<td>• microsoft.photos.exe</td>
</tr>
<tr>
<td>• msaccess.exe</td>
</tr>
<tr>
<td>• mspub.exe</td>
</tr>
<tr>
<td>• nginx.exe</td>
</tr>
<tr>
<td>• notepad++.exe</td>
</tr>
<tr>
<td>• outlook.exe</td>
</tr>
<tr>
<td>• powerpnt.exe</td>
</tr>
<tr>
<td>• SLMail.exe</td>
</tr>
<tr>
<td>• soffice.exe</td>
</tr>
<tr>
<td>• telnet.exe</td>
</tr>
<tr>
<td>• unrar.exe</td>
</tr>
<tr>
<td>• vboxservice.exe</td>
</tr>
<tr>
<td>• vboxsvc.exe</td>
</tr>
<tr>
<td>• vboxtray.exe</td>
</tr>
<tr>
<td>• video.ui.exe</td>
</tr>
<tr>
<td>• visio.exe</td>
</tr>
<tr>
<td>• vlc.exe</td>
</tr>
<tr>
<td>• vmware-authd.exe</td>
</tr>
<tr>
<td>• vmware-hostd.exe</td>
</tr>
<tr>
<td>• vmware-hostd.exe</td>
</tr>
<tr>
<td>• vmware-vmx.exe</td>
</tr>
</tbody>
</table>
**Windows Processes Protected by Exploit Security Policy**

- eqnedt32.exe
- excel.exe
- flashfxp.exe
- fltldr.exe
- fontdrvhost.exe
- foxit reader.exe
- foxitreader.exe
- groovemonitor.exe
- hxmail.exe
- i_view32.exe
- infopath.exe
- pptview.exe
- qttask.exe
- quicktimeplayer.exe
- rar.exe
- reader_sl.exe
- realconverter.exe
- realplay.exe
- realsched.exe
- skype.exe
- skypeapp.exe
- skypehost.exe
- vpreview.exe
- vprintproxy.exe
- w3wp.exe
- winrar.exe
- winword.exe
- wireshark.exe
- wmplayer.exe
- wmpnetwk.exe
- xpsrchvw.exe

**Operating System Exploit Protection**

- ctfmon.exe
- dllhost.exe
- lsass.exe
- msmpeng.exe
- runtimebroker.exe
- spoolsv.exe
- svchost.exe
- taskeng.exe
- taskhost.exe
- wmpipvse.exe
- wmiprvse.exe
- wwahost.exe

**Mac Processes Protected by Exploit Security Policy**

**Browser Exploits Protection**

- com.apple.safariservices
- com.apple.webkit.plugin
- com.apple.webkit.plugin.64
- com.apple.webkit.webcontent
- firefox
- firefox-bin
- google chrome helper
- google chrome
- plugin-container
- safari
- seamonkey

**Logical Exploits Protection**

- adobereader
- app drive for google drive
- app drop for dropbox
- app for dropbox
- app for facebook
- app for google drive
- app for googledocs
- app for instagram
- app for linkedin
- app for youtube
- com.apple.safariservices
- com.apple.webkit.plugin
- com.apple.webkit.plugin.64
- com.apple.webkit.webcontent
- document writer
- firefox
- firefox-bin
- google chrome helper
- google chrome
- itunes helper
- itunes
- mail+ for yahoo
- microsoft excel
- microsoft outlook
- microsoft powerpoint
- microsoft remote desktop
- microsoft word
- minwriterfree
- parallels client
- pdf reader pro free
- pdf reader x
- plugin-container
- quicktime player
- safari
- seamonkey
- slack
- sonicwall mobile connect
- textwrangler
- vlc
- vmware fusion services
- vmware fusion
- vpn shield
- winmail.dat file viewer

**Known Vulnerable Processes Protection**
### Mac Processes Protected by Exploit Security Policy

<table>
<thead>
<tr>
<th>Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>adobereader</td>
</tr>
<tr>
<td>airmail</td>
</tr>
<tr>
<td>app drive for google drive</td>
</tr>
<tr>
<td>app drop for dropbox</td>
</tr>
<tr>
<td>app for dropbox</td>
</tr>
<tr>
<td>app for facebook</td>
</tr>
<tr>
<td>app for google drive</td>
</tr>
<tr>
<td>app for googledocs</td>
</tr>
<tr>
<td>app for instagram</td>
</tr>
<tr>
<td>app for linkedin</td>
</tr>
<tr>
<td>app for youtube</td>
</tr>
<tr>
<td>bbedit</td>
</tr>
<tr>
<td>c-lion</td>
</tr>
<tr>
<td>cisco anyconnect secure mobility client</td>
</tr>
<tr>
<td>com.apple.cloudphotosconfiguration</td>
</tr>
<tr>
<td>document writer</td>
</tr>
<tr>
<td>itunes helper</td>
</tr>
<tr>
<td>itunes</td>
</tr>
<tr>
<td>jump desktop</td>
</tr>
<tr>
<td>mail</td>
</tr>
<tr>
<td>mail+ for yahoo</td>
</tr>
<tr>
<td>messages</td>
</tr>
<tr>
<td>microsoft excel</td>
</tr>
<tr>
<td>microsoft outlook</td>
</tr>
<tr>
<td>microsoft powerpoint</td>
</tr>
<tr>
<td>microsoft remote desktop</td>
</tr>
<tr>
<td>microsoft word</td>
</tr>
<tr>
<td>miniwriterfree</td>
</tr>
<tr>
<td>parallels client</td>
</tr>
<tr>
<td>pdf reader pro free</td>
</tr>
<tr>
<td>pdf reader x</td>
</tr>
<tr>
<td>photos</td>
</tr>
<tr>
<td>photoshop</td>
</tr>
<tr>
<td>quickbooks</td>
</tr>
<tr>
<td>quicktime player</td>
</tr>
<tr>
<td>signal</td>
</tr>
<tr>
<td>slack</td>
</tr>
<tr>
<td>sonicwall mobile connect</td>
</tr>
<tr>
<td>telegram</td>
</tr>
<tr>
<td>textmate</td>
</tr>
<tr>
<td>textwrangler</td>
</tr>
<tr>
<td>thunderbird</td>
</tr>
<tr>
<td>vlc</td>
</tr>
<tr>
<td>vmware fusion services</td>
</tr>
<tr>
<td>vmware fusion</td>
</tr>
<tr>
<td>vpn shield</td>
</tr>
</tbody>
</table>

### Linux Processes Protected by Exploit Security Policy

#### Known Vulnerable Processes Protection

<table>
<thead>
<tr>
<th>Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>anacron</td>
</tr>
<tr>
<td>apache2</td>
</tr>
<tr>
<td>authproxy</td>
</tr>
<tr>
<td>bluetoothd</td>
</tr>
<tr>
<td>charon</td>
</tr>
<tr>
<td>chronyd</td>
</tr>
<tr>
<td>couriertcpd</td>
</tr>
<tr>
<td>cron</td>
</tr>
<tr>
<td>crond</td>
</tr>
<tr>
<td>cupsd</td>
</tr>
<tr>
<td>cyrus_pop3d</td>
</tr>
<tr>
<td>danted</td>
</tr>
<tr>
<td>dhcpd</td>
</tr>
<tr>
<td>dovecot</td>
</tr>
<tr>
<td>exim</td>
</tr>
<tr>
<td>ftpd</td>
</tr>
<tr>
<td>ibserver</td>
</tr>
<tr>
<td>identd</td>
</tr>
<tr>
<td>lighttpd</td>
</tr>
<tr>
<td>kamailio</td>
</tr>
<tr>
<td>mailman</td>
</tr>
<tr>
<td>master</td>
</tr>
<tr>
<td>mongodb</td>
</tr>
<tr>
<td>mysqld</td>
</tr>
<tr>
<td>mysqld_safe</td>
</tr>
<tr>
<td>named</td>
</tr>
<tr>
<td>ndsd</td>
</tr>
<tr>
<td>nginx</td>
</tr>
<tr>
<td>ndmdb</td>
</tr>
<tr>
<td>nsqd</td>
</tr>
<tr>
<td>node</td>
</tr>
<tr>
<td>openbsd</td>
</tr>
<tr>
<td>php</td>
</tr>
<tr>
<td>php5-fpm</td>
</tr>
<tr>
<td>pmmasterd</td>
</tr>
<tr>
<td>pop2d</td>
</tr>
<tr>
<td>pop3d</td>
</tr>
<tr>
<td>postgres</td>
</tr>
<tr>
<td>proftpd</td>
</tr>
<tr>
<td>qmgr</td>
</tr>
<tr>
<td>rpcbind</td>
</tr>
<tr>
<td>rsync</td>
</tr>
<tr>
<td>rsyslogd</td>
</tr>
<tr>
<td>ruby</td>
</tr>
<tr>
<td>samba</td>
</tr>
<tr>
<td>saned</td>
</tr>
<tr>
<td>sendmail</td>
</tr>
<tr>
<td>sendmail.sendmail</td>
</tr>
<tr>
<td>smartd</td>
</tr>
<tr>
<td>smbld</td>
</tr>
<tr>
<td>snmpd</td>
</tr>
<tr>
<td>squid</td>
</tr>
<tr>
<td>squid3</td>
</tr>
<tr>
<td>starter</td>
</tr>
<tr>
<td>syslog-ng</td>
</tr>
<tr>
<td>tinyproxy</td>
</tr>
<tr>
<td>vsftpd</td>
</tr>
<tr>
<td>wicked-dhcp4</td>
</tr>
<tr>
<td>wickedddhcp6</td>
</tr>
<tr>
<td>winbindd</td>
</tr>
<tr>
<td>xinetd</td>
</tr>
</tbody>
</table>
Add a New Malware Security Profile

Malware security profiles allow you to configure the action Cortex XDR agents take when known malware and unknown files try to run on Windows, Mac, Linux, and Android endpoints.

By default, the Cortex XDR agent will receive the default profile that contains a pre-defined configuration for each malware protection capability supported by the platform. To fine-tune your Malware security policy, you can override the configuration of each capability to block the malicious behavior or file, allow but report it, or disable the module. For each setting you override, clear the option to Use Default.

To configure a Malware security profile:

STEP 1 | Add a new profile.
1. From Cortex XDR, select Endpoints > Policy Management > Profiles > + New Profile.
2. Select the platform to which the profile applies and Malware as the profile type.

STEP 2 | Identify the profile.
1. Enter a unique Profile Name to identify the profile. The name can contain only letters, numbers, or spaces, and must be no more than 30 characters. The name you choose will be visible from the list of profiles when you configure a policy rule.
2. To provide additional context for the purpose or business reason that explains why you are creating the profile, enter a profile Description. For example, you might include an incident identification number or a link to a help desk ticket.

STEP 3 | Configure the Cortex XDR agent to examine executable files, macros, or DLL files on Windows endpoints, Mach-O files on Mac endpoints, ELF files on Linux endpoints, or APK files on Android endpoints.
1. Configure the Action Mode—the behavior of the Cortex XDR agent—when malware is detected:
   - Block—Block attempts to run malware.
   - Report—Report but do not block malware that attempts to run.
   - (Android only) Prompt—Enable the Cortex XDR agent to prompt the user when malware is detected and allow the user to choose to allow malware, dismiss the notification, or uninstall the app.
   - Disabled—Disable the module and do not examine files for malware.
2. Configure additional actions to examine files for malware.
   By default, Cortex XDR uses the settings specified in the default malware security profile and displays the default configuration in parenthesis. When you select a setting other than the default, you override the default configuration for the profile.
   - (Windows only) Quarantine Malicious Executables—By default, the Cortex XDR agent blocks malware from running but does not quarantine the file. Enable this option to quarantine files depending on the verdict issuer (local analysis, WildFire, or both local analysis and WildFire. The quarantine feature is not available for malware identified in network drives.
   - Upload <file_type> files for cloud analysis—Enable the Cortex XDR agent to send unknown files to Cortex XDR, and for Cortex XDR to send the files to WildFire for analysis. With macro analysis, the Cortex XDR agent sends the Microsoft Office file containing the macro. The file types that the Cortex XDR agent analyzes depend on the platform type. WildFire accepts files up to 100MB in size.
   - Treat Grayware as Malware—Treat all grayware with the same Action Mode you configure for malware. Otherwise, if this option is disabled, grayware is considered benign and is not blocked.
   - Local Analysis—
• **Action on Unknown to WildFire**—Select the behavior of the Cortex XDR agent when an unknown file tries to run on the endpoint (**Allow**, **Run Local Analysis**, or **Block**). With local analysis, the Cortex XDR agent uses embedded machine learning to determine the likelihood that an unknown file is malware and issues a local verdict for the file. If you block unknown files but do not run local analysis, unknown files remain blocked until the Cortex XDR agent receives an official WildFire verdict.

• **(Windows only) Examine Office Files From Network Drives**—Enable the Cortex XDR agent to examine Microsoft Office files in network drives when they contain a macro that attempts to run. If this option is disabled, the Cortex XDR agent will not examine macros in network drives.

  *(Windows only)* As part of the anti-malware security flow, the Cortex XDR agent leverages the OS capability to identify revoked certificates for executables and DLL files that attempt to run on the endpoint by accessing the Windows Certificate Revocation List (CRL). To allow the Cortex XDR agent access the CRL, you must enable internet access over port 80 for Windows endpoints running Traps 6.0.3 and later releases, Traps 6.1.1 and later releases, or Cortex XDR 7.0 and later releases. If the endpoint is not connected to the internet, or you experience delays with executables and DLLs running on the endpoint, please contact Palo Alto Networks support.

3. **(Optional) Whitelist files and folders from examination.**

   1. +Add a file or folder.
   2. Enter the path and press **Enter** or click the check mark when done. You can also use a wildcard to match files and folders containing a partial name. Use ? to match a single character or * to match any string of characters. To match a folder, you must terminate the path with * to match all files in the folder (for example, `c:\temp\*`).
   3. Repeat to add additional files or folders.

4. Whitelist signers from examination.

   When a file that is signed by a whitelisted signer attempts to run,

   1. +Add a trusted signer.
   2. Enter the name of the trusted signer (Windows) or the SHA1 hash of the certificate that signs the file (Mac) and press **Enter** or click the check mark when done. You can also use a wildcard to match a partial name for the signer. Use ? to match any single character or * to match any string of characters.
   3. Repeat to add additional folders.

**STEP 4 | (Windows, Mac, and Linux only) Configure Behavioral Threat Protection.**

Behavioral threat protection requires Traps agent 6.0 or a later release for Windows endpoints, and Traps 6.1 or later versions for Mac and Linux endpoints.

With Behavioral threat protection, the agent continuously monitors endpoint activity to identify and analyze chains of events—known as causality chains. This enables the agent to detect malicious activity in the chain that could otherwise appear legitimate if inspected individually. A causality chain can include any sequence of network, process, file, and registry activities on the endpoint. For more information on data collection for Behavioral Threat Protection, see Endpoint Data Collected by Cortex XDR.

Palo Alto Networks researchers define the causality chains that are malicious and distribute those chains as behavioral threat rules. When the Cortex XDR agent detects a match to a behavioral threat protection rule, the Cortex XDR agent carries out the configured action (default is Block). In addition, the Cortex XDR agent reports the behavior of the entire event chain up to the process, known as the causality group owner (CGO), that the Cortex XDR agent identified as triggering the event sequence.

To configure Behavioral Threat Protection
1. Define the **Action mode** to take when the Cortex XDR agent detects malicious causality chains:
   - **Block** (default)—Block all processes and threads in the event chain up to the CGO.
   - **Report**—Allow the activity but report it to Cortex XDR.
   - **Disabled**—Disable the module and do not analyze or report the activity.

2. Define whether to quarantine the CGO when the Cortex XDR agent detects a malicious event chain.
   - **Enabled**—Quarantine the CGO if the file is not signed by a highly trusted signer. When the CGO is signed by a highly trusted signer or powershell.exe, wscript.exe, cscript.exe, mshta.exe, excel.exe, word.exe or powerpoint.exe, the Cortex XDR agent parses the command-line arguments and instead quarantines any scripts or files called by the CGO.
   - **Disabled** (default)—Do not quarantine the CGO of an event chain nor any scripts or files called by the CGO.

3. (Optional) Whitelist files that you do not want the Cortex XDR agent to terminate when a malicious causality chain is detected.
   1. **+Add** a file path.
   2. Enter the file path you want to exclude from evaluation. Use ? to match a single character or * to match any string of characters.
   3. Click the check mark to confirm the file path.
   4. Repeat the process to whitelist any additional file paths.

**STEP 5** (Windows only) Configure **Ransomware Protection** to define the **Action mode** to take when the Cortex XDR agent detects ransomware activity locally on the endpoint or in pre-defined network folders.
   - **Block** (default)—Block the activity.
   - **Report**—Allow the activity but report it to Cortex XDR.
   - **Disabled**—Disable the module and do not analyze or report the activity.

When enabled, the Ransomware module **Protection Mode** is **Normal** by default. If you suspect your network has been infected with ransomware and need to provide better coverage, apply the **Aggressive** protection mode.

*The Aggressive mode exposes more applications in your environment to the Cortex XDR agent decoy files to increase the module coverage. This could result in more cases where benign applications and users will report that the decoy files are visible on the endpoint.*

**STEP 6** (Windows only) Configure the Cortex XDR agent to **Prevent Malicious Child Process Execution**.

1. Select the **Action Mode** to take when the Cortex XDR agent detects malicious child process execution:
   - **Block**—Block the activity.
   - **Report**—Allow the activity but report it to Cortex XDR.

2. To allow specific processes to launch child processes for legitimate purposes, whitelist the child process with optional execution criteria.
   - **+Add** and then specify the whitelist criteria including the **Parent Process Name**, **Child Process Name**, and **Command Line Params**. Use ? to match a single character or * to match any string of characters.
If you are adding child process evaluation criteria based on a specific security event, the event indicates both the source process and the command line parameters in one line. Copy only the command line parameter for use in the profile.

STEP 7 | (Windows and Mac only) Enable endpoint file scanning.

Periodic scanning enables you to scan endpoints on a reoccurring basis without waiting for malware to run on the endpoint.

1. Configure the **Action Mode** for the Cortex XDR agent to periodically scan the endpoint for malware: **Enabled** to scan at the configured intervals, **Disabled** (default) if you don't want the Cortex XDR agent to scan the endpoint.
2. To configure the scan schedule, set the frequency (**Run Weekly** or **Run Monthly**) and day and time at which the scan will run on the endpoint.
   
   Just as with an on-demand scan, a scheduled scan will resume after a reboot, process interruption, or operating system crash.
3. (Windows only) To include removable media drives in the scheduled scan, enable the Cortex XDR agent to **Scan Removable Media Drives**.
4. **Whitelist Folders** from examination.
   
   1. Add (+) a folder.
   2. Enter the folder path. Use ? to match a single character or * to match any string of characters in the folder path (for example, C:\*\temp).
   3. Press Enter or click the check mark when done.
   4. Repeat to add additional folders.

STEP 8 | (Windows Vista and later Windows releases) Enable **Password Theft Protection**.
Select **Enabled** to enable the Cortex XDR agent to prevent attacks that use the Mimikatz tool to extract passwords from memory. When set to **Enabled**, the Cortex XDR agent silently prevents attempts to steal credentials (no notifications are provided when these events occur). The Cortex XDR agent enables this protection module following the next endpoint reboot. If you don’t want to enable the module, select **Disabled**.

*This module is supported with Traps agent 5.0.4 and later release. For agents managed by Cortex XDR, the default is Enabled.*

**STEP 9 | (Linux only) Configure Reverse Shell Protection.**

The Reverse Shell Protection module enables the Cortex XDR agent to detect and optionally block attempts to redirect standard input and output streams to network sockets.

1. Define the **Action Mode** to take when the Cortex XDR agent detects the malicious behavior.
   - **Block**—Block the activity.
   - **Report**—Allow the activity but report it to Cortex XDR.
   - **Disabled**—Disable the module and do not analyze or report the activity.

2. **(Optional)** Whitelist processes that must redirect streams to network sockets.
   - +Add a connection.
   - Enter the path of the process, and the local and remote IP address and ports.

   Use a wildcard to match a partial path name. Use a * to match any string of characters (for example, */bash*). You can also use a * to match any IP address or any port.

   ![Reverse Shell Protection](image)

   3. Press **Enter** or click the check mark when done.
   4. Repeat to add additional folders.

**STEP 10 | Save the changes to your profile.**

**STEP 11 | Apply Security Profiles to Endpoints.**

You can do this in two ways: You can **Create a new policy rule using this profile** from the right-click menu or you can launch the new policy wizard from **Policy Rules**.

**WildFire Analysis Concepts**

- File Forwarding
- File Type Analysis
- Verdicts
- Local Verdict Cache
File Forwarding

Cortex XDR sends unknown samples for in-depth analysis to WildFire. WildFire accepts up to 1,000,000 sample uploads per day and up to 1,000,000 verdict queries per day from each Cortex XDR tenant. The daily limit resets at 23:59:00 UTC. Uploads that exceed the sample limit are queued for analysis after the limit resets. WildFire also limits sample sizes to 100MB. For more information, see the WildFire documentation.

For samples that the Cortex XDR agent reports, the agent first checks its local cache of hashes to determine if it has an existing verdict for that sample. If the Cortex XDR agent does not have a local verdict, the Cortex XDR agent queries Cortex XDR to determine if WildFire has previously analyzed the sample. If the sample is identified as malware, it is blocked. If the sample remains unknown after comparing it against existing WildFire signatures, Cortex XDR forwards the sample for WildFire analysis.

File Type Analysis

The Cortex XDR agent analyzes files based on the type of file, regardless of the file’s extension. For deep inspection and analysis, you can also configure your Cortex XDR to forward samples to WildFire. A sample can be:

- Any Portable Executable (PE) file including (but not limited to):
  - Executable files
  - Object code
  - FON (Fonts)
  - Microsoft Windows screensaver (.scr) files
- Microsoft Office files containing macros opened in Microsoft Word (winword.exe) and Microsoft Excel (excel.exe):
  - Microsoft Office 2003 to Office 2016—.doc and .xls
  - Microsoft Office 2010 and later releases—.docm, .docx, .xlsm, and .xlsx
- Dynamic-link library file including (but not limited to):
  - .dll files
  - .ocx files
  - Android application package (APK) files
  - Mach-o files
  - Linux (ELF) files

For information on file-examination settings, see Add a New Malware Security Profile.

Verdicts

WildFire delivers verdicts to identify samples it analyzes as safe, malicious, or unwanted (grayware is considered obtrusive but not malicious):

- **Unknown**—Initial verdict for a sample for which WildFire has received but has not analyzed.
- **Benign**—The sample is safe and does not exhibit malicious behavior.
- **Malware**—The sample is malware and poses a security threat. Malware can include viruses, worms, Trojans, Remote Access Tools (RATs), rootkits, botnets, and malicious macros. For files identified as malware, WildFire generates and distributes a signature to prevent against future exposure to the threat.
- **Grayware**—The sample does not pose a direct security threat, but might display otherwise obtrusive behavior. Grayware typically includes adware, spyware, and Browser Helper Objects (BHOs).

When WildFire is not available or integration is disabled, the Cortex XDR agent can also assign a local verdict for the sample using additional methods of evaluation: When the Cortex XDR agent performs local analysis on a file, it uses machine learning to determine the verdict. The Cortex XDR agent can also compare the signer of a file with a local list of trusted signers to determine whether a file is malicious:

- Local analysis verdicts:
• **Benign**—Local analysis determined the sample is safe and does not exhibit malicious behavior.
• **Malware**—The sample is malware and poses a security threat. Malware can include viruses, worms, Trojans, Remote Access Tools (RATs), rootkits, botnets, and malicious macros.

**Trusted signer verdicts:**
• **Trusted**—The sample is signed by a trusted signer.
• **Not Trusted**—The sample is not signed by a trusted signer.

**Local Verdict Cache**

The Cortex XDR agent stores hashes and the corresponding verdicts for all files that attempt to run on the endpoint in its local cache. The local cache scales in size to accommodate the number of unique executable files opened on the endpoint. On Windows endpoints, the cache is stored in the `C:\ProgramData\Cyvera\LocalSystem` folder on the endpoint. When service protection is enabled (see Add a New Agent Settings Profile), the local cache is accessible only by the Cortex XDR agent and cannot be changed.

Each time a file attempts to run, the Cortex XDR agent performs a lookup in its local cache to determine if a verdict already exists. If known, the verdict is either the official WildFire verdict or manually set as a hash exception. Hash exceptions take precedence over any additional verdict analysis.

If the file is unknown in the local cache, the Cortex XDR agent queries Cortex XDR for the verdict. If Cortex XDR receives a verdict request for a file that was already analyzed, Cortex XDR immediately responds to the Cortex XDR agent with the verdict.

If Cortex XDR does not have a verdict for the file, it queries WildFire and optionally submits the file for analysis. While the Cortex XDR agent attempts waits for an official WildFire verdict, it can use File Analysis and Protection Flow to evaluate the file. After Cortex XDR receives the verdict it responds to the Cortex XDR agent that requested the verdict.

For information on file-examination settings, see Add a New Malware Security Profile.

**Add a New Restrictions Security Profile**

Restrictions security profiles limit the surface of an attack on a Windows endpoint by defining where and how your users can run files.

By default, the Cortex XDR agent will receive the default profile that contains a pre-defined configuration for each restrictions capability. To customize the configuration for specific Cortex XDR agents, configure a new Restrictions security profile and assign it to one or more policy rules.

To define a Restrictions security profile:

**STEP 1 | Add a new profile.**

1. From Cortex XDR, select **Endpoints > Policy Management > Profiles > + New Profile**.
2. Select the platform to which the profile applies and **Restrictions** as the profile type.
3. Click **Next**.

**STEP 2 | Define the basic settings.**

1. Enter a unique **Profile Name** to identify the profile. The name can contain only letters, numbers, or spaces, and must be no more than 30 characters. The name you choose will be visible from the list of profiles when you configure a policy rule.
2. To provide additional context for the purpose or business reason that explains why you are creating the profile, enter a profile **Description**. For example, you might include an incident identification number or a link to a help desk ticket.

**STEP 3 | Configure each of the Restrictions Endpoint Protection Capabilities.**

1. Configure the action to take when a file attempts to run from a specified location.
- **Block**—Block the file execution.
- **Notify**—Allow the file to execute but notify the user that the file is attempting to run from a suspicious location. The Cortex XDR agent also reports the event to Cortex XDR.
- **Report**—Allow the file to execute but report it to Cortex XDR.
- **Disabled**—Disable the module and do not analyze or report execution attempts from restricted locations.

2. Whitelist or blacklist files, as needed.

   The type of protection capability determines whether the capability supports a whitelist, blacklist, or both. With a whitelist, the action mode you configure applies to all the paths except for those that you specify. With a blacklist, the action applies only to the paths that you specify.

   1. Add (+Add) a file or folder.
   2. Enter the path and press **Enter** or click the check mark when done. You can also use a wildcard to match a partial name for the folder and environment variables. Use `?` to match any single character or `*` to match any string of characters. To match a folder, you must terminate the path with `*` to match all files in the folder (for example, `c:\temp\*`).
   3. Repeat to add additional folders.

**STEP 4** | Save the changes to your profile.

**STEP 5** | Apply Security Profiles to Endpoints.

You can do this in two ways: You can **Create a new policy rule using this profile** from the right-click menu or you can launch the new policy wizard from **Policy Rules**.

### Manage Security Profiles

After you customize your Endpoint Security Profiles, you can manage them from the Profiles page, as needed.

- View information about your security profiles
- Edit a security profile
-Duplicate a security profile
-View the security profile rules that use a security profile
- Populate a new policy rule with a security profile
- Delete a security profile

- View information about your security profiles.

The following table displays the fields that are available on the Profiles page in alphabetical order. The table includes both default fields and additional fields that are available in the column manager.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Created By</td>
<td>Administrative user who created the security profile.</td>
</tr>
<tr>
<td>Created Time</td>
<td>Date and time at which the security profile was created.</td>
</tr>
<tr>
<td>Description</td>
<td>Optional description entered by an administrator to describe the security profile.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Modification Time</td>
<td>Date and time at which the security profile was modified.</td>
</tr>
<tr>
<td>Modified By</td>
<td>Administrative user who modified the security profile.</td>
</tr>
<tr>
<td>Name</td>
<td>Name provided to identify the security profile.</td>
</tr>
<tr>
<td>Platform</td>
<td>Platform type of the security profile.</td>
</tr>
<tr>
<td>Summary</td>
<td>Summary of security profile configuration.</td>
</tr>
<tr>
<td>Type</td>
<td>Security profile type.</td>
</tr>
<tr>
<td>Usage Count</td>
<td>Number of policy rules that use the security profile.</td>
</tr>
</tbody>
</table>

- **Edit a security profile.**
  1. From **Endpoints > Policy Management > Profiles**, right-click the security profile and select **Edit**.
  2. Make your changes and then **Save** the security profile.

- **Duplicate a security profile.**
  1. From **Endpoints > Policy Management > Profiles**, right-click the security profile and select **Save as New**.
  2. Make your changes and then **Create** the security profile.
  3. Populate a new policy rule with a security profile.

- **View the security policy rules that use a security profile.**
  From **Endpoints > Policy Management > Profiles**, right-click the security profile and select **View security policies**.
  Cortex XDR displays the policy rules that use the profile.

- **Populate a new policy rule with a security profile.**
  1. From **Endpoints > Policy Management > Profiles**, right-click the security profile and **Create a new policy rule using this profile**.
  Cortex XDR automatically populates the **Platform** selection based on your security profile configuration and assigns the security profile based on the security profile type.
  2. Enter a descriptive **Policy Name** and optional description for the policy rule.
  3. Assign any additional security profiles that you want to apply to your policy rule, and select **Next**.
  4. Select the target endpoints for the policy rule or use the filters to define criteria for the policy rule to apply, and then select **Next**.
  5. Review the policy rule summary, and if everything looks good, select **Done**.

- **Delete a security profile.**
  1. If necessary, delete or detach any policy rules that use the profile before attempting to delete it.
  2. From **Endpoints > Policy Management > Profiles**, identify the security profile that you want to remove.
The Usage Count should have a 0 value.
3. Right-click the security profile and select Delete.
4. Confirm the deletion and you are done.
Customizable Agent Settings

Each Agent Settings Profile provides a tailored list of settings that you can configure for the platform you select.

In addition to the customizable Agent Settings Profiles, you can also set:

- **Global Agent Configurations** that apply to all the endpoints in your network.
- **Hardened Endpoint Security** protections that leverage existing mechanisms and added capabilities to reduce the attack surface on your endpoints.

The following table describes these customizable settings and indicates which platforms support the setting (a dash (—) indicates the setting is not supported).

<table>
<thead>
<tr>
<th>Setting</th>
<th>Windows</th>
<th>Mac</th>
<th>Linux</th>
<th>Android</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Agent Profiles</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Disk Space</strong></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>—</td>
</tr>
<tr>
<td>Customize the amount of disk space the Cortex XDR agent uses to store logs and information about events.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>User Interface</strong></td>
<td>✓</td>
<td>✓</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Determine whether and how end users can access the Cortex XDR console.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Traps Tampering Protection</strong></td>
<td>✓</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Prevent users from tampering with the Cortex XDR agent components by restricting access.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Uninstall Password</strong></td>
<td>✓</td>
<td>✓</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Change the default uninstall password to prevent unauthorized users from uninstalling the Cortex XDR agent software.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Windows Security Center Configuration</strong></td>
<td>✓</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Configure your Windows Security Center preferences to allow registration with the Microsoft Security Center, to allow registration with automated Windows</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Setting</td>
<td>Windows</td>
<td>Mac</td>
<td>Linux</td>
<td>Android</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>---------</td>
<td>-----</td>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>patch installation, or to disable registration.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Forensics</strong></td>
<td>✓</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Change forensic data collection and upload preferences.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Enhanced Data Collection</strong></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>—</td>
</tr>
<tr>
<td>Upload data collected about endpoint activity for EDR to the Cortex Data Lake for Cortex apps usage. This capability requires an Advanced Endpoint Protection XDR license and allocation of log storage in Cortex Data Lake</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Response Actions</strong></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>—</td>
</tr>
<tr>
<td>Manual response actions that you can take on the endpoint after a malicious file, process, or behavior is detected. For example, you can terminate a malicious process, isolate the infected endpoint from the network, quarantine a malicious file, or perform additional action as necessary to remediate the endpoint.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Content Updates</strong></td>
<td>✓</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Configure how the Cortex XDR agent performs content updates on the endpoint: whether to download the content directly from Cortex XDR or from a peer agent, whether to perform immediate or delayed updates, and whether to perform automatic content updates or continue using the current content version.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Agent Auto Upgrade</strong></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>—</td>
</tr>
<tr>
<td>Enable the agent to perform automatic</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Setting</td>
<td>Windows</td>
<td>Mac</td>
<td>Linux</td>
<td>Android</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------</td>
<td>-----</td>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>upgrades whenever a new agent version is released. You can choose to upgrade only to minor versions in the same line, only to major versions, or both.</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Upload Using Cellular Data</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enable Android endpoints to send unknown APK files for inspection as soon as a user connects to a cellular network.</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>✓</td>
</tr>
</tbody>
</table>

**Global Agent Configurations**

<table>
<thead>
<tr>
<th>Setting</th>
<th>Windows</th>
<th>Mac</th>
<th>Linux</th>
<th>Android</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Global Uninstall Password</strong></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>—</td>
</tr>
<tr>
<td>Set the uninstall password for all agents in the system.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Content Bandwidth Management</strong></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>—</td>
</tr>
<tr>
<td>Configure the total bandwidth to allocate for content update distribution within your organization.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Agent Auto Upgrade</strong></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>—</td>
</tr>
<tr>
<td>Configure the Cortex XDR agent auto upgrade scheduler and number of parallel upgrades.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Vulnerability Assessment and Application Inventory</strong></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>—</td>
</tr>
<tr>
<td>Configure the Cortex XDR agent to collect information about applications installed on the endpoint and compile an application inventory. Additionally, for Linux endpoints, this enables you to perform vulnerability assessment of the endpoint.</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>—</td>
</tr>
<tr>
<td>*Only Application Inventory is supported for this operating system</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Only Application Inventory is supported for this operating system
Add a New Agent Settings Profile

Agent Settings Profiles enable you to customize Cortex XDR agent settings for different platforms and groups of users.

STEP 1 | Add a new profile.
1. From Cortex XDR, select Endpoints > Policy Management > Profiles > + New Profile.
2. Select the platform to which the profile applies and Agent Settings as the profile type.
3. Click Next.

STEP 2 | Define the basic settings.
1. Enter a unique Profile Name to identify the profile. The name can contain only letters, numbers, or spaces, and must be no more than 30 characters. The name you choose will be visible from the list of profiles when you configure a policy rule.
2. To provide additional context for the purpose or business reason that explains why you are creating the profile, enter a profile Description. For example, you might include an incident identification number or a link to a help desk ticket.

STEP 3 | (Windows, Mac, and Linux only) Configure the Disk Space to allot for Cortex XDR agent logs. Specify a value in MB from 100 to 10,000 (default is 5,000).

STEP 4 | (Windows and Mac only) Configure User Interface options for the Cortex XDR console.
By default, Cortex XDR uses the settings specified in the default agent settings profile and displays the default configuration in parenthesis. When you select a setting other than the default, you override the default configuration for the profile.
- **Tray Icon**—Choose whether you want the Cortex XDR agent icon to be Visible (default) or Hidden in the notification area (system tray).
- **XDR Agent Console Access**—Enable this option to allow access to the Cortex XDR console.
- **XDR Agent User Notifications**—Enable this option to operate display notifications in the notifications area on the endpoint. When the Cortex XDR agent operates in silent mode where the Cortex XDR agent does not display any notifications in the notification area. If you enable notifications, you can use the default notification messages, or provide custom text (up to 50 characters) for each notification type. You can also customize a notification footer.
- **Live Terminal User Notifications**—Enable this option to display a pop-up on the endpoint when you initiate a Live Terminal session.

STEP 5 | (Android only) Configure network usage preferences.
When the option to Upload Using Cellular Data is enabled, the Cortex XDR agent uses cellular data to send unknown apps to the Cortex XDR for inspection. Standard data charges may apply. When this option is disabled, the Cortex XDR agent queues any unknown files and sends them when the endpoint connects to a Wi-Fi network. If configured, the data usage setting on the Android endpoint takes precedence over this configuration.

STEP 6 | (Windows only) Configure Agent Security options that prevent unauthorized access or tampering with the Cortex XDR agent components.
Use the default agent settings or customize them for the profile. To customize agent security capabilities:
1. Enable XDR Agent Tampering Protection.
2. By default, the Cortex XDR agent protects all agent components, however you can configure protection more granularly for Cortex XDR agent services, processes, files, and registry values. With
Traps 5.0.6 and later releases, when protection is enabled, access will be read-only. In earlier Traps releases, enabling protection disables all access to services, processes, files, and registry values.

**STEP 7 | (Windows and Mac only)** Set an **Uninstall Password**.

Define and confirm a password the user must enter to uninstall the Cortex XDR agent. The uninstall password is encrypted using encryption algorithm (PBKDF2) when transferred between Cortex XDR and Cortex XDR agents.

The default uninstall password is **Password1**. A new password must satisfy the following requirements:

- Contain eight or more characters.
- Contain English letters, numbers, or any of the following symbols: ! () - _ ` ~ @ # '

**STEP 8 | (Windows only)** Configure **Windows Security Center Integration**.

The Windows Security Center is a reporting tool that monitors the system health and security state of Windows endpoints on Windows 7 and later releases. When **Enabled**, the Cortex XDR agent registers with the Windows Security Center as an official Antivirus (AV) software product. When registration is **Disabled**, the Cortex XDR agent does not register to the Windows Action Center. As a result, Windows Action Center could indicate that Virus protection is **Off**, depending on other security products that are installed on the endpoint.

*For Cortex XDR agent 5.0 release only, if you want to register the agent to the Windows Security Center but prevent from Windows to automatically install Meltdown/Spectra vulnerability patches on the endpoint, change the setting to Enabled (No Patches).*

**STEP 9 | (Windows only)** Configure **Forensics** alert data collection options.

When the Cortex XDR agent alerts on process-related activity on the endpoint, the Cortex XDR agent collects the contents of memory and other data about the event in what is known as a alert data dump file. You can customize the **Alert Data Dump File Size**—**Small**, **Medium**, or **Full** (the largest and most complete set of information)—and whether to **Automatically Upload Alert Data Dump File** to Cortex XDR. During event investigation, if automatic uploading of the alert data dump file was disabled, you can manually retrieve the data.

**STEP 10 | (Windows, Mac, and Linux only)** Enable the Cortex XDR agent to **Monitor and Collect Enhanced Endpoint Data** for use by apps on the Cortex platform.

*Event monitoring and data collection requires:*

- A Cortex XDR Pro per Endpoint license.
- A supported agent version—Traps agent 6.0 or a later release for Windows endpoints and Traps agent 6.1 or later releases for Mac and Linux endpoints.
- Log storage allocated to EDR endpoint data in your Cortex Data Lake instance.

By default, the Cortex XDR agent collects information about events that occur on the endpoint. If you enable Behavioral Threat Protection in a Malware Security profile, the Cortex XDR agent also collects information about all active file, process, network, and registry activity on an endpoint (see **Endpoint Data Collected by Cortex XDR**).

When you enable the Cortex XDR agent to monitor and collect enhanced endpoint data, you enable Cortex XDR to share the detailed endpoint information with other Cortex apps. The information can help to provide the endpoint context when a security event occurs so that you can gain insight on the overall event scope during investigation. The event scope includes all activities that took place during an attack, the endpoints that were involved, and the damage caused. When disabled, the Cortex XDR agent will not share endpoint activity logs.
STEP 11 | *(Windows only)* **Response Actions.**

If you need to isolate an endpoint but want to allow access for a specific application (for example, communication between the VDI process and a VDI server), add the process to the **Network Isolation Whitelist**.

If your Cortex XDR agents communicate with Cortex XDR through a proxy, you must whitelist the following Cortex XDR agent processes along with the IP address of the proxy server:

- `C:\Program Files\Palo Alto Networks\Traps\tlaservice.exe`
- `C:\Program Files\Palo Alto Networks\Traps\cyveraservice.exe`

This enables the Cortex XDR agent to maintain communication with Cortex XDR after you isolate the endpoint.

When you whitelist a specific application from network isolation, the Cortex XDR agent continues to block some internal system processes. This is because some applications, for example ping.exe, can use other processes to facilitate network communication. As a result, if the Cortex XDR agent continues to block a whitelisted application, you may need to perform additional network monitoring to determine the process that facilitates the communication, and then whitelist that process.

1. Add an entry to the whitelist.
2. Specify the **Process Path** you want to allow and the IPv4 or IPv6 address of the endpoint. Use the * wildcard on either side to match any process or IP address. For example, specify * as the process path and an IP address to allow any process to run on the isolated endpoint with that IP address. Conversely, specify * as the IP address and a specific process path to allow the process to run on any isolated endpoint that receives this profile.
3. Click the check mark when finished.

STEP 12 | *(Windows only)* **Specify the Content Configuration** for your Cortex XDR agents.

You have several option to configure how your Cortex XDR agent retrieves new content.

- **Download Source**—Cortex XDR deploys serverless peer-to-peer P2P content distribution to Cortex XDR agents in your LAN network by default to reduce bandwidth loads. Within the six hour randomization window during which the Cortex XDR agent attempts to retrieve the new content version, it will broadcast its peer agents on the same subnet twice: once within the first hour, and once again during the following five hours. If the agent did not retrieve the new content from other agents in both queries, it will retrieve it from Cortex XDR directly. If you do not want to allow P2P content distribution, select the **Cortex Server** download source to allow all Cortex XDR agents in your network to retrieve the content directly from the Cortex XDR server on their following heartbeat.

To enable P2P, you must enable UDP and TCP over the defined **PORT** in **Content Download Source**. By default, Cortex XDR uses port 33221. You can configure another port number.

**Limitations in the content download process:**

- When you install the Cortex XDR agent, the agent retrieves the latest content update version available. A freshly installed agent can take between five to ten minutes (depending on your network and content update settings) to retrieve the content for the first time. During this time, your endpoint is not protected.
- When you upgrade a Cortex XDR agent to a newer Cortex XDR agent version, if the new agent cannot use the content version running on the endpoint, then the new content update will start within one minute in P2P and within five minutes from Cortex XDR.
• **Content Auto-update**—By default, the Cortex XDR agent always retrieves the most updated content and deploys it on the endpoint so it is always protected with the latest security measures. However, you can **Disable** the automatic content download. Then, the agent stops retrieving content updates from the Cortex XDR Server and keeps working with the current content on the endpoint.

  - If you disable content updates for a newly installed agent, the agent will retrieve the content for the first time from Cortex XDR and then disable content updates on the endpoint.
  - When you add a Cortex XDR agent to an endpoints group with disabled content auto-upgrades policy, then the policy is applied to the added agent as well.

• **Content Rollout**—The Cortex XDR agent can retrieve content updates **Immediately** as they are available, or after a pre-configured **Delayed** period. When you delay content updates, the Cortex XDR agent will retrieve the content according to the configured delay. For example, if you configure a delay period of two days, the agent will not use any content released in the last 48 hours.

  If you disable or delay automatic-content updates provided by Palo Alto Networks, it may affect the security level in your organization.

**STEP 13 | Enable Agent Auto Upgrade** for your Cortex XDR agents.

To ensure your endpoints are always up-to-date with the latest Cortex XDR agent release, enable automatic agent upgrades. For increased flexibility, you can choose to apply automatic upgrades to major releases only, to minor releases only, or to both. It can take up to 15 minutes for new and updated auto-upgrade profile settings to take effect on your endpoints.

Automatic agent upgrades are not supported with non-persistent VDI and temporary sessions.

To control the agent auto upgrade scheduler and number of parallel upgrades in your network, see Global Agent Configurations.

Automatic upgrades are not supported with non-persistent VDI and temporary sessions.

**STEP 14 | Enable Network Location Configuration** for your Cortex XDR agents.

(Requires Cortex XDR agents 7.1 and later releases) If you configure host firewall rules in your network, you must enable Cortex XDR to determine the network location of your device, as follows:

1. **A domain controller (DC) connectivity test**—When **Enabled**, the DC test checks whether the device is connected to the internal network or not. If the device is connected to the internal network, then it is in the organization. Otherwise, if the DC test failed or returned an external domain, Cortex XDR proceeds to a DNS connectivity test.

2. **A DNS test**—In the DNS test, the Cortex XDR agent submits a DNS name that is known only to the internal network. If the DNS returned the pre-configured internal IP, then the device is within the organization. Otherwise, if the DNS IP cannot be resolved, then the device is located elsewhere. Enter the **IP Address** and **DNS Server Name** for the test.

If the Cortex XDR agent detects a network change on the endpoint, the agent triggers the device location test, and re-calculates the policy according to the new location.

**STEP 15 | Save** the changes to your profile.

**STEP 16 | Apply Security Profiles to Endpoints.**

You can do this in two ways: You can **Create a new policy rule using this profile** from the right-click menu or you can launch the new policy wizard from **Policy Rules**.
Global Agent Configurations

On top of customizable Agent Settings Profiles for each Operating System and different endpoint targets, you can set global Agent Configurations that apply to all the endpoints in your network.

**STEP 1 |** From Cortex XDR, select 😡 > Settings > Agent Configuration.

**STEP 2 |** Set global uninstall password.

The uninstall password is required to remove a Cortex XDR agent and to grant access to agent security component on the endpoint. You can use the default uninstall Password1 defined in Cortex XDR or set a new one. This global uninstall password applies to all the endpoints (excluding mobile) in your network and can be defined only once upon initial startup. If you want to change it later on, you must set a new uninstall password in a specific Agent Settings profile which applies to a specific operating system and group of target agents. After you set a password, Save the configuration.

**STEP 3 |** Configure content bandwidth allocated for all endpoints.

To control the amount of bandwidth allocated in your network to Cortex XDR content updates, assign a Content bandwidth management value between 20-10,000 Mbps. Cortex XDR will verify that agents attempting to download the content update are within the allocated bandwidth before beginning the distribution. If the bandwidth has reached its cap, the download will be refused and the agents will attempt again at a later time. After you set the bandwidth, Save the configuration.

**STEP 4 |** Configure the Cortex XDR agent auto upgrade scheduler and number of parallel upgrades.

If Agent Auto Upgrades are enabled for your Cortex XDR agents, you can control the automatic upgrade process in your network:

- **Number of agents per batch**—Set the number of parallel agent upgrades, while the minimum is 500 agents.
- **Task scheduler**—You can schedule the upgrade task for specific days of the week and a specific time range. The minimum range is four hours.

**STEP 5 |** Enable Cortex XDR to perform a Vulnerability Assessment of the endpoint and compile a full Application Inventory.

When enabled, the Cortex XDR agent collects information about applications installed on the endpoint, which provides you a full application inventory of your network. For Linux endpoints and applications only, Cortex XDR provides you with CVE data, so you can further analyze the vulnerability status of all endpoints.

Endpoint Data Collected by Cortex XDR

When the Cortex XDR agent alerts on endpoint activity, the agent collects a minimum set of data about the endpoint as described in Data Collected for All Alerts.

When you enable behavioral threat protection or EDR data collection in your endpoint security policy, the Cortex XDR agent can also continuously monitor endpoint activity for malicious event chains identified by Palo Alto Networks. The endpoint data that the Cortex XDR agent collects when you enable these capabilities varies by the platform type:

- Additional Endpoint Data Collected for Windows Endpoints
- Windows Event Logs
- Additional Endpoint Data Collected for Mac Endpoints
- Additional Endpoint Data Collected for Linux Endpoints
Data Collected for All Alerts

When Cortex XDR raises an alert on an endpoint, the Cortex XDR agent collects the following data and sends it to Cortex XDR.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Absolute Timestamp</td>
<td>Kernel system time</td>
</tr>
<tr>
<td>Relative Timestamp</td>
<td>Uptime since the computer booted</td>
</tr>
<tr>
<td>Thread ID</td>
<td>ID of the originating thread</td>
</tr>
<tr>
<td>Process ID</td>
<td>ID of the originating process</td>
</tr>
<tr>
<td>Process Creation Time</td>
<td>Part of process unique ID per boot session (PID + creation time)</td>
</tr>
<tr>
<td>Sequence ID</td>
<td>Unique integer per boot session</td>
</tr>
<tr>
<td>Primary User SID</td>
<td>Unique identifier of the user</td>
</tr>
<tr>
<td>Impersonating User SID</td>
<td>Unique identifier of the impersonating user, if applicable</td>
</tr>
</tbody>
</table>

Additional Endpoint Data Collected for Windows Endpoints

<table>
<thead>
<tr>
<th>Category</th>
<th>Events</th>
<th>Attributes</th>
</tr>
</thead>
</table>
| Executable metadata (Traps 6.1 and later)     | Process start | • File size  
• File access time                                    |
| Files                                         |        | • Create  
• Write  
• Delete  
• Rename  
• Move  
• Modification (Traps 6.1 and later)  
• Symbolic links (Traps 6.1 and later) | • Full path of the modified file before and after modification  
• SHA256 and MD5 hash for the file after modification  
• SetInformationFile for timestamps (Traps 6.1 and later)  
• File set security (DACL) information (Traps 6.1 and later)  
• Resolve hostnames on local network (Traps 6.1 and later)  
• Symbolic-link/hard-link and reparse point creation (Traps 6.1 and later) |
| Image (DLL)                                   | Load   | • Full path  
• Base address  
• Target process-id/thread-id  
• Image size  
• Signature (Traps 6.1 and later) |
<table>
<thead>
<tr>
<th>Category</th>
<th>Events</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process</td>
<td>• Create</td>
<td>• SHA256 and MD5 hash for the DLL <em>(Traps 6.1 and later)</em></td>
</tr>
<tr>
<td></td>
<td>• Terminate</td>
<td>• File size <em>(Traps 6.1 and later)</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• File access time <em>(Traps 6.1 and later)</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Process ID (PID) of the parent process</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• PID of the process</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Full path</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Command line arguments</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Integrity level to determine if the process is running with elevated privileges</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Hash (SHA256 and MD5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Signature or signing certificate details</td>
</tr>
<tr>
<td>Thread</td>
<td>Injection</td>
<td>• Thread ID of the parent thread</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Thread ID of the new or terminating thread</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Process that initiated the thread if from another process</td>
</tr>
<tr>
<td>Network</td>
<td>• Accept</td>
<td>• Source IP address and port</td>
</tr>
<tr>
<td></td>
<td>• Connect</td>
<td>• Destination IP address and port</td>
</tr>
<tr>
<td></td>
<td>• Create</td>
<td>• Failed connection</td>
</tr>
<tr>
<td></td>
<td>• Listen</td>
<td>• Protocol (TCP/UDP)</td>
</tr>
<tr>
<td></td>
<td>• Close</td>
<td>• Resolve hostnames on local network</td>
</tr>
<tr>
<td></td>
<td>• Bind</td>
<td>• Network connection ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• IPv6 connection status (true/false)</td>
</tr>
<tr>
<td>Network Protocols</td>
<td>• DNS request and UDP response</td>
<td>• Origin country</td>
</tr>
<tr>
<td></td>
<td>• HTTP connect</td>
<td>• Remote IP address and port</td>
</tr>
<tr>
<td></td>
<td>• HTTP disconnect</td>
<td>• Local IP address and port</td>
</tr>
<tr>
<td></td>
<td>• HTTP proxy parsing</td>
<td>• Destination IP address and port if proxy connection</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Network connection ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• IPv6 connection status (true/false)</td>
</tr>
<tr>
<td>Network Statistics</td>
<td>• On-close statistics</td>
<td>• Upload volume on TCP link</td>
</tr>
<tr>
<td></td>
<td>• Periodic statistics</td>
<td>• Download volume on TCP link</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Traps sends statistics on connection close and periodically while connection is open</td>
</tr>
<tr>
<td>Registry</td>
<td>• Registry value:</td>
<td>• Registry path of the modified value or key</td>
</tr>
<tr>
<td></td>
<td>• Deletion</td>
<td></td>
</tr>
</tbody>
</table>
### Category: Events

- Set
- Registry key:
  - Creation
  - Deletion
  - Rename
  - Addition
  - Modification (set information)
- Restore
- Save

### Attributes

- Name of the modified value or key
- Data of the modified value

### Category: Session

- Log on
- Log off
- Connect
- Disconnect

### Attributes

- Interactive log-on to the computer
- Session ID
- Session State (equivalent to the event type)
- Local (physically on the computer) or remote (connected using a terminal services session)

### Category: Host Status

- Boot
- Suspend
- Resume

### Attributes

- Host name
- OS Version
- Domain
- Previous and current state

### User Presence (Traps 6.1 and later)

**User Detection**

Detection when a user is present or idle per active user session on the computer.

### Windows Event Logs

See the [Windows Event Logs](#) table for the list of Windows Event Logs that the agent can collect.

### Table 1: Windows Event Logs

<table>
<thead>
<tr>
<th>Path</th>
<th>Provider</th>
<th>Event IDs</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>EMET</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Application</td>
<td>Windows Error Reporting</td>
<td></td>
<td>WER events for application crashes only</td>
</tr>
<tr>
<td>Application</td>
<td>Microsoft-Windows-User Profiles Service</td>
<td>1511, 1518</td>
<td>User logging on with temporary profile (1511), Cannot create profile using temporary profile (1518)</td>
</tr>
<tr>
<td>Path</td>
<td>Provider</td>
<td>Event IDs</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-------------------------------</td>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Application</td>
<td>Application Error</td>
<td>1000</td>
<td>Application crash/hang events, similar to WER/1001. These include full path to faulting EXE/Module</td>
</tr>
<tr>
<td>Application</td>
<td>Application Hang</td>
<td>1002</td>
<td>Application crash/hang events, similar to WER/1001. These include full path to faulting EXE/Module</td>
</tr>
<tr>
<td>Microsoft-Windows-CAPI2/Operational</td>
<td></td>
<td>11, 70, 90</td>
<td>CAPI events Build Chain (11), Private Key accessed (70), X509 object (90)</td>
</tr>
<tr>
<td>Microsoft-Windows-DNS-Client/Operational</td>
<td></td>
<td>3008</td>
<td>DNS Query Completed (3008) without local machine name resolution events and without empty name resolution events</td>
</tr>
<tr>
<td>Microsoft-Windows-DriverFrameworks-UserMode/Operational</td>
<td></td>
<td>2004</td>
<td>Detect User-Mode drivers loaded - for potential BadUSB detection</td>
</tr>
<tr>
<td>Microsoft-Windows-PowerShell/Operational</td>
<td></td>
<td>4103, 4104, 4105, 4106</td>
<td>PowerShell execute block activity (4103), Remote Command (4104), Start Command (4105), Stop Command (4106)</td>
</tr>
<tr>
<td>Microsoft-Windows-TaskScheduler/Operational</td>
<td>Microsoft-Windows-TaskScheduler</td>
<td>106, 129, 141, 142, 200, 201</td>
<td></td>
</tr>
<tr>
<td>Microsoft-Windows-TerminalServices-RDPClient/Operational</td>
<td></td>
<td>1024</td>
<td>Log attempted TS connect to remote server</td>
</tr>
<tr>
<td>Microsoft-Windows-Windows Defender/Operational</td>
<td></td>
<td>1006, 1009</td>
<td>Modern Windows Defender event provider Detection events (1006 and 1009)</td>
</tr>
<tr>
<td>Microsoft-Windows-Windows Defender/Operational</td>
<td></td>
<td>1116, 1119</td>
<td>Modern Windows Defender event provider Detection events (1116 and 1119)</td>
</tr>
<tr>
<td>Security</td>
<td></td>
<td>4698, 4702</td>
<td></td>
</tr>
<tr>
<td>Path</td>
<td>Provider</td>
<td>Event IDs</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------------</td>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Security</td>
<td></td>
<td>4778, 4779</td>
<td>TS Session reconnect (4778), TS Session disconnect (4779)</td>
</tr>
<tr>
<td>Security</td>
<td></td>
<td>5140</td>
<td>Network share object access without IPC$ and Netlogon shares</td>
</tr>
<tr>
<td>Security</td>
<td></td>
<td>5140, 5142, 5144, 5145</td>
<td>Network Share create (5142), Network Share Delete (5144), A network share object was checked to see whether client can be granted desired access (5145), Network share object access (5140)</td>
</tr>
<tr>
<td>Security</td>
<td></td>
<td>4616</td>
<td>System Time Change (4616)</td>
</tr>
<tr>
<td>Security</td>
<td></td>
<td>4624</td>
<td>Local logons without network or service events</td>
</tr>
<tr>
<td>Security</td>
<td></td>
<td>1100, 1102</td>
<td>Security Log cleared events (1102), EventLog Service shutdown (1100)</td>
</tr>
<tr>
<td>Security</td>
<td></td>
<td>4647</td>
<td>User initiated logoff</td>
</tr>
<tr>
<td>Security</td>
<td></td>
<td>4634</td>
<td>User logoff for all non-network logon sessions</td>
</tr>
<tr>
<td>Security</td>
<td></td>
<td>4624</td>
<td>Service logon events if the user account isn't LocalSystem, NetworkService, LocalService</td>
</tr>
<tr>
<td>Security</td>
<td></td>
<td>5142, 5144</td>
<td>Network Share create (5142), Network Share Delete (5144)</td>
</tr>
<tr>
<td>Security</td>
<td></td>
<td>4688</td>
<td>Process Create (4688)</td>
</tr>
<tr>
<td>Security</td>
<td>Microsoft-Windows-Eventlog</td>
<td></td>
<td>Event log service events specific to Security channel</td>
</tr>
<tr>
<td>Security</td>
<td></td>
<td>4672</td>
<td>Special Privileges (Admin-equivalent Access) assigned to new logon, excluding LocalSystem</td>
</tr>
<tr>
<td>Security</td>
<td></td>
<td>4732</td>
<td>New user added to local security group</td>
</tr>
<tr>
<td>Security</td>
<td></td>
<td>4728</td>
<td>New user added to global security group</td>
</tr>
<tr>
<td>Security</td>
<td></td>
<td>4756</td>
<td>New user added to universal security group</td>
</tr>
</tbody>
</table>
## Path Provider Event IDs Description

<table>
<thead>
<tr>
<th>Path</th>
<th>Provider</th>
<th>Event IDs</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security</td>
<td></td>
<td>4733</td>
<td>User removed from local Administrators group</td>
</tr>
<tr>
<td>Security</td>
<td></td>
<td>4886, 4887, 4888</td>
<td>Certificate Services received certificate request (4886), Approved and Certificate issued (4887), Denied request (4888)</td>
</tr>
<tr>
<td>Security</td>
<td></td>
<td>4720, 4722, 4725, 4726</td>
<td>New User Account Created(4720), User Account Enabled (4722), User Account Disabled (4725), User Account Deleted (4726)</td>
</tr>
<tr>
<td>Security</td>
<td></td>
<td>4624</td>
<td>Network logon events</td>
</tr>
<tr>
<td>Security</td>
<td></td>
<td>4880, 4881, 4896, 4898</td>
<td>CA Service Stopped (4880), CA Service Started (4881), CA DB row(s) deleted (4896), CA Template loaded (4898)</td>
</tr>
<tr>
<td>Security</td>
<td></td>
<td>4634</td>
<td>Logoff events - for Network Logon events</td>
</tr>
<tr>
<td>Security</td>
<td></td>
<td>6272, 6280</td>
<td>RRAS events – only generated on Microsoft IAS server</td>
</tr>
<tr>
<td>Security</td>
<td></td>
<td>4689</td>
<td>Process Terminate (4689)</td>
</tr>
<tr>
<td>Security</td>
<td></td>
<td>4648, 4776</td>
<td>Local credential authentication events (4776), Logon with explicit credentials (4648)</td>
</tr>
</tbody>
</table>

### Additional Endpoint Data Collected for Mac Endpoints

<table>
<thead>
<tr>
<th>Category</th>
<th>Events</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Files</td>
<td>• Create • Write • Delete • Rename • Move • Open</td>
<td>• Full path of the modified file before and after modification • SHA256 and MD5 hash for the file after modification</td>
</tr>
<tr>
<td>Process</td>
<td>• Start • Stop</td>
<td>• Process ID (PID) of the parent process • PID of the process • Full path • Command line arguments</td>
</tr>
</tbody>
</table>
### Category | Events | Attributes
--- | --- | ---
**Network** | • Accept  
• Connect  
• Connect Failure  
• Disconnect  
• Listen  
• Statistics | • Integrity level to determine if the process is running with elevated privileges  
• Hash (SHA256 and MD5)  
• Signature or signing certificate details  
• Source IP address and port  
• Destination IP address and port  
• Failed connection  
• Protocol (TCP/UDP)  
• Aggregated send/receive statistics for the connection

### Additional Endpoint Data Collected for Linux Endpoints

| Category | Events | Attributes |
--- | --- | ---
**Files** | • Create  
• Open  
• Write  
• Delete  
• Copy  
• Move (rename)  
• Change owner (chown)  
• Change mode (chmod) | • Full path of the file  
• Hash of the file  

*For specific files only and only if the file was written.*  
• Full paths of both the original and the modified files  
• Full path of the file  
• Newly set owner/attributes

**Network** | • Listen  
• Accept  
• Connect  
• Connect failure  
• Disconnect | • Source IP address and port for explicit binds  
• Destination IP address and port  
• Failed TCP connections  
• Protocol (TCP/UDP)

**Process** | • Start  
• Stop | • PID of the child process  
• PID of the parent process  
• Full image path of the process  
• Command line of the process  
• Hash of the image (SHA256 & MD5)  
• PID of the stopped process
Apply Security Profiles to Endpoints

Cortex XDR provides out-of-the-box protection for all registered endpoints with a default security policy customized for each supported platform type. To tune your security policy, you customize settings in a security profile and attach the profile to a policy. Each policy that you create must apply to one or more endpoints or endpoint groups.

**STEP 1 |** From Cortex XDR, create a policy rule.
Do either of the following:
- Select **Endpoints > Policy Management > Policy Rules > + New Policy** to begin a rule from scratch.
- Select **Endpoints > Policy Management > Profiles**, right-click the profile you want to assign and **Create a new policy rule using this profile**.

**STEP 2 |** Define a **Policy Name** and optional **Description** that describes the purpose or intent of the policy.

**STEP 3 |** Select the **Platform** for which you want to create a new policy.

**STEP 4 |** Select the desired **Exploit**, **Malware**, **Restrictions**, and **Agent Settings** profiles you want to apply in this policy.

If you do not specify a profile, the Cortex XDR agent uses the default profile.

**STEP 5 |** Click **Next**.

**STEP 6 |** Use the filters to assign the policy to one or more endpoints or endpoint groups.
Cortex XDR automatically applies a filter for the platform you selected. To change the platform, go **Back** to the general policy settings.

**STEP 7 |** Click **Done**.

**STEP 8 |** In the **Policy Rules** table, change the rule position, if needed, to order the policy relative to other policies.
The Cortex XDR agent evaluates policies from top to bottom. When the Cortex XDR agent finds the first match it applies that policy as the active policy. To move the rule, select the arrows and drag the policy to the desired location in the policy hierarchy.

Right-click to View Policy Details, Edit, Save as New, Disable, and Delete.
Exceptions Security Profiles

To allow full granularity, Cortex XDR allows you to create exceptions from your baseline policy. These exceptions allow you to remove specific folders or paths from exemption or disable specific security modules. In Cortex XDR, you can configure the following types of policy exceptions:

<table>
<thead>
<tr>
<th>Exception Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process exceptions</td>
<td>Define an exception for a specific process for one or more security modules.</td>
</tr>
<tr>
<td>Support exceptions</td>
<td>Import an exception from the Cortex XDR Support team.</td>
</tr>
<tr>
<td>Behavioral Threat Protection Rule</td>
<td>An exception disabling a specific BTP rule across all processes.</td>
</tr>
<tr>
<td>Digital Signer Exception</td>
<td><em>(Windows only)</em> An exception adding a digital signer to the list of allowed signers.</td>
</tr>
<tr>
<td>Java Deserialization Exception</td>
<td><em>(Linux only)</em> An exception allowing specific Java executable (jar, class).</td>
</tr>
</tbody>
</table>

There are two types of exceptions you can create:

- Policy exceptions that apply to specific policies and endpoints (see Add a New Exceptions Security Profile)
- Global exceptions that apply to all policies (see Add a Global Endpoint Policy Exception)

To help you manage and assess your BIOC/IOC rules, Cortex XDR automatically creates a System Generated rule exception if the same BIOC/IOC rule is detected by the same initiator hash within a 3 day timeframe on 100 different endpoints.

Each time a BIOC/IOC alert is detected, the 3 day timeframe begins counting down. If after 3 days without an alert, the 3 day timeframe is reset. For example:

<table>
<thead>
<tr>
<th>Day Number</th>
<th>BIOC/IOC Detections</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example A</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>98 Detections</td>
<td>No exception created</td>
</tr>
<tr>
<td>2</td>
<td>1 Detection</td>
<td>No exception created</td>
</tr>
<tr>
<td>4</td>
<td>1 Detection</td>
<td>System Generated exception created</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Day Number</th>
<th>BIOC/IOC Detections</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example B</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>98 Detections</td>
<td>No exception created</td>
</tr>
<tr>
<td>2</td>
<td>1 Detection</td>
<td>No exception created</td>
</tr>
</tbody>
</table>
Add a New Exceptions Security Profile

You can configure exceptions that apply to specific groups of endpoints or you can Add a Global Endpoint Policy Exception. Use the following workflow to create an endpoint-specific exception:

**STEP 1 | Add a new profile.**
1. From Cortex XDR, select **Endpoints > Policy Management > Profiles > + New Profile**.
2. Select the platform to which the profile applies and **Exceptions** as the profile type.
3. Click **Next**.

**STEP 2 | Define the basic settings.**
1. Enter a unique **Profile Name** to identify the profile. The name can contain only letters, numbers, or spaces, and must be no more than 30 characters. The name you choose will be visible from the list of profiles when you configure a policy rule.
2. To provide additional context for the purpose or business reason that explains why you are creating the profile, enter a profile **Description**. For example, you might include an incident identification number or a link to a help desk ticket.

**STEP 3 | Configure the exceptions profile.**

**To configure a Process Exception:**
1. Select the operating system.
2. Enter the name of the process.
3. Select one or more **Endpoint Protection Modules** that will allow this process to run. The modules displayed on the list are the modules relevant to the operating system defined for this profile. To apply the process exception on all security modules, select **Select all**. To apply the process exception on all exploit security modules, select **Disable Injection**.
4. Click the adjacent arrow.
5. After you've added all processes, click **Create**.

You can return to the Process Exception profile from the **Endpoints Profile** page at any point and edit the settings, for example if you want to add or remove more security modules.

**To configure a Support Exception:**
1. Import the **json** file you received from Palo Alto Networks support team by either browsing for it in your files or by dragging and dropping the file on the page.
2. Click **Create**.

**To configure module specific exceptions:**

<table>
<thead>
<tr>
<th>Day Number</th>
<th>BIOC/IOC Detections</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>99 Detections</td>
<td>No exception created since detections were not within the 3 day timeframe</td>
</tr>
</tbody>
</table>
• Behavioral Threat Protection Rule Exception—When you view an alert for a Behavioral Threat event which you want to allow in your network from now on, right-click the alert and Create alert exception. Cortex XDR displays the alert data (Platform and Rule name). Select Exception Scope: Profile and select the exception profile name. Click Add.

• Digital Signer Exception—When you view an alert for a Digital Signer Restriction which you want to allow in your network from now on, right-click the alert and Create alert exception. Cortex XDR displays the alert data (Platform, Signer, and Generating Alert ID). Select Exception Scope: Profile and select the exception profile name. Click Add.

• Java Deserialization Exception—When you identify a Suspicious Input Deserialization alert that you believe to be benign and want to suppress future alerts, right-click the alert and Create alert exception. Cortex XDR displays the alert data (Platform, Process, Java executable, and Generating Alert ID). Select Exception Scope: Profile and select the exception profile name. Click Add.

At any point, you can click the Generating Alert ID to return to the original alert from which the exception was originated. You cannot edit module specific exceptions.

STEP 4 | Apply Security Profiles to Endpoints.

If you want to remove an exceptions profile from your network, go to the Profiles page, right-click and select Delete

Add a Global Endpoint Policy Exception

As an alternative to adding an endpoint-specific exception in policy rules, Cortex XDR enables you to define and manage global exceptions that apply across all of your network. On the Global Exception page, you can manage all the global exceptions in your network for all platforms. Global exceptions together with Exceptions Security Profiles constitute the sum of all the exceptions allowed within your network policy rules.

• Add a Global Process Exception
• Add a Global Support Exception
• Add a Global Behavioral Threat Protection Rule Exception
• Add a Global Digital Signer Exception
• Add a Global Java Deserialization Exception

Add a Global Process Exception

STEP 1 | Go to Endpoints > Policy Management > Policy Exceptions.

STEP 2 | Select Process exceptions.

1. Select the operating system.
2. Enter the name of the process.
3. Select one or more Endpoint Protection Modules that will allow this process to run. The modules displayed on the list are the modules relevant to the operating system defined for this profile. To apply the process exception on all security modules, Select all. To apply the process exception on all exploit security modules, select Disable Injection. Click the adjacent arrow to add the exception.

STEP 3 | After you add all exceptions, Save your changes.

The new process exception is added to the Global Exceptions in your network and will be applied across all rules and policies. To edit the exception, select it and click the edit icon. To delete it, select it and click the delete icon.

Add a Global Support Exception

STEP 1 | Go to Endpoints > Policy Management > Policy Exceptions.

STEP 2 | Select Support exceptions.

Import the json file you received from Palo Alto Networks support team by either browsing for it in your files or by dragging and dropping the file on the page.
STEP 3 | Click Save.

The new support exception is added to the Global Exceptions in your network and will be applied across all rules and policies.

Add a Global Behavioral Threat Protection Rule Exception

STEP 1 | Right-click the alert and select Create alert exception.

STEP 2 | Review the alert data (Platform and rule name) and select Exception Scope: Global.

STEP 3 | Click Add.

The relevant BTP exception is added to the Global Exceptions in your network and will be applied across all rules and policies. At any point, you can click the Generating Alert ID to return to the original alert from which the exception was originated. To delete a specific global exception, select it and click X. You cannot edit global exceptions generated from a BTP security event.
Add a Global Digital Signer Exception

**STEP 1** | Right-click the alert and select **Create alert exception**.
Review the alert data (Platform, signer, and alert ID) and select **Exception Scope: Global**.

You selected 'Digital Signer Restriction' alert to create an exception with the following parameters:

- **Platform**: Windows
- **Signers**: Wen Jia Liu
- **Generating Alert**: 244

**STEP 2** | Click **Add**.

The relevant digital signer exception is added to the **Global Exceptions** in your network and will be applied across all rules and policies. At any point, you can click the **Generating Alert ID** to return to the original alert from which the exception was originated. To delete a specific global exception, select it and click **X**. You cannot edit global exceptions generated from a digital signer restriction security event.

Add a Global Java Deserialization Exception

**STEP 1** | Right-click the alert and select **Create alert exception**.
Review the alert data (Platform, Process, Java executable, and alert ID) and select **Exception Scope: Global**.

<table>
<thead>
<tr>
<th>Platform</th>
<th>Signers</th>
<th>Generating Alert ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows</td>
<td>Wen Jia Liu</td>
<td>244</td>
</tr>
</tbody>
</table>

**Digital Signer Exception (1)**

An exception adding a digital signer to the list of allowed signers.
STEP 2 | Click Add.

The relevant digital signer exception is added to the **Global Exceptions** in your network and will be applied across all rules and policies. At any point, you can click the Generating Alert ID to return to the original alert from which the exception was originated. To delete a specific global exception, select it and click X. You cannot edit global exceptions generated from a digital signer restriction security event.
## Hardened Endpoint Security

Cortex XDR enables you to extend the security on your endpoints beyond the Cortex XDR agent built-in prevention capabilities to provide an increased coverage of network security within your organization. By leveraging existing mechanisms and added capabilities, the Cortex XDR agent can enforce additional protections on your endpoints to provide a comprehensive security posture.

The following table describes these additional protections and indicates which platforms support the setting. A dash (—) indicates the setting is not supported.

*Hardened endpoint security capabilities are not supported for Android endpoints.*

<table>
<thead>
<tr>
<th>Module</th>
<th>Windows</th>
<th>Mac</th>
<th>Linux</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Device Control</strong></td>
<td>✓</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Protects endpoints from loading malicious files from USB-connected removable devices (CD-ROM, disk drives, floppy disks and portable devices drives).</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Host Firewall</strong></td>
<td>✓</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Protects endpoints from attacks originating in network communications to and from the endpoint.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Disk Encryption Using BitLocker</strong></td>
<td>✓</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Grants visibility into endpoints that encrypt their hard drives using BitLocker, the Microsoft Windows built-in encryption tool.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Application Inventory</strong></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Compiles a complete list of all applications installed in your network.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Vulnerability Assessment</strong></td>
<td>—</td>
<td>—</td>
<td>✓</td>
</tr>
<tr>
<td>Identifies and quantifies the security vulnerabilities (CVEs) that exist for applications installed on your endpoints.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Device Control

By default, all external USB devices are allowed to connect to your Cortex XDR endpoints. To protect Windows endpoints from connecting USB-connected removable devices—such as disk drives, CD-ROM drives, floppy disk drives, and other portable devices—that can contain malicious files, Cortex XDR provides device control.

For example, with device control, you can:

- Block all supported USB-connected devices for an endpoint group
- Block a USB device type but whitelist a specific vendor from that list that will be accessible from the endpoint
- Temporarily block only some USB device types on an endpoint

Before you start applying device control policy rules, ensure you meet the following requirements and refer to these known limitations:

- The endpoint is running a Cortex XDR agent 7.0 or later release
- The endpoint is a Windows endpoint

You cannot enforce device control on VDI endpoints.

To apply device control in your organization, you define device control profiles that determine which device types Cortex XDR blocks and which it permits. There are two types of profiles:

<table>
<thead>
<tr>
<th>Profile</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Configuration Profile</strong></td>
<td>Allow or block these USB-connected device type groups:</td>
</tr>
<tr>
<td></td>
<td>• Disk Drives</td>
</tr>
<tr>
<td></td>
<td>• CD-Rom Drives</td>
</tr>
<tr>
<td></td>
<td>• Portable Devices</td>
</tr>
<tr>
<td></td>
<td>• Floppy Disk Drives</td>
</tr>
<tr>
<td></td>
<td><em>Cortex XDR relies on the device class assigned by the operating system.</em></td>
</tr>
<tr>
<td></td>
<td>Add a New Configuration Profile.</td>
</tr>
<tr>
<td></td>
<td>For example, you can define a Configuration profile to block the use of any USB Disk Drives on endpoints, and define an Exceptions profile to provide granular permissions to specific devices, products, or vendors.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Exceptions Profile</th>
<th>Whitelist specific devices according to device types and vendor. You can further specify a specific product and/or product serial number.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Add a New Exceptions Profile.</td>
</tr>
</tbody>
</table>
After you configure a device control profile, Apply Device Control Profiles to Your Endpoints.

**Add a New Configuration Profile**

**STEP 1 |** Log in to Cortex XDR.

   Go to Endpoints > Policy management > Extension Profiles and select + New Profile > Device Configuration.

**STEP 2 |** Fill in the General Information.

   Assign the profile Name and add an optional Description. The profile Type and Platform are set by Cortex XDR.

**STEP 3 |** Configure the Device Configuration.

   For each group of device types, select whether to allow or block them on the endpoints. To use the default option defined by Palo Alto Networks, leave Use Default selected.

   *Currently, the default is set to Use Default (Allow) however Palo Alto Networks may change the default definition at any time.*

**STEP 4 |** Save your profile.

   When you’re done, Create your device profile definitions.

   If needed, you can edit, delete, or duplicate your profiles.

   *You cannot edit or delete the default profiles pre-defined in Cortex XDR.*

**STEP 5 |** *(Optional)* To define exceptions to your Device Configuration profile, Add a New Exceptions Profile.

**STEP 6 |** Apply Device Control Profiles to Your Endpoints.

**Add a New Exceptions Profile**

**STEP 1 |** Log in to Cortex XDR.

   Go to Endpoints > Policy management > Extension Profiles and select + New Profile > Device Exceptions.

**STEP 2 |** Fill in the General Information.

   Assign the profile Name and add an optional Description. The profile Type and Platform are set by the system.

**STEP 3 |** Configure Device Exceptions.

   You can whitelist devices according to different sets of identifiers—vendor, product, and serial numbers.

   - *(Disk Drives only)* Permission—Select the permissions you want to grant: Read or Read + Write.
   - **Type**—Select the Device Type you want to whitelist (Disk Drives, CD-Rom, Portable, or Floppy Disk).
   - **Vendor**—Select a specific vendor from the list or enter the vendor ID in hexadecimal code.
   - *(Optional)* **Product**—Select a specific product (filtered by the selected vendor) to whitelist only this product, or add your product ID in hexadecimal code.
- **(Optional) Serial Number**—Enter a specific serial number (pertaining to the selected product) to whitelist only devices with this serial number.

**STEP 4 | Save your profile.**

When you're done, **Create** your device exceptions profile.

If needed, you can later edit, delete, or duplicate your profiles.

*You cannot edit or delete the predefined profiles in Cortex XDR.*

**STEP 5 | Apply Device Control Profiles to Your Endpoints.**

**Apply Device Control Profiles to Your Endpoints**

After you defined the required profiles for Device Configuration and Exceptions, you must configure Device Control Policies and enforce them on your endpoints. Cortex XDR applies Device Control policies on endpoints from top to bottom, as you've ordered them on the page. The first policy that matches the endpoint is applied. If no policies match, the default policy that enables all devices is applied.

**STEP 1 | Log in to Cortex XDR.**

Go to **Endpoints > Policy management > Extension Policy Rules** and select **+ New Policy.**

**STEP 2 | Configure settings for the Device Control policy.**

1. Assign a policy name and optional description.
2. Assign the Device Type profile you want to use in this rule.
3. If desired, assign an Device Exceptions profile.
4. Click **Next**.
5. Select the target endpoints on which to enforce the policy.
   - Use filters or manual endpoint selection to define the exact target endpoints of the policy rules.
6. Click **Done**.

**STEP 3 | Configure policy hierarchy.**

Drag and drop the policies in the desired order of execution. The default policy that enables all devices on all endpoints is always the last one on the page and is applied to endpoints that don’t match the criteria in the other policies.

**STEP 4 | Save the policy hierarchy.**

After the policy is saved and applied to the agents, Cortex XDR enforces the device control policies on your environment.

**STEP 5 | (Optional) Manage your policy rules.**

In the **Protection Policy Rules** table: you can view and edit the policy you created and the policy hierarchy.

1. View your policy hierarchy.
2. Right-click to **View Policy Details, Edit, Save as New, Disable**, and **Delete**.

**STEP 6 | Monitor device control violations.**
After you apply Device Control rules in your environment, use the **Endpoints > Device Control Violations** page to monitor all instances where end users attempted to connect restricted USB-connected devices and Cortex XDR blocked them on the endpoint. All violation logs are displayed on the page. You can sort the results, and use the filters menu to narrow down the results. For each violation event Cortex XDR logs the event details (timestamp, host, user, and IP address) and the device details that are available (type, product, vendor, and serial number).

If you see a violation for which you’d like to define an exception on the device that triggered it, right-click the violation and select one of the following options:

- **Add device to permanent exceptions**—Select this option to add the device to the Permanent Exceptions whitelist in your network. This ensures that the device will always be allowed in your network.
- **Add device to temporary exceptions**—Select this option to temporarily allow the device on the selected endpoint or on all endpoints in your network for a time frame that you choose.
- **Allow device to a profile exception**—Select this option to allows the device within an existing exceptions profile.

**STEP 7 | Tune your device control exceptions.**

To better deploy device control in your network and allow further granularity, you can whitelist devices on your network and allow them access to your endpoints. Device control exceptions are configured per device and you must select the device category, vendor, and type of permission that you want to allow on the endpoint. Optionally, to limit the exception to a specific device, you can also include the product and/or serial number.

Cortex XDR enables you to configure the following exceptions:

<table>
<thead>
<tr>
<th>Exception Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Permanent Exceptions</td>
<td>Permanent exceptions whitelist the device in your network across all Device Control policies and profiles. You can create them directly from the violation event that blocked the device, or through the Permanent Exceptions list.</td>
</tr>
<tr>
<td></td>
<td>Create a Permanent Exception.</td>
</tr>
<tr>
<td>Temporary Exceptions</td>
<td>Temporary exceptions whitelist the device for a specific time period up to 30 days. You create a temporary exception directly from the violation event that blocked the device.</td>
</tr>
<tr>
<td></td>
<td>Create a Temporary Exception.</td>
</tr>
<tr>
<td>Profile Exceptions</td>
<td>Profile exceptions whitelist the device in an existing exceptions profile. You create a profile exception directly from the violation event that blocked the device.</td>
</tr>
<tr>
<td></td>
<td>Create a Profile Exception.</td>
</tr>
</tbody>
</table>

1. **Create a Permanent Exception.**

   Permanent device control exceptions are managed in the Permanent Exception list.

   - If you know in advance which device you’d like to allow throughout your network, create a general exception from the list:
1. Go to **Endpoints > Policy Management > Extensions** and select **Device Permanent Exceptions** on the left menu. The list of existing Permanent Exceptions is displayed.

2. Select: **Type, Permission, and Vendor**.

3. (Optional) Select a specific product and/or enter a specific serial number for the device.

4. Click the adjacent arrow and **Save**. The exception is added to the Permanent Exceptions list and will be applied in the next heartbeat.

- Otherwise, you can create a permanent exception directly from the violation event that blocked the device in your network:
  1. On the **Device Control Violations** page, right-click the violation event triggered by the device you want to permanently allow.
  2. Select **Add device to permanent exceptions**. Review the exception data and change the defaults if necessary.
  3. Click **Save**.

2. **Create a Temporary Exception.**

   1. On the **Device Control Violations** page, right-click the violation event triggered by the device you want to temporarily allow.
   2. Select **Add device to temporary exceptions**. Review the exception data and change the defaults if necessary. For example, you can configure the exception to this endpoint only or to all endpoints in your network, or set which device identifiers will be included in the exception.
   3. Configure the exception **TIME FRAME** by defining the number of days or number of hours during which the exception will be applied, up to 30 days.
   4. Click **Save**. The exception is added to the Device Temporary Exceptions list and will be applied in the next heartbeat.

3. **Create an Exception within a Profile.**

   1. On the **Device Control Violations** page, right-click the violation event triggered by the device you want to add to a Device Exceptions profile.
   2. Select the **PROFILE** from the list.
   3. Click **Save**. The exception is added to the Exceptions Profile and will be applied in the next heartbeat.

### Host Firewall

The Cortex XDR host firewall enables you to control inbound and outbound communications on your Windows endpoints. To use the host firewall, you set rules that allow or block the traffic on the devices and apply them to your endpoints using Cortex XDR host firewall policy rules. Host firewall rules can be set on the endpoint according to the following conditions:

- Communication direction (Inbound or outbound)
- IP address or ranges
- Ports or ranges
- Protocol (ICMP, TCP, UCP, and ICMPv6)
- Program path

Additionally, you can configure different sets of rules based on the current location of your endpoints - within or outside your organization network.

- **Enable Network Location Configuration**
- **Add a New Host Firewall Profile**
- **Apply Host Firewall Profiles to Your Endpoints**
- **Monitor the Host Firewall Activity on your Endpoint**
Before you start applying host firewall policy rules, ensure you meet the following requirements and refer to these known limitations:

- The endpoint is running a Cortex XDR agent 7.1 or later release
- The endpoint is a Windows endpoint

Enable Network Location Configuration

If you want to apply location based host firewall rules, you must first enable network location configuration in your Agent Settings Profile.

When enabled, Cortex XDR performs the following to determine the endpoint location:

1. A domain controller (DC) connectivity test to check whether the device is connected to the internal network or not. If the device is connected to the internal network, then it is in the organization. Otherwise, if the DC test failed or returned an external domain, Cortex XDR proceeds to a DNS connectivity test.

2. In the DNS test, the Cortex XDR agent submits a DNS name that is known only to the internal network. If the DNS returned the pre-configured internal IP, then the device is within the organization. Otherwise, if the DNS IP cannot be resolved, then the device is located outside.

In every heartbeat, and if the Cortex XDR agent detects a network change on the endpoint, the agent triggers the device location test and re-calculates the policy according to the new location.

Add a New Host Firewall Profile

STEP 1 | Log in to Cortex XDR.

Go to Endpoints > Policy Management > Extensions Profiles and select + New Profile > Host Firewall.

STEP 2 | Fill-in the general information for the new profile.

- Assign a name and an optional description to the profile.
- By default, host firewall profile rules are based on the current location of your device. Configure two sets of rules: a set of External Rules that apply when the device is located outside the internal organization network, and a set of Internal Rules that apply when the device is located within the internal organization network. If you disable the Location Based option, your policy will apply the internal set of rules only, and that will be applied to the device regardless of its location.
STEP 3 | Create host firewall rules.

Click +New Rule. A host firewall rule allows or blocks the communication to and/or from the endpoint. Additionally, you can fine tune the rule by applying the action to specific IPs, ports, protocols, or software programs. To create a new rule, configure the following:

**New External Rule**

*Define the action, direction and at least one more setting to create a new rule*

<table>
<thead>
<tr>
<th>ACTION</th>
<th>LOCAL ADDRESS</th>
<th>LOCAL PORT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select</td>
<td>Any</td>
<td>Any</td>
</tr>
<tr>
<td>IPv4</td>
<td>IP address or a range with a hyphen</td>
<td>Any</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>REMOTE ADDRESS</th>
<th>REMOTE PORT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Any</td>
<td>Any</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DIRECTION</th>
<th>PROTOCOL</th>
<th>PATH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select</td>
<td>Any</td>
<td>Any</td>
</tr>
</tbody>
</table>

- **Action**—Select whether to Allow or Block the communication on the endpoint.
- **Specific IPs and Ports**—*(Optional)* Configure the rule for specific local or remote IPs and/or Ports. You can also set a range of IP addresses.
- **Direction**—Select the direction of the communication this rule applies to:
  - **Inbound**—Communication to the endpoint.
  - **Outbound**—Communication from the endpoint.
  - **Both**—The rule applies to both inbound and outbound communication.
- **Protocol**—*(Optional)* Select a specific protocol you want this rule to apply to.
- **Path**—*(Optional)* Enter the full path and name of a program you want the rule to apply to. If you use system variables in the path definition, you must re-enforce the policy on the endpoint every time the directories and/or system variables on the endpoint change.

If the profile is location based, you can define both internal and external rules at this point. You can also copy a rule from one set to another.

STEP 4 | Save your profile.

When you’re done, Create your host firewall profile.
STEP 5 | Apply Host Firewall Profiles to Your Endpoints.

Apply Host Firewall Profiles to Your Endpoints

After you defined the required host firewall profiles, you must configure the Protection Policies and enforce them on your endpoints. Cortex XDR applies Protection policies on endpoints from top to bottom, as you’ve ordered them on the page. The first policy that matches the endpoint is applied. If no policies match, the default policy that enables all communication to and form the endpoint is applied.

STEP 1 | Log in to Cortex XDR.

Go to Endpoints > Policy Management > Extensions Policy Rules > +New Policy.

STEP 2 | Configure settings for the host firewall policy.

1. Assign a policy name and optional description.
   - The platform will automatically be assigned to Windows.
2. Assign the host firewall profile you want to use in this rule.
3. If desired, assign Device Configuration and/or Device Exceptions and/or Host Firewall profiles. If none are assigned, the default profiles will be applied.
4. Click Next.
5. Select the target endpoints on which to enforce the policy.
   - Use filters or manual endpoint selection to define the exact target endpoints of the policy rules.
6. Click Done.

Alternatively, you can associate the host firewall profile to an existing policy. Right-click the policy and select Edit. Select the Host Firewall profile and click Next. If needed, you can edit other settings in the rule (such as target endpoints, description, etc.) When you’re done, click Done.

STEP 3 | Configure policy hierarchy.

Drag and drop the policies in the desired order of execution.

STEP 4 | Save the policy hierarchy.

After the policy is saved and applied to the agents, Cortex XDR enforces the host firewall policies on your environment.

Monitor the Host Firewall Activity on your Endpoint

The Cortex XDR Host Firewall leverages the Microsoft Windows Filtering Platform (WFP). To monitor the communication on your endpoint, you can use a monitoring tool such as Network Shell (netsh), the Microsoft Windows command-line utility for monitoring the network communication on an endpoint.

Additionally, to view only the communication events on the endpoint to which the Cortex XDR host firewall rules were applied, you can run the Cytool firewall show command.

Disk Encryption Using BitLocker

Cortex XDR provides full visibility into your Windows endpoints that were encrypted using BitLocker, and lists all the encrypted drives. Additionally, you can apply BitLocker encryption or decryption on the endpoints system drive by creating disk encryption rules and policies.

- Monitor the Endpoint Encryption Status in Cortex XDR
- Configure a Disk Encryption Profile
- Apply Disk Encryption Profile to Your Endpoints
Before you start applying disk encryption policy rules, ensure you meet the following requirements and refer to these known limitations:

- **Endpoint pre-requisites:**
  - The endpoint is running a Microsoft Windows version that supports BitLocker.
  - The endpoint is within the organization network domain.
  - The endpoint is running a Cortex XDR agent 7.1 or later release.
  - To allow the agent to encrypt the endpoint, Trusted Platform Module (TPM) must be supported and enabled on the endpoint.
  - To allow the agent to access the encryption recovery key backup, Active Directory Domain Services must be enabled on the endpoint.

- **Disk encryption scope:**
  - You can enforce XDR disk encryption policy rules only on the Operating System volume.

- **Group Policy configuration:**
  - Make sure the GPO configuration applying to the endpoint enables **Save BitLocker recovery information to AD DS for operating system drives**.
  - Make sure your Cortex XDR disk encryption policy does not conflict with the GPO configuration to **Choose drive encryption method and cipher strength**.

**Monitor the Endpoint Encryption Status in Cortex XDR**

You can monitor the **Encryption Status** of an endpoint in the new **Endpoints > Disk Encryption Visibility** table. The table lists both system and custom drives that were encrypted using BitLocker for each endpoint, including the encryption method.

The following table describes both the default and additional optional fields that you can view in the **Disk Encryption Visibility** table per endpoint. The fields are in alphabetical order.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Endpoint ID</td>
<td>Unique ID assigned by Cortex XDR that identifies the endpoint.</td>
</tr>
<tr>
<td>Encryption Status</td>
<td>The endpoint encryption status can be:</td>
</tr>
</tbody>
</table>

![Disk Encryption Visibility Table](image-url)
Field | Description
--- | ---
Endpoint Name | Hostname of the endpoint.
Endpoint Status | The status of the endpoint. For more details, see View Details About an Endpoint.
IP Address | Last known IPv4 or IPv6 address of the endpoint.
Last Reported | Date and time of the last change in the agent’s status. For more details, see View Details About an Endpoint.
MAC Address | The MAC address of the endpoint.
Operating System | The platform running on the endpoint.
OS Version | Name of the operating system version running on the endpoint.
Volume Status | Lists all the disks on the endpoint along with the status per volume, Decrypted or Encrypted.

You can also monitor the endpoint Encryption Status in your Endpoint Administration table. If the Encryption Status is missing from the table, add it.

Configure a Disk Encryption Profile

**STEP 1 | Log in to Cortex XDR.**

Go to Endpoints > Policy Management > Extensions Profiles and select + New Profile > Disk Encryption.
STEP 2 | Fill-in the general information for the new profile.
Assign a name and an optional description to the profile.

STEP 3 | Enable disk encryption.
To enable the Cortex XDR agent to apply disk encryption rules using BitLocker, Enable the Disk encryption BitLocker management option.

STEP 4 | Configure Encryption details.

You can encrypt and decrypt only system drives.

• Encrypt or decrypt the system drives.
• Encrypt the entire disk or only the used disk space.

STEP 5 | Specify the Encryption methods per operating system.
For each operating system (Windows 7, Windows 8-10, Windows 10 (1511) and above), select the encryption method from the corresponding list.

You must select the same encryption method configured by the Microsoft Windows Group Policy in your organization for the target endpoints. Otherwise, if you select a different encryption method than the one already applied through the Windows Group Policy, Cortex XDR will display errors.

STEP 6 | Save your profile.
When you're done, Create your disk encryption profile.

STEP 7 | Apply Disk Encryption Profile to Your Endpoints.

Apply Disk Encryption Profile to Your Endpoints

After you defined the required disk encryption profiles, you must configure the Protection Policies and enforce them on your endpoints. Cortex XDR applies Protection policies on endpoints from top to bottom,
as you’ve ordered them on the page. The first policy that matches the endpoint is applied. If no policies match, the default policy that enables all communication to and form the endpoint is applied.

**STEP 1 | Log in to Cortex XDR.**

Go to **Endpoints > Policy Management > Extensions Policy Rules > +New policy.**

**STEP 2 | Configure settings for the disk encryption policy.**

1. Assign a policy name and optional description.
   - The platform will automatically be assigned to Windows.
2. Assign the disk encryption profile you want to use in this rule.
3. If desired, assign **Device Configuration** and/or **Device Exceptions** profiles and/or **Host Firewall** profiles. If none are assigned, the default profiles will be applied.
4. Click **Next**.
5. Select the target endpoints on which to enforce the policy.
   - Use filters or manual endpoint selection to define the exact target endpoints of the policy rules.
6. Click **Done**.

Alternatively, you can associate the disk encryption profile to an existing policy. Right-click the policy and select **Edit**. Select the **Disk Encryption** profile and click **Next**. If needed, you can edit other settings in the rule (such as target endpoints, description, etc.) When you’re done, click **Done**.

**STEP 3 | Configure policy hierarchy.**

Drag and drop the policies in the desired order of execution.

**STEP 4 | Save** the policy hierarchy.

After the policy is saved and applied to the agents, Cortex XDR enforces the disk encryption policies on your environment.

**STEP 5 | Now, Monitor the Endpoint Encryption Status in Cortex XDR**

**Application Inventory**

Cortex XDR compiles an application inventory of all the applications installed in your network by collecting from each Cortex XDR agent the list of installed applications.

The following are prerequisites to enable Cortex XDR to retrieve a list of all the applications installed on your endpoints:

- Provision an active Cortex XDR Pro per Endpoint license
- Ensure that you are running a Cortex XDR agent 7.1 or later release

*Application inventory is supported for Windows, Mac, and Linux endpoints. However, because Cortex XDR performs Vulnerability Assessment for Linux endpoints only, no CVE information is available for Windows or Mac applications in the Application Inventory, and Windows and Mac applications are marked as Unsupported Platform.*

- Enable **EDR data collection** on the endpoint
- Enable **application data collection** on the endpoint.

Any new application installed on the endpoint will appear in Cortex XDR with 24 hours. Alternatively, you can re-scan the endpoint to retrieve the most updated list.
During the first few days of the feature roll-out and until Cortex XDR collects the application data from all endpoints in your network, you will see only partial information in Application Inventory and a system notification that indicates the data is still being collected. When Cortex XDR completes the data collection, it will stop displaying the system notification.

From Endpoints > Vulnerability Assessment, select Apps on the upper-right bar. For each application, Cortex XDR displays the following default and optional values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Affected endpoints</td>
<td>The number of endpoints that are currently affected by this CVE.</td>
</tr>
<tr>
<td>Application name</td>
<td>The name of the application affected by this CVE.</td>
</tr>
<tr>
<td>CVEs</td>
<td>A list of all CVEs that exist on applications that are installed on the endpoint.</td>
</tr>
<tr>
<td>(Linux endpoints only)</td>
<td></td>
</tr>
<tr>
<td>Platform</td>
<td>A list of all platforms on which the application is installed.</td>
</tr>
<tr>
<td>Severity</td>
<td>The severity level (High, Medium, or Low) of the CVE as ranked in the NIST database.</td>
</tr>
<tr>
<td>(Linux endpoints only)</td>
<td></td>
</tr>
<tr>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Severity score</strong></td>
<td>The CVE severity score based on the NIST Common Vulnerability Scoring System (CVSS). Click the score to see the full CVSS description.</td>
</tr>
<tr>
<td>(Linux endpoints only)</td>
<td></td>
</tr>
<tr>
<td><strong>Version</strong></td>
<td>The version of the installed application.</td>
</tr>
</tbody>
</table>

To view the details of all the endpoints in your network on which an application is installed, right click the application and View Endpoints.

- **The number of affected endpoints in the application inventory is updated every four hours.**

  Because Cortex XDR agents report their application inventory to Cortex XDR at different times within this four-hour window, the number of affected endpoints in the application inventory are sometimes different (and less accurate) than the number of endpoints you see when you view the endpoints list.

**Vulnerability Assessment**

Use the vulnerability assessment to identify and quantify security vulnerabilities for the applications installed on Linux endpoints in your network. Cortex XDR provides full visibility in to all applications installed on each endpoint, detects the presence of existing Common Vulnerabilities and Exposures (CVEs), and retrieves the latest data for each CVE from the NIST National Vulnerability Database to help you with your analysis and prioritization. You can use the vulnerability assessment in Cortex XDR to easily mitigate and patch these vulnerabilities on all endpoints in your network.

The following are prerequisites to enable Cortex XDR to perform a vulnerability assessment of your Linux endpoints:

- Provision an active Cortex XDR Pro per Endpoint license
- Ensure that you are running a Cortex XDR agent 7.1 or later release
- Ensure the endpoint is a Linux endpoint
- Enable EDR data collection on the endpoint
- Enable vulnerability assessment for your Cortex XDR agent.

To detect existing vulnerabilities, the Cortex XDR agent provides Cortex XDR with the name and version of all applications installed on the endpoint. Every four hours, Cortex XDR correlates the network application inventory with the data from the NIST public database and displays the information in Cortex XDR. If Cortex XDR detects a new CVE during data correlation, it creates an alert and generates an incident in Cortex XDR (only one alert per CVE). The alerts help you to recognize new risks in your network, to follow-up and remediate them, and to associate other security events with security patching problems in your network.

Additionally, you can use Cortex XDR to evaluate the extent and severity of each CVE in your network, gain full visibility in to the risks to which each endpoint is exposed, and assess the vulnerability status of an installed application in your network.
During the first few days of this feature roll-out and until Cortex XDR collects the application data from all endpoints in your network, you will see only partial information in Vulnerability Assessment and a system notification that indicates the data is still being collected. When Cortex XDR completes the data collection, it will stop displaying the system notification.

**CVE Analysis**

To evaluate the extent and severity of each CVE across your endpoints, you can drill down in to each CVE in Cortex XDR and view all the endpoints and applications in your environment that are impacted by the CVE. Cortex XDR retrieves the latest information from the NIST public database every 24 hours.

From **Endpoints > Vulnerability Assessment**, select **CVEs** on the upper-right bar. For each vulnerability, Cortex XDR displays the following default and optional values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Affected endpoints</td>
<td>The number of endpoints that are currently affected by this CVE.</td>
</tr>
<tr>
<td>Applications</td>
<td>The names of the applications affected by this CVE.</td>
</tr>
<tr>
<td>CVE</td>
<td>The name of the CVE.</td>
</tr>
<tr>
<td>Description</td>
<td>The general NIST description of the CVE.</td>
</tr>
<tr>
<td>Platforms</td>
<td>The name and version of the operating system affected by this CVE.</td>
</tr>
<tr>
<td>Severity</td>
<td>The severity level (High, Medium, or Low) of the CVE as ranked in the NIST database.</td>
</tr>
<tr>
<td>Severity score</td>
<td>The CVE severity score based on the NIST Common Vulnerability Scoring System (CVSS). Click the score to see the full CVSS description.</td>
</tr>
</tbody>
</table>

For detailed information about the endpoints in your network that are impacted by a CVE, right-click the CVE and **View affected endpoints**.
To learn more about the application in your network that is impacted by a CVE, right-click the CVE and View application.

**Endpoint Analysis**

To help you assess the vulnerability status of an endpoint, Cortex XDR provides a full list of all installed applications and existing CVEs per endpoint and also assigns each endpoint a vulnerability severity score that reflects the highest NIST vulnerability score detected on the endpoint. This information helps you to determine the best course of action for remediating each endpoint.

From **Endpoints > Vulnerability Assessment**, select **Endpoints** on the upper-right bar. For each endpoint, Cortex XDR displays the following default and optional values:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVEs</td>
<td>A list of all CVEs that exist on applications that are installed on the endpoint. <em>Cortex XDR displays a maximum of 500 CVEs per endpoint. If your endpoint has more than 500 CVEs, you must address some of them to reduce the number of CVEs and rescan the endpoint. Then, additional CVEs can be displayed.</em></td>
</tr>
<tr>
<td>Endpoint ID</td>
<td>Unique ID assigned by Cortex XDR that identifies the endpoint.</td>
</tr>
<tr>
<td>Endpoint name</td>
<td>Hostname of the endpoint.</td>
</tr>
<tr>
<td>Last Reported Timestamp</td>
<td>The date and time of the last time the Cortex XDR agent started the process of reporting its application inventory to Cortex XDR.</td>
</tr>
<tr>
<td>MAC address</td>
<td>The MAC address associated with the endpoint.</td>
</tr>
<tr>
<td>IP address</td>
<td>The IP address associated with the endpoint.</td>
</tr>
<tr>
<td>Platform</td>
<td>The name of the platform running on the endpoint.</td>
</tr>
<tr>
<td>Severity</td>
<td>The severity level (High, Medium, or Low) of the CVE as ranked in the NIST database.</td>
</tr>
<tr>
<td>Severity score</td>
<td>The CVE severity score based on the NIST Common Vulnerability Scoring System (CVSS). Click the score to see the full CVSS description.</td>
</tr>
</tbody>
</table>

You can perform the following actions from Cortex XDR as you investigate and remediate your endpoints:

- **View a complete list of all applications installed on an endpoint**—Right-click the endpoint and View installed applications. This list includes the application name, version, and installation path on the endpoint. If an installed application has known vulnerabilities, Cortex XDR also displays the list of CVEs and the highest **Severity**.
• **(Windows only) Isolate an endpoint from your network**—Right-click the endpoint and **Isolate** the endpoint before or during your remediation to allow the Cortex XDR agent to communicate only with Cortex XDR.

• **Retrieve an updated list of applications installed on an endpoint**—Right-click the endpoint and **Rescan** endpoint.

**Application Analysis**

You can assess the vulnerability status of a Linux application in your network using the Cortex XDR **Application Inventory**. Cortex XDR assigns to each Linux application on the list its existing CVEs, and assigns each application a vulnerability severity score that reflects the highest NIST vulnerability score detected for the application.

To perform vulnerability assessment of an application, see **Application Inventory**.
Investigation and Response

- Cortex XDR Indicators
- Search Queries
- Investigate Incidents
- Investigate Alerts
- Investigate Endpoints
- Investigate Files
- Response Actions
Cortex XDR Indicators

When you identify a threat, you can define specific indicators for which you want Cortex XDR to raise alerts. You can define rules for the following types of indicators:

- **Behavioral indicators of compromise (BIOCs)**—Identifying threats based on their behaviors can be quite complex. As you identify specific network, process, file, or registry activity that indicates a threat, you create BIOCs that can alert you when the behavior is detected. See Working with BIOCs.

- **Indicators of compromise (IOCs)**—Known artifacts that are considered malicious or suspicious. IOCs are static and based on criteria such as SHA256 hashes, IP addresses and domains, file names, and paths. You create IOC rules based on information that you gather from various threat-intelligence feeds or that you gather as a result of an investigation within Cortex XDR. See Working with IOCs.

After you create an indicator rule, you can Manage Existing Indicators from Cortex XDR.

Working with BIOCs

Behavioral indicators of compromise (BIOCs) enable you to alert and respond to behaviors—tactics, techniques, and procedures. Instead of hashes and other traditional indicators of compromise, BIOC rules detect the behavior of processes, registry, files, and network activity.

To enable you to take advantage of the latest threat research, Cortex XDR automatically receives preconfigured rules from Palo Alto Networks. These global rules are delivered to all tenants with content updates. In cases where you need to override a global BIOC rule, you can disable it or set a rule exception. You can also configure additional BIOC rules as you investigate threats on your network and endpoints. BIOC rules are highly customizable: you can create a BIOC rule that is simple or quite complex.

As soon as you create or enable a BIOC rule, the app begins to monitor input feeds for matches. Cortex XDR also analyzes historical data collected in the Cortex Data Lake. Whenever there is a match, or hit, on a BIOC rule, Cortex XDR logs an Cortex XDR Alerts.

- **BIOC Rule Details**
- **Create a BIOC Rule**
- **Manage Existing Indicators**
- **Manage Global BIOC Rules**

**BIOC Rule Details**

From Rules > BIOC, you can view all user-defined and preconfigured behavioral indicator of compromise (BIOC) rules. To search for a specific BIOC rule, you can filter by one or more fields in the BIOC rules table. From the BIOC page, you can also manage or clone existing rules.
The following table describes the fields that are available for each BIOC rule in alphabetical order.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong># OF HITS</strong></td>
<td>The number of hits (matches) on this behavior.</td>
</tr>
<tr>
<td><strong>BEHAVIOR</strong></td>
<td>A schematic of the behavior of the rule.</td>
</tr>
<tr>
<td><strong>COMMENT</strong></td>
<td>Free-form comments specified when the BIOC was created or modified.</td>
</tr>
<tr>
<td><strong>EXCEPTIONS</strong></td>
<td>Exceptions to the BIOC rule. When there's a match on the exception, the event will not trigger an alert.</td>
</tr>
<tr>
<td><strong>INSERTION DATE</strong></td>
<td>Date and time when the BIOC rule was created.</td>
</tr>
<tr>
<td><strong>MITRE ATT&amp;CK TACTIC</strong></td>
<td>Displays the type of MITRE ATT&amp;CK tactic the BIOC rule is attempting to trigger on.</td>
</tr>
<tr>
<td><strong>MITRE ATT&amp;CK TECHNIQUE</strong></td>
<td>Displays the type of MITRE ATT&amp;CK technique the BIOC rule is attempting to trigger on.</td>
</tr>
<tr>
<td><strong>MODIFICATION DATE</strong></td>
<td>Date and time when the BIOC was last modified.</td>
</tr>
<tr>
<td><strong>NAME</strong></td>
<td>Unique name that describes the rule. Global BIOC rules defined by Palo Alto Networks are indicated with a blue dot and cannot be modified or deleted.</td>
</tr>
<tr>
<td><strong>RULE ID</strong></td>
<td>Unique identification number for the rule.</td>
</tr>
<tr>
<td><strong>TYPE</strong></td>
<td>Type of BIOC rule:</td>
</tr>
<tr>
<td></td>
<td>• Collection</td>
</tr>
<tr>
<td></td>
<td>• Credential Access</td>
</tr>
<tr>
<td></td>
<td>• Dropper</td>
</tr>
<tr>
<td></td>
<td>• Evasion</td>
</tr>
</tbody>
</table>
Create a BIOC Rule

Create a Rule from Scratch

Create a Rule from Scratch

To define a BIOC, you configure the entity and any related activity or characteristics. An entity can be a specific process, registry, file, or network host. An entity activity can describe the various actions that are relevant to that type of entity. For example, for a Registry entity, the actions are: Write, Rename, and Delete. If you can identify a threat by additional attributes, you can also specify those characteristics as additional entity information in the BIOC. For example, for a Process, you can add a process name, command-line argument used to call the process, or a user name.

The following table describes the type of process and actions you can create a BIOC rule for:

<table>
<thead>
<tr>
<th>BIOC Rule Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event Log</td>
<td>Events relating to Windows Event Log.</td>
</tr>
<tr>
<td>File</td>
<td>Events relating to file create, write, read, and rename according to the file name and path.</td>
</tr>
<tr>
<td>Image Load</td>
<td>Events relating to module IDs of processes.</td>
</tr>
<tr>
<td>BIOC Rule Type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Network</td>
<td>Events relating to incoming and outgoing network, filed IP addresses, port, host name, and protocol.</td>
</tr>
<tr>
<td>New Generation (NG) Network</td>
<td>Events relating to a combination of firewall and endpoint logs over the network.</td>
</tr>
<tr>
<td>Process</td>
<td>Events relating to execution and injection of a process name, hash, path, and CMD</td>
</tr>
<tr>
<td>Registry</td>
<td>Events relating to registry write, rename and delete according to registry path.</td>
</tr>
</tbody>
</table>

To create a BIOC rule:

**STEP 1** | From Cortex XDR, select Rules > BIOC.

**STEP 2** | Select + Add Rule.

**STEP 3** | Configure the BIOC criteria.

Define any relevant activity or characteristics for the entity type. Creating a new BIOC rule is similar to the way that you create a search with Query Builder.

**STEP 4** | **Test** your BIOC rule.

Rules that you do not refine enough can create thousands of alerts. As a result, it is highly recommended that you test the behavior of a new or edited BIOC rule before you save it. For example, if a rule will return thousands of hits because you negated a single parameter, it is a good idea to test the rule before you save it and make it active.

*Cortex XDR automatically disables BIOC rules that reach 5000 or more hits over a 24 hour period.*

When you test the rule, Cortex XDR immediately searches for rule matches across all your Cortex Data Lake data. If there are surprises, now is the time to see them and adjust the rule definition.

*For the purpose of showing you the expected behavior of the rule before you save it, Cortex XDR tests the BIOC on historical logs. After you save a BIOC rule, it will operate on both historical logs (up to 10,000 hits) and new data received from your log sensors.*

**STEP 5** | **Save** your BIOC rule.

**STEP 6** | Enter a descriptive name to identify the BIOC rule.

**STEP 7** | Specify the SEVERITY you want to associate with the alert.

**STEP 8** | Select a rule TYPE which describes the activity.

**STEP 9** | Enter any additional comments such as why you created the BIOC.

**STEP 10** | Click OK.
Import Rules

You can use the import feature of Cortex XDR to import BIOCs from external feeds or that you previously exported. The export/import capability is useful for rapid copying of BIOCs across different Cortex XDR instances.

You can only import files that were exported from Cortex XDR. You cannot edit an exported file.

STEP 1 | From Cortex XDR, select Rules > BIOC.

STEP 2 | Select Import Rules.

STEP 3 | Drag and drop the file on the import rules dialog or browse to a file.

STEP 4 | Click Import.

Cortex XDR loads any BIOC rules. This process may take a few minutes depending on the size of the file.

STEP 5 | Refresh the BIOC Rules page to view matches (# of Hits) in your historical data.

STEP 6 | To investigate any matches, view the Alerts page and filter the Alert Name by the name of the BIOC rule.

Manage Global BIOC Rules

Cortex XDR checks for the latest update of global BIOC rules. If there are no new global BIOC rules, the app displays a content status of Content up to date next to the BIOC rules table heading. A dot to the left of the rule name indicates a global BIOC rule. You can also view the optional Source column to see which rules are pushed by Palo Alto Networks.

- Get the latest global BIOC rules.
- Copy a global BIOC rule.
- Add a Rule Exception.

- Get the latest global BIOC rules.
  1. Navigate to Rules > BIOC.
  2. To view the content details, hover over the status to show the global rules version number and last check date.
The content status displays the date when the content was last updated, either automatically or manually by an administrator.

4. If the status displays *Could not check update*, click the status to check for updates manually. The last updated date changes when the download is successful.

- **Copy a global BIOC rule.**
  You cannot directly modify a global rule, but you can copy global rules as a template to create new rules.

- **Add a Rule Exception.**
  Although you cannot edit global rules, you can add exceptions to the rule.

**Working with IOCs**

IOCs provide the ability to alert on known malicious objects on endpoints across the organization. You can load IOC lists from various threat-intelligence sources into the Cortex XDR app or define them individually.

You can define the following types of IOCs:

- Full path
- File name
- Domain
- Destination IP address
- MD5 hash
- SHA256 hash

After you define or load IOCs, the app checks for matches in the endpoint data collected from Cortex XDR agents. Checks are both retroactive and ongoing: The app looks for IOC matches in all data collected in the past and continues to evaluate new any new data it receives in the future.

Alerts for IOCs are identified by a source type of IOC (see Cortex XDR Alerts for more information).

- IOC Rule Details
- Create an IOC Rule
- Manage Existing Indicators

**IOC Rule Details**

From the Rules > IOC page, you can view all indicators of compromise (IOCs) configured from or uploaded to the Cortex XDR app. To filter the number of IOC rules you see, you can create filter by one or more fields in the IOC rules table. From the IOC page, you can also manage or clone existing rules.
The following table describes the fields that are available for each IOC rule in alphabetical order.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td># OF HITS</td>
<td>The number of hits (matches) on this indicator.</td>
</tr>
<tr>
<td>CLASS</td>
<td>The IOC's class. For example, 'Malware'.</td>
</tr>
<tr>
<td>COMMENT</td>
<td>Free-form comments specified when the IOC was created or modified.</td>
</tr>
<tr>
<td>EXPIRATION DATE</td>
<td>The date and time at which the IOC will be removed automatically.</td>
</tr>
<tr>
<td>INDICATOR</td>
<td>The indicator value itself. For example, if the indicator type is a destination IP address, this could be an IP address such as 1.1.1.1.</td>
</tr>
<tr>
<td>INSERTION DATE</td>
<td>Date and time when the IOC was created.</td>
</tr>
<tr>
<td>MODIFICATION DATE</td>
<td>Date and time when the IOC was last modified.</td>
</tr>
</tbody>
</table>
| RELIABILITY | Indicator's reliability level:  
  • A - Completely Reliable  
  • B - Usually Reliable  
  • C - Fairly Reliable  
  • D - Not Usually Reliable  
  • E - Unreliable                                                                 |
<p>| REPUTATION  | Indicator's reputation level. One of Unknown, Good, Bad, or Suspicious.                                                                |
| RULE ID     | Unique identification number for the rule.                                                                                                |</p>
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEVERITY</td>
<td>IOC severity that was defined when the IOC was created.</td>
</tr>
</tbody>
</table>
| SOURCE  | User who created this IOC, or the file name from which it was created, or one of the following keywords:  
          • Public API—the indicator was uploaded using the Insert Simple Indicators, CSV or Insert Simple Indicators, JSON REST APIs.  
          • XSOAR TIM—the indicator was retrieved from XSOAR. |
| STATUS  | Rule status: Enabled or Disabled.                                           |
| TYPE    | Type of indicator: Full path, File name, Host name, Destination IP, MD5 hash. |
| VENDORS | A list of threat intelligence vendors from which this IOC was obtained.     |

Create an IOC Rule

There are two options for creating new IOC rules:

- Configure a single IOC.
- Upload a file, one IOC per line, that contains up to 20,000 IOCs. For example, you can upload multiple file paths and MD5 hashes for an IOC rule. To help you format the upload file in the syntax that Cortex XDR will accept, you can download the example file.

If you have a Cortex XDR Pro per Endpoint license, you can upload IOCs using REST APIs in either CSV or JSON format.

**STEP 1** From Cortex XDR, select Rules > IOC.

**STEP 2** Select + Add IOC.

**STEP 3** Configure the IOC criteria.
If after investigating a threat, you identify a malicious artifact, you can create an alert for the Single IOC right away.

1. Configure the INDICATOR value on which you want to match.
2. Configure the IOC TYPE. Options are Full Path, File Name, Domain, Destination IP, and MD5 or SHA256 Hash.
3. Configure the SEVERITY you want to associate with an alert for the IOC: Informational, Low, Medium, or High.
4. (Optional) Enter a comment that describes the IOC.
5. (Optional) Enter the IOC’s REPUTATION.
6. (Optional) Enter the IOC’s RELIABILITY.
7. (Optional) Enter an EXPIRATION for the IOC.
8. Click Create.

If you want to match on multiple indicators, you can upload the criteria in a CSV file.

1. Select Upload File.
2. Drag and drop the CSV file containing the IOC criteria in the drop area of the Upload File dialog or browse to the file.
Cortex XDR supports a file with multiple IOCs in a pre-configured format. For help determining the format syntax, Cortex XDR provides an example text file that you can download.

3. Configure the **SEVERITY** you want to associate with an alert for the IOCs: Informational, Low, Medium, or High.

4. Define the **DATA FORMAT** of the IOCs in the CSV file. Options are Mixed, Full Path, File Name, Domain, Destination IP, and MD5 or SHA256 Hash.

5. Click **Upload**.

**STEP 4 | (Optional)** Define any expiration criteria for your IOC rules.

If desired, you can also configure additional expiration criteria per IOC type to apply to all IOC rules. In most cases, IOC types like Destination IP or Host Name are considered malicious only for a short period of time since they are soon cleaned and then used by legitimate services, from which time they only cause false positives. For these types of IOCs, you can set a short expiration period. The expiration criteria you define for an IOC type will apply to all existing rules and additional rules that you create in the future.

1. Select **Settings**.
2. Set the expiration for any relevant IOC type. Options are Never, 1 week, 1 month, 3 months, or 6 months.
3. Click **Save**.

**Manage Existing Indicators**

After you create an indicator rule, you can take the following actions:

- View Alerts Triggered by a Rule
- Edit a Rule
- Export a Rule (BIOC Only)
- Copy a Rule
- Disable or Remove a Rule
- Add a Rule Exception

**View Alerts Triggered by a Rule**

As your IOC and BIOC rules trigger alerts, Cortex XDR displays the total **# OF HITS** for the rule in the on the BIOC orIOC rules page. To view the associated alerts triggered by a rule:

**STEP 1 |** Select **RULES** and the type of rule (BIOC or IOC).

**STEP 2 |** Right-click anywhere in the rule, and then select **View associated alerts**.

Cortex XDR displays a filtered query of alerts associated with the Rule ID.

**Edit a Rule**

After you create a rule, it may be necessary to tweak or change the rule settings. You can open the rule configuration from the Rules page or from the pivot menu of an alert triggered by the rule. To edit the rule from the Rules page:

**STEP 1 |** Select **RULES** and the type of rule (BIOC or IOC).

**STEP 2 |** Locate the rule you want to edit.

**STEP 3 |** Right click anywhere in the rule and then select **Edit**.
STEP 4 | Edit the rule settings as needed.
       If you make any changes, Test and then Save the rule.

STEP 5 | Adjust the schedule settings as needed, and then click OK.

Export a Rule (BIOC Only)

STEP 1 | Select RULES > BIOC.

STEP 2 | Select the rules that you want to export.

STEP 3 | Right click any of the rows, and select Export selected.
       The exported file is not editable, however you can use it as a source to import rules at a later date.

Copy a Rule

You can use an existing rule as a template to create a new one. Global BIOC rules cannot be deleted or altered, but you can copy a global rule and edit the copy. See Manage Global BIOC Rules.

STEP 1 | Select RULES and the type of rule (BIOC or IOC).

STEP 2 | Locate the rule you want to copy.

STEP 3 | Right click anywhere in the rule row and then select Copy to create a duplicate rule.

Disable or Remove a Rule

If you no longer need a rule you can temporarily disable or permanently remove it.

You cannot delete global BIOC delivered with content updates.

STEP 1 | Select RULES and the type of rule (BIOC or IOC).

STEP 2 | Locate the rule that you want to change.

STEP 3 | Right click anywhere in the rule row and then select Remove to permanently delete the rule, or Disable to temporarily stop the rule. If you disable a rule you can later return to the rule page to Enable it.

Add a Rule Exception

If you want to create a rule to take action on specific behaviors but also want to exclude one or more indicators from the rule, you can create a rule exception. An indicator can include the SHA256 hash of a process, process name, process path, vendor name, user name, causality group owner (CGO) full path, or process command-line arguments. For more information about these indicators, see Cortex XDR Indicators. For each exception, you also specify the rule scope to which exception applies.

Cortex XDR only supports exceptions with one attribute. See Add an Alert Exclusion Policy to create advanced exceptions based on your filtered criteria.

STEP 1 | From Cortex XDR, select Rules > Rule Exceptions.
STEP 2 | Select + New Exception.

STEP 3 | Configure the indicators and conditions for which you want to set the exception.

STEP 4 | Choose the scope of the exception, whether the exception applies to IOCs, BIOCs, or both.

STEP 5 | Save the exception.

By default, activity matching the indicators does not trigger any rule. As an alternative, you can select one or more rules. After you save the exception, the Exceptions count for the rule increments. If you later edit the rule, you will also see the exception defined in the rule summary.
Search Queries

- Cortex XDR Query Builder
- Cortex XDR Query Center
- Cortex XDR Scheduled Queries
- Quick Launcher
- Research a Known Threat

Cortex XDR Query Builder

The Query Builder is a powerful search tool at the heart of Cortex XDR that you can use to investigate any lead quickly, expose the root cause of an alert, perform damage assessment, and hunt for threats from your data sources. With Query Builder, you can build complex queries for entities and entity attributes so that you can surface and identify connections between them. The Query Builder searches the raw data from the logs received by the Cortex Data Lake for the entities and attributes you specify.

The Query Builder provides queries for the following types of entities:

- **Process**—Search on process execution and injection by process name, hash, path, command-line arguments, and more. See Create a Process Query.
- **File**—Search on file creation and modification activity by file name and path. See Create a File Query.
- **Network**—Search network activity by IP address, port, host name, protocol, and more. See Create a Network Query.
- **Registry**—Search on registry creation and modification activity by key, key value, path, and data. See Create a Registry Query.
- **Event Log**—Search Windows event logs by username, log event ID, log level, and message. See Create an Event Log Query.
• **NG Network**—Search security event logs by firewall logs, endpoint raw data over your network. See Create an NG Network Query.
• **All Actions**—Search across all network, registry, file, and process activity by endpoint or process. See Query Across All Entities.

The **Query Builder** also provides flexibility for both on-demand query generation and scheduled queries.

**Native Search**

To search across all available logs and data in Cortex XDR, you can use the text-based Native Search. To facilitate simple and complex text-based queries, you can enter fields based on the log’s metadata hierarchy (core fields, vendor fields, or log types) the operator, the field value, and the timeframe. For simplicity, the Native Search provides auto-completion—based on the known log fields—as you type. You can also use Regex (except for with IP addresses and ranges) and wildcards in your queries and can string together multiple queries using **and** or **or**.

For examples of text-based queries, see [Native Search Examples](#).

**Core Fields for Native Search**

When you specify core fields without any other search criteria, the Native Search queries the field value across all data and logs that contain that field type. To further refine the results and specify context, you can combine core fields with other criteria such as **vendor** or **log type**. You can build queries in Native Search for any of the following core fields:

- ip
- source_ip
- destination_ip
- hash
- host_name
- user_name
- process_name
- process_path

**Vendor Fields for Native Search**

To search for logs or data from a specific vendor, you can refine your query by vendor and product. The query fields are hierarchical. To construct a query, separate each field in the hierarchy with periods. Examples of vendor fields include:
• Search for results from all Palo Alto Networks products—**PANW**
• Search for results from Cisco ASA firewalls—**Cisco.ASA**

<table>
<thead>
<tr>
<th>Vendor</th>
<th>Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>PANW</td>
<td>NGFW</td>
</tr>
<tr>
<td></td>
<td>Cortex Agent</td>
</tr>
<tr>
<td>Checkpoint</td>
<td>FW1/VPN1</td>
</tr>
<tr>
<td>Cisco</td>
<td>ASA</td>
</tr>
<tr>
<td></td>
<td>Firepower</td>
</tr>
<tr>
<td>Okta</td>
<td>MFA</td>
</tr>
<tr>
<td>Microsoft</td>
<td>Azure AD</td>
</tr>
<tr>
<td>Corelight</td>
<td>Corelight sensor</td>
</tr>
<tr>
<td>Fortinet</td>
<td>Fortigate</td>
</tr>
</tbody>
</table>

**Log Types for Native Search**

You can construct queries for the following types of logs and log subtypes.

<table>
<thead>
<tr>
<th>Log Type</th>
<th>Log Subtype</th>
</tr>
</thead>
<tbody>
<tr>
<td>process_actions</td>
<td>• process_executed</td>
</tr>
<tr>
<td></td>
<td>• process_injected</td>
</tr>
<tr>
<td>registry_actions</td>
<td>• key_created</td>
</tr>
<tr>
<td></td>
<td>• key_renamed</td>
</tr>
<tr>
<td></td>
<td>• key_deleted</td>
</tr>
<tr>
<td></td>
<td>• key_created</td>
</tr>
<tr>
<td></td>
<td>• value_set</td>
</tr>
<tr>
<td></td>
<td>• value_deleted</td>
</tr>
<tr>
<td>file_actions</td>
<td>• file_created</td>
</tr>
<tr>
<td></td>
<td>• file_deleted</td>
</tr>
<tr>
<td></td>
<td>• file_renamed</td>
</tr>
<tr>
<td></td>
<td>• file_written</td>
</tr>
<tr>
<td></td>
<td>• file_read</td>
</tr>
<tr>
<td>network_connections</td>
<td>• outbound_connection</td>
</tr>
<tr>
<td></td>
<td>• inbound_connection</td>
</tr>
<tr>
<td></td>
<td>• failed_connection</td>
</tr>
<tr>
<td>event_logs</td>
<td>• endpoint_eventlog</td>
</tr>
<tr>
<td></td>
<td>• dc_eventlog</td>
</tr>
<tr>
<td>Log Type</td>
<td>Log Subtype</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>authentication</td>
<td>• successful_authentication</td>
</tr>
<tr>
<td></td>
<td>• failed_authentication</td>
</tr>
<tr>
<td>image_load</td>
<td>• image_load_success</td>
</tr>
<tr>
<td></td>
<td>• change_page_protection</td>
</tr>
</tbody>
</table>

**Operators**

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>=</td>
<td>Show results equal to a value</td>
</tr>
<tr>
<td>!=</td>
<td>Show results that are not equal to a value.</td>
</tr>
<tr>
<td>=~</td>
<td>Show results that are equal to a Regex pattern match. Not supported with IP addresses or ranges.</td>
</tr>
<tr>
<td>!~</td>
<td>Show results that are not equal to a Regex pattern match. Not supported with IP addresses or ranges.</td>
</tr>
<tr>
<td>contains</td>
<td>Show results that contain a value</td>
</tr>
<tr>
<td>not contains</td>
<td>Show results that do not contain a value.</td>
</tr>
<tr>
<td>in (list, range)</td>
<td>Show results including one or more matches in a list or range. Not supported with IP addresses or ranges.</td>
</tr>
<tr>
<td>not in (list, range)</td>
<td>Show results excluding one or more matches in a list or range. Not supported with IP addresses or ranges.</td>
</tr>
</tbody>
</table>

**Native Search Examples**

**Search**

- `logtype = file AND subtype IN ("file create", "file delete") and hostname contains SF`
- `network connections AND palo alto networks.app id = facebook`
- `okta.sso AND ip != 10.0.*`
- `palo alto networks.file create.file name =~ ".+?"
- `event log AND (palo alto networks.event log id = 41783 OR hostname =~ la^xcortex xdr agent AND palo alto networks.dst process name CONTAINS chrome`
Create a Process Query

From the **Query Builder** you can investigate connections between processes, child processes, and endpoints.

For example, you can create a process query to search for processes executed on a specific endpoint.

To build a process query:

**STEP 1** | From Cortex XDR, select **INVESTIGATION > Query Builder**.

**STEP 2** | Select **PROCESS**.

**STEP 3** | Enter the search criteria for the process query.

- **Process action**—Select the type of process action you want to search: On process **Execution**, **Injection** into another process, or both.
- **Process attributes**—Define any additional process attributes for which you want to search.
  
  Use a pipe (|) to separate multiple values. Use an asterisk (*) to match any string of characters.
  
  By default, Cortex XDR will return results that match the attribute you specify. To exclude an attribute value, toggle the operator from = to !=. Attributes are:

  - **NAME**—Name of the process. For example, `notepad.exe`.
  - **CMD**—Command-line used to initiate the process including any arguments, up to 128 characters.
• **SIGNATURE**—Signing status of the process: Signature Unavailable, Signed, Invalid Signature, Unsigned, Revoked, Signature Fail.

• **PATH**—Path to the process. For example, `C:\windows\system32\notepad.exe`.

• **PID**—Process ID.

• **SIGNER**—Signer of the process.

To specify an additional exception (match this value except), click the + to the right of the value and specify the exception value.

**STEP 4 | (Optional) Limit the scope to a specific acting process:**

Specify one or more of the following attributes for the acting (parent) process.

Use a pipe (|) to separate multiple values. Use an asterisk (*) to match any string of characters.

- **NAME**—Name of the parent process.
- **PATH**—Path to the parent process.
- **CMD**—Command-line used to initiate the parent process including any arguments, up to 128 characters.
- **MD5**—MD5 hash value of the parent process.
- **SHA256**—SHA256 hash value of the process.
- **USER NAME**—User who executed the process.
- **SIGNATURE**—Signing status of the parent process: Signed, Unsigned, N/A, Invalid Signature, Weak Hash
- **SIGNER**—Entity that signed the certificate of the parent process.
- **PID**—Process ID of the parent process.

• **Run search on process, Causality and OS actors**—The causality actor—also referred to as the causality group owner (CGO)—is the parent process in the execution chain that the Cortex XDR agent identified as being responsible for initiating the process tree. The OS actor is the parent process that creates an OS process on behalf of a different initiator. By default, this option is enabled to apply the same search criteria to initiating processes. To configure different attributes for the parent or initiating process, clear this option.

**STEP 5 | (Optional) Limit the scope to an endpoint or endpoint attributes:**

Specify one or more of the following attributes:

- **HOST**—HOST NAME, HOST IP address, HOST OS, or HOST MAC ADDRESS.
- **PROCESS**—NAME, PATH, CMD, MD5, SHA256, USER NAME, SIGNATURE, or PID

Use a pipe (|) to separate multiple values. Use an asterisk (*) to match any string of characters.

**STEP 6 | Specify the time period for which you want to search for events.**

Options are: Last 24H (hours), Last 7D (days), Last 1M (month), or select a Custom time period.
STEP 7 | Choose when to run the query.

Select the calendar icon to schedule a query to run on or before a specific date, Run in background to run the query as resources are available, or Run to run the query immediately and view the results in the Query Center.

STEP 8 | When you are ready, View the Results of a Query.

Create a File Query

From the Query Builder you can investigate connections between file activity and endpoints.

Some examples of file queries you can run include:

- Files modified on specific endpoints.
- Files related to process activity that exist on specific endpoints.

To build a file query:

STEP 1 | From Cortex XDR, select INVESTIGATION > Query Builder.

STEP 2 | Select FILE.

STEP 3 | Enter the search criteria for the file events query.

- File activity—Select the type or types of file activity you want to search: Create, Read, Rename, Delete, or Write.
- File attributes—Define any additional process attributes for which you want to search. Use a pipe (|) to separate multiple values (for example notepad.exe|chrome.exe). By default, Cortex XDR will return the events that match the attribute you specify. To exclude an attribute value, toggle the = option to =! . Attributes are:
  - NAME—File name.
  - PATH—Path of the file.
  - PREVIOUS NAME—Previous name of a file.
  - PREVIOUS PATH—Previous path of the file.
To specify an additional exception (match this value except), click the + to the right of the value and specify the exception value.

**STEP 4** *(Optional)* Limit the scope to a specific acting process:

Specify one or more of the following attributes for the acting (parent) process.

Use a pipe (|) to separate multiple values. Use an asterisk (*) to match any string of characters.

- **NAME**—Name of the parent process.
- **PATH**—Path to the parent process.
- **CMD**—Command-line used to initiate the parent process including any arguments, up to 128 characters.
- **MD5**—MD5 hash value of the parent process.
- **SHA256**—SHA256 hash value of the process.
- **USER NAME**—User who executed the process.
- **SIGNATURE**—Signing status of the parent process: Signed, Unsigned, N/A, Invalid Signature, Weak Hash
- **SIGNER**—Entity that signed the certificate of the parent process.
- **PID**—Process ID of the parent process.
- **Run search on process, Causality and OS actors**—The causality actor—also referred to as the causality group owner (CGO)—is the parent process in the execution chain that the Cortex XDR agent identified as being responsible for initiating the process tree. The OS actor is the parent process that creates an OS process on behalf of a different initiator. By default, this option is enabled to apply the same search criteria to initiating processes. To configure different attributes for the parent or initiating process, clear this option.

**STEP 5** *(Optional)* Limit the scope to an endpoint or endpoint attributes:

Specify one or more of the following attributes:

- **HOST**—HOST NAME, HOST IP address, HOST OS, or HOST MAC ADDRESS.
- **PROCESS**—NAME, PATH, CMD, MD5, SHA256, USER NAME, SIGNATURE, or PID

Use a pipe (|) to separate multiple values. Use an asterisk (*) to match any string of characters.

**STEP 6** Specify the time period for which you want to search for events.

Options are: **Last 24H** (hours), **Last 7D** (days), **Last 1M** (month), or select a **Custom** time period.

**STEP 7** Choose when to run the query.
Select the calendar icon to schedule a query to run on or before a specific date. Run in background to run the query as resources are available, or Run to run the query immediately and view the results in the Query Center.

**STEP 8 | When you are ready, View the Results of a Query.**

**Create a Network Query**

From the Query Builder you can investigate connections between network activity, acting processes, and endpoints.

Some examples of network queries you can run include:

- Network connections to or from a specific IP address and port number.
- Processes that created network connections.
- Network connections between specific endpoints.

To build a network query:

**STEP 1 |** From Cortex XDR, select INVESTIGATION > Query Builder.

**STEP 2 |** Select NETWORK.

**STEP 3 |** Enter the search criteria for the network events query.

- Network traffic type—Select the type or types of network traffic alerts you want to search: Incoming, Outgoing, or Failed.
- Network attributes—Define any additional process attributes for which you want to search. Use a pipe (|) to separate multiple values (for example 80|8080). By default, Cortex XDR will return the events that match the attribute you specify. To exclude an attribute value, toggle the = option to =!. Options are:
  - REMOTE COUNTRY—Country from which the remote IP address originated.
  - REMOTE IP—Remote IP address related to the communication.
  - REMOTE PORT—Remote port used to make the connection.
• **LOCAL IP**—Local IP address related to the communication. Matches can return additional data if a machine has more than one NIC.
• **LOCAL PORT**—Local port used to make the connection.
• **PROTOCOL**—Network transport protocol over which the traffic was sent.

To specify an additional exception (match this value except), click the + to the right of the value and specify the exception value.

**STEP 4 | (Optional)** Limit the scope to a specific acting process:

Specify one or more of the following attributes for the acting (parent) process.

Use a pipe (|) to separate multiple values. Use an asterisk (*) to match any string of characters.

• **NAME**—Name of the parent process.
• **PATH**—Path to the parent process.
• **CMD**—Command-line used to initiate the parent process including any arguments, up to 128 characters.
• **MD5**—MD5 hash value of the parent process.
• **SHA256**—SHA256 hash value of the process.
• **USER NAME**—User who executed the process.
• **SIGNATURE**—Signing status of the parent process: Signed, Unsigned, N/A, Invalid Signature, Weak Hash
• **SIGNER**—Entity that signed the certificate of the parent process.
• **PID**—Process ID of the parent process.
• **Run search on process, Causality and OS actors**—The causality actor—also referred to as the causality group owner (CGO)—is the parent process in the execution chain that the Cortex XDR agent identified as being responsible for initiating the process tree. The OS actor is the parent process that creates an OS process on behalf of a different initiator. By default, this option is enabled to apply the same search criteria to initiating processes. To configure different attributes for the parent or initiating process, clear this option.

**STEP 5 | (Optional)** Limit the scope to an endpoint or endpoint attributes:

Specify one or more of the following attributes:

• **HOST**—HOST NAME, HOST IP address, HOST OS, or HOST MAC ADDRESS.
• **PROCESS**—NAME, PATH, CMD, MD5, SHA256, USER NAME, SIGNATURE, or PID

Use a pipe (|) to separate multiple values. Use an asterisk (*) to match any string of characters.

**STEP 6 |** Specify the time period for which you want to search for events.

Options are: **Last 24H** (hours), **Last 7D** (days), **Last 1M** (month), or select a **Custom** time period.
STEP 7 | Choose when to run the query.

Select the calendar icon to schedule a query to run on or before a specific date, **Run in background** to run the query as resources are available, or **Run** to run the query immediately and view the results in the Query Center.

STEP 8 | When you are ready, **View the Results of a Query**.

Create an Image Load Query

From the Query Builder you can investigate connections between image load activity, acting processes, and endpoints.

Some examples of image load queries you can run include:

- Module load into process events by module path or hash.

To build an image load query:

**STEP 1** | From Cortex XDR, select **INVESTIGATION > Query Builder**.

**STEP 2** | Select **IMAGE LOAD**.

**STEP 3** | Enter the search criteria for the image load activity query.

- Type of image activity: **All**, **Image Load**, or **Change Page Protection**.
- Identifying information about the image module: **Full Module Path**, **Module MD5**, or **Module SHA256**.

By default, Cortex XDR will return the activity that matches all the criteria you specify. To exclude a value, toggle the = option to =!.

**STEP 4** | **(Optional)** Limit the scope to a specific acting process:
Specify one or more of the following attributes for the acting (parent) process.

Use a pipe (|) to separate multiple values. Use an asterisk (*) to match any string of characters.

- **NAME**—Name of the parent process.
- **PATH**—Path to the parent process.
- **CMD**—Command-line used to initiate the parent process including any arguments, up to 128 characters.
- **MD5**—MD5 hash value of the parent process.
- **SHA256**—SHA256 hash value of the process.
- **USER NAME**—User who executed the process.
- **SIGNATURE**—Signing status of the parent process: Signed, Unsigned, N/A, Invalid Signature, Weak Hash
- **SIGNER**—Entity that signed the certificate of the parent process.
- **PID**—Process ID of the parent process.
- **Run search on process, Causality and OS actors**—The causality actor—also referred to as the causality group owner (CGO)—is the parent process in the execution chain that the Cortex XDR agent identified as being responsible for initiating the process tree. The OS actor is the parent process that creates an OS process on behalf of a different initiator. By default, this option is enabled to apply the same search criteria to initiating processes. To configure different attributes for the parent or initiating process, clear this option.

**STEP 5 | (Optional)** Limit the scope to an endpoint or endpoint attributes:

Specify one or more of the following attributes:

- **HOST**—HOST NAME, HOST IP address, HOST OS, or HOST MAC ADDRESS.
- **PROCESS**—NAME, PATH, CMD, MD5, SHA256, USER NAME, SIGNATURE, or PID

Use a pipe (|) to separate multiple values. Use an asterisk (*) to match any string of characters.

**STEP 6 | Specify the time period for which you want to search for events.**

Options are: Last 24H (hours), Last 7D (days), Last 1M (month), or select a Custom time period.

**STEP 7 | Choose when to run the query.**

Select the calendar icon to schedule a query to run on or before a specific date, Run in background to run the query as resources are available, or Run to run the query immediately and view the results in the Query Center.

**STEP 8 | When you are ready, View the Results of a Query.**
Create a Registry Query

From the Query Builder you can investigate connections between registry activity, processes, and endpoints.

Some examples of registry queries you can run include:

- Modified registry keys on specific endpoints.
- Registry keys related to process activity that exist on specific endpoints.

To build a registry query:

**STEP 1** | From Cortex XDR, select **INVESTIGATION > Query Builder**.

**STEP 2** | Select **REGISTRY**.

**STEP 3** | Enter the search criteria for the registry events query.  
- Registry action—Select the type or types of registry actions you want to search: **Key Create**, **Key Delete**, **Key Rename**, **Value Set**, or **Value Delete**.  
- Registry attributes—Define any additional registry attributes for which you want to search. By default, Cortex XDR will return the events that match the attribute you specify. To exclude an attribute value, toggle the = option to =!. Attributes are:
  - **KEY NAME**—Registry key name.  
  - **DATA**—Registry key data value.  
  - **REGISTRY FULL KEY**—Full registry key path.  
  - **KEY PREVIOUS NAME**—Name of the registry key before modification.  
  - **VALUE NAME**—Registry value name.  

To specify an additional exception (match this value except), click the + to the right of the value and specify the exception value.

**STEP 4** | (Optional) Limit the scope to a specific acting process:
Specify one or more of the following attributes for the acting (parent) process.

Use a pipe (|) to separate multiple values. Use an asterisk (*) to match any string of characters.

- **NAME**—Name of the parent process.
- **PATH**—Path to the parent process.
- **CMD**—Command-line used to initiate the parent process including any arguments, up to 128 characters.
- **MD5**—MD5 hash value of the parent process.
- **SHA256**—SHA256 hash value of the process.
- **USER NAME**—User who executed the process.
- **SIGNATURE**—Signing status of the parent process: Signed, Unsigned, N/A, Invalid Signature, Weak Hash
- **SIGNER**—Entity that signed the certificate of the parent process.
- **PID**—Process ID of the parent process.
- **Run search on process, Causality and OS actors**—The causality actor—also referred to as the causality group owner ( CGO)—is the parent process in the execution chain that the Cortex XDR agent identified as being responsible for initiating the process tree. The OS actor is the parent process that creates an OS process on behalf of a different initiator. By default, this option is enabled to apply the same search criteria to initiating processes. To configure different attributes for the parent or initiating process, clear this option.

**STEP 5 | (Optional)** Limit the scope to an endpoint or endpoint attributes:

Specify one or more of the following attributes:

- **HOST**—HOST NAME, HOST IP address, HOST OS, or HOST MAC ADDRESS.
- **PROCESS**—NAME, PATH, CMD, MD5, SHA256, USER NAME, SIGNATURE, or PID

Use a pipe (|) to separate multiple values. Use an asterisk (*) to match any string of characters.

**STEP 6 |** Specify the time period for which you want to search for events.

Options are: Last 24H (hours), Last 7D (days), Last 1M (month), or select a Custom time period.

**STEP 7 |** Choose when to run the query.

Select the calendar icon to schedule a query to run on or before a specific date, Run in background to run the query as resources are available, or Run to run the query immediately and view the results in the Query Center.

**STEP 8 |** When you are ready, View the Results of a Query.
Create an Event Log Query

From the Query Builder you can search Windows event log attributes and investigate event logs across endpoints with an Cortex XDR agent installed.

Some examples of event log queries you can run include:
- Critical level messages on specific endpoints.
- Message descriptions with specific keywords on specific endpoints.

To build a file query:

**STEP 1** | From Cortex XDR, select INVESTIGATION > Query Builder.

**STEP 2** | Select EVENT LOG.

**STEP 3** | Enter the search criteria for your Windows event log query.

Define any event attributes for which you want to search. By default, Cortex XDR will return the events that match the attribute you specify. To exclude an attribute value, toggle the = option to =!. Attributes are:

- PROVIDER NAME—The provider of the event log.
- USERNAME—The username associated with the event.
- EVENT ID—The unique ID of the event.
- LEVEL—The event severity level.
- MESSAGE—The description of the event.

To specify an additional exception (match this value except), click the + to the right of the value and specify the exception value.

**STEP 4** | (Optional) Limit the scope to an endpoint or endpoint attributes:
Specify one or more of the following attributes:

- **HOST**—HOST NAME, HOST IP address, HOST OS, or HOST MAC ADDRESS.
- **PROCESS**—NAME, PATH, CMD, MD5, SHA256, USER NAME, SIGNATURE, or PID

Use a pipe (|) to separate multiple values. Use an asterisk (*) to match any string of characters.

**STEP 5** | Specify the time period for which you want to search for events.
Options are: Last 24H (hours), Last 7D (days), Last 1M (month), or select a Custom time period.

**STEP 6** | Choose when to run the query.
Select the calendar icon to schedule a query to run on or before a specific date, Run in background to run the query as resources are available, or Run to run the query immediately and view the results in the Query Center.

**STEP 7** | When you are ready, View the Results of a Query.

**STEP 8** | Specify the time period for which you want to search for events.
Options are: Last 24H (hours), Last 7D (days), Last 1M (month), or select a Custom time period.

**STEP 9** | Choose when to run the query.
Select the calendar icon to schedule a query to run on or before a specific date, Run in background to run the query as resources are available, or Run to run the query immediately and view the results in the Query Center.

**STEP 10** | When you are ready, View the Results of a Query.

*Create an NG Network Query*

From the Query Builder you can investigate network events stitched across endpoints and the Palo Alto Networks next-generation firewalls logs.
Some examples of network queries you can run include:

- Source and destination of a process.
- Network connections that included a specific App ID
- Processes that created network connections.
- Network connections between specific endpoints.

To build a network query:

**STEP 1 |** From Cortex XDR, select **INVESTIGATION > Query Builder.**

**STEP 2 |** Select **NG NETWORK.**

**STEP 3 |** Enter the search criteria for the network events query.

- Network attributes—Define any additional process attributes for which you want to search. Use a pipe (|) to separate multiple values (for example 80 | 8080). By default, Cortex XDR will return the events that match the attribute you specify. To exclude an attribute value, toggle the = option to ≠!. Options are:
  - **APP ID**—App ID of the network.
  - **PROTOCOL**—Network transport protocol over which the traffic was sent.
  - **FW SERIAL ID**—Firewall serial ID.
  - **FW DEVICE NAME**—Firewall device name.
  - **FW RULE**—Firewall rule.
  - **SESSION UPLOAD SIZE**—Amount of data uploaded during a session.
  - **SESSION DOWNLOAD SIZE**—Amount of data downloaded during a session.

**STEP 4 | (Optional)** To limit the scope to a specific source, click the + to the right of the value and specify the exception value.
Specify one or more attributes for the source.

Use a pipe (|) to separate multiple values. Use an asterisk (*) to match any string of characters.

- **HOST NAME**—Name of the source.
- **HOST IP**—IP address of the source.
- **HOST OS**—Operating system of the source.
- **PROCESS NAME**—Name of the process.
- **PROCESS PATH**—Path to the process.
- **CMD**—Command-line used to initiate the process including any arguments, up to 128 characters.
- **MD5**—MD5 hash value of the process.
- **SHA256**—SHA256 hash value of the process.
- **PROCESS USER NAME**—User who executed the process.
- **SIGNATURE**—Signing status of the parent process: Signature Unavailable, Signed, Invalid Signature, Unsigned, Revoked, Signature Fail.
- **PID**—Process ID of the parent process.
- **IP**—IP address of the process.
- **PORT**—Port number of the process.
- **USER ID**—ID of the user who executed the process.
- **Run search for both the process and the Causality actor**—The causality actor—also referred to as the causality group owner (CGO)—is the parent process in the execution chain that XDR app identified as being responsible for initiating the process tree. Select this option if you want to apply the same search criteria to the causality actor. If you clear this option, you can then configure different attributes for the causality actor.

**STEP 5 | (Optional) Limit the scope to a destination.**

Use a pipe (|) to separate multiple values. Use an asterisk (*) to match any string of characters.

Specify one or more of the following attributes:

- **REMOTE IP**—IP address of the destination.
- **COUNTRY**—Country of the destination.
- Destination **TARGET HOST, NAME, PORT, HOST NAME, PROCESS USER NAME, HOST IP, CMD, HOST OS, MD5, PROCESS PATH, USER ID, SHA256, SIGNATURE, or PID**

**STEP 6 | Specify the time period for which you want to search for events.**
Options are: Last 24H (hours), Last 7D (days), Last 1M (month), or select a Custom time period.

STEP 7 | Choose when to run the query.

Select the calendar icon to schedule a query to run on or before a specific date, Run in background to run the query as resources are available, or Run to run the query immediately and view the results in the Query Center.

STEP 8 | When you are ready, View the Results of a Query.

Create an Authentication Query

From the Query Builder you can investigate authentication activity across all ingested authentication logs and data.

Some examples of authentication queries you can run include:

- Authentication logs by severity
• Authentication logs by event message
• Authentication logs for a specific source IP address

To build an authentication query:

**STEP 1 |** From Cortex XDR, select **INVESTIGATION > Query Builder.**

**STEP 2 |** Select **AUTHENTICATION.**

**STEP 3 |** Enter the search criteria for the authentication query.

   By default, Cortex XDR will return the activity that matches all the criteria you specify. To exclude a value, toggle the = option to =!.

**STEP 4 |** Choose when to run the query.

   Select the calendar icon to schedule a query to run on or before a specific date, **Run in background** to run the query as resources are available, or **Run** to run the query immediately and view the results in the **Query Center.**

**STEP 5 |** When you are ready, **View the Results of a Query.**

### Query Across All Entities

From the **Query Builder** you can perform a simple search for hosts and processes across all file events, network events, registry events, process events, and Windows event logs.

Some examples of queries you can run across all entities include:

• All activities on a host
• All activities initiated by a process on a host.

To build a query:

**STEP 1 |** From Cortex XDR, select **INVESTIGATION > Query Builder.**

**STEP 2 |** Select **ALL ACTIONS.**

**STEP 3 |** *(Optional)* Limit the scope to a specific acting process:
Specify one or more of the following attributes for the acting (parent) process.

Use a pipe (|) to separate multiple values. Use an asterisk (*) to match any string of characters.

- **NAME**—Name of the parent process.
- **PATH**—Path to the parent process.
- **CMD**—Command-line used to initiate the parent process including any arguments, up to 128 characters.
- **MD5**—MD5 hash value of the parent process.
- **SHA256**—SHA256 hash value of the process.
- **USER NAME**—User who executed the process.
- **SIGNATURE**—Signing status of the parent process: Signed, Unsigned, N/A, Invalid Signature, Weak Hash.
- **SIGNER**—Entity that signed the certificate of the parent process.
- **PID**—Process ID of the parent process.
- **Run search on process, Causality and OS actors**—The causality actor—also referred to as the causality group owner (CGO)—is the parent process in the execution chain that the Cortex XDR agent identified as being responsible for initiating the process tree. The OS actor is the parent process that creates an OS process on behalf of a different initiator. By default, this option is enabled to apply the same search criteria to initiating processes. To configure different attributes for the parent or initiating process, clear this option.

**STEP 4** | *(Optional)* Limit the scope to an endpoint or endpoint attributes:

Specify one or more of the following attributes:

- **HOST**—HOST NAME, HOST IP address, HOST OS, or HOST MAC ADDRESS.
- **PROCESS**—NAME, PATH, CMD, MD5, SHA256, USER NAME, SIGNATURE, or PID

Use a pipe (|) to separate multiple values. Use an asterisk (*) to match any string of characters.

**STEP 5** | Specify the time period for which you want to search for events.

Options are: **Last 24H** (hours), **Last 7D** (days), **Last 1M** (month), or select a **Custom** time period.

**STEP 6** | Choose when to run the query.

Select the calendar icon to schedule a query to run on or before a specific date, **Run in background** to run the query as resources are available, or **Run** to run the query immediately and view the results in the Query Center.

**STEP 7** | When you are ready, **View the Results of a Query**.
Cortex XDR Query Center

From the **Query Center** you can manage and view the results of all simple and complex queries created from the **Query Builder**. The Query Center displays information about the query including the query parameters and allows you to adjust and rerun queries as needed.

The following table describes the fields that are available for each query in alphabetical order.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CREATED BY</td>
<td>User who created or scheduled the query.</td>
</tr>
<tr>
<td>NUM OF RESULTS</td>
<td>Number of results returned by the query.</td>
</tr>
<tr>
<td>QUERY DESCRIPTION</td>
<td>The query parameters used to run the query.</td>
</tr>
<tr>
<td>QUERY ID</td>
<td>Unique identifier of the query.</td>
</tr>
<tr>
<td>QUERY NAME</td>
<td>For saved queries, the <strong>Query Name</strong> identifies the query specified by the administrator. For scheduled queries, the <strong>Query Name</strong> identifies the auto-generated name of the parent query. Scheduled queries also display an icon to the left of the name to indicate that the query is reoccurring.</td>
</tr>
<tr>
<td>QUERY STATUS</td>
<td>Status of the query:</td>
</tr>
<tr>
<td></td>
<td>• Queued—The query is queued and will run when there is an available slot.</td>
</tr>
<tr>
<td></td>
<td>• Running</td>
</tr>
<tr>
<td></td>
<td>• Failed</td>
</tr>
<tr>
<td></td>
<td>• Partially completed—The query was stopped after exceeding the maximum number of permitted results (10,000). To reduce the number of results returned, you can adjust the query settings and rerun.</td>
</tr>
<tr>
<td></td>
<td>• Stopped—The query was stopped by an administrator.</td>
</tr>
<tr>
<td></td>
<td>• Completed</td>
</tr>
<tr>
<td></td>
<td>• Deleted—The query was pruned.</td>
</tr>
<tr>
<td>RESULTS SAVED</td>
<td>Yes or No.</td>
</tr>
</tbody>
</table>
Manage Your Queries

From the Query Center, you can view details about and results for all manual and scheduled queries. The Query Center also provides management functions that allow you to modify, rerun, schedule, and remove queries. Similar to the Alerts page, you can also refresh the page to view updated status for queries, filter available queries based on fields in the query table, and manage the fields presented in the Query Center.

- View the Results of a Query
- Rename a Query
- Modify a Query
- Rerun or Schedule a Query to Run
- Manage Scheduled Queries

View the Results of a Query

After you run a query, you can view the events that match your search criteria. To view the results:

**STEP 1** Select INVESTIGATION > Query Center.

**STEP 2** Locate the query for which you want to view the results.

If necessary, use the Filter to reduce the number of queries Cortex XDR – Investigation and Response displays.

**STEP 3** Right click anywhere in the query row and then select Show results.

Cortex XDR displays the results in a new window.
STEP 4 | (Optional) In the Results table, you can export the results to a tab-separated values (TSV) file. and right-click:

- Analyze and open the causality view.
- Investigate in Timeline and open the Timeline page.
- View event log message to view the event details

STEP 5 | (Optional) If you want to refine your results, you can Modify a query from the query results.

### Modify a Query

After you run a query you might find you need to change your search parameters such as to narrow the search results or correct a search parameter. There are two ways you can modify a query: You can edit it in the Query Center, or you can edit it from the results page. Both methods populate the criteria you specified in the original query in a new query which you can modify and save.

- Create a query based on an existing query.
  1. Select INVESTIGATION > Query Center.
  2. Right click anywhere in the query and then select Save as a new query.
  3. If desired, enter a descriptive name to identity the query.
  4. Then modify the search parameters as desired.
  5. Choose when to run the query.

    Select the calendar icon to schedule a query to run on or before a specific date, Run in background to run the query as resources are available, or Run to run the query immediately and view the results in the Query Center.

- Modify an existing query from the Query Center.
  1. Select INVESTIGATION > Query Center.
2. Right click anywhere in the query and then **Edit a query.**
3. Modify the search parameters as desired.
4. Choose when to run the query.

Select the calendar icon to schedule a query to run on or before a specific date, **Run in background** to run the query as resources are available, or **Run** to run the query immediately and view the results in the **Query Center.**

- **Modify a query from the query results.**
  1. **View the Results of a Query.**
  2. At the top of the query, click the pencil icon to the right of the query parameters.
     Cortex XDR opens the query settings page.
  3. Modify the search parameters as desired.
  4. Choose when to run the query.

Select the calendar icon to schedule a query to run on or before a specific date, **Run in background** to run the query and review the result at a later time, or **Run** to run the query immediately and view the results in the **Query Center.**

**Rerun or Schedule a Query to Run**

If you want to rerun a query, you can either schedule it to run on or before a specific date, or you can rerun it immediately. Cortex XDR will create a new query in the Query Center. When the query completes, Cortex XDR displays a notification in the notification bar.

- **Rerun a query immediately.**
  1. Select **INVESTIGATION** > **Query Center.**
  2. Right click anywhere in the query and then select **Rerun Query.**
     Cortex XDR initiates the query immediately.

- **Schedule a query to run:**
  1. Select **INVESTIGATION** > **Query Center.**
  2. Right click anywhere in the query and then select **Schedule.**
  3. Choose the desired schedule option and the date and time the query should run:

  - Run one time query on a specific date
• Run query by date and time—Schedule a reoccurring query at a frequency of your choice.
4. Click OK to schedule the query.
   Cortex XDR creates a new query and schedules it to run on or by the selected date and time.
5. View the status of the scheduled query on the Cortex XDR Scheduled Queries page.
   At any time, you can view or make changes to the query on the Scheduled Queries page. For example, you can edit the frequency, view when the query will next run, or disable the query.

Rename a Query
If needed, you can rename a query at any time. If you later rerun the query, the new query will run using the new name. You can also edit the name of a query when you Modify a Query.

STEP 1 | Select INVESTIGATION > Query Center.
STEP 2 | Right click anywhere in the query and then select Rename.
STEP 3 | Enter the new query name and click OK.

Quick Launcher
The Quick Launcher provides a quick, in-context shortcut that you can use to search for information, perform common investigation tasks, or initiate response actions from any place in the Cortex XDR app. The tasks that you can perform with the Quick Launcher include:
• Search for host, username, IP address, domain, filename, or filepath, timestamp
• Blacklist or whitelist processes by SHA256? hash
• Add domains or IP addresses to the EDL blocklist
• Create a new IOC for an IP address, domain, hash, filename, or filepath
• Isolate an endpoint
• Open a terminal to a given endpoint
• Initiate a malware scan on an endpoint
You can bring up the Quick Launcher either using keyboard shortcuts—Ctrl-Shift+X on Windows or CMD+Shift+X on macOS—or using the Quick Launcher icon located in the top navigation bar. You can also prepopulate searches in Quick Launcher by selecting text in the app or selecting a node in the Causality or Timeline Views.
By default, Cortex XDR opens the Quick Launcher in the center of the page. To change the default position, drag the Quick Launcher to another preferred location. The next time you open the Quick Launcher, it opens in the previous location. To close the Quick Launcher, click Esc or click out of the Quick Launcher dialog.

Cortex XDR Scheduled Queries
From the Scheduled Queries page, you can easily view all scheduled and reoccurring queries created from the Query Builder. The Scheduled Queries page displays information about the query including the query parameters and allows you to adjust or modify the schedule as needed. To edit a query schedule, right click the query and select the desired action.
The following table describes the fields that are available for each query in alphabetical order.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CREATED BY</td>
<td>User who created or scheduled the query.</td>
</tr>
<tr>
<td>NEXT EXECUTION</td>
<td>Next execution time if the query is scheduled to run at a specific frequency. If the query was only scheduled to run at a specific time and date, this field will show None.</td>
</tr>
<tr>
<td>QUERY DESCRIPTION</td>
<td>The query parameters used to run the query.</td>
</tr>
<tr>
<td>QUERY ID</td>
<td>Unique identifier of the query.</td>
</tr>
<tr>
<td>QUERY NAME</td>
<td>For saved queries, the Query Name identifies the query specified by the administrator. For scheduled queries, the Query Name identifies the auto-generated name of the parent query. Scheduled queries also display an icon to the left of the name to indicate that the query is reoccurring.</td>
</tr>
<tr>
<td>SCHEDULE TIME</td>
<td>Frequency or time at which the query was scheduled to run.</td>
</tr>
<tr>
<td>TIMESTAMP</td>
<td>Date and time the query was created.</td>
</tr>
</tbody>
</table>

Manage Scheduled Queries

From the Scheduled Queries page, you can perform additional actions to manage your scheduled and reoccurring queries.
View Completed Queries

To view completed queries:

**STEP 1** | Select INVESTIGATION > Scheduled Queries.

**STEP 2** | Locate the scheduled query for which you want to view previous executions.

If necessary, use the **Filter** to reduce the number of queries Cortex XDR displays.

**STEP 3** | Right click anywhere in the query row and then select **Show all executed instances**.

Cortex XDR filters the queries on the **Query Center** and displays the results in a new window.

Edit the Query Frequency

**STEP 1** | Select INVESTIGATION > Scheduled Queries.

**STEP 2** | Locate the scheduled query that you want to edit.

If necessary, use the **Filter** to reduce the number of queries Cortex XDR displays.

**STEP 3** | Right click anywhere in the query row and then select **Edit**.

**STEP 4** | Adjust the schedule settings as needed, and then click **OK**.

Disable or Remove a Query

If you no longer need a query you can temporarily disable or permanently remove it.

**STEP 1** | Select INVESTIGATION > Scheduled Queries.

**STEP 2** | Locate the scheduled query that you want to change.

If necessary, use the **Filter** to reduce the number of queries Cortex XDR displays.

**STEP 3** | Right click anywhere in the query row and then select **Remove** to permanently remove the scheduled query, or **Disable** to temporarily stop the query from running at the scheduled time.

If you disable a query you can later return to the **Scheduled Queries** page and **Enable** it.

Rename a Scheduled Query

**STEP 1** | Select INVESTIGATION > Scheduled Queries.
STEP 2 | Locate the scheduled query that you want to change.  
If necessary, use the Filter to reduce the number of queries Cortex XDR displays.

STEP 3 | Right click anywhere in the query row and then select Rename.

STEP 4 | Edit the query name as desired, and then click OK.

Research a Known Threat

This topic describes what steps you can take to investigate a lead. A lead can be:
- An alert from a non-Palo Alto Networks system with information relevant to endpoints or firewalls.
- Information from online articles or other external threat intelligence that provides well-defined characteristics about the threat.
- Users or hosts that have been reported as acting abnormally.

STEP 1 | Use the threat intelligence you have to build a query using Cortex XDR Query Builder.  
For example, if external threat intelligence indicates a confirmed threat that involves specific files or behaviors, search for those characteristics.

STEP 2 | View the Results of a Query and refine as needed to filter out noise.  
See Modify a Query.

STEP 3 | Select an event of interest, and open the Causality View.  
Review the chain of execution and data, navigate through the processes on the tree, and analyze the information.

STEP 4 | Open the Timeline View to view the sequence of events over time.

STEP 5 | Inspect the information again, and identify any characteristics you can use to Create a BIOC Rule.  
If you can create a BIOC rule, test and tune it as needed.
Investigate Incidents

An attack event can affect several users or hosts and raise different types of alerts caused by a single event. You can track incidents, assign analysts to investigate, and document the resolution. For a record log of all actions taken by analysts in the incident, see Monitor Administrative Activity.

Use the following steps to investigate an incident:

STEP 1 | Select Incidents.

STEP 2 | From the Incidents table, locate the incident you want to investigate.

There are several ways you can filter or sort incidents:

- In the Status column for New incidents to view only the incidents that have not yet been investigated.
- In the Severity column, identify the incidents with the highest threat impact.
- In the Incident Sources column, filter according to the sources that raised the alerts which make up the incident.

After you locate an incident you want to investigate, right-click it and select View Incident.

The Incident details page aggregates all alerts, insights, and affected assets and artifacts from those alerts in a single location. From the Incident details page you can manage the alert and investigate an event within the context and scope of a threat. Select the pencil icon to edit the incident name and description.

STEP 3 | Assign an incident to an analyst.
Select the assignee (or Unassigned in the case of a new incident) below the incident description and begin typing the analyst’s email address for automated suggestions. Users must have logged into the app to appear in the auto-generated list.

**STEP 4 | Assign an incident status.**

Select the incident status *New* to update the status from New to Under Investigation to indicate which incidents have been reviewed and to filter by status in the incidents table.

**STEP 5 | Review the details of the incident, such as alerts and insights related to the event, and affected assets and artifacts.**

- **Investigate Key Artifacts.**
  
  Key Artifacts list files and file hashes, signers, processes, domains, and IP addresses that are related to the threat event. Each alert type contains certain key artifacts, and the app weighs and sorts alerts into Incidents based on the key artifacts. Different key artifacts have different weights according to their impact and case. The app analyzes the alert type, related causality chains, and key artifacts to determine which incident has the highest correlation with the alert, and the Cortex XDR app groups the alert with that incident.

  The app also displays any available threat intelligence for the artifact. The Threat Intelligence column in the Key Artifacts panel lists the WildFire (WF) verdicts associated with each artifact and identifies any malware with a red malware icon. If you also integrate additional threat intelligence, this section can also display VirusTotal (VT) scores and AutoFocus (AF) tags. For additional information, see External Integrations.

  Right-click a file or process under Key Artifacts to view the entire artifact report from the threat intelligence source.

  - View VirusTotal and AutoFocus reports.
  - Add to Whitelist. Artifacts added to whitelist are displayed with ✓
  - Add to Blacklist. Artifacts added to blacklist are displayed with ✗
  - Open Hash View to display detailed information about the files and processes relating to the hash.
  - Open IP Address View to display detailed information about the IP address.

- **Investigate Key Assets.**

  Key Assets identify the scope of endpoints and users affected by the threat. Right-click an asset to filter alerts by that asset.

- **Investigate Alerts.**

  Incidents are created only through high or medium severity alerts. Low and informational severity alerts are categorized as Insights and are available on the Insights tab. In the incident, review the alerts and, if additional context is required, review the related insights. You can also view high, medium, and low severity alerts in the main Alerts table.

  During your investigation, you can also perform additional management of alerts, which include:

  - Analyze an Alert
  - View the Alert Causality
  - Timeline View
  - Copy Alerts
  - Build an Alert Exclusion Policy from Alerts in an Incident

**STEP 6 | (Optional) Change the incident severity.**
The default severity is based on the highest alert in the incident. To manually change the severity select Actions > Change Incident Severity and choose the new severity. The smaller severity bubble indicates the original severity.

STEP 7 | Track and share your investigation progress.

Add notes or comments to track your investigative steps and any remedial actions taken.

- Select the Incident Notepad (📝) to add and edit the incident notes. You can use notes to add code snippets to the incident or add a general description of the threat.
- Use the comments to coordinate the investigation between analysts and track the progress of the investigation. Select the comments (💬) to view or manage comments. Collapse the comment threads for an overview of the discussion. If needed, Search to find specific words or phrases in the comments.

STEP 8 | Resolve the incident.

After the incident is resolved:
1. Set the status to Resolved.
   Select the status from the Incident details or select Actions > Change Incident Status.
2. Select the reason the resolution was resolved.
3. Add a comment that explains the reason for closing the incident.
4. Select OK.

The Cortex XDR app no longer adds new alerts to the resolved incident and instead adds incoming alerts to a new incident.

External Integrations

Cortex XDR supports the following integrations.

<table>
<thead>
<tr>
<th>Integration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threat Intelligence</td>
<td></td>
</tr>
</tbody>
</table>
## Integration

<table>
<thead>
<tr>
<th>Integration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>WildFire®</strong></td>
<td>Cortex XDR automatically includes WildFire threat intelligence in incident and alert investigation. WildFire detects known and unknown threats, such as malware. The WildFire verdict contains detailed insights into the behavior of identified threats. The WildFire verdict displays next to relevant <strong>Key Artifacts</strong> in the incidents details page.</td>
</tr>
<tr>
<td><strong>AutoFocus™</strong></td>
<td>AutoFocus groups conditions and indicators related to a threat with a <em>tag</em>. Tags can be user-defined or come from threat-research team publications and are divided into classes, such as <em>exploit</em>, <em>malware family</em>, and <em>malicious behavior</em>. See the <a href="#">AutoFocus Administrator’s Guide</a> for more information on AutoFocus tags. To view AutoFocus tags in Cortex XDR incidents, you must obtain the license key for the service and add it to the Cortex XDR Configuration. When you add the service, the relevant tags display in the incident details page under <strong>Key Artifacts</strong>.</td>
</tr>
<tr>
<td><strong>VirusTotal</strong></td>
<td>VirusTotal provides aggregated results from over 70 antivirus scanners, domain blacklist services, and user contributions. The VirusTotal score is represented as a fraction, where, for example, a score of <em>34/52</em> means out of 52 queried services, 34 services determined the artifact to be malicious. To view VirusTotal threat intelligence in Cortex XDR incidents, you must obtain the license key for the service and add it to the Cortex XDR Configuration. When you add the service, the relevant VirusTotal (VT) score displays in the incident details page under <strong>Key Artifacts</strong>.</td>
</tr>
</tbody>
</table>

## Incident Management

| Cortex XSOAR | Cortex XSOAR enables automated and coordinated threat response with the ability to adjust and test response playbooks. When used with Cortex XDR, you can manage incidents from the Cortex XSOAR interface and leverage the Cortex XDR Causality Analytics Engine and detection capabilities. Changes to one app are reflected in the other. |
| Third-party ticketing systems | To manage incidents from the application of your choice, you can use the Cortex XDR API Reference to send alerts and alert details to an external receiver. After you generate your API key and set up the API to query Cortex XDR, external apps can receive incident updates, request additional data about incidents, and make changes such as to set the status and change the severity, or assign an owner. To get started, see the Cortex XDR API Reference. |
Create an Incident Starring Configuration

To help you focus on the incidents that matter most, you can create an incident starring configuration that categorizes and stars incidents when alerts contain attributes that you decide are important. After you define an incident starring configuration, Cortex XDR adds a star indicator to any incidents that contain alerts that match the configuration.

You can then sort or filter the Incidents table for incidents containing starred alerts. In addition, you can also choose whether to display all incidents or only starred incidents on the Incidents Dashboard.

**STEP 1 |** In Cortex XDR, select Incidents > Starred Alerts.

**STEP 2 |** + Add Starring Configuration

**STEP 3 |** Enter a Configuration Name to identify your starring configuration.

**STEP 4 |** Enter a descriptive Comment that identifies the reason or purpose of the starring configuration.

**STEP 5 |** Use the alert filters to build the match criteria for the policy. You can also right-click a specific value in the alert to add it as match criteria. The app refreshes to show you which alerts in the incident would be included.
STEP 6 | **Create** the policy and confirm the action.

If you later need to make changes, you can view, modify, or delete the exclusion policy from the **Incidents > Starred Incidents** page.
Investigate Artifacts and Assets

- IP Address View
- Hash View

Investigate an IP Address

The IP Address View provides a powerful way to investigate and take action on IP addresses by reducing the number of steps it takes to collect, research, and threat hunt related incidents. Cortex XDR automatically aggregates and displays a summary of all the information Cortex XDR and threat intelligence services have regarding a specific IP address over a defined 24-hour or 7-day time frame.

To help you determine whether an IP address is malicious, the IP Address View displays an interactive visual representation of the collected activity for a specific IP address.

To investigate an IP address:

**STEP 1 | Open the IP View for an IP address.**

You can access the view from every IP address in Cortex XDR console by either right-click > Open IP View, selecting the IP address and using the keyboard shortcut Ctrl/CMD+Shift+E combination, or searching for a specific IP address in the Quick Launcher.

**STEP 2 | Review the overview for the IP address.**

The overview displays network operations, incidents, actions, and threat intelligence information relating to a specific IP address and provides a summary of the network operations and processes related to the IP.

1. Review the auto generated summary of the number of network operations and processes related to the IP that occurred over the past 7 days.
2. Review the location of the IP address.
• **External**—IP address is located outside of your organization. Displays the country flag if the location information is available.
• **Internal**—IP address is from within your organization. The XDR Agent icon is displayed if the corresponding endpoint identified by the IP address has an agent is installed at that point in time.

3. **Identify the IOC severity.**

The color of the IP address value is color-coded to indicate the IOC severity.
• Low—Blue
• Medium—Yellow
• High—Red

4. **Review any available threat intelligence for the IP address.**

Depending on the threat intelligence sources that you integrate with Cortex XDR, you can review any of the following threat intelligence.
• **Virus Total** score and report

  Requires a license key. Navigate to > Settings > Integrations > Threat Intelligence.
• **Whois** identification data for the specific IP address.
• **IOC** Rule, if applicable, including the IOC **Severity**, **Number of hits**, and **Source**.
• **EDL** IP address if the IP address was added to an EDL.

5. **Review any related incidents:**

**Recent Open Incidents** lists the last 3 incidents which contain the specific IP address as part of the incident **Key Artifacts** according to the **Last Updated** timestamp. To dive deeper into specific incidents, you can select the Incident ID. If more than three incidents are displayed, select **View All**. Cortex XDR displays **Recently Updated Incidents** which filters incidents for those that contain the IP address.

**STEP 3 | Filter the IP address information you want to visualize.**

Select from the following criteria to refine the scope of your IP address information you want visualized. Each selection aggregates the displayed data.

<table>
<thead>
<tr>
<th>Filter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Primary</strong></td>
<td>The main set of values you want to display. The values depend on the selected <strong>Connection Type</strong>.</td>
</tr>
<tr>
<td></td>
<td>• <strong>All Aggregations</strong>—Summary of all the related IP data.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Destination/Source Country</strong></td>
</tr>
<tr>
<td></td>
<td>• <strong>Destination/Source Port</strong></td>
</tr>
<tr>
<td></td>
<td>• <strong>Destination/Source IP</strong></td>
</tr>
<tr>
<td></td>
<td>• <strong>Destination/Source Process</strong></td>
</tr>
<tr>
<td></td>
<td>• <strong>App-ID</strong></td>
</tr>
<tr>
<td><strong>Secondary</strong></td>
<td>The set of values you want to apply as the secondary set of aggregations. Must differ than your <strong>Primary</strong> selection:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Destination Country</strong></td>
</tr>
<tr>
<td></td>
<td>• <strong>Destination/Source Port</strong></td>
</tr>
<tr>
<td></td>
<td>• <strong>Destination/Source IP</strong></td>
</tr>
<tr>
<td>Filter</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
|             | • Destination/Source Process  
|             | • App-ID  
| Node Size   | The node size to display for the type of values.  
|             | • Number of Connections  
|             | • Total Traffic  
|             | • Total Download  
|             | • Total Upload  
| Showing     | The number of the Primary and Secondary aggregated values are incoming or outgoing connections.  
|             | • Top 5  
|             | • Top 3  
|             | • Bottom 5  
|             | • Bottom 3  
| Connection Type | Type of connection you want to display your defined set of values.  
|             | • Incoming  
|             | • Outgoing  
| Timeframe   | Time period over which to display your defined set of values.  
|             | • 24 Hours  
|             | • 7 Days  

Select ![to apply your selections and update the information displayed in the visualization pane. If necessary, Refresh to retrieve data.](image)

**STEP 4** After reviewing the available information for the IP address, take action if desired:

Depending on the current IOC and EDL status, select Actions to:

• Edit Rule  
• Disable Rule  
• Delete Rule  
• Add to EDL

**Hash View**

The file and process Hash View provides a powerful way to investigate and take action on SHA256 hash processes and files by reducing the number of steps it takes to collect, research, and threat hunt related incidents. The Hash View automatically aggregates and displays a summary of all the information Cortex XDR and threat intelligence services have regarding a specific SHA256 hash over a defined 7 or 30 day timeframe.

The Hash View allows you to drill down on each of the process executions, file operations, incidents, actions, and threat intelligence reports relating to the hash.
The Hash View is accessible from every SHA256 hash in Cortex XDR console by either right-click > Open Hash View, selecting the hash and using the keyboard shortcut Ctrl/CMD+Shift+E combination, or searching for a specific hash in the Quick Launcher.

### The Hash View

The hash view is comprises five sections:

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Available Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) Overview</td>
<td>Displays the hash value and signature information.</td>
<td>•  [Add alias]—Add an alias name to the hash. Maximum of 64 characters.</td>
</tr>
<tr>
<td></td>
<td>[Add alias] represents the Wildfire report verdict according to the following color codes:</td>
<td>•  [Add comment]—Add a comment to the hash ID. Maximum of 256 characters.</td>
</tr>
<tr>
<td></td>
<td>• Blue—Benign</td>
<td>•  [Open Causality View]—Toggle to display file and process hash data during a 7 or 30 day time frame in the Investigation and Recently Updated Incidents sections.</td>
</tr>
<tr>
<td></td>
<td>• Yellow—Grayware</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Red—Malware</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Light gray—Unknown verdict</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Dark gray—The verdict is inconclusive</td>
<td></td>
</tr>
<tr>
<td>(2) Investigation</td>
<td>Displays an auto generated summary of the number of file activity and process executions related to the hash over the selected time frame.</td>
<td>•  Toggle between Recent Execution and Recent File Operation—View the 5 most recent processes or files per distinct hosts.</td>
</tr>
<tr>
<td></td>
<td>The table lists the top 5 most recent processes and file operations that include the hash ID during the selected time frame. Data is displayed on a distinct host.</td>
<td>•  Recent Executions Tab—Select an specific process—Opens the process execution event Causality View.</td>
</tr>
</tbody>
</table>

The Hash View is accessible from every SHA256 hash in Cortex XDR console by either right-click > Open Hash View, selecting the hash and using the keyboard shortcut Ctrl/CMD+Shift+E combination, or searching for a specific hash in the Quick Launcher.
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Available Actions</th>
</tr>
</thead>
</table>
|                                 | For example, if the hash is found in events and files on different hosts, each host is displayed on a separate table row. | - Select **Search all process Executions**—Runs a query on the hash and timetsamp of the process execution according to the 7/30 day toggle, and opens the Cortex XDR console query **Results** table.  
- **Recent File Operation Tab**  
  - Select a specific file operation—Opens the file event **Causality View**.  
  - Select **Search all File Operations**—Runs a query on the hash and timetsamp according to the 7/30 day toggle of the file operation, and opens the Cortex XDR console query **Results** table. |
| (3) Recently Updated Incidents  | Displays the top 3 incidents which contain the hash as part of the incident **Key Artifacts** according to the **Last Updated** timestamp. | - Select a specific incident—Opens the Cortex XDR console **Incident View**.  
  - If more than three incidents are displayed, select **View All**—Opens a **Recently Updated Incidents** window, displaying the **Incidents** table filtered according to all the incidents that contain the hash. |
| (4) Actions                     | Lists the actions taken on the hash.                                       | - Select an action **Blacklisted**, **Whitelisted**, or **Quarantined**—Opens the Cortex XDR console **Action Center** table filtered according to selected action initiated on the hash.  
  - Depending on the current hash status, you can  
    - Add to Whitelist  
    - Add to Blacklist  
    - Create IOC rule |
| (5) Threat Intelligence         | Displays the following threat intelligence service results if available:  
  - **Virus Total** Score—Requires a license key. Navigate to [VirusTotal](https://www.virustotal.com) | - Select a threat intelligence service to view the report results:  
  - **VirusTotal**—Opens the web based report. |
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Available Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Settings &gt; Integrations &gt; Threat Intelligence</strong></td>
<td>and enter your <strong>License Key</strong>.</td>
<td><strong>AutoFocus</strong>—Each tag opens a summary window of the tag <strong>Source</strong>, <strong>Tag Class</strong>, and <strong>Description</strong></td>
</tr>
<tr>
<td>• <strong>AutoFocus</strong> Tags—Requires a license key. Navigate to <strong>Settings &gt; Integrations &gt; Threat Intelligence</strong> and enter your License Key.</td>
<td><strong>WildFire</strong>—Opens the Cortex XDR console <strong>WildFire Analysis Report</strong> window.</td>
<td></td>
</tr>
<tr>
<td>• <strong>Wildfire Verdict</strong></td>
<td><strong>IOC</strong>—Opens the Cortex XDR console <strong>IOC Rules</strong> table, filtered according to the IOC rule containing the hash.</td>
<td></td>
</tr>
</tbody>
</table>
Investigate Alerts

- Cortex XDR Alerts
- Triage Alerts
- Manage Alerts
- Alert Exclusions
- Causality View
- Network Causality View
- Timeline View
- Analytics Alert View

Cortex XDR Alerts

The **Alerts** page displays a table of all alerts in Cortex XDR.

The **Alerts** page consolidates non-informational alerts from your detection sources to enable you to efficiently and effectively triage the events you see each day. By analyzing the alert, you can better understand the cause of what happened and the full story with context to validate whether an alert requires additional action. Cortex XDR supports saving 2M alerts per 4000 agents or 20 terabyte, half of the alerts are allocated for informational alerts, and half for severity alerts.

To view detailed information for an alert, you can also view details in the **Causality View** and **Timeline View**. From these views you can also view related informational alerts that are not presented on the **Alerts** page.

By default, the **Alerts** page displays the alerts that it received over the last seven days (to modify the time period, use the page filters). Every 12 hours, Cortex XDR enforces a cleanup policy to remove the oldest alerts that exceed the maximum alerts limit.

The following table describes both the default fields and additional optional fields that you can add to the alerts table using the **column manager** and lists the fields in alphabetical order.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status Indicator ( • )</td>
<td>Identifies whether there is enough endpoint data to analyze an alert.</td>
</tr>
<tr>
<td></td>
<td>Check box to select one or more alerts on which to perform actions. Select multiple alerts to assign all</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>ACTION</td>
<td>Action taken by the alert sensor with action status displayed in parenthesis:</td>
</tr>
<tr>
<td></td>
<td>• Detected</td>
</tr>
<tr>
<td></td>
<td>• Detected (Download)</td>
</tr>
<tr>
<td></td>
<td>• Detected (Post Detected)</td>
</tr>
<tr>
<td></td>
<td>• Detected (Prompt Allow)</td>
</tr>
<tr>
<td></td>
<td>• Detected (Reported)</td>
</tr>
<tr>
<td></td>
<td>• Detected (Scanned)</td>
</tr>
<tr>
<td></td>
<td>• Prevented (Blocked)</td>
</tr>
<tr>
<td></td>
<td>• Prevented (Prompt Block)</td>
</tr>
<tr>
<td>AGENT OS SUBTYPE</td>
<td>The operating system subtype of the agent from which the alert was triggered.</td>
</tr>
<tr>
<td>ALERT ID</td>
<td>A unique identifier that Cortex XDR assigns to each alert.</td>
</tr>
</tbody>
</table>
| ALERT NAME          | Module that triggered the alert. If the alert was generated by Cortex XDR, the Alert Name will be the specific Cortex XDR rule that created the alert (BIOC or IOC rule name). If from an external system, it will carry the name assigned to it by Cortex XDR. Alerts that match an alert starring policy also display a purple star.  

For alerts coming from firewalls, if duplicate alerts with the same name and host are raised within 24 hours, they are aggregated and identified by a +n tag.

<p>| ALERT SOURCE        | Source of the alert: BIOC, Analytics BIOC, IOC, XDR Agent, Firewall, or Analytics.                                                            |
| APP ID              | Related App-ID for an alert. App-ID is a traffic classification system that determines what an application is irrespective of port, protocol, encryption (SSH or SSL) or any other evasive tactic used by the application. When known, you can also pivot to the Palo Alto Networks Applipedia entry that describes the detected application. |
| APP CATEGORY        | APP-ID category name associated with a firewall alert.                                                                                         |
| APP SUBCATEGORY     | APP-ID subcategory name associated with a firewall alert.                                                                                       |</p>
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>APP TECHNOLOGY</td>
<td>APP-ID technology name associated with a firewall alert.</td>
</tr>
<tr>
<td>CATEGORY</td>
<td>Alert category based on the alert source. An example of an XDR Agent alert category is Exploit Modules. An example of a BIOC alert category is Evasion. If a URL filtering category is known, this field also displays the name of the URL filtering category.</td>
</tr>
<tr>
<td>CGO CMD</td>
<td>Command-line arguments of the Causality Group Owner.</td>
</tr>
<tr>
<td>CGO MD5</td>
<td>The MD5 value of the CGO which initiated the alert.</td>
</tr>
<tr>
<td>CGO NAME</td>
<td>The name of the process that started the causality chain based on Cortex XDR causality logic.</td>
</tr>
<tr>
<td>CGO SIGNATURE</td>
<td>Signing status of the CGO:</td>
</tr>
<tr>
<td></td>
<td>• Unsigned</td>
</tr>
<tr>
<td></td>
<td>• Signed</td>
</tr>
<tr>
<td></td>
<td>• Invalid Signature</td>
</tr>
<tr>
<td></td>
<td>• Unknown</td>
</tr>
<tr>
<td>CGO SIGNER</td>
<td>The name of the software publishing vendor that signed the file in the causality chain that led up to the alert.</td>
</tr>
<tr>
<td>CID</td>
<td>Unique identifier of the causality instance generated by Cortex XDR.</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>Text summary of the event including the alert source, alert name, severity, and file path. For alerts triggered by BIOC and IOC rules, Cortex XDR displays detailed information about the rule.</td>
</tr>
<tr>
<td>DESTINATION ZONE NAME</td>
<td>The destination zone of the connection for firewall alerts.</td>
</tr>
<tr>
<td>DOMAIN</td>
<td>The domain on which an alert was triggered.</td>
</tr>
<tr>
<td>EMAIL RECIPIENT</td>
<td>The email recipient value of a firewall alerts triggered on the content of a malicious email.</td>
</tr>
<tr>
<td>EMAIL SENDER</td>
<td>The email sender value of a firewall alerts triggered on the content of a malicious email.</td>
</tr>
<tr>
<td>EMAIL SUBJECT</td>
<td>The email subject value of a firewall alerts triggered on the content of a malicious email.</td>
</tr>
<tr>
<td>EVENT TYPE</td>
<td>The type of event on which the alert was triggered:</td>
</tr>
<tr>
<td></td>
<td>• File Event</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Field</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td></td>
<td>• Injection Event</td>
</tr>
<tr>
<td></td>
<td>• Load Image Event</td>
</tr>
<tr>
<td></td>
<td>• Network Event</td>
</tr>
<tr>
<td></td>
<td>• Process Execution</td>
</tr>
<tr>
<td></td>
<td>• Registry Event</td>
</tr>
<tr>
<td>EXCLUDED</td>
<td>Whether the alert is excluded by an exclusion configuration.</td>
</tr>
<tr>
<td>EXTERNAL ID</td>
<td>The alert ID as recorded in the detector from which this alert was sent.</td>
</tr>
<tr>
<td>FILE PATH</td>
<td>When the alert triggered on a file (the Event Type is File) this is the path to the file on the endpoint. If not, then N/A.</td>
</tr>
<tr>
<td>FILE MACRO SHA256</td>
<td>SHA256 hash value of an Microsoft Office file macro</td>
</tr>
<tr>
<td>FILE MD5</td>
<td>MD5 hash value of the file.</td>
</tr>
<tr>
<td>FILE SHA256</td>
<td>SHA256 hash value of the file.</td>
</tr>
<tr>
<td>FW NAME</td>
<td>Name of firewall device relevant for firewall alerts.</td>
</tr>
<tr>
<td>FW RULE ID</td>
<td>The firewall rule ID that matches the network traffic that triggered the firewall alert.</td>
</tr>
<tr>
<td>FW RULE NAME</td>
<td>The firewall rule name that matches the network traffic that triggered the firewall alert.</td>
</tr>
<tr>
<td>FW SERIAL NUMBER</td>
<td>The firewall serial number of the firewall alerts.</td>
</tr>
<tr>
<td>HOST NAME</td>
<td>The endpoint or server on which this alert triggered.</td>
</tr>
<tr>
<td>HOST OS</td>
<td>Operating system of the endpoint or server on which this alert triggered.</td>
</tr>
<tr>
<td>INITIATED BY</td>
<td>The name of the process that initiated an activity such as a network connection or registry change.</td>
</tr>
<tr>
<td>INITIATOR MD5</td>
<td>The MD5 value of the process which initiated the alert.</td>
</tr>
<tr>
<td>INITIATOR SHA256</td>
<td>The SHA256 hash value of the initiator.</td>
</tr>
<tr>
<td>INITIATOR CMD</td>
<td>Command-line used to initiate the process including any arguments.</td>
</tr>
<tr>
<td>INITIATOR SIGNATURE</td>
<td>Signing status of the process that initiated the activity:</td>
</tr>
<tr>
<td></td>
<td>• Unsigned</td>
</tr>
<tr>
<td></td>
<td>• Signed</td>
</tr>
<tr>
<td></td>
<td>• Invalid Signature</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>INITIATOR SIGNER</td>
<td>Signer of the process that triggered the alert.</td>
</tr>
<tr>
<td>LOCAL IP</td>
<td>If the alert triggered on network activity (the Event Type is Network Connection) this is the IP address of the host that triggered the alert. If not, then N/A.</td>
</tr>
<tr>
<td>LOCAL PORT</td>
<td>If the alert triggered on network activity (the Event Type is Network Connection) this is the port on the endpoint that triggered the alert. If not, then N/A.</td>
</tr>
<tr>
<td>MAC ADDRESS</td>
<td>The MAC address on which the alert was triggered.</td>
</tr>
<tr>
<td>MITRE ATT&amp;CK TACTIC</td>
<td>Displays the type of MITRE ATT&amp;CK tactic the alert is attempting to trigger on.</td>
</tr>
<tr>
<td>MITRE ATT&amp;CK TECHNIQUE</td>
<td>Displays the type of MITRE ATT&amp;CK technique the alert is attempting to trigger on.</td>
</tr>
<tr>
<td>NGFW VSYS NAME</td>
<td>Name of the firewall virtual system of a firewall triggered alert.</td>
</tr>
<tr>
<td>OS PARENT CREATED BY</td>
<td>Name of the parent operating system that created the alert.</td>
</tr>
<tr>
<td>OS PARENT CMD</td>
<td>Command-line used to by the parent operating system to initiate the process including any arguments.</td>
</tr>
<tr>
<td>OS PARENT SIGNATURE</td>
<td>Signing status of the operating system of the activity:</td>
</tr>
<tr>
<td></td>
<td>• Unsigned</td>
</tr>
<tr>
<td></td>
<td>• Signed</td>
</tr>
<tr>
<td></td>
<td>• Invalid Signature</td>
</tr>
<tr>
<td></td>
<td>• Unknown</td>
</tr>
<tr>
<td>OS PARENT SIGNER</td>
<td>Parent operating system signer.</td>
</tr>
<tr>
<td>OS PARENT SH256</td>
<td>Parent operating system SHA256 hash value.</td>
</tr>
<tr>
<td>OS PARENT ID</td>
<td>Parent operating system ID.</td>
</tr>
<tr>
<td>PROCESS EXECUTION SIGNATURE</td>
<td>Signature status of the process that triggered the alert:</td>
</tr>
<tr>
<td></td>
<td>• Unsigned</td>
</tr>
<tr>
<td></td>
<td>• Signed</td>
</tr>
<tr>
<td></td>
<td>• Invalid Signature</td>
</tr>
<tr>
<td></td>
<td>• Unknown</td>
</tr>
<tr>
<td>PROCESS EXECUTION SIGNER</td>
<td>Signer of the process that triggered the alert.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>REGISTRY DATA</td>
<td>If the alert triggered on registry modifications (the Event Type is Registry) this is the registry data that triggered the alert. If not, then N/A.</td>
</tr>
<tr>
<td>REGISTRY FULL KEY</td>
<td>If the alert triggered on registry modifications (the Event Type is Registry) this is the full registry key that triggered the alert. If not, then N/A.</td>
</tr>
<tr>
<td>REMOTE HOST</td>
<td>If the alert triggered on network activity (the Event Type is Network Connection) this is the the remote host name that triggered the alert. If not, then N/A.</td>
</tr>
<tr>
<td>REMOTE IP</td>
<td>The remote IP address of a network operation that triggered the alert.</td>
</tr>
<tr>
<td>REMOTE PORT</td>
<td>The remote port of a network operation that triggered the alert.</td>
</tr>
<tr>
<td>SEVERITY</td>
<td>The severity that was assigned to this alert when it was triggered (or modified): Informational, Low, Medium, High, or Unknown. For BIOC and IOCs, you define the severity when you create the rule. Insights are low and informational severity alerts that do not raise incidents, but provide additional details when investigating an event.</td>
</tr>
<tr>
<td>STARRED</td>
<td>Whether the alert is starred by starring configuration.</td>
</tr>
<tr>
<td>SOURCE ZONE NAME</td>
<td>The source zone name of the connection for firewall alerts.</td>
</tr>
<tr>
<td>TARGET FILE SHA256</td>
<td>The SHA256 hash value of an external DLL file that triggered the alert.</td>
</tr>
<tr>
<td>TARGET PROCESS CMD</td>
<td>The command-line of the process whose creation triggered the alert.</td>
</tr>
<tr>
<td>TARGET PROCESS NAME</td>
<td>The name of the process whose creation triggered the alert.</td>
</tr>
<tr>
<td>TARGET PROCESS SHA256</td>
<td>The SHA256 value of the process whose creation triggered the alert.</td>
</tr>
<tr>
<td>TIMESTAMP</td>
<td>The date and time when the alert was triggered.</td>
</tr>
<tr>
<td>URL</td>
<td>The URL destination address of the domain triggering the firewall alert.</td>
</tr>
<tr>
<td>XFF</td>
<td>X-Forwarded-For value from the HTTP header of the IP address connecting with a proxy.</td>
</tr>
</tbody>
</table>
From the **Alerts** page, you can also perform additional actions to manage alerts and pivot on specific alerts for deeper understanding of the cause of the event.

- Manage Alerts
- Causality View
- Timeline View
- Analytics Alert View

**Triage Alerts**

When the Cortex XDR app displays a new alert on the **Alerts** page, use the following steps to investigate and triage the alert:

**STEP 1 |** Review the data shown in the alert such as the command-line arguments (CMD), process info, etc.

For more information about the alert fields, see Cortex XDR Alerts.

**STEP 2 |** **Analyze** the chain of execution in the **Causality View**.

When the app correlates an alert with additional endpoint data, the Alerts table displays a green dot to the left of the alert row to indicate the alert is eligible for analysis in the Causality View. If the alert has a gray dot, the alert is not eligible for analysis in the Causality View. This can occur when there is no data collected for an event, or the app has not yet finished processing the EDR data. To view the reason analysis is not available, hover over the gray dot.

**STEP 3 |** Review the **Timeline View** of review the sequence of events over time.

The timeline is available for alerts that have been stitched with endpoint data.

**STEP 4 |** If deemed malicious, consider responding by isolating the endpoint from the network.

**STEP 5 |** RemEDIATE the endpoint and return the endpoint from isolation.

**STEP 6 |** Inspect the information again to identify any behavioral details that you can use to **Create a BIOC Rule**.

If you can create a BIOC rule, test and tune the logic for the rule, and then save it.

**Manage Alerts**

From the **Alerts** page, you can manage the alerts you see and the information Cortex XDR displays about each alert.
• Copy Alerts
• Analyze an Alert
• Create Profile Exceptions
• View Generating BIOC Rule
• Retrieve Alert Details
• Add an Alert Exclusion Policy
• Forward Alerts to an External Service

Copy Alerts

There are two ways you can copy an alert into memory: you can copy the URL of the alert record, or you can copy the value for an alert field. With either option, you can paste the contents of memory into an email to send. This is helpful if you need to share or discuss a specific alert with someone. If you copy a field value, you can also easily paste it into a search or begin a query.

• Create a URL for an alert record:
  1. From the Alerts page, right-click the alert you want to send.
  2. Select Copy alert URL.
     Cortex XDR saves the URL to memory.
  3. Paste the URL into an email or use as needed to share the alert.

• Copy a field value in an alert record:
  1. From the Alerts page, right-click the field in the alert that you want to copy.
  2. Select Copy.
     Cortex XDR saves the field contents to memory.
  3. Paste the value into an email or use as needed to share information from the alert.

Analyze an Alert

To help you understand the full context of an alert, Cortex XDR provides a powerful analysis view that empowers you to make a thorough analysis very quickly.

The Causality View is available for XDR agent alerts that are based on endpoint data and for alerts raised on network traffic logs that have been stitched with endpoint data.

To view the analysis:

STEP 1 | From the Alerts page, locate the alert you want to analyze.

STEP 2 | Right-click anywhere in the alert, and select Analyze.
Cortex XDR opens the alert in the Causality View.

STEP 3 | Review the chain of execution and available data for the process and, if available, navigate through the processes tree.

Create Profile Exceptions

Quickly create exception for Window processes, BTP, and JAVA deserialization alerts directly from the Alerts table.

STEP 1 | Right-click an alert of source XDR Agent, category Exploit, and select Create alert exception.
Cortex XDR opens a Create Alert Exception window detailing the exception parameters.

**STEP 2** | Select an **Exception Scope**:
- **Global** - Applies the exception across your organization.
- **Profile** - Select an existing profile or click and enter a **Profile Name** to create a new profile.

**STEP 3** | Add.

**STEP 4** | *(Optional)* View your profile exceptions.
1. Navigate to **Endpoints > Policy Management > Profiles**.
2. In the **Profiles** table, locate the OS in which you created your global or profile exception and right-click to view or edit the exception properties.

**View Generating BIOC Rule**
Easily view the BIOC or IOC rules that generated alerts directly from the Alerts table.

**STEP 1** | From the **Alerts** page, locate alerts with **Alert Sources**: XDR BIOC and XDR IOC.

**STEP 2** | Right-click the row, and select **View generating rule**.
Cortex XDR opens the BIOC rule that generated the alert in the **BIOC Rules** page. If the rule has been deleted, an empty table is displayed.

**STEP 3** | Review the rule, if necessary, right-click to perform available actions.

**Retrieve Alert Details**
To easily access additional information relating to an alert:

**STEP 1** | From the **Alerts** page, locate the alert for which you want to retrieve information.

**STEP 2** | Right-click anywhere in the alert, and select one of the following options:
- Retrieve alert data
- Retrieve related files
- View full endpoint details

**STEP 3** | Navigate to **Response > Action Center** to view retrieval status.

**Alert Exclusions**
The **Alert Exclusions** page displays all alert exclusions in Cortex XDR.

An **alert exclusion** is a policy that contains a set of alert match criteria that you want to suppress from Cortex XDR. You can **Add an Alert Exclusion Policy** from scratch or you can base the exclusion off of alerts that you investigate in an incident. After you create an exclusion policy, Cortex XDR hides any future alerts
that match the criteria from incidents and search query results. If you choose to apply the policy to historic results as well as future alerts, the app identifies any historic alerts as grayed out.

The following table describes both the default fields and additional optional fields that you can add to the alert exclusions table and lists the fields in alphabetical order.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check box</td>
<td>Check box to select one or more alert exclusions on which you want to perform actions.</td>
</tr>
<tr>
<td>BACKWARD SCAN STATUS</td>
<td>Exclusion policy status for historic data, either enabled if you want to apply the policy to previous alerts or disabled if you don’t want to apply the policy to previous alerts.</td>
</tr>
<tr>
<td>COMMENT</td>
<td>Administrator-provided comment that identifies the purpose or reason for the exclusion policy.</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>Text summary of the policy that displays the match criteria.</td>
</tr>
<tr>
<td>MODIFICATION DATE</td>
<td>Date and time when the exclusion policy was created or modified.</td>
</tr>
<tr>
<td>NAME</td>
<td>Descriptive name provided to identify the exclusion policy.</td>
</tr>
<tr>
<td>POLICY ID</td>
<td>Unique ID assigned to the exclusion policy.</td>
</tr>
<tr>
<td>STATUS</td>
<td>Exclusion policy status, either enabled or disabled.</td>
</tr>
<tr>
<td>USER</td>
<td>User that last modified the exclusion policy.</td>
</tr>
<tr>
<td>USER EMAIL</td>
<td>Email associated with the administrative user.</td>
</tr>
</tbody>
</table>

Add an Alert Exclusion Policy

Through the process of triaging alerts or resolving an incident, you can determine a specific alert does not indicate a threat. If you do not want Cortex XDR to display alerts that match certain criteria, you can create an alert exclusion policy. After you create an exclusion policy, Cortex XDR hides any future alerts that match the criteria, and excludes the alerts from incidents and search query results. If you choose to apply the policy to historic results as well as future alerts, the app identifies any historic alerts as grayed out.

If an incident contains only alerts with exclusions, Cortex XDR changes the incident status to Resolved - False Positive and sends an email notification to the incident assignee (if set).

There are two ways to create an exclusion policy. You can define the exclusion criteria when you investigate an incident or you can create an alert exclusion from scratch.

- Build an Alert Exclusion Policy from Alerts in an Incident
- Build an Alert Exclusion Policy from Scratch
Build an Alert Exclusion Policy from Alerts in an Incident

If after reviewing the incident details, if you want to suppress one or more alerts from appearing in the future, create an exclusion policy based on the alerts in the incident. When you create an incident from the incident view, you can define the criteria based on the alerts in the incident. If desired, you can also Create Alert Exclusions from scratch.

STEP 1 | From the Incident view in Cortex XDR, select Actions > Create Exclusion Policy.

STEP 2 | Enter a POLICY NAME to identify your alert exclusion.

STEP 3 | Enter a descriptive COMMENT that identifies the reason or purpose of the alert exclusion policy.

STEP 4 | Use the alert filters to add any the match criteria for the alert exclusion policy.

You can also right-click a specific value in the alert to add it as match criteria. The app refreshes to show you which alerts in the incident would be excluded. To see all matching alerts including those not related to the incident, clear the option to Show only alerts in the named incident.

STEP 5 | Create the exclusion policy and confirm the action.

If you later need to make changes, you can view, modify, or delete the exclusion policy from the Incidents > Alert Exclusions page.

Build an Alert Exclusion Policy from Scratch

STEP 1 | Select Incidents > Exclusions.

STEP 2 | Select + Add Exclusion Policy.

STEP 3 | Enter a Policy Name to identify the exclusion policy.

STEP 4 | Enter any comments to explain the purpose or intent behind the policy.

STEP 5 | Define the exclusion criteria.

Use either the filters at the top to build your exclusion criteria. Or, to use existing alert values to populate your exclusion criteria, right click the value, and select Add rows with <value> to policy.

As you define the criteria, the app filters the results to display matches.

STEP 6 | Review the results.
The alerts in the table will be excluded from appearing in the app after the policy is created and optionally, any existing alert matches will be grayed out.

此操作是不可逆的：如果禁用或删除该策略，所有已排除的警报将保持排除状态。

**STEP 7 | Create** and then select **Yes** to confirm the alert exception policy.

**Causality View**

The **Causality View** provides a powerful way to analyze and respond to alerts. The scope of the Causality View is the **Causality Instance (CI)** to which this alert pertains. The Causality View presents the alert (generated by Cortex XDR or sent to Cortex XDR from a supported alert source such as the Cortex XDR agent) and includes the entire process execution chain that led up to the alert. On each node in the CI chain, Cortex XDR provides information to help you understand what happened around the alert.

The Causality View comprises five sections:

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Context</td>
<td>Summarizes information about the alert you are analyzing, including the host name, the process name on which the alert was raised, and the host IP and MAC address. For alerts raised on endpoint data or activity, this section also displays the endpoint connectivity status and operating system.</td>
</tr>
<tr>
<td>Host Isolation</td>
<td>You can choose to isolate the host, on which the alert was triggered, from the network or initiate a live terminal session to the host to continue investigation and remediation.</td>
</tr>
<tr>
<td>CI Chain</td>
<td>Includes the graphical representation of the Causality Instance (CI) along with other information and capabilities to enable you to conduct your analysis. The Causality View presents a single CI chain. The CI chain is built from processes nodes, events, and alerts.</td>
</tr>
</tbody>
</table>
The chain presents the process execution and might also include events that these processes caused and alerts that were triggered on the events or processes. The Causality Group Owner (CGO) is displayed on the left side of the chain. The CGO is the process that is responsible for all the other processes, events and alerts in the chain. You need the entire CI to fully understand why the alert occurred.

The Causality View provides an interactive way to view the CI chain for an alert. You can move it, extend it, and modify it. To adjust the appearance of the CI chain, you can enlarge/shrink the chain for easy viewing using the size controls on the right. You can also move the chain around by selecting and dragging it. To return the chain to its original position and size, click in the lower-right of the CI graph.

The process node displays icons to indicate when an RPC protocol or code injection event were executed on another process from either a local or remote host.

- Injected Node
- Remote IP

From any process node, you can also right-click to display additional actions that you can perform during your investigation:

- **Show parents and children**—If the parent is not presented by default, you can display it. If the process has children, XDR app displays the number of children beneath the process name and allows you to display them for additional information.
- **Hide branch**—Hide a branch from the Causality View.
- **Blacklist, whitelist, terminate, or quarantine a process**—If after investigating the activity in the CI chain, you want to take action on the process, you can select the desired action. Blacklist and whitelist process across your organization.

In the causality view of a **Detection (Post Detected)** type alert, you can also **Terminate process by hash**.

- Depending on the type of node; file, process, or IP address open the artifact view:
  - **Open Hash View** to display detailed information about the files and processes relating to the hash.
  - **Open IP View** to display detailed information about the IP address.

The color of a process node also correlates to the WildFire verdict.

- Blue—Benign.
### Section Description

- Yellow—Grayware.
- Red—Malware.
- Light gray—Unknown verdict.
- Dark gray—The verdict is inconclusive.

To view and download the WildFire report, in the Entity Data section, click 🗂️.

#### Entity Data

Provides additional information about the entity that you selected. The data varies by the type of entity but typically identifies information about the entity related to the cause of the alert and the circumstances under which the alert occurred.

For example, device type, device information, remote IP address.

#### Events Table

Displays all related events for the process node which matches the alert criteria that were not triggered in the alert table but are informational.

For the Behavioral Threat Protection table, right-click to whitelist, blacklist, terminate, and quarantine a process.

💡 To view statistics for files on VirusTotal, you can pivot from the Initiator MD5 or SHA256 value of the file on the Files tab.

---

### Network Causality View

The *Network Causality View* provides a powerful way to analyze and respond to the stitched firewall and endpoint alerts. The scope of the Causality View is the *Causality Instance (CI)* to which this alert pertains. The Causality View presents the network processes that triggered the alert, generated by Cortex XDR, Palo Alto Networks Next-Generation Firewalls, and supported alert source such as the Cortex XDR agent.

The network causality view includes the entire process execution chain that led up to the alert. On each node in the CI chain, Cortex XDR provides information to help you understand what happened around the alert.
The CI chain visualizes the firewall logs, endpoint files, and network connections that triggered alerts connected to a security event.

*The network causality view displays only the information it collects from the detectors. It is possible that the CI may not show some of the firewall or agent processes.*

The Network Causality View comprises five sections:

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Context</strong></td>
<td>Summarizes information about the alert you are analyzing, including the host name, the process name on which the alert was raised, and the host IP address. For alerts raised on endpoint data or activity, this section also displays the endpoint connectivity status and operating system.</td>
</tr>
<tr>
<td><strong>Host Isolation</strong></td>
<td>You can choose to isolate the host, on which the alert was triggered, from the network or initiate a live terminal session to the host to continue investigation and remediation.</td>
</tr>
</tbody>
</table>
| **CI Chain** | Includes the graphical representation of the Causality Instance (CI) along with other information and capabilities to enable you to conduct your analysis. The Causality View presents a CI chain for each of the processes and the network connection. The CI chain is built from processes nodes, events, and alerts. The chain presents the process execution and might also include events that these processes caused and alerts that were **...**
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>triggered on the events or processes. The Causality Group Owner (CGO) is displayed on the left side of the chain. The CGO is the process that is responsible for all the other processes, events and alerts in the chain. You need the entire CI to fully understand why the alert occurred.</td>
</tr>
<tr>
<td></td>
<td>The Causality View provides an interactive way to view the CI chain for an alert. You can move it, extend it, and modify it. To adjust the appearance of the CI chain, you can enlarge/shrink the chain for easy viewing using the size controls on the right. You can also move the chain around by selecting and dragging it. To return the chain to its original position and size, click in the lower-right of the CI graph.</td>
</tr>
<tr>
<td></td>
<td>From any process node, you can also right-click to display additional actions that you can perform during your investigation: The color of a process node also correlates to the WildFire verdict.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Show parents and children</strong>—If the parent is not presented by default, you can display it. If the process has children, XDR app displays the number of children beneath the process name and allows you to display them for additional information.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Hide branch</strong>—Hide a branch from the Causality View.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Blacklist, whitelist, terminate, or quarantine a process</strong>—If after investigating the activity in the CI chain, you want to take action on the process, you can select the desired action. Blacklist and whitelist process across your organization.</td>
</tr>
<tr>
<td></td>
<td>In the causality view of a Detection (Post Detected) type alert, you can also <strong>Terminate process by hash</strong>.</td>
</tr>
<tr>
<td></td>
<td>The color of a process node also correlates to the WildFire verdict.</td>
</tr>
<tr>
<td></td>
<td>• Blue—Benign.</td>
</tr>
<tr>
<td></td>
<td>• Yellow—Grayware.</td>
</tr>
<tr>
<td></td>
<td>• Red—Malware.</td>
</tr>
<tr>
<td></td>
<td>• Light gray—Unknown verdict.</td>
</tr>
<tr>
<td></td>
<td>• Dark gray—The verdict is inconclusive.</td>
</tr>
<tr>
<td></td>
<td>To view and download the WildFire report, in the Entity Data section, click .</td>
</tr>
<tr>
<td>Entity Data</td>
<td>Provides additional information about the entity that you selected. The data varies by the type of entity but typically identifies information about the entity related to the cause of the alert and the circumstances under which the alert occurred.</td>
</tr>
</tbody>
</table>
### Timeline View

The Timeline provides a forensic timeline of the sequence of events, alerts, and informational BIOCs involved in an attack. While the Causality View of an alert surfaces related events and processes that Cortex XDR identifies as important or interesting, the Timeline displays all related events, alerts, and informational BIOCs over time.

Cortex XDR presents the Timeline in four parts:

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CGO (and process instances that are part of the CGO)</td>
<td>Cortex XDR displays the Causality Group Owner (CGO) and the host on which the CGO ran in the top left of the timeline. The CGO is the parent process in the execution chain that Cortex XDR identified as being responsible for initiating the process tree. In the example above, wscript.exe is the CGO and the host it ran on was HOST488497. You can also click the blue corner of the CGO to view and filter related processes from the Timeline. This will add or remove the process and related events or alerts associated with the process from the Timeline.</td>
</tr>
<tr>
<td>Section</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Timespan</td>
<td>By default, Cortex XDR displays a 24-hour period from the start of the investigation and displays the start and end time of the CGO at either end of the timescale. You can move the slide bar to the left or right to focus on any time-gap within the timescale. You can also use the time filters above the table to focus on set time periods.</td>
</tr>
<tr>
<td>Activity</td>
<td>Depending on the type of activities involved in the CI chain of events, the activity section can present any of the following three lanes across the page:</td>
</tr>
<tr>
<td></td>
<td>- Alerts—The alert icon indicates when the alert occurred.</td>
</tr>
<tr>
<td></td>
<td>- BIOC s—The category of the alert is displayed on the left (for example: tampering or lateral movement). Each BIOC event also indicates a color associated with the alert severity. An informational severity can indicate something interesting has happened but there weren’t any triggered alerts. These events are likely benign but are byproducts of the actual issue.</td>
</tr>
<tr>
<td></td>
<td>- Event information—The event types include process execution, outgoing or incoming connections, failed connections, data upload, and data download. Process execution and connections are indicated by a dot. One dot indicates one connection while many dots indicates multiple connections. Uploads and Downloads are indicated by a bar graph that shows the size of the upload and download.</td>
</tr>
<tr>
<td></td>
<td>The lanes depict when activity occurred and provide additional statistics that can help you investigate. For BIOC and Alerts, the lanes also depict activity nodes—highlighted with their severity color: high (red), medium (yellow), low (blue), or informational (gray)—and provide additional information about the activity when you hover over the node.</td>
</tr>
<tr>
<td>Related events, alerts, and informational BIOC s</td>
<td>Cortex XDR displays all the alerts, BIOC s (triggered and informational), and events a in this table. Clicking on a node in the activity area of the Timeline filters the results you see here. Similar to other pages in Cortex XDR, you can create filters to search for specific events.</td>
</tr>
</tbody>
</table>

**Analytics Alert View**

The *analytics alert view* provides a detailed summary of the behavior that triggered an Analytics or Analytics BIOC alert. This view also provides a visual depiction of the behavior and additional information you can use to assess the alert. This includes the endpoint on which the activity was initiated, the user that performed the action, the technique the analytics engine observed, and activity and interactions with other hosts inside or outside of your network.
Figure 1: Analytics View of an Analytics Alert

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Context</td>
<td>For Analytics alerts, the analytics view indicates the endpoint for which the alert was raised.</td>
</tr>
<tr>
<td></td>
<td>For Analytics BIOC alerts, the Analytics view summarizes information about the alert, including the source host name, IP address, the process name on which the alert was raised, and the corresponding process ID.</td>
</tr>
<tr>
<td>2. Alert summary</td>
<td><em>(Analytics alerts only)</em> Describes the behavior that triggered the alert and activity impact.</td>
</tr>
<tr>
<td>3. Graphic summary</td>
<td>Similar to the Causality View, the analytics view provides a graphic representation of the activity that triggered the alert and an interactive way to view the chain of behavior for an Analytics alert. You can move the graphic, extend it, and modify it. To adjust the appearance, you can enlarge/shrink the chain for easy viewing using the size controls on the right. You can also move the chain around by selecting and dragging it. To return the chain to its original position and size, click in the lower-right of the graph. The activity depicted in the graphic varies depending on the type of alert:</td>
</tr>
<tr>
<td></td>
<td>• Analytics alerts—You can view a summary of the aggregated activity including the source host, the anomalous activity, connection count, and the destination host. You can also select the host to view any relevant profile information.</td>
</tr>
<tr>
<td></td>
<td>• Analytics BIOC alerts—You can view the specific event behavior including the causality group owner that initiated the activity and related process nodes. To view the summary of the specific event, you can select the above the process node.</td>
</tr>
<tr>
<td>4. Alert description</td>
<td>The alert description provides details and statistics related to the activity. Beneath the description, you can also view the alert name, severity assigned</td>
</tr>
<tr>
<td>Section</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>to the alert, time of the activity, alert tactic (category) and type, and links to the MITRE summary of the attack tactic.</td>
</tr>
<tr>
<td>5. Events table</td>
<td>Displays events related to the alert.</td>
</tr>
<tr>
<td>6. Response actions</td>
<td>Actions you can take in response to an Analytics alert. These actions can include isolating a host from the network, initiating a live terminal session, running a Pathfinder scan, and adding an IP address or domain name to an external dynamic list (EDL) that is enforceable in your Palo Alto Networks firewall security policy.</td>
</tr>
</tbody>
</table>
Investigate Endpoints

*Endpoint investigation requires either a Cortex XDR Prevent or a Cortex XDR Pro per Endpoint license.*

- Action Center
- View Details About an Endpoint
- Retrieve Files from an Endpoint
- Retrieve Support Logs from an Endpoint
- Scan an Endpoint for Malware
- Run a Pathfinder Scan

Action Center

The **Action Center** lists and tracks the progress of all investigation, response, and maintenance actions you performed on your Cortex XDR endpoints. For actions that may take a while to complete, the **Action Center** tracks the action progress and displays the action status and current progress description for each stage. For example, after initiating an agents upgrade action, Cortex XDR monitors all stages from Pending request, to In Progress, and until the action status is Completed and Cortex XDR reports back the number of agents for which the action was complete successfully and the number of agents on which the action failed.

By default, all actions are displayed in the **Action Center**. To narrow down the results, click **Filter** on the top right.

The following table describes both the default and additional optional fields that you can view from the **Action Center** and lists the fields in alphabetical order.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action Type</td>
<td>Type of action initiated on the endpoint (for example Agent Upgrade).</td>
</tr>
<tr>
<td>Created By</td>
<td>The name of the user who initiated the action.</td>
</tr>
<tr>
<td>Creation Timestamp</td>
<td>Date and time the action was created.</td>
</tr>
<tr>
<td>Description</td>
<td>Includes the action scope of affected endpoints and additional data relevant for each of the specific actions, such as agent version, file path, and file hash.</td>
</tr>
</tbody>
</table>
### Field | Description
--- | ---
Expiration Date | Time the action will expire. To set an expiration the action must apply to one or more endpoints. By default, Cortex XDR assigns a 30-day expiration limit to the following actions:
- Agent Uninstall
- Agent Upgrade
- Files Retrieval
- Isolate
- Cancel Endpoint Isolation
Additional actions such as malware scans, quarantine, and endpoint data retrieval are assigned a 4-day expiration limit.
After the expiration limit, the status for any remaining Pending actions on endpoints change to Expired and these endpoints will not perform the action.

### Status | The status the action is currently at:
--- | ---
Pending | No endpoint has started to perform the action yet.
In Progress | At least one endpoint has started to perform the action.
Canceled | The action was canceled before any endpoint has started performing it.
Expired | The action expired before any endpoint has started performing it.
Completed with Partial Success | The action was completed on all endpoints. However, some endpoints did not complete it successfully. Depending on the action type, it may have failed, been canceled, expired, or failed to retrieve all data.
Failed | The action failed on all endpoints.
Completed Successfully | The action was completed successfully on all endpoints.

For more details on an action, such as endpoint name, IP address, and endpoint status, right-click and select Additional data. You can also export the action additional data to a TSV file.

### Manage Endpoint Actions

There are two ways you can initiate an endpoint action. You can Initiate an Endpoint Action from the Action Center or you can initiate an action when you View Details About an Endpoint. Then, to monitor the progress and status of an endpoint action, you can Monitor Endpoint Actions from the Action Center.

**Initiate an Endpoint Action**

You can create new administrative actions using the Action Center wizard in three easy steps:

1. Select the action type and configure its parameters.
2. Define the target agents for this action.
3. Review and confirm the action summary.
Define an Action

STEP 1 | Log in to Cortex XDR.
       Go to Response > Action Center > +New Action.

STEP 2 | Select the action you want to initiate and follow the required steps and parameters you need to define for each action.
       Cortex XDR displays only the endpoints eligible for the action you want to perform.

STEP 3 | Review the action summary.
       Cortex XDR will inform you if any of the agents in your action scope will be skipped. Click Done.

STEP 4 | Track your action.
       Track the new action in the Action Center. The action status is updated according to the action progress, as listed in the table above.

Monitor Endpoint Actions

STEP 1 | Log in to Cortex XDR.
       Go to Response > Action Center.

STEP 2 | Select the relevant view.
       Use the left-side menu on the Action Center page to monitor the different actions according to their type:
• **All**—Lists all the administrative actions that were created in your network, including time of creation, action type and description, action status, the name of the user who initiated the action, and the action expiration date, if it exists.

• **Quarantine**—Lists only actions initiated to quarantine files on endpoints, including the file hash, file name, file path and scope of target agents included in this action.

• **Blacklist**—Lists only actions initiated to blacklist files, including file hash, status and any existing comments.

• **Whitelist**—Lists only actions initiated to whitelist files, including file hash, status and any existing comments.

**STEP 3 | Filter the results.**

To further narrow the results, use the *Filters* menu on the top of the page.

**STEP 4 | Take further actions.**

After inspecting an action log, you may want to take further action. Right-click the action and select one of the following (where applicable):

• **View additional data**—Display more relevant details for the action, such as file paths for quarantined files or operating systems for agent upgrades.

• **Cancel for Pending endpoints**—Cancel the original action for agents that are still in Pending status.

• **Download output**—Download a zip file with the files received from the endpoint for actions such as file and data retrieval.

• **Rerun**—Launch the Create new action wizard populated with the same details as the original action.

• **Run on additional agents**—Launch the action wizard populated with the details as the original action except for the agents which you have to fill in.

• **Restore**—Restore quarantined files.

**View Details About an Endpoint**

The **Endpoints > Endpoint Management > Endpoint Administration** page provides a central location from which you can view and manage the endpoints on which the Cortex XDR agent is installed. The right-click pivot menu that is available for each endpoint displays the actions you can perform.

The following table describes the list of actions you can perform on your endpoints.

<table>
<thead>
<tr>
<th>Field</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Endpoint Control</td>
<td>• Perform Heartbeat</td>
</tr>
</tbody>
</table>
Field | Action
-----|-----
| • Change Endpoint Alias  
• Upgrade Agent Version  

⚠️ *You cannot upgrade VDI endpoints.*

| Security Operations | • Retrieve Endpoint Files  
• Initiate Malware Scan  
• Abort Malware Scan  
• Initiate Live Terminal  
• Isolate Endpoint  

| Endpoint Data | • View Incidents  
• View Endpoint Policy  
• View Actions  
• View Endpoint Logs  

The following table describes both the default and additional optional fields that you can view in the Endpoints table and lists. The table lists the fields in alphabetical order.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Check box to select one or more endpoints on which to perform actions.</td>
</tr>
<tr>
<td>Active Directory</td>
<td>Lists all Active Directory Groups and Organizational Units to which the user belongs.</td>
</tr>
<tr>
<td>Assigned Policy</td>
<td>Policy assigned to the endpoint.</td>
</tr>
</tbody>
</table>
| Auto Upgrade Status | When Agent Auto Upgrades are enabled, indicates the action status is either:  
• **In progress**—Indicates that the Cortex XDR agent upgrade is in progress on the endpoint.  
• **Up to date**—Indicates that the current Cortex XDR agent version on the endpoint is up to date.  
• **Failure**—Indicates that the Cortex XDR agent upgrade failed after three retries.  
• **Not configured**—Indicates that automatic agent upgrades are not configured for this endpoint.  
• **Pending**—Indicates that the Cortex XDR agent version running on the endpoint is not up to date, and the agent is waiting for the upgrade message from Cortex XDR. |
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Field</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>• <strong>Not supported</strong> — Indicates this endpoint type does not support automatic agent upgrades. Relevant for VDI, TS, or Android endpoints.</td>
<td></td>
</tr>
<tr>
<td>Content Auto Update</td>
<td>Indicates whether automatic content updates are <strong>Enabled</strong> or <strong>Disabled</strong> for the endpoint. See Agent Settings profile.</td>
</tr>
<tr>
<td>Content Rollout Delay (days)</td>
<td>If you configured delayed content rollout, the number of days for delay is displayed here. See Agent Settings profile.</td>
</tr>
<tr>
<td>Content Version</td>
<td>Content update version used with the Cortex XDR agent.</td>
</tr>
<tr>
<td>Disabled Capabilities</td>
<td>A list of the capabilities that were disabled on the endpoint. Options are:</td>
</tr>
<tr>
<td>• <strong>Live Terminal</strong></td>
<td></td>
</tr>
<tr>
<td>• <strong>Script Execution</strong></td>
<td></td>
</tr>
<tr>
<td>• <strong>File Retrieval</strong></td>
<td></td>
</tr>
<tr>
<td>You can disable these capabilities during the Cortex XDR agent installation on the endpoint. Disabling any of these actions is irreversible, so if you later want to enable the action on the endpoint, you must uninstall the Cortex XDR agent and install a new package on the endpoint.</td>
<td></td>
</tr>
<tr>
<td>Domain</td>
<td>Domain or workgroup to which the endpoint belongs, if applicable.</td>
</tr>
<tr>
<td>Endpoint Alias</td>
<td>If you assigned an alias to represent the endpoint in Cortex XDR, the alias is displayed here. To set an endpoint alias, right-click the endpoint name, and select <strong>Change endpoint alias</strong>. The alias can contain any of the following characters: a-zA-Z, 0-9, !@#$%^&amp;()-'{}~_.</td>
</tr>
<tr>
<td>Endpoint ID</td>
<td>Unique ID assigned by Cortex XDR that identifies the endpoint.</td>
</tr>
<tr>
<td>Endpoint Isolated</td>
<td>Isolation status, either:</td>
</tr>
<tr>
<td>• <strong>Isolated</strong> — The endpoint has been isolated from the network with communication permitted to only Cortex XDR and to any whitelisted IP addresses and processes.</td>
<td></td>
</tr>
<tr>
<td>• <strong>Not Isolated</strong> — Normal network communication is permitted on the endpoint.</td>
<td></td>
</tr>
<tr>
<td>• <strong>Pending Isolation</strong> - The isolation action has reached the server and is pending contact with the endpoint.</td>
<td></td>
</tr>
<tr>
<td>• <strong>Pending Isolation Cancelation</strong> - The cancel isolation action has reached the server and is pending contact with the endpoint.</td>
<td></td>
</tr>
<tr>
<td>Endpoint Name</td>
<td>Hostname of the endpoint.</td>
</tr>
<tr>
<td>Endpoint Status</td>
<td>Registration status of the Cortex XDR agent on the endpoint:</td>
</tr>
<tr>
<td>• <strong>Connected</strong> — The Cortex XDR agent has checked in within 10 minutes.</td>
<td></td>
</tr>
<tr>
<td>• <strong>Disconnected</strong> — The Cortex XDR agent has checked in within the defined inactivity window: between 10 minutes and 30 days for standard</td>
<td></td>
</tr>
</tbody>
</table>
### Field | Description
--- | ---
endpoints, and between 10 minutes and 90 minutes for VDI and temporary sessions.
  - **Connection Lost**—The Cortex XDR agent has not checked in within 30 to 180 days for standard endpoints, and between 90 minutes and 6 hours for VDI and temporary sessions.
  - **Uninstalled**—The Cortex XDR agent has been uninstalled from the endpoint.

#### Endpoint Type
Type of endpoint: Mobile, Server, or Workstation.

#### Endpoint Version
Versions of the Cortex XDR agent that runs on the endpoint.

#### First Seen
Date and time the Cortex XDR agent first checked in (registered) with Cortex XDR.

#### Golden Image ID
For endpoints with a System Type of Golden Image, the image ID is a unique identifier for the golden image.

#### Group Names
Endpoint Groups to which the endpoint is a member, if applicable. See Define Endpoint Groups.

#### Incompatibility Mode
Cortex XDR agent incompatibility status, either:
  - **Agent Incompatible**—The Cortex XDR agent is incompatible with the environment and cannot recover.
  - **OS Incompatible**—The Cortex XDR agent is incompatible with the operating system.

When Cortex XDR agents are compatible with the operating system and environment, this field is blank.

#### Isolation Date
Date and time of when the endpoint was Isolated. Displayed only for endpoints in Isolated or Pending Isolation Cancelation status.

#### Install Date
Date and time at which the Cortex XDR agent was first installed on the endpoint.

#### Installation Package
Installation package name used to install the Cortex XDR agent.

#### Installation Type
Type of installation:
  - Standard
  - VDI
  - Golden Image
  - Temporary Session

#### IP
Last known IPv4 or IPv6 address of the endpoint.

#### Is EDR Enabled
Whether EDR data is enabled on the endpoint.
### Field | Description
--- | ---
Last Proxy | The IP address and port number of proxy that was last used for communication between this agent and Cortex XDR.
Last Scan | Date and time of the last malware scan on endpoint.
Last Seen | Date and time of the last change in an agent’s status. This can occur when Cortex XDR receives a periodic status report from the agent (once an hour), a user performed a manual Check In, or a security event occurred. 

> Changes to the agent status can take up to ten minutes to display on the Cortex XDR.

Last Used Proxy | Last proxy used on the endpoint.
Last Used Proxy Port | Last proxy port used on endpoint.
MAC | The endpoint MAC address that corresponds to the IP address.
Operating System | Name of operating system.
Operational Status | Cortex XDR agent operational status, either:
- **Protected**—Indicates that the Cortex XDR agent is running as configured and did not report any exceptions to Cortex XDR.
- **Partially protected**—Indicates that the Cortex XDR agent reported Cortex XDR one or more exceptions.
- **Unprotected**—Indicates the Cortex XDR agent was shut down.
OS Description | Operating system version name.
OS Type | Name of the operating system.
OS Version | Operating system version number.
Platform | Platform architecture.
Proxy | IP address and port number of the configured proxy server.
Scan Status | Malware scan status, either:
- **None** - No scan initiated
- **Pending** - Scan in process.
- **Complete Successfully** - Scan completed.
- **Pending Cancelation** - Scan was aborted, waiting for cancellation action to reach endpoint.
Users | User that was last logged into the endpoint. On Android endpoints, the Cortex XDR app obtains the user from the email prefix specified during app activation.
Retrieve Files from an Endpoint

If during investigation you want to retrieve files from one or more endpoints, you can initiate a files retrieval request from Cortex XDR.

For each files retrieval request, Cortex XDR supports up to:

- 20 files
- 500MB in total size
- 10 different endpoints

The request instructs the agent to locate the files on the endpoint and upload them to Cortex XDR. The agent collects all requested files into one archive and includes a log in JSON format containing additional status information. When the files are successfully uploaded, you can download them from the Action Center.

To retrieve files from one or more endpoints:

STEP 1 | Log in to Cortex XDR.

Go to Response > Action Center > + New Action.

STEP 2 | Select Files Retrieval and click Next.

STEP 3 | Select the operating system and enter the paths for the files you want to retrieve, pressing ADD after each completed path. You cannot define a path using environment variables on Mac and Linux endpoints.

STEP 4 | Click Next.

STEP 5 | Select the target endpoints (up to 10) from which you want to retrieve files.

If needed, Filter the list of endpoints. For more information, refer to Filter Page Results.

STEP 6 | Click Next.

STEP 7 | Review the action summary and click Done when finished.
To track the status of a files retrieval action, return to the Action Center. Cortex XDR retains retrieved files for up to 30 days.

If at any time you need to cancel the action, you can right-click it and select **Cancel for pending endpoint**. You can cancel the retrieval action only if the endpoint is still in **Pending** status and no files have been retrieved from it yet. The cancellation does not affect endpoints that are already in the process of retrieving files.

**STEP 8** | To view additional data and download the retrieved files, right-click the action and select **Additional data**.

This view displays all endpoints from which files are being retrieved, including their IP Address, Status, and Additional Data such as error messages of names of files that were not retrieved.

**STEP 9** | When the action status is **Completed Successfully**, you can right-click the action and download the retrieved files logs.

Cortex XDR retains retrieved files for up to 30 days.

---

**Disable File Retrieval**

If you want to prevent Cortex XDR from retrieving files from an endpoint running the Cortex XDR agent, you must disable this capability during agent installation. Disabling this capability is irreversible from the Cortex XDR server. If you later want to re-enable this capability on the endpoint, you must re-install the Cortex XDR agent. See the Cortex XDR agent administrator’s guide for more information.

---

**Retrieve Support Logs from an Endpoint**

When you need to send additional forensic data to Palo Alto Networks Technical Support, you can initiate a request to retrieve all support logs and alert data dump files from an endpoint. After Cortex XDR receives the logs, you can then download and send them to Technical Support.

**STEP 1** | Log in to Cortex XDR.

Go to **Response > Action Center > + New Action**.

**STEP 2** | Select **Retrieve Support File** and click **Next**.

**STEP 3** | Select the target endpoints (up to 10) from which you want to retrieve logs.

*If needed, Filter the list of endpoints. For more information, refer to Filter Page Results.*

**STEP 4** | Click **Next**.

**STEP 5** | Review the action summary and click **Done** when finished.

In the next heart beat, the agent will retrieve the request to package and send all logs to Cortex XDR.

**STEP 6** | To track the status of a support log retrieval action, return to the Action Center.

When the status is **Completed Successfully**, you can right-click the action and download the support logs. Cortex XDR retains retrieved files for up to 30 days.

If at any time you need to cancel the action, you can right-click it and select **Cancel for pending endpoint**. You can cancel the retrieval action only if the endpoint is still in **Pending** status and no
files have been retrieved from it yet. The cancellation does not affect endpoints that are already in the process of retrieving files.

**STEP 7** | To view additional data and download the support logs, right-click the action and select Additional data.
You will see all endpoints from which files are being retrieved, including their IP Address, Status, and Additional Data.

**STEP 8** | When the action status is Completed Successfully, you can right-click the action and download the retrieved logs.
Cortex XDR retains retrieved files for up to 30 days.

**Scan an Endpoint for Malware**

In addition to blocking the execution of malware, the Cortex XDR agent can scan your Windows endpoints and attached removable drives for dormant malware that is not actively attempting to run.

During malware scan, the Cortex XDR leverages WildFire to examine executable files, Office files containing macros, and DLLs only. When a malicious file is detected, the Cortex XDR agent reports the malware to Cortex XDR so that you can manually take additional action to remove the malware before it is triggered and attempts to harm the endpoint. In addition, if you enable the Cortex XDR agent to quarantine malicious files, the agent can also automatically quarantine the files detected as malware by WildFire during the scan. Unsupported file types excluded from the scan are blocked on the endpoint by other Cortex XDR protection capabilities.

You can scan your endpoints for malware in two ways:

- **Initiate a Full Scan from Cortex XDR**—You can enable automatic periodic scanning of endpoints as part of a malware security profile (see Add a New Malware Security Profile) and you can run an on-demand scan on one or more endpoints. If you enable scanning in a malware security profile, any new agents to which the profile applies will trigger an initial scan on the endpoint. After the initial scan, the Cortex XDR agent will perform periodic scans per the settings you defined in the profile. When a system scan is triggered on an endpoint, the Cortex XDR agent collects all executable files, all Office files containing macros, and all DLLs and sends them in batches to Cortex XDR. Cortex XDR then submits the files to WildFire to determine whether any of the files are malware. System scanning is persistent—it resumes after reboots, process interruptions, and operating system crashes. If the endpoint was offline when the scan was scheduled, the scan will begin when the endpoint comes online. If the Cortex XDR agent loses connectivity, the Cortex XDR agent pauses the scan until it receives a verdict from Cortex XDR. However, if the local analysis service (tlaservice) stops or the scan state is corrupted, the Cortex XDR agent cancels the scan. After the scan is complete, you can view the scan results in the Action Center. The results can help you quickly identify any files or endpoints that require remediation.

- **Initiate a Scan from the Endpoint**—When a user-initiated scan is triggered on an endpoint, the Cortex XDR agent collects hashes and sends them to Cortex XDR. Cortex XDR then submits the hashes to WildFire to determine whether any of the files are malware. Cortex XDR then logs a security event for each file that WildFire returns with a malware verdict. If the hash is unknown to WildFire, it is considered non-malicious for the time being and the file is submitted to WildFire for in-depth analysis. When a malicious file is detected during the custom scan, the event is reported to Cortex XDR directly and is visible in the Alerts table as Detected (Scanned).
Initiate a Full Scan from Cortex XDR

You can initiate full scans of one or more endpoints from either Endpoint Administration or the Action Center. After initiating a scan, you can monitor the progress from Response > Action Center. From both locations, you can also abort an in-progress scan. The time a scan takes to complete depends on the number of endpoints, connectivity to those endpoints, and the number of files for which Cortex XDR needs to obtain verdicts.

To initiate a scan from Cortex XDR:

**STEP 1** | Log in to Cortex XDR.
Select **Response > Action Center > +New Action.**

**STEP 2** | Select **Malware Scan.**

**STEP 3** | Click **Next.**

**STEP 4** | Select the target endpoints (up to 100) on which you want to scan for malware.

Scanning is available on Windows and Mac endpoints only. Cortex XDR automatically filters out any endpoints for which scanning is not supported. Scanning is also not available for inactive endpoints.

*If needed, Filter the list of endpoints by attribute or group name.*

**STEP 5** | Click **Next.**

**STEP 6** | Review the action summary and click **Done** when finished.

Cortex XDR initiates the action at the next heart beat and sends the request to the agent to initiate a malware scan.

**STEP 7** | To track the status of a scan, return to the Action Center.

When the status is **Completed Successfully**, you can view the scan results.

**STEP 8** | View the scan results.

After a Cortex XDR agent completes a scan, it reports the results to Cortex XDR.

To view the scan results for a specific endpoint:

1. On **Action Center**, when the scan status is complete, right-click the scan action and select **Additional data**.

   Cortex XDR displays additional details about the endpoint.

2. Right-click the endpoint for which you want to view the scan results and select **View related security events**.

   Cortex XDR displays a filtered list of malware alerts for files that were detected on the endpoint during the scan.

Initiate a Scan from the Endpoint

You can initiate a scan for Cortex XDR agents 7.1 or later releases running on Windows endpoints only. For more information, see the Cortex XDR agent administrator’s guide for Windows.
Run a Pathfinder Scan

When Cortex XDR observes problematic traffic coming from an endpoint that does not use or support the Cortex XDR agent, it uses Pathfinder to investigate the endpoint. At any time, you can also initiate a Pathfinder scan for a particular device. If N2PA (network-to-process association) monitoring is enabled, Pathfinder also automatically performs periodic scanning for devices that have displayed suspicious behavior.

From Cortex XDR, you can view the status for all of these Pathfinder scan types. You can view both in-progress and queued Pathfinder scans, a history of the scans Pathfinder has performed, and a list of devices that are undergoing N2PA monitoring. You can export the Scan History and Hosts Under N2PA Monitoring lists to a flat-text file for the purposes of viewing them in a spreadsheet application.

To enable N2PA monitoring, click the gear on the top menu bar, select Configuration > Pathfinder and enable Network to Process Association (N2PA). (For details, see Pathfinder Configuration).

STEP 1 | Initiate a Pathfinder scan for a particular endpoint.
1. Open the Analytics View of an Analytics or BIOC Analytics alert.
2. Select Actions > Pathfinder.
3. Use your preconfigured credentials or configure an alternate account under which to run the Pathfinder scan.
4. Scan the host.
STEP 2 | View the status of the scan.

Select Analytics Management > Pathfinder.

Cortex XDR displays any on-going and queued Pathfinder Scans, the Pathfinder scan history, and suspicious endpoints that are undergoing periodic scanning (N2PA monitoring).
Investigate Files

- Manage File Execution
- Manage Quarantined Files
- Hash View

Manage File Execution

You can manage file execution on your endpoints using lists of blacklisted and whitelisted file hashes. If you trust a certain file and know it to be benign, you can add the file hash to the whitelist and allow it to be executed on all your endpoints regardless of the WildFire or local analysis verdict. Similarly, if you want to always block a file from running on any of your endpoints, you can add the associated hash to the blacklist. Whitelisting and blacklisting of files takes precedence of any other policy rules that may have otherwise been applied to these files. In the **Action Center** in Cortex XDR, you can monitor whitelisting and blacklisting actions performed in your networks and add/remove file from these lists.

**STEP 1** | Log in to Cortex XDR.
Go to **Response > Action Center > + New Action**.

**STEP 2** | Select either **Add to Blacklist** or **Add to Whitelist**.

**STEP 3** | Enter the SHA256 hash of the file and click ![right-click symbol].
You can add up to 100 file hashes at once. You can add a comment that will be added to all the hashes you added in this action.

**STEP 4** | Click **Next**.

**STEP 5** | Review the summary and click **Done**.
In the next heart beat, the agent will retrieve the updated lists from Cortex XDR.

**STEP 6** | You are automatically redirected to the **Blacklist** or **Whitelist** that corresponds to the action in the **Action Center**.

**STEP 7** | To manage the file hashes on the **Blacklist** or the **Whitelist**, right-click the file and select one of the following:

- **Disable**—The file hash remains on the list but will not be applied on your Cortex XDR agents.
- **Move to Blacklist** or **Move to Whitelist**—Removes this file hash from the current list and adds it to the opposite one.
- **Delete**—Delete the file hash from the list altogether, meaning this file hash will no longer be applied to your endpoints.
- **Open in VirusTotal**—Directs you to the VirusTotal analysis of this hash.
Manage Quarantined Files

When the Cortex XDR agent detects malware on a Windows endpoint, you can take additional precautions to quarantine the file. When the Cortex XDR agent quarantines malware, it moves the file from the location on a local or removable drive to a local quarantine folder (%PROGRAMDATA%\Cyvera\Quarantine) where it isolates the file. This prevents the file from attempting to run again or causing any harm to your endpoints.

To evaluate whether an executable file is considered malicious, the Cortex XDR agent calculates a verdict using information from the following sources in order of priority:

- Hash exception policy
- WildFire threat intelligence
- Local analysis

Quarantining a file in Cortex XDR can be done in one of two ways:

- You can enable the Cortex XDR agent to automatically quarantine malicious executables by configuring quarantine settings in the Malware security profile.
- You can quarantine a specific file from the causality card.

- View the quarantined files in your network.
Select Response > Action Center > Quarantine. For each quarantined file, you can view the file Hash, File Name, and File Path.

- Review details about quarantined files.
  On the Quarantine page, right click the row and select Additional Data. You will see the Host Name, Domain, File Path, Quarantine Source, and Quarantine Date per file.

- Open the quarantined file in VirusTotal.
  If you’d like to review the quarantined file inspection results on VirusTotal, right click the file and select Open in VirusTotal. You will be redirected in a new browser tab to the VirusTotal site and view all analysis details on the selected quarantined file.

Import File Hash Exceptions

The Action Center page displays information on quarantined, whitelisted, and blacklisted files. To import hashes from the Endpoint Security Manager or from external feeds, you can initiate an action.

**STEP 1 |** From Cortex XDR, select Response > Action Center > + New Action

**STEP 2 |** Select Import Hash Exceptions.

**STEP 3 |** Drag your Verdict Override_Exports.csv file to the drop area.
If necessary, resolve any conflicts encountered during the upload and retry.

**STEP 4 |** Click **Next** twice.

**STEP 5 |** Review the action summary, and click **Done**.

Cortex XDR imports and then distributes your hashes to the whitelist and blacklist based on the assigned verdict.
Response Actions

- Initiate a Live Terminal Session
- Isolate an Endpoint
- Run Scripts on an Endpoint
- Add an IP Address or Domain to the Cortex XDR EDL

Initiate a Live Terminal Session

To investigate and respond to security events on endpoints, you can use the *Live Terminal* to initiate a remote connection to an endpoint. The Cortex XDR agent facilitates the connection using a remote procedure call. Live Terminal enables you to manage remote endpoints. Investigative and response actions that you can perform include the ability to navigate and manage files in the file system, manage active processes, and run the operating system or Python commands.

Live Terminal is supported for endpoints that meet the following requirements:

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Requirements</th>
</tr>
</thead>
</table>
| Windows          | • Traps 6.1 or a later release  
                  • Windows 7 SP1 or a later release  
                  • Windows update patch for WinCRT (KB 2999226)—To verify the Hotfixes that are installed on the endpoint, run the `systeminfo` command from a command prompt.  
                  • PowerShell 5.0 or a later release  
                  • Endpoint activity reported within the last 90 minutes (as identified by the Last Seen time stamp in the endpoint details). |
| Mac              | • Cortex XDR agent 7.0 or a later release  
                  • macOS 10.12 or a later release  
                  • Endpoint activity reported within the last 90 minutes (as identified by the Last Seen time stamp in the endpoint details). |
| Linux            | • Cortex XDR agent 7.0 or a later release  
                  • Any Linux supported release  
                  • Endpoint activity reported within the last 90 minutes (as identified by the Last Seen time stamp in the endpoint details). |

If the endpoint supports the necessary requirements, you can initiate a Live Terminal session from the Endpoints page. You can also initiate a Live Terminal as a response action from a security event. If the endpoint is inactive or does not meet the requirements, the option is disabled.

After you terminate the Live Terminal session, you also have the option to save a log of the session activity. All logged actions from the Live Terminal session are available for download as a text file report when you close the live terminal session.

**STEP 1** | Start the session.

From a security event or endpoint details, select Response > Live Terminal. It can take the Cortex XDR agent a few minutes to facilitate the connection.
STEP 2 | Use the Live Terminal to investigate and take action on the endpoint.

- Manage Processes
- Manage Files
- Run Operating System Commands
- Run Python Commands and Scripts

STEP 3 | When you are done, Disconnect the Live Terminal session.

You can optionally save a session report containing all activity you performed during the session.

The following example displays a sample session report:

```
Live Terminal Session Summary
Initiated by user username@paloaltonetworks.com on target TrapsClient1 at Jun 27th 2019 14:17:45
Jun 27th 2019 13:56:13 Live Terminal session has started [success]
Jun 27th 2019 14:00:45 Kill process calc.exe (4920) [success]
Jun 27th 2019 14:11:46 Live Terminal session end request [success]
Jun 27th 2019 14:11:47 Live Terminal session has ended [success]

No artifacts marked as interesting
```

Manage Processes

From the Live Terminal you can monitor processes running on the endpoint. The Task Manager displays the task attributes, owner, and resources used. If you discover an anomalous process while investigating the cause of a security event, you can take immediate action to terminate the process or the whole process tree, and block processes from running.

STEP 1 | From the Live Terminal session, open the Task Manager to navigate the active processes on the endpoint.

You can toggle between a sorted list of processes and the default process tree view ([`.]). You can also export the list of processes and process details to a comma-separated values file.

If the process is known malware, the row displays a red indicator and identifies the file using a malware attribute.

STEP 2 | To take action on a process, right-click the process:

- **Terminate process**—Terminate the process or entire process tree.
- **Suspend process**—To stop an attack while investigating the cause, you can suspend a process or process tree without killing it entirely.
- **Resume process**—Resume a suspended process.
- **Open in VirusTotal**—VirusTotal aggregates known malware from antivirus products and online scan engines. You can scan a file using the VirusTotal scan service to check for false positives or verify suspected malware.
- **Get WildFire verdict**—WildFire evaluates the file hash signature to compare it against known threats.
- **Get file hash**—Obtain the SHA256 hash value of the process.
- **Download Binary**—Download the file binary to your local host for further investigation and analysis. You can download files up to 200MB in size.
• **Mark as Interesting**—Add an Interesting tag to a process to easily locate the process in the session report after you end the session.
• **Remove from Interesting**—If no threats are found, you can remove the Interesting tag.
• **Copy Value**—Copy the cell value to your clipboard.

**STEP 3** | **Select Disconnect** to end the **Live Terminal** session.

Choose whether to save the remote session report including files and tasks marked as interesting. Administrator actions are not saved to the endpoint.

---

**Manage Files**

The **File Explorer** enables you to navigate the file system on the remote endpoint and take remedial action to:

• Create, manage (move or delete), and download files, folders, and drives, including connected external drives and devices such as USB drives and CD-ROM.

  *Network drives are not supported.*

• View file attributes, creation and last modified dates, and the file owner.
• Investigate files for malicious content.

To navigate and manage files on a remote endpoint:

**STEP 1** | From the Live Terminal session, open the **File Explorer** to navigate the file system on the endpoint.

**STEP 2** | Navigate the file directory on the endpoint and manage files.

To locate a specific file, you can:

• Search for any filename rows on the screen from the search bar.
• Double click a folder to explore its contents.

**STEP 3** | Perform basic management actions on a file.

• View file attributes
• Rename files and folders
• Export the table as a CSV file
• Move and delete files and folders

**STEP 4** | Investigate files for malware.

Right-click a file to take investigative action. You can take the following actions:

• **Open in VirusTotal**—VirusTotal aggregates known malware from antivirus products and online scan engines. You can scan a file using the VirusTotal scan service to check for false positives or verify suspected malware.
• **Get WildFire verdict**—WildFire evaluates the file hash signature to compare it against known threats.
• **Get file hash**—Obtain the SHA256 hash value of the file.
• **Download Binary**—Download the file binary to your local host for further investigation and analysis. You can download files up to 200MB in size.
• **Mark as Interesting**—Add an Interesting tag to any file or directory to easily locate the file. The files you tag are recorded in the session report to help you locate them after you end the session.
• **Remove from Interesting**—If no threats are found, you can remove the Interesting tag.
• **Copy Value**—Copies the cell value to your clipboard.

**STEP 5 |** Select **Disconnect** to end the live terminal session.

Choose whether to save the live terminal session report including files and tasks marked as interesting. Administrator actions are not saved to the endpoint.

**Run Operating System Commands**

The Live Terminal provides a command-line interface from which you can run operating system commands on a remote endpoint. Each command runs independently and is not persistent. To chain multiple commands together so as to perform them in one action, use `&&` to join commands. For example:

```
cd c:\windows\temp\ && <command1> && <command2>
```

*On Windows endpoints, you cannot run GUI-based cmd commands like `winver` or `appwiz.cpl`*

**STEP 1 |** From the Live Terminal session, select **Command Line**.

**STEP 2 |** Run commands to manage the endpoint.

Examples include file management or launching batch files. You can enter or paste the commands, or you can upload a script. After you are done, you can save the command session output to a file.

**STEP 3 |** When you are done, **Disconnect** the Live Terminal session.

Choose whether to save the live terminal session report including files and tasks marked as interesting. Administrator actions are not saved to the endpoint.
Run Python Commands and Scripts

The Live Terminal provides a Python command line interface that you can use to run Python commands and scripts.

The Python command interpreter uses Unix command syntax and supports Python 3 with standard Python libraries. To issue Python commands or scripts on the endpoint, follow these steps:

**STEP 1** | From the Live Terminal session, select **Python** to start the python command interpreter on the remote endpoint.

**STEP 2** | Run Python commands or scripts as desired.

You can enter or paste the commands, or you can upload a script. After you are done, you can save the command session output to a file.

**STEP 3** | When you are done, **Disconnect** the Live Terminal session.

Choose whether to save the live terminal session report including files and tasks marked as interesting. Administrator actions are not saved to the endpoint.

Disable Live Terminal Sessions

If you want to prevent Cortex XDR from initiating Live Terminal remote sessions on an endpoint running the Cortex XDR agent, you must disable this capability during agent installation. Disabling this capability is irreversible from the Cortex XDR server. If you later want to re-enable this capability on the endpoint, you must re-install the Cortex XDR agent.

Isolate an Endpoint

This capability is supported on Windows endpoints with Traps agent 6.0 and later releases.

When you isolate an endpoint, you halt all network access on the endpoint except for traffic to Cortex XDR. This can prevent a compromised endpoint from communicating with other endpoints thereby reducing an attacker’s mobility on your network. After the Cortex XDR agent receives the instruction to isolate the endpoint and carries out the action, the Cortex XDR console shows an Isolated check-in status. To ensure an endpoint remains in isolation, agent upgrades are not available for isolated endpoints.

For VDI sessions, using the network isolation response action can disrupt communication with the VDI host management system thereby halting access to the VDI session. As a result, before using the response action you must whitelist the VDI processes and corresponding IP addresses in the Agent Settings profile (Response Actions > Network Isolation Whitelist).

**STEP 1** | From Cortex XDR, initiate an action to isolate an endpoint.

Go to **Response > Action Center > + New Action** and select **Isolate**.

You can also initiate the action (for one or more endpoints) from the Isolation page of the Action Center or from **Endpoints > Endpoint Management > Endpoint Administration**.

**STEP 2** | Select **Isolate**.

**STEP 3** | Click **Next**.
STEP 4 | Select the target endpoint that you want to isolate from your network.

If needed, Filter the list of endpoints. To learn how to use the Cortex XDR filters, refer to Filter Page Results.

STEP 5 | Click Next.

STEP 6 | Review the action summary and click Done when finished.

In the next heart beat, the agent will receive the isolation request from Cortex XDR.

STEP 7 | To track the status of an isolation action, select Response > Action Center > Isolation.

If after initiating an isolation action, you want to cancel, right-click the action and select Cancel for pending endpoint. You can cancel the isolation action only if the endpoint is still in Pending status and has not been isolated yet.

STEP 8 | After you remediate the endpoint, cancel endpoint isolation to resume normal communication.

You can cancel isolation from the Actions Center (Isolation page) or from Endpoints > Endpoint Management > Endpoint Administration. From either place right-click the endpoint and select Endpoint Control > Cancel Endpoint Isolation.

---

Run Scripts on an Endpoint

For enhanced endpoint remediation and endpoint management, you can run Python 3.7 scripts on your endpoints directly from Cortex XDR. For commonly used actions, Cortex XDR provides pre-canned scripts you can use out-of-the-box. You can also write and upload your own Python scripts and code snippets into Cortex XDR for custom actions. Cortex XDR enables you to manage, run, and track the script execution on the endpoints, as well as store and display the execution results per endpoint.

The following are pre-requisites to executing scripts on your endpoints:

- Cortex XDR Pro Per Endpoint license
- Endpoints running the Cortex XDR agent 7.1 and later releases. Since the agent uses its built-in capabilities and many available Python modules to execute the scripts, no additional setup is required on the endpoint.
- Role in the hub with the following permissions to run and configure scripts:
  - Run Standard scripts
  - Run High-risk scripts
  - Script configuration (required to upload a new script, run a snippet, and edit an existing script)
  - Scripts (required to view the Scripts Library and the script execution results)

Running snippets requires both Run High-risk scripts and Script configuration permissions. Additionally, all scripts are executed as System User on the endpoint.

---

Use the following work flow to start running scripts on your endpoints:

- Manage All Scripts in the Scripts Library
- Upload Your Scripts
- Run a Script on Your Endpoints
- Track Script Execution and View Results
- Troubleshoot Script Execution
- Disable Script Execution
Manage All Scripts in the Scripts Library

All your scripts are available in the Action Center > Scripts Library, including pre-canned scripts provided by Palo Alto Networks and custom scripts that you uploaded. From the Scripts Library, you can view the script code and meta data.

The following table describes both the default and additional optional fields that you can view in the Scripts Library per script. The fields are in alphabetical order.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compatible OS</td>
<td>The operating systems the script is compatible with.</td>
</tr>
<tr>
<td>Created By</td>
<td>Name of the user who created the script. For pre-canned scripts, the user name is Palo Alto Networks.</td>
</tr>
<tr>
<td>Description</td>
<td>The script description is an optional field that can be filled-in when creating, uploading, or editing a script.</td>
</tr>
<tr>
<td>Id</td>
<td>Unique ID assigned by Cortex XDR that identifies the script.</td>
</tr>
<tr>
<td>Modification Date</td>
<td>Last date and time in which the script or its attributes were edited in Cortex XDR.</td>
</tr>
<tr>
<td>Name</td>
<td>The script name is a mandatory field that can be filled-in when creating, uploading, or editing a script.</td>
</tr>
<tr>
<td>Outcome</td>
<td>• High-risk—Scripts that may potentially harm the endpoint.</td>
</tr>
<tr>
<td></td>
<td>• Standard—Scripts that do not have a harmful impact on the endpoint.</td>
</tr>
<tr>
<td>Script FileSHA256</td>
<td>The SHA256 of the code file.</td>
</tr>
</tbody>
</table>

From the Scripts Library, you can perform the following additional actions:

- **Download script**—To see exactly what the script does, right-click and Download the Python code file locally.
- **View / Download definitions file**—To view or download the script meta-data, right-click the script and select the relevant option.
- **Run**—To run the selected script, right-click and select Run. Cortex XDR redirects you to the Action Center with the details of this script already populating the new action fields.
- **Edit**—To edit the script code or meta-data, right-click and Edit. This option is not available for pre-canned scripts provided by Palo Alto Networks.

By default, Palo Alto Networks provides you with a variety of pre-canned scripts that you can use out-of-the-box. You can view the script, download the script code and meta-data, and duplicate the script, however you cannot edit the code or definitions of pre-canned scripts.

The following table lists the pre-canned scripts provided by Palo Alto Networks, in alphabetical order. New pre-canned scripts are continuously uploaded into Cortex XDR through content updates, and are labeled New for a period of three days.
<table>
<thead>
<tr>
<th>Script name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>delete_file</td>
<td>Delete a file on the endpoint according to the full path.</td>
</tr>
<tr>
<td>file_exists</td>
<td>Search for a specific file on the endpoint according to the full path.</td>
</tr>
<tr>
<td>get_process_list</td>
<td>List CPU and memory for all processes running on the endpoint.</td>
</tr>
<tr>
<td>list_directories</td>
<td>List all the directories under a specific path on the endpoint, You can limit the number of levels you want to list.</td>
</tr>
<tr>
<td>process_kill_cpu</td>
<td>Set a minimum CPU value and kill all process on the endpoint that are using higher CPU.</td>
</tr>
<tr>
<td>process_kill_mem</td>
<td>Set a minimum RAM usage in bytes and kill all process on the endpoint that are using higher private memory.</td>
</tr>
<tr>
<td>process_kill_name</td>
<td>Kill all processes by a given name.</td>
</tr>
<tr>
<td>*registry_delete</td>
<td>(Windows) Delete a Registry key or value on the endpoint.</td>
</tr>
<tr>
<td>*registry_get</td>
<td>(Windows) Retrieve a Registry value from the endpoint.</td>
</tr>
<tr>
<td>*registry_set</td>
<td>(Windows) Set a Registry value from the endpoint.</td>
</tr>
</tbody>
</table>

*Since all scripts are running under System context, you cannot perform any Registry operations on user-specific hives (HKEY_CURRENT_USER of a specific user).*

**Upload Your Scripts**

You can write and upload additional scripts to the **Scripts Library**.

To upload a new script:

**STEP 1** | From **Action Center** > **Scripts Library** select **+New Script**.
Drag and drop your script file, or browse and select it. During the upload, Cortex XDR parses your script to ensure you are using only Python modules supported by Cortex XDR. Click Supported Modules if you want to view the supported modules list. If your script is using unsupported Python modules, or if your script is not using proper indentation, Cortex XDR will require that you fix it. You can use the editor to update your script directly in Cortex XDR.

**STEP 2 | Add meta-data to your script.**

You can fill-in the fields manually, and also upload an existing definitions file in the supported format to automatically fill-in some or all of the definition. To view the manifest format and create your own, see Creating a Script Manifest.

- **General**—The general script definitions include: name and description, risk categorization, supported operating systems, and timeout in seconds.

  ![General script definitions](Image)

  

- **Input**—Set the starting execution point of your script code. To execute the script line by line, select Just run. Alternatively, to set a specific function in the code as the entry point, select Run by entry point. Select the function from the list, and specify for each function parameter its type.
• **Output**—If your script returns an output, Cortex XDR displays that information in the script results table.
  
  • **Single parameter**—If the script returns a single parameter, select the **Output type** from the list and the output will be displayed as is. To detect the type automatically, select **Auto Detect**.
  
  • **Dictionary**—If the script returns more than a single value, select **Dictionary** from the **Output type** list. By default, Cortex XDR displays in the script results table the dictionary value as is. To improve the script results table display and be able to filter according to the returned value, you can assign a user friendly name and type to some or all of your dictionary keys, and Cortex XDR will use that in the results table instead.

To retrieve files from the endpoint, add to the dictionary the **files_to_get** key to include an array of paths from which files on the endpoint will be retrieved from the endpoint.

**STEP 3** | **When you are done, Create** the new script.

The new script is uploaded to the **Scripts Library**.
Creating a Script Manifest

The script manifest file you upload into Cortex XDR has to be a single-line textual file, in the exact format explained below. If your file is structured differently, the manifest validation will fail and you will be required to fix the file.

For the purpose of this example, we are showing each parameter in a new line. However, when you create your file, you must remove any \n or \t characters.

This is an example of the manifest file structure and content:

```
{
  "name": "script name",
  "description": "script description",
  "outcome": "High Risk|Standard",
  "platform": "Windows,macOS,Linux",
  "timeout": 600,
  "entry_point": "entry_point_name",
  "entry_point_definition": {
    "input_params": [
      {"name": "registry_hkey", "type": "string"},
      {"name": "registry_key_path", "type": "number"},
      {"name": "registry_value", "type": "number"}],
    "output_params": [{"type": "JSON", "value": [
      {"name": "output_auto_detect", "friendly_name": "name1", "type": "auto_detect"},
      {"name": "output_boolean", "friendly_name": "name2", "type": "boolean"},
      {"name": "output_number", "friendly_name": "name3", "type": "number"},
      {"name": "output_string", "friendly_name": "name4", "type": "string"},
      {"name": "output_ip", "friendly_name": "name5", "type": "ip"}]}
  }
}
```

Always use lower case for variable names.

**STEP 1** | Fill-in the script name and description.
You can use letters and digits. Avoid the use of special characters.

**STEP 2** | Categorize the script.
If a script is potentially harmful, set it as High—Risk to limit the user roles that can run it. Otherwise, set it as Standard.

**STEP 3** | Assign the platform.
Enter the name of the operating system this script supports. The options are Windows, macOS, and Linux. If you need to define more than one, use a comma as a separator.

**STEP 4** | Set the script timeout.
Enter the number of seconds after which Cortex XDR agent halts the script execution on the endpoint.

**STEP 5** | Configure the script input and output.
To Run by entry point, you must specify the entry point name, and all input and output definitions.
The available parameter types are:
- auto_detect
• boolean
• number
• string
• ip
• number_list
• string_list
• ip_list

To set the script to Just run, leave both **Entry_point** and **Entry_point_definitions** empty:

```json
{
  "name":"script name",
  "description":"script description",
  "outcome":"High Risk|Standard",
  "platform":"Windows,macOS,Linux",
  "timeout":600,
  "entry_point":"
  "entry_point_definition":{}
}
```

### Run a Script on Your Endpoints

Follow this high-level workflow to run scripts on your endpoints that perform actions, or retrieve files and data from the endpoint back to Cortex XDR.

**STEP 1 |** Initiate a new action to run a script.

From **Action Center** > **+New Action**, select **Run Script**.

**STEP 2 |** Select an existing script or add a code snippet.

1. To run an existing script, start typing the script name or description in the search field, or scroll down and select it from the list. Set the script timeout in seconds and any other script parameters, if they exist. Click **Next**

2. Alternatively, you can insert a **Code Snippet**. Unlike scripts, snippets are not saved in the Cortex XDR **Scripts Library** and cannot receive input or output definitions. Write you snippet in the editor, fill-in the timeout in seconds, and click **Next**

**STEP 3 |** Select the target endpoints.

Select the target endpoints on which to execute the script. When you’re done, click **Next**.

**STEP 4 |** Review the summary and run script.

Cortex XDR displays the summary of the script execution action. If all the details are correct, **Run** the script and proceed to **Track Script Execution and View Results**. Alternatively, to track the script execution progress on all endpoints and view the results in real-time, **Run in interactive mode**.

**Run Scripts in Interactive Mode**

When you need to run several scripts on the same target scope of endpoints, or when you want to view and inspect the results of those scripts immediately and interactively, you can run your scripts in **Interactive Mode**. In this mode, Cortex XDR enables you to track the execution progress on all endpoints in real-time, run more scripts or code snippets as you go, and view the results of these scripts all in one place.
In Interactive Mode, Cortex XDR displays general information that includes the scope of target endpoints and a list of all the scripts that are being executed in this session. For each script on the executed scripts list, you can view the following:

- The script name, date and time the script execution action was initiated, and a list of input parameters.
- A progress bar that indicates in real-time the number of endpoints for which the script execution is In Progress, Failed, or Completed. When you hover over the progress bar, you can drill-down for more information about the different sub-statues included in each group. Similarly, you can also view this information on the scripts list to the left in the form of a pie chart that is dynamically updated per script as it is being executed.

Cortex XDR does not include disconnected endpoints in the visualization of the script execution progress bar or pie chart. If a disconnected endpoint later gets connected, Cortex XDR will execute the script on that endpoint and the graphic indicators will change accordingly to reflect the additional run and its status.

- Dynamic script results that are continuously updated throughout the script execution progress. Cortex XDR lists the results, and graphically aggregates results only if they have a small variety of values. When both views are available, you can switch between them.

While in Interactive Mode, you can continuously execute more scripts and add code snippets that will be immediately executed on the target endpoints scope. Cortex XDR logs all the scripts and code snippets you execute in Interactive Mode, and you can later view them in the Action Center.

- To add another script, select the script from the Cortex XDR scripts library, or start typing a Code Snippet. Set the script timeout and input parameters as necessary, and Run when you are done. The script is added to the executed scripts list and its runtime data is immediately displayed on screen.

**Track Script Execution and View Results**

After you run a script, you see the script execution action in the Action Center.
From the Action Center, you can:

- Track Script Execution Status
- Cancel or Abort Script Execution
- View Script Execution Results
- Open Script Interactive Mode
- Rerun a Script

**Track Script Execution Status**

All script execution actions are logged in the Action Center. The Status indicates the action progress, which includes the general action status and the breakdown by endpoints included in the action. The following table lists the possible status of a script execution action for each endpoint, in alphabetical order:

<table>
<thead>
<tr>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aborted</td>
<td>The script execution action was aborted after it was already In Progress on</td>
</tr>
<tr>
<td></td>
<td>the endpoint.</td>
</tr>
<tr>
<td>Canceled</td>
<td>The script execution action was canceled from Cortex XDR before the Cortex</td>
</tr>
<tr>
<td></td>
<td>XDR agent pulled the request from the server.</td>
</tr>
<tr>
<td>Completed Successfully</td>
<td>The script was executed successfully on the endpoint with no exceptions.</td>
</tr>
<tr>
<td>Expired</td>
<td>Script execution actions expire after four days. After an action expires,</td>
</tr>
<tr>
<td></td>
<td>the status of any remaining Pending actions on endpoints change to Expired</td>
</tr>
<tr>
<td></td>
<td>and these endpoints will not receive the action.</td>
</tr>
<tr>
<td>Failed</td>
<td>A script can fail due to these reasons:</td>
</tr>
<tr>
<td></td>
<td>The Cortex XDR agent failed to execute the script.</td>
</tr>
<tr>
<td></td>
<td>Exceptions occurred during the script execution.</td>
</tr>
<tr>
<td></td>
<td>To understand why the script execution failed, see Troubleshoot Script</td>
</tr>
<tr>
<td></td>
<td>Execution.</td>
</tr>
<tr>
<td>In Progress</td>
<td>The Cortex XDR agent pulled the script execution request.</td>
</tr>
<tr>
<td>Status</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Pending</td>
<td>The Cortex XDR agent has not yet pulled the script execution request from the Cortex XDR server.</td>
</tr>
<tr>
<td>Pending Abort</td>
<td>The Cortex XDR agent is in the process of executing the script, and has not pulled the abort request from the Cortex XDR server yet.</td>
</tr>
<tr>
<td>Timeout</td>
<td>The script execution reached its configured time out and the Cortex XDR agent stopped the execution on the endpoint.</td>
</tr>
</tbody>
</table>

**Cancel or Abort Script Execution**

Depending on the current status of the script execution action on the target endpoints, you can cancel or abort the action for **Pending** and **In Progress** actions:

- **When the script execution action is Pending**, the Cortex XDR agent has not pulled the request yet from Cortex XDR. When you cancel a pending action, the Cortex XDR server pulls back the pending request and updates the action status as **Canceled**. To cancel the action for all pending endpoints, go to the **Action Center**, right-click the action and **Cancel for pending endpoints**. Alternatively, to cancel a pending action for specific endpoints only, go to **Action Center** > **Additional data** > **Detailed Results**, right-click the endpoint(s) and **Cancel pending action**.

- **When the script execution action is In Progress**, the Cortex XDR agent has begun running the script on the endpoint. When you abort an in progress action, the Cortex XDR agent halts the script execution on the endpoint and updates the action status as **Aborted**. To abort the action for all In Progress endpoints and cancel the action for any Pending endpoints, go to the **Action Center**, right-click the action and **Abort and cancel execution**. Alternatively, to abort an in progress action for specific endpoints only, go to **Action Center** > **Additional data** > **Detailed Results**, right-click the endpoint(s) and **Abort for endpoint in progress**.

**View Script Execution Results**

Cortex XDR logs all script execution actions, including the script results and specific parameters used in the run. To view the full details about the run, including returned values, right-click the script and select **Additional data**.

The script results are divided into two sections. On the upper bar, Cortex XDR displays the script meta-data that includes the script name and entry point, the script execution action status, the parameter values used in this run and the target endpoints scope. You can also download the exact code used in this run as a py file.

In the main view, Cortex XDR displays the script execution results in two formats:

- **Aggregated results**—A visualization of the script results. Cortex XDR automatically aggregates only results that have a small variety of values. To see how many of the script results were aggregated successfully, see the counts on the toggle (for example, aggregated results 4/5). You can filter the results to adjust the endpoints considered in the aggregation. You can also generate a PDF report of the aggregated results view.
Main results view—A detailed table listing all target endpoints and their details.

In addition the endpoint details (name, IP, domain, etc), the following table describes both the default and additional optional fields that you can view per endpoint. The fields are in alphabetical order.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returned values</td>
<td>If your script returned values, the values are also listed in the additional data table according to your script output definitions.</td>
</tr>
<tr>
<td>Execution timestamp</td>
<td>The date and time the Cortex XDR agent started the script execution on the endpoint. If the execution has not started yet, this field is empty.</td>
</tr>
<tr>
<td>Failed files</td>
<td>The number of files the Cortex XDR agent failed to retrieve from the endpoint.</td>
</tr>
</tbody>
</table>
### Field and Description Table

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retention date</td>
<td>The date after which the retrieved file will no longer be available for download in Cortex XDR. The value is 90 days from the execution date.</td>
</tr>
<tr>
<td>Retrieved files</td>
<td>The number of files the Cortex XDR successfully retrieved from the endpoint.</td>
</tr>
<tr>
<td>Status</td>
<td>See the list of statuses and their descriptions in Track Script Execution Status.</td>
</tr>
<tr>
<td>Standard output</td>
<td>The returned stdout</td>
</tr>
</tbody>
</table>

For each endpoint, you can right-click and download the script stdout, download retrieved files if there are any, and view returned exceptions if there are any. You can also Export to file to download the detailed results table in TSV format.

### Open Script Interactive Mode

In Interactive Mode, Cortex XDR enables you to dynamically track the script execution progress on all target endpoints and view the results as they are being received in real-time. Additionally, you can start executing more scripts on the same scope of target endpoints.

To initiate Interactive Mode for an already running script:

- From the Action Center, right-click the execution action of the relevant script and select Open in interactive mode.

### Rerun a Script

Cortex XDR allows you to select a script execution action and rerun it. When you rerun a script, Cortex XDR uses the same parameters values, target endpoints, and defined timeout that were defined for the previous run. However, if the target endpoints in the original run were defined using a filter, then that filter will be recalculated when you rerun the script. Cortex XDR will use the current version of the script. If since the previous run the script has been deleted, or the supported operating system definition has been modified, you will not be able to rerun the script.

To rerun a script:

**STEP 1** | From the Action Center, right-click the script you want to rerun and select Rerun.  
You are redirected to the final summary stage of the script execution action.

**STEP 2** | Run the script.  
To run the script with the same parameters and on the same target endpoints as the previous run, click Done. To change any of the previous run definitions, navigate through the wizard and make the necessary changes. Then, click Done. The script execution action is added to the Action Center.

### Troubleshoot Script Execution

To understand why a script returned Failed execution status, you can do the following:

1. **Check script exceptions**—If the script generated exceptions, you can view them to learn why the script execution failed. From the Action Center, right-click the Failed script and select Additional data.
the Script Results table, right-click an endpoint for which the script execution failed and select View exceptions. The Cortex XDR agent executes scripts on Windows endpoints as a SYSTEM user, and on Mac and Linux endpoints as a root user. These context differences could cause differences in behavior, for instance when using environment variables.

2. **Validate custom scripts**—When a custom script you uploaded failed and the reason the script failed is still unclear from the exceptions, or if the script did not generate any exceptions, try to identify whether it failed due to an error in Cortex XDR or due to an error in the script. To identify the error source, execute the script without the Cortex XDR agent on the same endpoint with regular Python 3.7 installation. If the script execution is unsuccessful, you should fix your script. Otherwise, if the script was executed successfully with no errors, please contact Palo Alto Networks support.

### Disable Script Execution

If you want prevent Cortex XDR from running scripts on a Cortex XDR agent, you must disable this capability during agent installation. Disabling script execution is irreversible from Cortex XDR. If you later want to re-enable this capability on the endpoint, you must re-install the Cortex XDR agent. See the *Cortex XDR Agent Administrator’s Guide* for more information.

### Add an IP Address or Domain to the Cortex XDR EDL

> The external dynamic list (EDL) requires a Cortex XDR Pro per TB license.

Cortex XDR hosts two block lists, to which you can add IP addresses and domains as you triage alerts. You can use a Cortex XDR EDL with a Palo Alto Networks firewall to provide an integrated response to malicious network activity. With a Cortex XDR EDL as the source of a firewall external dynamic list, the firewall can control user access to IP addresses and domains that the app has found to be associated with an alert.

Before you can add to the EDL, you must configure the Cortex XDR EDL.

**STEP 1** | Open the Add to EDL dialog.

There are two ways you can open the dialog: From the Response menu or in the analyze view of an Analytics alert select Actions > Add to EDL.
STEP 2 | Enter the IP address or domain you want to add.

STEP 3 | Click the arrow to the right of the entry.

STEP 4 | Repeat these steps to add additional IP addresses or domains and then click Add when finished.
Broker VM Overview

The Palo Alto Networks Broker is a secured virtual machine (VM), integrated with Cortex XDR, that bridges your network and Cortex XDR. By setting up the broker, you establish a secure connection in which you can route your endpoints, and collect and forward logs and files for analysis.

The Broker can be leveraged for running different services separately on the VM using the same Palo Alto Networks authentication. Once installed, the broker automatically receives updates and enhancements from Cortex XDR, providing you with new capabilities without having to install a new VM.
Set up Broker VM

The Palo Alto Networks Broker is a secured virtual machine (VM), integrated with Cortex XDR, that bridges your network and the Cortex XDR app. By setting up the broker, you establish a secure connection in which you can route your endpoints, and collect and forward logs and files for analysis.

The Broker can be leveraged for running different services separately on the VM using the same Palo Alto Networks authentication. Once installed, the broker automatically receives updates and enhancements from Cortex XDR, providing you with new capabilities without having to install a new VM.

- Configure the Broker VM
- Activate the Agent Proxy
- Activate the Syslog Collector
- Activate the Windows Event Collector

Configure the Broker VM

To set up the broker virtual machine (VM), you need to deploy an image created by Palo Alto Networks on your network or AWS/Azure cloud environments and activate the available applications. You can set up several broker VMs for the same tenant to support larger environments. Ensure each environment matches the necessary requirements.

Before you set up the broker VM, verify you meet the following requirements:

- Hardware: For standard installation use 4-core processor, 8GB RAM, 512GB disk. For Agent Proxy only, you can use 2-core processor.

  The Broker VM comes with 512GB, you should deploy thin provisioning, meaning that the hard disk can grow up to 512GB but will do so only if needed.

- VM compatible with:

<table>
<thead>
<tr>
<th>Image Type</th>
<th>Infrastructure</th>
</tr>
</thead>
<tbody>
<tr>
<td>OVA</td>
<td>VMware ESXi 6.0 or later</td>
</tr>
<tr>
<td>VMDK</td>
<td>AWS</td>
</tr>
</tbody>
</table>

Create an Amazon Web Services AMI image.

<table>
<thead>
<tr>
<th>VHD</th>
<th>Hyper-V 2012 or later</th>
</tr>
</thead>
<tbody>
<tr>
<td>VHD (Azure)</td>
<td>Azure</td>
</tr>
</tbody>
</table>

Create an Azure compatible VM.

- Enable communication between the Broker Service, and other Palo Alto Networks services and apps. Confirm your Cortex XDR version to ensure you enable the appropriate connections.

<table>
<thead>
<tr>
<th>FQDN, Protocol, and Port</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Required for All Cortex XDR Versions</td>
<td></td>
</tr>
</tbody>
</table>
### FQDN, Protocol, and Port

<table>
<thead>
<tr>
<th>FQDN, Protocol, and Port</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Default</strong></td>
<td></td>
</tr>
<tr>
<td>• rolex.usg.edu</td>
<td>NTP server for clock synchronization between the syslog collector and other apps and services. The broker VM provides default servers you can use, or you can define an NTP server of your choice. If you remove the default servers, and do not specify a replacement, the broker VM uses the time of the host ESX.</td>
</tr>
<tr>
<td>• ntp2.netwrx1.com</td>
<td></td>
</tr>
<tr>
<td>• 0.north-america.pool.ntp.org</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>dl.magnifier.paloaltonetworks.com</td>
<td>VM and analytics engine package upgrades.</td>
</tr>
<tr>
<td>HTTPS over TCP port 443</td>
<td></td>
</tr>
<tr>
<td>pathfinder-docker.magnifier.paloaltonetworks.com</td>
<td>VM docker images required by package upgrades.</td>
</tr>
<tr>
<td>HTTPS over TCP port 443</td>
<td></td>
</tr>
<tr>
<td>bintray-cdn.paloaltonetworks.com</td>
<td>Server used to distribute broker upgrade package.</td>
</tr>
<tr>
<td>HTTPS over TCP port 443</td>
<td></td>
</tr>
<tr>
<td><strong>Required for Cortex XDR 2.0 and later</strong></td>
<td></td>
</tr>
<tr>
<td>br-&lt;XDR tenant&gt;.xdr.&lt;region&gt;.paloaltonetworks.com</td>
<td>Broker Service server depending on the region of your deployment, either us or eu.</td>
</tr>
<tr>
<td>HTTPS over TCP port 443</td>
<td></td>
</tr>
<tr>
<td>distributions-prod-us.traps.paloaltonetworks.com</td>
<td>Information needed to communicate with your Cortex XDR tenant. Used by tenants deployed in all regions.</td>
</tr>
<tr>
<td>HTTPS over TCP port 443</td>
<td></td>
</tr>
<tr>
<td><strong>Required for TMS or Cortex XDR 1.0</strong></td>
<td></td>
</tr>
<tr>
<td>apitrusted.paloaltonetworks.com</td>
<td>PKI server used for passing an one-time password and receiving a certificate.</td>
</tr>
<tr>
<td>HTTPS over TCP port 443</td>
<td></td>
</tr>
<tr>
<td>api.paloaltonetworks.com</td>
<td>PKI server used for certificate renewal and revocation.</td>
</tr>
<tr>
<td>mTLS over TCP port 443</td>
<td></td>
</tr>
<tr>
<td>brokerservice-&lt;region&gt;.paloaltonetworks.com</td>
<td>Broker Service server depending on the region of your deployment, either us or eu.</td>
</tr>
<tr>
<td>mTLS over TCP port 443</td>
<td></td>
</tr>
</tbody>
</table>

**Enable Access to Cortex XDR** from the broker VM to allow communication between agents and the Cortex XDR app.

Configure your broker VM as follows:

**STEP 1** | In Cortex XDR, select > Settings > Broker VMs.
STEP 2 | **Download** and install one of the following broker images:

- OVA
- VHD
- VHD (Azure)—Cortex XDR supports Azure compatible VM.
- VMDK—Convert Cortex XDR VMDK image to Amazon Web Services AMI.

STEP 3 | **Generate Token** and copy to your clipboard.

The token is valid only for 24 hours. A new token is generated each time you select Generate Token.

STEP 4 | Navigate to **https://<broker_vm_ip_address>/**.

STEP 5 | Log in with the password **!nitialPassw0rd** and then define your own unique password.

The password must contain a minimum of eight characters, contain letters and numbers, and at least one capital letter and one special character.

STEP 6 | Configure your broker VM settings:
1. In the Network Interface section, review the pre-configured Name, IP address, and MAC Address, select the Address Allocation: DHCP (default) or Static, and select to either to Disable or set as Admin the network address as the broker VM web interface.

- If you choose Static, define the following and Save your configurations:
  - Static IP address
  - Netmask
  - Default Gateway
  - DNS Server

2. (Optional) Configure a Proxy Server.
   - Select the proxy Type: HTTP, SOCKS4 or SOCKS5
   - Enter the proxy Address, Port and an optional User and Password. Select the pencil icon to enter the password.
   - Save your configurations.
3. (Requires Broker VM 8.0 and later) (Optional) In the NTP section, configure your NTP servers. Enter the server addresses according to the information detailed in the grant communications table. You can enter a server address or IP address.

4. (Requires Broker VM 8.0 and later) (Optional) In the SSH Access section, Enable or Disable SSH connections to the broker VM. SSH access is authenticated using a public key, provided by the user. Using a public key grants remote access to colleagues and Cortex XDR support who the private key. You must have App Administrator role permissions to configure SSH access.

   To enable connection, generate an RSA Key Pair, enter the public key in the SSH Public Key section and Save your configuration.

5. (Requires Broker VM 8.0 and later) (Optional) Collect and Download Logs. Your XDR logs will download automatically after approximately 30 seconds.

STEP 7 | Register and enter your unique Token, created in Cortex XDR console.

Registration of the Broker VM can take up to 30 seconds.

After a successful registration, a registered notification will appear.
Configure the Broker VM settings in order to connect to the Cortex XDR server

In order to connect the Broker VM to the Cortex XDR server, please select the relevant network interfaces and define their respective IP allocation method. You may also choose to configure a proxy to the broker VM, directing traffic from all interfaces to this proxy. Once all the configuration is in place, click Register to enter the token and complete the process.

Broker VM GQLNH1907 is registered to the Cortex XDR server

You are directed to Cortex XDR > Settings > Broker VMs. The Broker VMs page displays your broker VM details and allows you to edit the defined configurations.

**STEP 8 | Activate** your Cortex XDR applications:

- Agent Proxy Collector
- Activate the Syslog Collector
- Activate the Windows Event Collector

**Create a Broker VM AMI Image**

After you download your Cortex XDR Broker VMDK image, you can convert the image to Amazon Web Services (AWS) AMI.

To convert the image:

**Set up AWS CLI**

(Optional) If you haven’t done so already, set up your AWS CLI as follows:

**STEP 1 | Install the AWS zip file by running the following command on your local machine:**

```
```

**STEP 2 | Connect to your AWS account by running:**

```
aws configure
```

**Create the AMI Image**

**STEP 1 | Navigate and log in to your AWS account.**

**STEP 2 | In the AWS Console, navigate to Services > Storage > S3 > Buckets.**

**STEP 3 | In the S3 buckets page, + Create bucket to upload your broker image to.**

**STEP 4 | Upload the Broker VM VMDK you downloaded from Cortex XDR to the AWS S3 bucket.**

Run

```
aws s3 cp -/<path/to/broker-vm-version.vmdk> s3://<your_bucket/broker-vm-version.vmdk>
```

**STEP 5 | Prepare a configuration file on your hard drive.**
For example:

```json
[  {    "Description": "<Broker VM Version>",    "Format": "vmdk",    "UserBucket": {      "S3Bucket": "<your_bucket>",      "S3Key": "<broker-vm-version.vmdk>"    }  }]
```

**STEP 6** | Create a AMI image from the VMDK file.

Run

```
aws ec2 import-image --description="<Broker VM Version>" --disk-containers="file:///path/to/configuration.json"
```

Creating an AMI image can take up to 60 minutes to complete.

To track the progress, use the task id value from the output and run:

```
aws ec2 describe-import-image-tasks --import-task-ids import-ami-<task-id>
```

Completed status output example:

```json
{
   "ImportImageTasks": [ {
      "Status": "completed",
      "SnapshotDetails": [
         { "Description": "Broker VM version",
           "DeviceName": "/dev/<name>",
           "DiskImageSize": 2976817664.0,
           "Format": "VMDK",
           "SnapshotId": "snap-1234567890",
           "Status": "completed",
           "UserBucket": { "S3Bucket": "broker-vm",
           "S3Key": "broker-vm-<version>.vmdk" } }
      ],
      "Status": "completed",
      "...
   }]
}
```

**STEP 7** | (Optional) After the AMI image has been created, you can define a new name for the image.

Navigate to Services > EC2 > IMAGES > AMIs and locate your AMI image using the task ID. Select the pencil icon to enter a new name.

Launch an Instance

**STEP 1** | Navigate to Services > EC2 > Instances.

**STEP 2** | Search for your AMI image and Launch the file.

**STEP 3** | In the Launch Instance Wizard define the instance according to your company requirements and Launch.

**STEP 4** | (Optional) In the Instances page, locate your instance and use the pencil icon to rename the instance Name.

**STEP 5** | Define HTTPS and SSH access to your instance.

Right-click your instance and navigate to Networking > Change Security Groups.
In the **Change Security Groups** pop-up, select HTTPS to be able to access the Broker VM Web UI, and SSH to allow for remote access when troubleshooting. Make sure to allow these connection to the broker from secure networks only.

> Assigning security groups can take up to 15 minutes.

**STEP 6 | Verify the broker VM has started correctly.**

Locate your instance, right-click and navigate to **Instance Settings > Get Instance Screenshot**.
You are directed to your broker VM console listing your broker details.

### Create a Broker VM Azure Image

After you download your Cortex XDR Broker **VHD (Azure)** image, you need to upload it to Azure as a storage blob.

To create the image:

**STEP 1 |** Decompress the downloaded VHD (Azure) image. Make sure you decompress the zipped hard disk file on a server that has more then 512GB of free space.

> Decompression can take up to a few hours.

**STEP 2 |** Create a new storage blob on your Azure account by uploading the VHD file.

**Uploading from Microsoft**

1. Update to Windows PowerShell 5.1.
2. Install .NET Framework 4.7.2 or later.
3. Configure NuGet.
   - `Install-PackageProvider -Name NuGet -MinimumVersion 2.8.5.201-Force`
4. Install azure cmdlets.
   - `Install-Module -Name Az -AllowClobber`
5. Connect to your Azure account.
   - `Connect-AzAccount`
6. Start the upload.

> Upload can take up to a few hours.

**Uploading from Ubuntu 18.04**

1. Install azure util.
   
   ```
curl -sL https://aka.ms/InstallAzureCLIDeb | sudo bash
   ```
2. Connect to Azure.
az login

3. Start the upload.

   Add-AzVhd -Destination "https://<bucket name>.blob.core.windows.net/<container name>/<desired vhd name> -LocalFilePath <decompressed vhd> -ResourceGroupName <resource group name>

STEP 3 | In the Azure home page, navigate to Azure services > Disks and +Add a new disk.

STEP 4 | In the Create a managed disk > Basics page define the following information:

   Project details
   • Resource group—Select your resource group.

   Disk details
   • Disk name—Enter a name for the disk object.
   • Region—Select your preferred region.
   • Source type—Select Storage Blob. Additional field are displayed, define as follows:
     • Source blob—Select Browse. You are directed to the Storage accounts page. From the navigation panel, select the bucket and then container to which you uploaded the Cortex XDR VHD image.
       In the Container page, Select your VHD image.
     • OS type—Select Linux
     • VM generation—Select Gen 1
   Review + create to check you settings.

STEP 5 | Create you broker VM disk.

   After deployment is complete Go to resource.

STEP 6 | In your created Disks page, Create VM.

STEP 7 | In the Create a virtual machine page, define the following:

   Instance details
   • (Optional)Virtual machine name—Enter the same name as the disk name you defined.
   • Size—Select the size according to your company guidelines.
     Select Next to navigate to the Networking tab.

   Network interface
   • NIC network security group—Select Advanced.
   • Configure network security group—Select HTTPS to be able to access the Broker VM Web UI, and SSH to allow for remote access when troubleshooting. Make sure to allow these connection to the broker from secure networks only.
   Review + create to check you settings.

STEP 8 | Create your VM.

   After deployment is complete Go to resource. You are directed to your VM page.

   Creating the VM can take up to 15 minutes. The broker VM Web UI is not accessible during this time.
Activate the Agent Proxy

After you have configured and registered your broker VM, activate your agent proxy collector application. You must have either Cortex XDR Prevent or Cortex XDR Pro per Endpoint licenses to activate the agent proxy.

The Agent Proxy is used for routing all the agent traffic via a centralized and controlled access point in your network. Each proxy on the broker VM can support up to 10,000 agents.

**STEP 1 |** In Cortex XDR, navigate to Cortex XDR > Cortex XDR > Settings > Broker VMs table and locate your broker VM.

**STEP 2 |** Right-click, select Applet Management > Activate Agent Proxy.

**STEP 3 |** From Cortex XDR, Create an Agent Installation Package and download it to the endpoint.

> The Broker Service is supported with Traps agent version 5.0.9 and Traps agent version 6.1.2 and later releases.

**STEP 4 |** Run the installation package on each endpoint according to the endpoint OS. During installation you must configure the IP address of the broker VM and use port 8888. See the Cortex XDR Agent Administrator’s Guide for installation instructions.

**STEP 5 |** After a successful activation, the Apps field displays the Agent Proxy - Active.

**STEP 6 |** In the Apps filed, select Agent Proxy to view the Resources.

Activate the Syslog Collector

After you have configured and registered your broker VM, activate your Syslog collector application. Activating the Syslog collector requires a Cortex XDR Pro per TB license.

The Syslog Collector allows you to collect syslog logs from within your network by listening to specific ports.

**STEP 1 |** In Cortex XDR, navigate to Cortex XDR > Settings > Broker VMs table and locate your broker VM.

**STEP 2 |** Right-click, select Applet Management > Activate Syslog Collector.

**STEP 3 |** In the Configure Syslog window, define the Port, Protocol, and Syslog Format. You can define the Syslog collector to listen to multiple ports and select the relevant Syslog format for each of the ports.
You are not permitted to configure port number between 0-1024 and 63000-65000, except for 514. In addition, 4369, 5671, 5672, 5986, 6379, 8000, 8888, 9100, 15672, 25672 are also not allowed.

**STEP 4** | **Activate** your configurations.

After a successful activation, the **Apps** field displays the **Syslog Collector - Active**.

**STEP 5** | In the **Apps** filed, select **Syslog Collector** to view the following applet metrics:

- **Connectivity Status**—Whether the applet is connected to Cortex XDR.
- **Logs Received** and **Logs Sent**—Number of logs received and sent by the applet per second over the last 24 hours. If the number of incoming logs received is larger than the number of logs sent, it could indicate a connectivity issue.
- **Resources**—Displays the amount of **CPU**, **Memory**, and **Disk** space the applet is using.

---

**Activate the Windows Event Collector**

*Use this workflow for Broker VM version 8.0 and later. For earlier Broker VM versions follow the process detailed in Set up a Windows Event Collector.*

After you have configured and registered your broker VM, activate your Windows Event Collector application.

The Windows Event Collector (WEC) runs on the broker VM collecting event logs from Domain Controllers (DCs). To enable collection of the event logs, you need to configure them as Windows Event Forwarders (WEFs), and establish trust between them and the WEC. Establishing trust between the WEFs and the WEC is achieved by mutual authentication over TLS using server and client certificates.

The WEF, a WinRM plugin, runs under the Network Service account. Therefore, you need to provide the WEFs with the relevant certificates and grant the account access permissions to the private key used for client authentication, for example, authenticate with WEC.

Ensure you meet the following prerequisites before activating the collector:
• Cortex XDR Pro per TB license
• You have knowledge of Windows Active Directory and Domain Controllers.
• Broker VM is registered in the DNS and its FQDN is resolvable from the DCs.
• DCs running on Windows Server 2012 or later.

**STEP 1** | In Cortex XDR, navigate to Cortex XDR > Settings > Broker VMs table and locate your broker VM.

**STEP 2** | Right-click, select Applet Management > Activate Windows Event Collector.

(Optional) If you already have an Windows Event Collector signed certificate, migrate your existing CA to the Cortex XDR console.

**STEP 3** | In the Activate Windows Event Collector window, enter your Broker VM FQDN as it will be defined in your Domain Name System (DNS). This enables connection between Cortex XDR and your Windows Event Collector.

**STEP 4** | Activate your configurations.

After a successful activation, the Apps field displays the Windows Event Collector – Active.

**STEP 5** | In the Windows Event Forwarder Configuration window:

• (copy) the Subscription Manage URL. This will be used when you configure the subscription manager in the GPO (Global Policy Object) on your DC.
• Define Client Certificate Export Password used to secure the downloaded Windows Event Forwarders (WEF) certificate used to establish connection between Cortex XDR and the Windows Event collector. You will need this password when the certificate is imported to the DC.

• Download the WEF certificate in a PFX format.
To view your Windows Event Forwarder Configuration details at any time, right-click and select **Applet Management > Configure Windows Event Forwarder**.

**STEP 6 | (Optional)** In the **Apps** field, select **Windows Event Collector** to view the following applet metrics:

- **Connectivity Status**—Whether the applet is connected to Cortex XDR.
- **Logs Received** and **Logs Sent**—Number of logs received and sent by the applet per second over the last 24 hours. If the number of incoming logs received is larger than the number of logs sent, it could indicate a connectivity issue.
- **Resources**—Displays the amount of **CPU**, **Memory**, and **Disk** space the applet is using.

![Windows Event Collector metrics](image)

**STEP 7 |** Install your WEF Certificate on the DC to establish connection.

1. Copy the PFX file you downloaded from the Cortex XDR console to your DC, double-click the file and import it to **Local Machine**.
2. Run **certlm.msc**.
3. Navigate to **Certificates > Personal** and verify the following:
   - In the **Personal > Certificates** folder, ensure the certificate has been imported.
   - In the **Trusted Root Certification Authorities** folder, ensure the CA was added.
4. Navigate to **Certificates > Personal > Certificates**.
5. Right-click the certificate and navigate to **All tasks > Manage Private Keys**.
6. In the **Permissions** window, select **Add** and in the **Enter the object name** section, enter **NETWORK SERVICE** followed by **OK**.

![Certificate management](image)
Verify the **Group or user names** appear.

**STEP 8** | Add the Network Service account to the DC Event Log Readers group.

1. To enable DCs to forward events, the Network Service account must be a member of the Active Directory Event Log Readers group. In PowerShell, execute the following command on the DC:

   ```
   C:\> net localgroup "Event Log Readers" "NT Authority\Network Service" /add
   ```

**STEP 9** | Create a WEF Group Policy which applies to every DC you want to configure as a WEF.

1. Open `gpmc.msc`.
2. Create a new Group Policy and name it **Windows Event Forwarding**.
3. In the **Group Policy Management** window, navigate to Domains > `<your domain name>` > Windows Event Forwarding, right-click and select **Edit**.
4. In the **Group Policy Management Editor**:
   - Set the WinRM service for automatic startup.
     - Navigate to **Computer Configuration > Policies > Windows Settings > Security Settings > System Services**, and double-click **Windows Remote Management**.
     - Mark **Define this policy setting** and select **Automatic**.
     - Enable collection of Broker VM supported Kerberos events; Kerberos pre-authentication, authentication, request, and renewal tickets.
     - Navigate to **Computer Configuration > Policies > Advanced Audit Policy Configuration > Audit Policy > Account Logon**.
     - Configure **Audit Kerberos Authentication Service** and **Audit Kerberos Service Ticket Operations** to **Success and Failure**.

5. Configure the subscription manager.

   Navigate to **Computer Configuration > Policies > Administrative Templates > Windows Components > Event Forwarding**, and double-click **Configure target Subscription Manager**.

   ![Configure target Subscription Manager](image)

   In the **Configure target Subscription Manager** window, and select **Show**
   - Mark **Enabled**.
   - Select **Show** and paste the **Subscription Manager URL** you copied from the Cortex XDR console.

6. Add Network Service to Event Log Readers group.

   Navigate to **Computer Configuration > Preferences > Control Panel Settings > Local Users and Groups**, right-click and select **New Local Group**.
In the Event Log Readers (built-in) Properties window:

- In Group name field, select Event Log Readers (built-in).
- In Members section, Add and enter in the Name field Network Service.

> You must type the name, it cannot select the name from the browse button.

- Ok.

7. Configure the Windows Firewall.

> If Windows Firewall is enabled on your DCs, you will have to define an outbound rule to enable the WEF to reach port 5986 on the WEC.


Configure the following:

- Type - Port
- TCP - Port 5986
- Allow the connection
- Mark Domain, disable Private and Public
- Name the rule Windows Event Forwarding
- Finish

**STEP 10** | Apply the WEF Group Policy.

Link the policy to the DC OU or the group of DCs you would like to configure as WEFs.

1. Navigate to Group Policy Management > <your domain name> > Domain Controllers, right-click and select Link an existing GPO....
2. Select the WEF Group Policy you created, Windows Event Forwarding.
3. In an administrative PowerShell console, execute the following command:

```
PS C:\Users\Administrator> gpupdate /force
PS C:\Users\Administrator> Restart-Service WinRM
```

**STEP 11 | Verify Windows Event Forwarding.**

1. In an administrative PowerShell console, run the following command:

```
PS C:\Users\Administrator> Get-WinEvent Microsoft-windows-WinRM/operational -MaxEvents10
```

2. Look for **WSMan operation EventDelivery completed successfully** messages. These indicate events forwarded successfully.

**Migrate Existing Windows Event Collector Certificate**

For users who are running broker VM version 8.0 and later, and have already have a signed Windows Event Collector certificate, it’s best to migrate your CA to the Cortex XDR console to better manage connection between the Windows Event Collector and Broker VM.

To migrate your exiting Windows Event Collector signed certificate to the Cortex XDR console:

**STEP 1 |** In Cortex XDR, navigate to Cortex XDR > Settings > Broker VMs table and locate your broker VM.

**STEP 2 |** Right click, select Applet Management > Windows Event Forwarder Migration.

**STEP 3 |** In the Windows Event Forwarder Migration window:

1. Securely import the signed certificate and key from your Linux server by copying and running in OpenSSL the **Run Export Command**. Make sure you enter your certificate and key file names.
2. Copy the auto-generated password. **Provide the following password** when running the OpenSSL command to authenticate import.
3. **Upload CA Certificate** by Drag and Drop or browse for your certificate.
4. **Upload** your certificate to the Cortex XDR console.
Cortex XDR displays an **Action Succeeded** notification.

![Image of Cortex XDR migration process]

After a successful migration, your certificates are managed and signed by Cortex XDR.

> It is recommended to delete the CA PFX file and private key from the secured host where the certificates were signed.

**Set Up a Windows Event Collector**

*Use this workflow for Broker VM version 7.4.5 and earlier. For later Broker VM versions follow the process detailed in Activate the Windows Event Collector.*

The Windows Event Collector (WEC) runs on the broker VM collecting event logs from Domain Controllers (DCs). To enable collection of the event logs, you need to configure them as Windows Event Forwarders (WEFs), and establish trust between them and the WEC. Establishing trust between the WEFs and the WEC is achieved by mutual authentication over TLS using server and client certificates.

The WEF, a WinRM plugin, runs under the Network Service account. Therefore, you need to provide the WEFs with the relevant certificates and grant the account access permissions to the private key used for client authentication, for example, authenticate with WEC.

Ensure you meet the following prerequisites:

- Cortex XDR Pro per TB license
- You have knowledge of Windows Active Directory and Domain Controllers.
- You have OpenSSL installed on a secure Linux or macOS host.
- Broker VM supports a working DNS name resolution and valid DNS domain zone records.
- DCs are running on Windows Server 2012 or later.

**STEP 1 | Generate CA and WEC certificates.**

1. On your secure Linux/macOS host, download the scripts and save each of the following files to the same directory:
   - `generate_certs.sh`
   - `openssl.conf`
The CA, WEC, and WEF private keys are generated on this host. Ensure you are working on a secure host and store the CA private key securely with password protection, so you are able to generate WEF certificates for any DC you would want to turn into a WEF in the future.

2. Set execution permission on `generate_certs.sh` file by running:

```
$ chmod +x generate_certs.sh
```

3. Run the script, providing the broker VM CN, as registered in the DNS, on which the WEC will be activated. You will be prompted for a password to protect the PFX file.

```
If you are running the script for the first time, use the `--create-ca` flag to also generate the CA certificate.
```

```
$ ./generate_certs.sh --create-ca --cn broker.etac-tlv.local
Creating the CA
It is recommended to protect the CA certificate/key pair from overriding/deleting it unintentionally. Set readOnly permissions? [y/n] y
Creating the cert
Packing all to a PFX
Enter Export Password:
Verifying - Enter Export Password:
Done exporting to /Users/test/Projects/WEC/out:
PFX: broker.etac-tlv.local.pfx (SHA1 Fingerprint=6A1DF3BE9C9875C1DC3167DE1805F6FBCC1D3861)
CA: ca.cert  (SHA1  Fingerprint=D9DFCC987F21839A65682DF527193F78296FBB2A)
```

After completing, the script prints the location of the output files along with their SHA1 hashes:

- PFX file containing the WEC key pair and the signing CA certificate.
- The CA certificate file in PEM format.

**STEP 2 | Activate WEC on the Cortex XDR Broker VM.**

1. In Cortex XDR app, navigate to 📑 > Settings > Broker VMs.
2. Locate the broker VM on which you want to activate WEC, right-click and select Activate Windows Event Collector.

3. In the Activate Windows Event Collector window, Browse to the WEC certificate PFX file you generated.
The PFX file contains the certificate and key pair of the WEC along with its certificate chain. Normally the same CA will sign both the WEC and the DCs’ certificates. If this is not the case - upload the CA file which will be used to validate the DCs' client certificate in the CA BUNDLE field.

4. On successful activation, copy the displayed subscription URL.

STEP 3 | Generate the WEF certificate.

1. In your secure Linux/macOS host, run the script using the copied subscription URL:

```
$ ./generate_certs.sh --cn ETAC-DC-2016.etac-tlv.local
Not creating a new CA cert/key pair. Existing ones will be used
Creating the cert
Packing all to a PFX
Enter Export Password:
Verifying - Enter Export Password:
Done exporting to /Users/test/Projects/WEC/out:
PFX: ETAC-DC-2016.etac-tlv.local.pfx (SHA1
Fingerprint=BFD922E214DB6A0F5C3A176118FA76C82895A8DF)
CA:  ca.cert  (SHA1
Fingerprint=D9DFCC987F21839A65682DF527193F78296FBB2B)
```

2. Repeat this step for each DC you want to configure as a WEF.

STEP 4 | Install WEF Certificate on the DC.

1. Copy the PFX file you created to your DC, double-click the file and import it to Local Machine.
2. Run certlm.msc.
3. Navigate to Certificates > Personal and verify the following:
   - In the Personal > Certificates folder, ensure the certificate has been imported.
   - In the Trusted Root Certification Authorities folder, ensure the CA was added.
4. Navigate to Certificates > Personal > Certificates.
5. Right-click the certificate and navigate to All tasks > Manage Private Keys.
6. In the Permissions window, select Add and in the Enter the object name section, enter NETWORK SERVICE followed by OK.

Verify the Group or user names appear.

**STEP 5** | Add the Network Service account to the DC Event Log Readers group.

1. To enable DCs to forward events, the Network Service account must be a member of the Active Directory Event Log Readers group. In PowerShell, execute the following command on the DC:

   ```
   C:\> net localgroup "Event Log Readers" "NT Authority\Network Service" /add
   ```
STEP 6 | Create a WEF Group Policy which applies to every DC you want to configure as a WEF.

1. Open `gpmc.msc`.
2. Create a new Group Policy and name it `Windows Event Forwarding`.
3. In the Group Policy Management window, navigate to Domains > <your domain name> > Windows Event Forwarding, right-click and select Edit.

4. In the Group Policy Management Editor:
   - Set the WinRM service for automatic startup.
     - Mark Define this policy setting and select Automatic.
   - Enable collection of Broker VM supported Kerberos events; Kerberos pre-authentication, authentication, request, and renewal tickets.
     - Configure Audit Kerberos Authentication Service and Audit Kerberos Service Ticket Operations to Success and Failure.

5. Configure the subscription manager.

   Navigate to Computer Configuration > Policies > Administrative Templates > Windows Components > Event Forwarding, and double-click Configure target Subscription Manager.

   In the Configure target Subscription Manager window, and select Show
   - Mark Enabled.
   - Select Show and paste the subscription URL you copied.

6. Add Network Service to Event Log Readers group.

   Navigate to Computer Configuration > Preferences > Control Panel Settings > Local Users and Groups, right-click and select New Local Group.
In the Event Log Readers (built-in) Properties window:

- In Group name field, select Event Log Readers (built-in).
- In Members section, Add and enter in the Name field Network Service.

**You must type the name, it cannot select the name from the browse button.**

- Ok.

7. Configure the Windows Firewall.

**If Windows Firewall is enabled on your DCs, you will have to define an outbound rule to enable the WEF to reach port 5986 on the WEC.**


Configure the following:

- Type - Port
- TCP - Port 5986
- Allow the connection
- Mark Domain, disable Private and Public
- Name the rule Windows Event Forwarding
- Finish

**STEP 7 | Apply the WEF Group Policy.**

Link the policy to the DC OU or the group of DCs you would like to configure as WEFs.

1. Navigate to Group Policy Management > <your domain name > Domain Controllers, right-click and select Link an existing GPO....
2. Select the WEF Group Policy you created, Windows Event Forwarding.
3. In an administrative PowerShell console, execute the following command:

```powershell
PS C:\Users\Administrator> gpupdate /force
PS C:\Users\Administrator> Restart-Service WinRM
```

**STEP 8 | Verify Windows Event Forwarding.**

1. In an administrative PowerShell console, run the following command:

```powershell
PS C:\Users\Administrator> Get-WinEvent Microsoft-windows-WinRM/operational -MaxEvents10
```

2. Look for **WSMan operation EventDelivery completed successfully** messages. These indicate events forwarded successfully.
Manage Your Broker VMs

After you configured the broker VMs, you can manage your broker VMs from the Cortex XDR console.

- View Broker VM Details
- Edit Your Broker VM Configuration
- Collect Broker VM Logs
- Reboot a Broker VM
- Upgrade a Broker VM
- Remove a Broker VM

View Broker VM Details

In Cortex XDR, navigate to Cortex XDR app > Settings > Broker VMs to view detailed information regarding your registered broker VMs.

The Broker VMs table enables you to monitor and manage your broker VM and applet connectivity status, version management, device details, and usage metrics.

The following table describes both the default fields and additional optional fields that you can add to the alerts table using the column manager and lists the fields in alphabetical order.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status Indicator (●)</td>
<td>Identifies in the following columns:</td>
</tr>
<tr>
<td></td>
<td>• DEVICE NAME—Whether the broker machine is registered and connected to Cortex XDR.</td>
</tr>
<tr>
<td></td>
<td>• VERSION—Whether the broker VM is running the latest version.</td>
</tr>
<tr>
<td></td>
<td>• APPS—Whether the available applications are connected to Cortex XDR.</td>
</tr>
<tr>
<td></td>
<td>Colors depict the following statuses:</td>
</tr>
<tr>
<td></td>
<td>• Black—Disconnected to Cortex XDR</td>
</tr>
<tr>
<td></td>
<td>• Red—Disconnected from Cortex X</td>
</tr>
<tr>
<td></td>
<td>• Orange—Past Version</td>
</tr>
<tr>
<td></td>
<td>• Green—Connected, Current Version</td>
</tr>
<tr>
<td></td>
<td>Check box to select one or more broker devices on which to perform actions.</td>
</tr>
<tr>
<td>APPS</td>
<td>List of active or inactive applets and the connectivity status for each.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CPU USAGE</td>
<td>CPU usage of the broker device in percentage synced every 5 minutes.</td>
</tr>
</tbody>
</table>
| CONFIGURATION STATUS| Broker VM configuration status. Status is defined by the following according to changes made to any of the broker VM configurations.  
  - **up to date**—Broker VM configuration changes made through the Cortex XDR console have been applied.  
  - **in progress**—Broker VM configuration changes made through the Cortex XDR console are being applied.  
  - **submitted**—Broker VM configuration changes made through the Cortex XDR console have reached the broker machine and awaiting implementation.  
  - **failed**—Broker VM configuration changes made through the Cortex XDR console have failed. Need to open a Palo Alto Networks support ticket. |
| DEVICE ID           | Device ID allocated to the broker machine by Cortex XDR after registration. |
| DEVICE NAME         | Same as the Device ID.                                                     |
| DISK USAGE          | Disk usage of the broker in portion of computer storage that is currently in use.  
  Notification about low disk space appear in the Notification Center. |
| EXTERNAL IP         | The IP interface the broker is using to communicate with the server.  
  For AWS and Azure cloud environments, the field displays the **Internal IP** value. |
| INTERNAL IP         | All IP addresses of the different interfaces on the device.               |
| MEMORY USAGE        | Memory usage of the broker device in percentage synced every 5 minutes.    |
| STATUS              | Connection status of the broker device. Status is defined by either **Connected** or **Disconnected**.  
 Disconnected broker devices do not display **CPU Usage**, **Memory Usage**, and **Disk Usage** information.  
  Notification about broker VM loosing connectivity to Cortex XDR appear in the Notification Center. |
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UPGRADE TIME</td>
<td>Timestamp of when the broker device was upgraded.</td>
</tr>
<tr>
<td>VERSION</td>
<td>Version number of the broker device. If the status indicator is not green, then the broker is not running the latest version. Notification about available new broker VM version appear in the Notification Center.</td>
</tr>
</tbody>
</table>

### Edit Your Broker VM Configuration

After configuring and registering your broker VM, navigate to Cortex XDR app > Settings > Broker VMs to edit existing configurations and define additional settings.

**STEP 1** | In the Broker VMs table, locate your broker VM, right click and select Configure.

If the broker VM is disconnected, you can only View the configurations.

**STEP 2** | In the Broker VM Configurations window, define the following settings:

- Edit the exiting Network Interfaces, Proxy Server, NTP Server, and SSH Access configurations.
- (Requires Broker VM 8.0 and later) Device Name
  Change the name of your broker VM device name by selecting the pencil icon. The new name will appear in the Broker VMs table.

- (Requires Broker VM 8.0 and later) (Optional) Internal Network
  Enter a network subnet to avoid the broker VM dockers colliding with your internal network. The Network Subnet must be formatted prefix/mask, for example 192.0.2.1/24.

- Auto Upgrade
  Enable or Disable automatic upgrade of the broker VM. By default, auto upgrade is enabled. If you disable auto-upgrade, new features and improvements will require manual upgrade.

- Monitoring
Enable or Disable of local monitoring of the broker VM usage statistics in Prometheus metrics format, allowing you to tap in and export data by navigating to http://<broker_vm_address>:9100/metrics/. By default, monitoring your broker VM is disabled.

- (For Broker VM 7.4.5 and earlier) Enable/Disable ssh Palo Alto Networks support team SSH access by using a Cortex XDR token.
  
  Enabling allows Palo Alto Networks support team to connect to the broker VM remotely, not the customer, with the generated password.

  Make sure you save the password before closing the window. The only way to re-generate a password is to disable ssh and re-enable.

STEP 3 | Save your changes.

Collect Broker VM Logs

Cortex XDR allows you to collect your broker VM logs directly from the Cortex XDR console.

STEP 1 | Navigate to Cortex XDR app > Settings > Broker VMs table.

STEP 2 | Locate your broker VM and right-click Collect Latest Logs.

Logs are generated automatically after approximately 30 seconds and are available for 24 hours after the logs have been downloaded.

Reboot a Broker VM

Cortex XDR allows you reboot your broker VM directly from the Cortex XDR console.

STEP 1 | Navigate to Cortex XDR app > Settings > Broker VMs table.

STEP 2 | Locate your broker VM and right-click Reboot.

Upgrade a Broker VM

Cortex XDR allows you to upgrade your broker VM directly from the Cortex XDR console.

STEP 1 | Navigate to Cortex XDR app > Settings > Broker VMs table.

STEP 2 | Locate your broker VM and right-click Upgrade Broker version.
Upgrading your broker VM takes approximately 5 minutes.

Remove a Broker VM

Cortex XDR allows you to remove a broker VM directly from the Cortex XDR console.

**STEP 1** | Navigate to Cortex XDR app > 🗝️ > Settings > Broker VMs table.

**STEP 2** | Locate your broker VM and right-click Remove Broker.
Broker VM Notifications

To help you monitor your broker VM version and connectivity effectively, Cortex XDR send notifications to your Cortex XDR console Notification Center.

Cortex XDR send the following notifications:

- **New Broker VM Version**—Notifies when a new broker VM version has been released.
  - If the broker VM Auto Upgrade is disabled, the notification includes a link to the latest release information. It is recommend you upgrade to the latest version.
  - If the broker VM Auto Upgrade is enabled, 12 hours after the release you are notified of the latest upgrade, or you are notified that the upgrade failed. In such a case, open a Palo Alto Networks Support Ticket.
- **Broker VM Connectivity**—Notifies when the broker VM has lost connectivity to Cortex XDR.
- **Broker VM Disk Usage**—Notifies when the broker VM is utilizing over 90% of the allocated disk space.
Analytics

> Analytics Concepts
> Analytics Management
Analytics Concepts

Network security professionals know that safeguarding a network requires a defense-in-depth strategy. This layered approach to network security means ensuring that software is always patched and current, while running hardware and software systems that are designed to keep attackers out. Many strategies exist to keep unwanted users out of a network, most of these work by stopping intrusion attempts at the network perimeter.

As good and necessary as those strategies and products are, they all can defend only against known threats. Systems that looks for malicious software, for example, traditionally do its work based on previously identified MD5 signatures. But authors of these viruses constantly make trivial modifications to these signatures of the virus to avoid virus scanners until their MD5 database is updated with the modified and newly discovered signatures.

In other words, defensive network systems are constantly trying to keep up with the best efforts of aggressive, nimble attackers. Your defensive network software must be 100% correct 100% of the time to prevent successful attacks. A determined attacker, on the other hand, must be successful only once to ruin your day.

Consequently, your network defense-in-depth strategy must include software and processes that are designed to detect and respond to an intruder who has successfully penetrated your systems. This is the position that Cortex XDR takes in your enterprise. The app efficiently and automatically identifies abnormal activity on your network while providing you with the exact information you need to rapidly evaluate potential threats and then isolate and remove those threats from your network before they can perform real damage.

- Analytics Engine
- Coverage of the MITRE Attack Tactics
- Analytics Detectors

Analytics Engine

The Cortex XDR™ app uses an analytics engine to examine logs and data from your sensors. The analytics engine retrieves logs from Cortex Data Lake to understand the normal behavior (creates a baseline) so that it can raise alerts when abnormal activity occurs. The analytics engine accesses your logs as they are streamed to Cortex Data Lake and analyzes the data as soon as it arrives. An Analytics alert is reported when the analytics engine determines an anomaly.

Coverage of the MITRE Attack Tactics

Network attacks follow predictable patterns. If you interfere with any portion of this pattern, the attack will be neutralized.
The analytics engine can alert on any of the following attack tactics as defined by the MITRE ATT&CK™ knowledge base of tactics.

<table>
<thead>
<tr>
<th>Tactic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execution</td>
<td>After attackers gain a foothold in your network, they can use various techniques to execute malicious code on a local or remote endpoint. The Cortex XDR app detects malware and grayware on your network using a combination of network activity, Pathfinder scans of your endpoints, endpoint data from your Cortex XDR agents, and evaluation of suspicious files using the WildFire® cloud service.</td>
</tr>
<tr>
<td>Persistence</td>
<td>To carry out a malicious action, an attacker can try techniques that maintain access in a network or on an endpoint. An attacker can initiate configuration changes—such as a system restart or failure—that require the endpoint to restart a remote access tool or open a backdoor that allows the attacker to regain access on the endpoint.</td>
</tr>
<tr>
<td>Discovery</td>
<td>After an attacker has access to a part of your network, discovery techniques to explore and identify subnets, and discover servers and the services that are hosted on those endpoints. The idea is to identify vulnerabilities within your network. The app detects attacks that use this tactic by looking for symptoms in your internal network traffic such as changes in connectivity patterns that including increased rates of connections, failed connections, and port scans.</td>
</tr>
<tr>
<td>Lateral Movement</td>
<td>To expand the footprint inside your network, and attacker uses lateral movement techniques to obtain credentials to gain additional access to more data in the network.</td>
</tr>
</tbody>
</table>
The analytics engine detects attacks during this phase by examining administrative operations (such as SSH, RDP, and HTTP), file share access, and user credential usage that is beyond the norm for your network. Some of the symptoms the app looks for are increased administrative activity, SMB usage, and remote code execution.

**Command and Control**

The command and control tactic allows an attacker to remotely issue commands to and endpoint and receive information from it. The analytics engine identifies intruders using this tactic by looking for anomalies in outbound connections, DNS lookups, and endpoint processes with bound ports. The app is looking for unexplained changes in the periodicity of connections and failed DNS lookups, changes in random DNS lookups, and other symptoms that suggest an attacker has gained initial control of a system.

**Exfiltration**

Exfiltration tactics are techniques to receive data from a network, such as valuable enterprise data. The app seeks to identify it by examining outbound connections with a focus on the volume of data being transferred. Increases in this volume are an important symptom of data exfiltration.

**Analytics Detectors**

The analytics engine for Cortex XDR retrieves logs from Cortex Data Lake to understand the normal behavior (creates a baseline) so that it can raise alerts when abnormal activity occurs. This analysis is highly sophisticated and performed on more than a thousand dimensions of data. Internally, the Cortex XDR app organizes its analytics activity into algorithms called *detectors*. Each detector is responsible for raising an alert when worrisome behavior is detected.

To raise alerts, each detector compares the recent past behavior to the expected baseline by examining the data found in your logs. A certain amount of log file time is required to establish a baseline and then a certain amount of recent log file time is required to identify what is currently happening in your environment.

There are four meaningful time intervals for Cortex XDR Analytics detectors:

<table>
<thead>
<tr>
<th>Time Interval</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detection Frequency</td>
<td>How often the Cortex XDR app runs the detector algorithm. Usually this is a short interval (10 minutes to 1 hour).</td>
</tr>
<tr>
<td>Learning Period</td>
<td>The shortest amount of log file time before the app can raise an alert. This is typically the time from when a detector first starts running and when you see an alert but, in some cases, detectors</td>
</tr>
</tbody>
</table>
pause after an upgrade as they enter a new learning period.
Most but not all detectors will wait until they have a learning period amount of time before they run. This learning period exists to give the detector enough data to establish a baseline, which in turn helps to avoid false positives.
The learning period is also referred to as the profiling or waiting period and, informally, it is also referred to as soak time.

Test Period
The amount of logging time that a detector uses to determine if unusual activity is occurring on your network. The detector compares test period data to the baseline created during the training period, and uses that comparison to identify abnormal behavior.

Training Period
The amount of logging time that the detector requires to establish a baseline, and to identify the behavioral limits beyond which an alert is raised. Because your network is not static in terms of its topology or usage, detectors are constantly updating the baselines that they require for their analytics. For this update process, the training period is how far back in time the detector goes to update and tune the baseline.
This period is also referred to as the baseline period.

When establishing a baseline, detectors compute limits beyond which network activity will require an alert. In some cases, detectors do not compute baseline limits; instead they are predetermined by Cortex XDR engineers. The engineers determine the values used for predetermined limits using statistical analysis of malicious activity recorded worldwide. The engineers routinely perform this statistical analysis and update the predetermined limits as needed with each release of the Cortex XDR.

These time periods are different for every Cortex XDR Analytics detector. The actual amount of logging data (measured in time) required to raise any given Cortex XDR Analytics alert is identified in the Cortex XDR Analytics Alert Reference.
Analytics Management

From Cortex XDR, you can manage settings for the analytics engine.

- Analytics Status
- Analytics Configuration Settings
- Analytics Management

Analytics Status

The Cortex XDR Analytics Status page provides information regarding the health and operational status of the analytics engine. Each tab on the Status page displays the health and/or status information about a specific aspect of the analytics engine:

- **Analytics System Status**—provides quick-reference on the overall health of the system.

  *If the system status is not OK, then the gear icon shows a red exclamation point. You can find additional details under the System Status tab.*

- **Analytics Log Status**—shows the number of logs recently received by Cortex XDR analytics engine.
- **Analytics Traffic Status**—shows the amount of network traffic observed by Cortex XDR analytics engine in the recent past.
- **Analytics Pathfinder Status**—shows the connection and scan status of the Pathfinder VM(s).
- **Analytics Traps Status**—shows the number of hosts from which the app receives Traps data.
- **Analytics Directory Sync Status**—shows the connection status of the Directory Sync Service paired with Cortex XDR.
- **Analytics Network Coverage Status**—provides a report on the networks that Cortex XDR analytics engine is monitoring, as well as relevant statistics on the IPs and traffic observed on each network.

These status pages are accessible from > Analytics Management > Status.

Analytics System Status

The Analytics System status tab provides quick-reference information that describes the overall health of the analytics engine. If the system status is not OK, then the gear icon shows a red exclamation point, and the System tab provides an alert that identifies the nature of the problem.

To receive emails notifications for system alerts (once per 24 hours), select the gear on the top menu bar of the Cortex XDR interface, and select Analytics Management > Configuration > System Alerts (see the System Alerts Configuration for details).
Analytics Log Status

The Analytics Log status tab provides information on the app log reception. Use this tab to see how many logs the app has received, how fast it is receiving them, and the types of logs it is receiving.

![Analytics Log Status Table]

Analytics Traffic Status

The Analytics Traffic status tab provides information about how much network traffic the app is observing. It provides statistics on overall traffic volume, as well as internal-to-internal and internal-to-external traffic.

If the numbers shown here seem lower than they should be, then check to ensure all your next-generation firewalls are configured to forward their logs to the Cortex Data Lake.

![Analytics Traffic Status Chart]

Analytics Pathfinder Status

The Pathfinder status tab provides information on your Pathfinder installation(s). If this tab does not show you the proper Pathfinder VM count, or if your Pathfinder scans are failing at a high rate, you should troubleshoot your Pathfinder configurations.

![Analytics Pathfinder Status Table]
Analytics Traps Status

The Traps status tab provides statistics about the number of Cortex XDR and Traps agents that are sending data for analysis by the analytics engine.

Analytics Directory Sync Status

The Analytics Directory Sync status tab indicates whether the Cortex XDR analytics engine is configured to use the Directory Sync Service. If the Cortex XDR analytics engine is configured to use the Directory Sync Service, this tab displays the health of that service.
Analytics Network Coverage Status

The Analytics Network Coverage status tab provides a report on the networks that the app is monitoring. For each monitored network, the report provides metrics for a specified time range that you define. Only networks that have had traffic during the reporting interval are shown in the report.

- Network Coverage Metrics
- Network Coverage Report Time Ranges
- Network Coverage Warnings
For a more detailed report on a given network, select the network name in the left hand column.

Reports are displayed in the Network Coverage tab. Alternatively, you can export reports to a CSV file for importation into a spreadsheet or similar software.

If you close your browser during report generation, the generation will continue in the background.

Network Coverage Metrics

Each Network Coverage report provides the following information:
<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>First IP</td>
<td>The first, or lowest, IP address in the network for which Cortex XDR has observed traffic. Based on data retrieved from traffic logs.</td>
</tr>
<tr>
<td>Last IP</td>
<td>The last, or highest, IP address in the network for which Cortex XDR has observed traffic. Based on data retrieved from traffic logs.</td>
</tr>
<tr>
<td>Total IPs Seen</td>
<td>The total number of IP addresses that Cortex XDR – Analytics has observed operating on the network. Based on data retrieved from traffic logs.</td>
</tr>
<tr>
<td>SSL %</td>
<td>The percentage of IP addresses on the network that are using the SSL protocol. Based on data retrieved from enhanced application logs.</td>
</tr>
<tr>
<td>HTTP %</td>
<td>The percentage of IP addresses on the network that are using the HTTP protocol. Based on data retrieved from URL logs.</td>
</tr>
<tr>
<td>DNS %</td>
<td>The percentage of IP addresses on the network that are using the DNS protocol. Based on data retrieved from enhanced application logs.</td>
</tr>
<tr>
<td>DHCP %</td>
<td>The percentage of IP addresses on the network that are using the DHCP protocol. Based on data retrieved from enhanced application logs.</td>
</tr>
<tr>
<td>Attempted Pathfinder Scans</td>
<td>The number of times that Pathfinder attempted to scan an endpoint on the network.</td>
</tr>
<tr>
<td>Successful Pathfinder Scans</td>
<td>The number of times that Pathfinder successfully scanned an endpoint on the network.</td>
</tr>
<tr>
<td>Average Throughput</td>
<td>The average number of bits per second sent across the network. Based on data retrieved from traffic logs.</td>
</tr>
</tbody>
</table>

**Network Coverage Report Time Ranges**

To generate a Network Coverage report, you must identify a time range. You can select from three predefined time ranges, or create a custom range.
Predefined time ranges are **Latest hour**, **Latest day**, and **Latest week**. These reports cover an hour, day, or week of activity from the last log record that Cortex XDR has received. In other words, they represent the *latest available* time range worth of activity. Because the app should be continually receiving logs, these time ranges should reflect time ranges up to the current time. However, if log reception has paused or stopped, then these time ranges will represent historical data. As an extreme example, if the app stopped receiving logs a month in the past, then **Latest Hour** will show you network activity represented by the last hour worth of logs from the previous month.

The custom time range allows you to generate reports over an interval that you define. Use this to create reports that cover up to 30 days. Log data must exist for the time range that you specify. If you attempt to select a start time for your report that is prior to the first log record that the app has, then Cortex XDR will indicate the problem with:

```
The start time cannot be earlier than: xxxxxxxx
```

where `xxxxxxxx` identifies the timestamp on the first log record that the app has. Similarly, if you select a time that is after the last log record that the app has, then the following message is shown:

```
The end time cannot be later than: xxxxxxxx
```

Finally, be aware that reports are rounded to a time interval that is determined by the report time range:

- Ten minutes for reports covering seven days or less.
- An hour for reports covering more than seven days.

*If Cortex XDR rounds your time range to any of these intervals, it always rounds up so that the time range you requested is fully contained in the report. It also provides the following informational message:*

```
Selected time ranges were modified to match available time resolution.
```

**Network Coverage Warnings**

It is possible for Cortex XDR to place a warning on individual cells in the network coverage report. Warnings indicate that the network traffic the app has observed is unexpected in some way. To see the warning text, hover your cursor over the warning icon.
Warning Text | Description
--- | ---
Low coverage | The app sees less than half of the hosts on the network performing traffic using the identified protocol. For example, if a network contains several hundred IPs but only 30% of those IPs are using HTTP, then the app would flag this as low coverage. The warning does not necessarily mean that there is a problem — HTTP traffic comprising only 30% of the total might be perfectly normal for that particular network — but Cortex XDR considers this unusual so it flags the issue in the report.

Received and transmitted traffic are not balanced | Cortex XDR is observing considerably more send traffic than receive, or more receive than send, which is unexpected for TCP/IP traffic (the difference is 25% or greater). This could indicate that the app is not receiving all the logs that it should receive. A possible reason for this is a misconfigured TAP interface or SPAN port on your next-generation firewalls.

Analytics Configuration Settings

Click [Analytics Management] > [Configuration] to access Pathfinder and analytics engine settings.

- **Pathfinder VMs Configuration**—Pair Pathfinder VMs with Cortex XDR and monitor Pathfinder VM status.
- **Pathfinder Configuration**—Configure the Pathfinder default configuration. To monitor Pathfinder scanning activity, instead select Management > Pathfinder.
- **Network Segments Configuration**—Configure which of your IP ranges Cortex XDR will monitor for Analytics alerts. Also report on the activities for each IP range the app is monitoring.
- **Directory Sync Configuration**—Tell Cortex XDR which Active Directory domain to use when paired with the Palo Alto Networks Directory Sync Service.
- **System Alerts Configuration**—Configure emailed system alerts. A system alert occurs if the system experiences any kind of an internal error.
- **EDL Configuration**—External block lists (EDL) can be used as the source for a Palo Alto Networks firewall external dynamic list (EDL). This allows you to enforce your security policy based on IP addresses and domains found to be associated with an alert.

Pathfinder VMs Configuration

The Analytics Pathfinder VMs page allows you to pair Pathfinder virtual machines installed on your network with your Cortex XDR app. This page also allows you to monitor the status of your Pathfinder VMs.
Pairing is the process of creating a network connection between your Cortex XDR Analytics system and the Pathfinder VM(s) running on your network. The pairing process requires a pairing token, which you obtain using the Generate pairing token button on this page.

This is one step in the workflow to Set Up Pathfinder.

The Pathfinder network connection is from Pathfinder to the Cortex XDR cloud service. Cortex XDR never needs to connect into your network.

Pathfinder Configuration

The Pathfinder page allows you to configure Pathfinder settings. Pathfinder is a virtual machine that you install on your network for the purpose of investigating your network endpoints for suspicious/malicious software and other artifacts.
When you first click on this tab, the **Default Configuration** screen is displayed. Use this screen to specify what and how Pathfinder scans:

- **Allow automatic scans for workstations** (On By Default)—Enable Pathfinder to automatically scan endpoints.
- **Allow automatic scans for servers** (Off By Default)—Enable Pathfinder to automatically scan servers.
- **Allow uploading suspicious files to WildFire** (On By Default)—Enable this option to allow Pathfinder to send suspicious files for WildFire analysis, or disable it so that Pathfinder sends WildFire only the file hash (MD5 and SHA-256).

When WildFire™ identifies a file as malware, the analytics engine generates a malware alert. However, the malware alert provides more expansive detection with this option is enabled. WildFire can identify known malware based on only the file hash, but must execute and observe the file itself to identify previously-unknown, zero-day malware.

*The Analytics Audit Log* records when Pathfinder sends a file to WildFire (select > Analytics Management > Management > Audit).

- **Enable N2PA 2-week monitoring for workstations** (Off By Default)—With N2PA monitoring enabled, Pathfinder activates Windows Instrumentation features on a suspicious device to start process logging for that endpoint. Pathfinder then periodically scans the device to collect the process logs and sends those logs to Cortex XDR. The app uses this additional forensic data to attribute network activity to a specific process running on the endpoint. The app displays the additional forensic data that N2PA collects along with device details, so that you can better isolate the origin of malicious network activity and take action. When a device ceases to display suspicious behavior, Pathfinder stops collecting the
event logs after a two week period. Otherwise, N2PA monitoring continues until suspicious activity is no longer detected with the device.

In the **Credentials** section, you can specify the login credentials that Pathfinder uses by default to scan your Microsoft Windows endpoints. Alternatively, you can select **Use local Pathfinder VM credentials** and instead specify these login credentials locally on the Pathfinder VM.

To locally configure a Pathfinder VM with the credentials it needs to scan your network, select the gear on the menu bar, select \( \text{Analytics Management} \) > \( \text{Configuration} \) > Pathfinder, and follow the steps to **Set Up Pathfinder**.

Navigate to the other tabs on this page to continue to configure Pathfinder:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Global Settings</strong></td>
<td>Allows you to configure how Pathfinder behaves relative to your entire network. Use this tab to indicate whether Pathfinder should automatically scan devices when it first discovers them (N2PA is never used for newly discovered devices). Also use this tab to identify the DNS domains that Pathfinder uses to locate, connect to, and scan a host.</td>
</tr>
<tr>
<td><strong>Per Asset Configuration</strong></td>
<td>Allows you to set Pathfinder configurations that are specific to a subset of devices on your network. You can override the default Pathfinder configuration on a per-asset basis using this page. This requires you to first have configured network assets using the Network Segments Configuration.</td>
</tr>
</tbody>
</table>

**Network Segments Configuration**

The Analytics **Network Segments** page allows you to view and manage the internal networks and mobile endpoints that the Cortex XDR analytics engine is monitoring.

This page lists the IP address ranges that Cortex XDR monitors. By default, Cortex XDR is configured to monitor the standard internal IP address ranges (for example, 10.x.x.x, 172.16.x.x, and 192.168.x.x). If you have non-standard internal IP address ranges that the app should monitor, or use GlobalProtect or Prisma Access to secure mobile endpoint traffic, add those IP address ranges to the table. For GlobalProtect and Prisma Access IP address ranges, also select **Reserve for VPN** to enable the Cortex XDR app to recognize the network segment as a VPN IP address pool.

To add a network segment to the table, click the + in the upper-right corner. Alternatively, you can export the table to CSV, modify it, and then reimport the modified file. See **Set up Cortex XDR Pro Overview** for more information on configuring network segments.
You can also use this page to edit existing network segment start and end IP addresses. If your edits result in an error (that is, you provide an illegal IP address or create overlapping ranges), the page will provide an error indicator for the improper values.

You can use this page to assign a Pathfinder VM to a network segment. Use the column in the segment table row to select the Pathfinder VM that you want to monitor the segment.

If the network segment is a GlobalProtect or Prisma Access VPN pool, select Reserved for VPN in the final column and do not assign a Pathfinder VM to the network segment.

To delete a segment, hover over the segment and click the trashcan icon, or delete the segment from your CSV file and then reimport it.

You cannot remove or edit the default IPv4 and IPv6 address ranges from the table. However, you can add ranges that are more specific than these defaults.

From the Network Segments page, you can access the IP Ranges Report page where you can generate reports on the various networks that Cortex XDR is monitoring. Use the Open network coverage report link to access this page.

For each network that Cortex XDR is monitoring, this report shows you relevant information such as the number of IPs discovered by Cortex XDR, Pathfinder scan activity (attempts and successes), and the amount of traffic seen on the network. Both the Network Segments page and the IP Ranges Report displays the percentage of DNS, DHCP, HTTP, and SSL traffic that the Palo Alto Networks firewall logs for each IP range (%DNS, %DHCP, %HTTP, %SSL). However, be aware that you must enable the firewall to send Enhanced Application Logs to the Cortex Data Lake in order for Cortex XDR to display the percent coverage for these types of application traffic.

Directory Sync Configuration

The Directory Sync Service is an optional service that enables you to add additional details to triage and alert information. To use the Directory Sync Service with Cortex XDR Analytics, you must first activate and configure Directory Sync. See the Directory Sync Service Getting Started Guide for instructions on how to do this.
Then, Set Up Directory Sync to work with Cortex XDR Analytics. As part of this task, use the Directory Sync page to tell Cortex XDR Analytics which Active Directory domain to use.

### System Alerts Configuration

Use the Cortex XDR Analytics System Alerts page to configure delivery of system alerts. A system alert occurs if the app experiences any kind of an internal error. To view current system status, select Analytics Management > Status.

### EDL Configuration

Cortex XDR hosts two block lists, to which you can add IP addresses and domains as you triage alerts. You can use a Cortex XDR external dynamic list (EDL) with a Palo Alto Networks firewall to provide an integrated response to malicious network activity. With a Cortex XDR EDL as the source of a firewall external dynamic list, the firewall can control user access to IP addresses and domains that the app has found to be associated with an alert.

The following steps describe how to set up a Palo Alto Networks firewall to use the Cortex XDR EDL as the source for an external dynamic list (EDL), and how to start building a block list.

Before you begin:

- Validate the firewall DNS configuration to make sure that it can resolve the Cortex XDR FQDN.
Ensure the firewall has a direct internet connection; if another network device resides between the firewall and the internet, make sure that device is configured to allow the traffic between the firewall and Cortex XDR.

**STEP 1 | Enable EDL.**

1. Select > Analytics Management > Configuration and then select EDL.

![EDL Configuration](image)

2. Enter login credentials that the Palo Alto Networks firewall should use to access the Cortex XDR EDL.

**STEP 2 | Record the IP Addresses Block List URL and the Domains Block List URL.** You will need these URLs in the coming steps to point the firewall to these lists.

![URLs](image)

*Test the URLs in a browser to confirm that they are active.*

**STEP 3 | Save** the EDL configuration.

**STEP 4 | Enable** the firewall to authenticate the Cortex XDR EDL.

1. Download and save the following root certificate: https://certs.godaddy.com/repository/gd-class2-root.crt.

2. On the firewall, select Device > Certificate Management > Certificates and Import the certificate. Make sure to give a descriptive name, and select OK to save the certificate.
4. Give the profile a descriptive name and Add the certificate to the profile.

5. Select OK to save the certificate profile.

**STEP 5 | Set the Cortex XDR EDL as the source for a firewall EDL.**

For more detailed information about how Palo Alto Networks firewall EDLs work, how you can use EDLs, and how to configure them, review how to Use an External Dynamic List in Policy.

1. On the firewall, select Objects > External Dynamic Lists and Add a new list.
2. Define the list Type as either IP List or Domain List.
3. Enter the IP Addresses Block List URL or the Domains Block List URL that you recorded in the last step as the list Source.
4. Select the Certificate Profile that you created in the last step.
5. Select Client Authentication and enter the username and password that the firewall must use to access the Cortex XDR EDL. These should be the same login credentials that you saved in Cortex XDR, when enabling the EDL in the first step.
6. Use the Repeat field to define how frequently the firewall retrieves the latest list from Cortex XDR.
7. Click OK to add the new EDL.

**STEP 6** | Select **Policies > Security** and **Add** or edit a security policy rule to add the Cortex XDR EDL as match criteria to a security policy rule.

Review the different ways you can **Enforce Policy on an External Dynamic List**; this topic describes the complete workflow to add an EDL as match criteria to a security policy rule.

1. Select **Policies > Security** and **Add** or edit a security policy rule.
2. In the **Destination** tab, select **Destination Zone** and select the external dynamic list as the **Destination Address**.
3. Click **OK** to save the security policy rule and **Commit** your changes.

You do not need to perform additional commit or make any subsequent configuration changes for the firewall to enforce the EDL as part of your security policy; even as you update the Cortex XDR EDL, the firewall will enforce the list most recently retrieved from Cortex XDR.

You can also use the Cortex XDR domain list as part of a URL Filtering profile; when attached to a security policy rule, a URL Filtering profile allows you to granularly control user access to the domains on the list.

**STEP 7** | Start building your Cortex XDR EDL.

1. Select **Triage** and select a host or user to see the associated alerts.
2. Select an alert that the host or user triggered.
3. Select **Actions** and **Add to Block List** to see the full list of IP addresses and domains associated with the alert. From this list, select the entries you want to add and then **Submit**.

Entries that are already included in a EDL appear as selected.

**STEP 8** | View and manage the Cortex XDR EDL.

Select **> Analytics Management > Management** and then select **EDL**. Two lists are displayed: one for IP addresses and one for domains. Here, you can delete any entries that you no longer want included on the lists.

**Analytics Management**

From the Cortex XDR **> Analytics Management > Management** menu, you can manage whitelist rules, view the audit log, and view the status of Pathfinder scans:
• Select **Pathfinder** to see ongoing and queued **Pathfinder Scans**, Pathfinder scan history, and suspicious endpoints that are undergoing periodic scanning (N2PA monitoring).
• Select **Audit** to see the **Analytics Audit Log**, which records the triage activity that has occurred in the Cortex XDR Analytics application lately.
• Select **EDL** to view **Analytics External Dynamic List**. You can add IP addresses and domains to the Cortex XDR block lists as you triage alerts; a Palo Alto Networks firewall can then dynamically enforce policy based on these lists.

**Pathfinder Scans**

When the analytics engine observes problematic traffic coming from an endpoint that does not have a supported Cortex XDR agent installed with endpoint monitoring and data collection enabled, it uses Pathfinder to investigate the endpoint. At any time, you can also initiate a Pathfinder scan for a particular device. If N2PA (network-to-process association) monitoring is enabled, Pathfinder also automatically performs periodic scanning for devices that have displayed suspicious behavior.

The **Analytics Management > Management > Pathfinder** page displays status for all Pathfinder scan types. You can view both in-progress and queued Pathfinder scans, a history of the scans Pathfinder has performed, and a list of devices that are undergoing N2PA monitoring. You can also export the **Scan History** and **Hosts Under N2PA Monitoring** lists to a flat-text file for the purposes of viewing them in a spreadsheet application.

• To initiate a Pathfinder scan for a particular device, select Actions > Pathfinder Scan from the Analyze view of an Analytics alert.

Endpoints that are protected and monitored by Cortex XDR agents do not require additional Pathfinder scans.
• To enable N2PA monitoring, click the gear on the top menu bar, select Configuration > Pathfinder and enable Network to Process Association (N2PA). (For details, see Pathfinder Configuration).

**Analytics Audit Log**

The Analytics **Audit** log lists the recent triage activity that you and your team has performed in Cortex XDR. This includes all changes to alert statuses and whitelist rules. The log identifies who performed the change, when the change occurred, and the host to which the change is related.
The Audit log separates security policy-related logs on the Analyst tab, and other configuration and management changes on the Configuration tab.

Analytics External Dynamic List

Cortex XDR™ hosts two built-in block lists, to which you can easily add IP addresses and domains that you find as you triage alerts. You can use these block lists as sources for a Palo Alto Networks firewall external dynamic list (EDL); this allows you to enforce security policy based on IP addresses and domains that the app has found to be associated with anomalous and malicious network activity.

Any IP addresses and domains associated with an alert can be added to a block list. To enable block lists, and configure a Palo Alto Networks firewall to use them as EDL sources, see EDL Configuration.
Monitoring

- Cortex XDR Dashboard
- Monitor Administrative Activity
  - Forward Your Management Audit Log
- Monitor Agent Activity
  - Forward Your Agent Audit Log
Cortex XDR Dashboard

The Dashboard screen is the first page you see in the Cortex XDR app when you log in.

1. Dashboard menu
2. Dashboard Widgets
3. Color theme toggle

The dashboard is comprised of Dashboard Widgets (2) that summarize information about your endpoint in graphical or tabular format. You can customize Cortex XDR to display Predefined Dashboards or create your own custom dashboard using the dashboard builder. You can toggle between your available dashboards using the dashboard menu (1).

In addition, the dashboard provides a color theme toggle (3) that enables you to switch the interface colors between light and dark.

Dashboard Widgets

Cortex XDR provides the following list of widgets to help you create dashboards and reports displaying summarized information about your endpoints.

Cortex XDR sorts widgets in the Cortex XDR app according to the following categories:

- Agent Management Widgets
- Incident Management Widgets
- Investigation Widgets
- User Defined Widgets
## Agent Management Widgets

<table>
<thead>
<tr>
<th>Widget Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent Content Version Breakdown</td>
<td>Displays the total number of registered agents and their distribution according to the installed content update version.</td>
</tr>
<tr>
<td>Agent Status Breakdown</td>
<td>Provides a summary of the total number of endpoint agents according to their status.</td>
</tr>
<tr>
<td>Agent Version Breakdown</td>
<td>Displays the total number of registered agents and their distribution according to agent versions.</td>
</tr>
<tr>
<td>Number of Installed Agents</td>
<td>Displays a timeline of the number of agents installed on endpoints over the last 24 hours, 7 days, or 30 Days.</td>
</tr>
<tr>
<td>Operating System Type Distribution</td>
<td>Displays the total number of registered agents and their distribution according to the operating system.</td>
</tr>
</tbody>
</table>

## Incident Management Widgets

<table>
<thead>
<tr>
<th>Widget Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incidents By Assignee</td>
<td>Displays the distribution of incidents according to users and then the number of aged and open incidents. Aged incidents have not been modified in seven days. Click a user to open a filtered view of incidents assigned to the selected user.</td>
</tr>
<tr>
<td>Incidents By Status</td>
<td>Provides a summary of the total current number of open incidents according to status. Click a status to open a filtered view of the incidents.</td>
</tr>
<tr>
<td>Widget Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Data Usage Breakdown</td>
<td>Displays a timeline of the consumption of Cortex XDR data in TB. Hover over the graph to see the amount at a specific time.</td>
</tr>
<tr>
<td>Detection By Actions</td>
<td>Displays the top five actions performed on alerts or incidents. In the upper right corner:</td>
</tr>
<tr>
<td></td>
<td>• Toggle between alerts and incidents</td>
</tr>
<tr>
<td></td>
<td>• Select to view the number of alert/incidents per action over the last 24 hours, 7 days, or 30 Days</td>
</tr>
<tr>
<td>Detections By Category</td>
<td>Displays the top five categories of alerts or incidents. In the upper right corner:</td>
</tr>
<tr>
<td></td>
<td>• Toggle between alerts and incidents</td>
</tr>
<tr>
<td></td>
<td>• Select to view the number of alert/incidents per category over the last 24 hours, 7 days, or 30 Days</td>
</tr>
<tr>
<td>Detection By Source</td>
<td>Displays the top five sources of alerts or incidents. In the upper right corner:</td>
</tr>
<tr>
<td></td>
<td>• Toggle between alerts and incidents</td>
</tr>
<tr>
<td></td>
<td>• Select to view the number of alert/incidents per source over the last 24 hours, 7 days, or 30 Days</td>
</tr>
<tr>
<td>Open Incidents</td>
<td>Displays a timeline of open incidents over time and the number of aged and open incidents. Aged incidents have not been modified in seven days.</td>
</tr>
<tr>
<td></td>
<td>Select the time scope in the upper right to view the number of open incidents over the last 24 hours, 7 days, or 30 Days.</td>
</tr>
<tr>
<td></td>
<td>Hover over the graph to view the number of open incidents on a specific day.</td>
</tr>
<tr>
<td>Open Incidents by Severity</td>
<td>Provides a summary of the total current number of open incidents according to severity.</td>
</tr>
<tr>
<td></td>
<td>Click a severity percentage to open a filtered view of the incidents.</td>
</tr>
<tr>
<td>Widget Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Response Action Breakdown</td>
<td>Displays the top response actions taken in the Action Center over the last 24 hours, 7 days, or 30 Days.</td>
</tr>
<tr>
<td>Top Hosts</td>
<td>Displays the top ten hosts with the highest number of incidents according to severity. Incidents are color-coded: red for high and yellow for medium. Click a host to open a filtered view of all open incidents for the selected host.</td>
</tr>
<tr>
<td>Top Incidents</td>
<td>Displays the top ten current incidents with the highest number of alerts according to severity. Alerts are color-coded: red for high and yellow for medium. Click a severity to open a filtered view of all open alerts for the selected incident.</td>
</tr>
</tbody>
</table>

**User Defined Widgets**

<table>
<thead>
<tr>
<th>Widget Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Free Text</td>
<td>Displays a text box allowing to insert free text.</td>
</tr>
<tr>
<td>Header</td>
<td>Displays a title containing the free text. For example, name and description of a report or dashboard, customer name, tenant ID, or date.</td>
</tr>
</tbody>
</table>

**Predefined Dashboards**

Cortex XDR comes with predefined dashboards that display widgets tailored to the dashboard type. You can select any of the predefined dashboards directly from the dashboard menu in Reporting > Dashboard. You can also select and rename a predefined dashboard in the Dashboard Builder available by clicking + New Dashboard. The types of dashboards that are available to you depend on your license type but can include:

- Agent Management Dashboard
- Incident Management Dashboard
- Security Manager Dashboard
Agent Management Dashboard

The Agent Management Dashboard displays at-a-glance information about the endpoints and agents in your deployment.

Support for the Agent Management Dashboard requires either a Cortex XDR Prevent or Cortex XDR Pro per Endpoint license.

The dashboard is comprised of the following Dashboard Widgets:

- Agent Status Breakdown
- Agent Content Version Breakdown (Top 5)
- Agent Version Breakdown (Top 5)
- Operating Type Distribution
- Top Hosts (Top 10)
The Incidents Management Dashboard provides a graphical summary of incidents in your environment, with incidents prioritized and listed by severity, assignee, incident age, and affected hosts.

The dashboard is comprised of the following Dashboard Widgets:

- Incidents by Assignee (Top 5 Assignees)
- Open Incidents
- Open Incidents By Severity
- Top Hosts (Top 10)
- Top Incidents (Top 10)

To filter a widget to display only incidents that match incident starring policies, select the star in the right corner. A purple star indicates that the widget is displaying only starred incidents. The starring filter is persistent and will continue to show the filtered results until you clear the star.
The Security Manager Dashboard widgets display general information about Cortex XDR incidents and agents. If you migrated from either Traps management service or the Endpoint Security Manager, you will notice similarities between the dashboards.

Support for Security Manager Dashboard requires either a Cortex XDR Prevent or Cortex XDR Pro per Endpoint license.

The dashboard is comprised of the following Dashboard Widgets:

- Agent Status Breakdown
- Agent Version Breakdown (Top 5)
- Incidents by Assignees
- Open Incidents
- Open Incidents by Severity
- Top Incidents (Top 10)

For incident-related widgets you can also filter the results to display only incidents that match incident starring policies. To apply the filter, select the star in the right corner of the widget. A purple star indicates that the widget is displaying only starred incidents. The starring filter is persistent and will continue to show the filtered results until you clear the star.

Build a Custom Dashboard

To create purposeful dashboards, you must consider the information that you and other analysts find important to your day to day operations. This consideration guides you in building a custom dashboard. When you create a dashboard, you can select widgets from the widget library and choose their placement on the dashboard.

STEP 1 | Select Reporting > Dashboards Manager > + New Dashboard.

STEP 2 | Enter a unique Dashboard Name and an optional Description of the dashboard.

STEP 3 | Choose the Dashboard Type.
You can use an existing dashboard as a template, or you can build a new dashboard from scratch.

**STEP 4** | Click **Next**.

**STEP 5** | Customize your dashboard.

To get a feel for how the data will look, Cortex XDR provides mock data. To see how the dashboard would look with real data in your environment, you can use the toggle above the dashboard to use **Real Data**.

Drag and drop widgets from the widget library to their desired position.

If necessary, remove unwanted widgets from the dashboard. To remove a widget, select the menu in the top right corner, and **Remove widget**.

For incident-related widgets, you can also select the star to display only incidents that match an incident starring configuration on your dashboard. A purple star indicates that the widget is displaying only starred incidents (see **Create an Incident Staring Configuration**).

**STEP 6** | When you have finished customizing your dashboard, click **Next**.

**STEP 7** | To set the custom dashboard as your default dashboard when you log in to Cortex XDR, **Define as default dashboard**.

**STEP 8** | To keep this dashboard visible only for you, select **Private**.

Otherwise, the dashboard is public and visible to all Cortex XDR app users with the appropriate roles to manage dashboards.

**STEP 9** | **Generate** your dashboard.
Manage Dashboards

From the Reporting > Dashboards Manager, you can view all custom and default dashboards. From the Dashboards Manager, you can also delete, edit, duplicate, disable, and perform additional management actions on your dashboards.

To manage an existing dashboard, right click the dashboard and select the desired action.

- **Delete** - Permanently delete a dashboard.
- **Edit** - Edit an existing dashboard. You cannot edit the default dashboards provided by Palo Alto Networks, but you can save it as a new dashboard.
- **Save as new** - Duplicate an existing template.
- **Disable** - Temporarily disable a dashboard. If the dashboard is public, this dashboard is also removed for all users.
- **Set as default** - Make the dashboard the default dashboard that displays when you (and other users, if the dashboard is public) log in to Cortex XDR.
- **Save as report template** - Save a report as a template.

Run or Schedule Reports

There are two ways to create a report template:

- Run a Report Based on a Dashboard
- Create a Report from Scratch

**Run a Report Based on a Dashboard**

**STEP 1 |** Select Reporting > Dashboards Manager.

**STEP 2 |** Right-click the dashboard from which you want to generate a report, and select *Save as report template*.

**STEP 3 |** Enter a unique *Report Name* and an optional *Description* of the report, then *Save* the template.

**STEP 4 |** Select Reporting > Report Templates.

**STEP 5 |** Run the report.

You can either *Generate Report* to run the report on-demand, or you can *Edit* the report template to define a schedule.

**STEP 6 |** After your report completes, you can download it from the Reporting > Reports page.

**Create a Report from Scratch**

**STEP 1 |** Select Reporting > Report Templates > *+ New Template*.

**STEP 2 |** Enter a unique *Report Name* and an optional *Description* of the report.

**STEP 3 |** Select the *Data Timeframe* for your report.

You can choose *Last 24H* (day), *Last 7D* (week), *Last 1M* (month), or you can choose a custom timeframe.
Custom timeframe is limited to one month.

STEP 4 | Choose the Report Type.
You can use an existing template, or you can build a new report from scratch.

STEP 5 | Click Next.

STEP 6 | Customize your report.
To get a feel for how the data will look, Cortex XDR provides mock data. To see how the report would look with real data in your environment, you can use the toggle above the report to use Real Data.
Drag and drop widgets from the widget library to their desired position.
If necessary, remove unwanted widgets from the template. To remove a widget, select the menu in the top right corner, and select Remove widget.
For incident-related widgets, you can also select the star to include only incidents that match an incident starring configuration in your report. A purple star indicates that the widget is displaying only starred incidents.

STEP 7 | When you have finished customizing your report template, click Next.

STEP 8 | If you are ready to run the report, select Generate now.

STEP 9 | To run the report on a regular Schedule, you can specify the time and frequency that Cortex XDR will run the report.

STEP 10 | Enter an optional Email Distribution or Slack workspace to send a PDF version of your report.
Select Add password for e-mailed report to set a password encryption.
Ensure you have #unique_320.

STEP 11 | Save Template.

STEP 12 | After your report completes, you can download it from the Reporting > Reports page.
Monitor Cortex XDR Incidents

The **Incidents** table lists all incidents in the Cortex XDR app.

### Incidents Table

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Check box to select one or more incidents on which to perform the following actions.</td>
</tr>
<tr>
<td></td>
<td>• Assign incidents to an analyst in bulk</td>
</tr>
<tr>
<td></td>
<td>• Change the status of multiple incidents</td>
</tr>
<tr>
<td></td>
<td>• Change the severity of multiple incidents</td>
</tr>
</tbody>
</table>

**Actions**

Manage multiple incidents with **Actions**.

**Alerts Breakdown**

The total number of incidents and number of alerts by severity.

**Assignee Email**

Email address associated with the assigned incident owner.

**Assigned To**

The user to which the incident is assigned. The assignee tracks which analysts are responsible for investigating the threat. Incidents that have not been assigned have a status of **Unassigned**.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creation Time</td>
<td>The time the first alert was added to a new incident.</td>
</tr>
<tr>
<td>Hosts</td>
<td>The number of hosts affected by the incident. Right-click the host count to view the list of hosts grouped by operating system.</td>
</tr>
<tr>
<td>Incident Description</td>
<td>The description is generated from the alert name from the first alert added to the incident, the host and user affected, or number of users and hosts affected.</td>
</tr>
<tr>
<td>Incident ID</td>
<td>A unique number to identify the incident.</td>
</tr>
<tr>
<td>Incident Name</td>
<td>A user-defined incident name.</td>
</tr>
<tr>
<td>Incident Sources</td>
<td>List of sources that raised high and medium severity alerts in the incident.</td>
</tr>
<tr>
<td>Last Updated</td>
<td>The last time a user took an action or an alert was added to the incident.</td>
</tr>
<tr>
<td>Resolve Comment</td>
<td>The user-added comment when the user changes the incident status to a Resolved status.</td>
</tr>
<tr>
<td>Severity</td>
<td>The highest alert in the incident or the user-defined severity.</td>
</tr>
<tr>
<td>Starred</td>
<td>The incident includes alerts that match your incident prioritization policy. Incidents that have alert matches include a star by the incident name in the Incident details view and a value of Yes in this field.</td>
</tr>
<tr>
<td>Status</td>
<td>Incidents have the status set to <strong>New</strong> when they are generated. To begin investigating an incident, set the status to <strong>Under Investigation</strong>. The Resolved status is subdivided into resolution reasons: • <strong>Resolved - Threat Handled</strong>  • <strong>Resolved - Known Issue</strong>  • <strong>Resolved - Duplicate Incident</strong>  • <strong>Resolved - False Positive</strong>  • <strong>Resolved - Auto Resolve</strong> - Auto-resolved by Cortex XDR when all of the alerts contained in an incident have been excluded.</td>
</tr>
<tr>
<td>Total Alerts</td>
<td>The total number of alerts in the incident.</td>
</tr>
<tr>
<td>Users</td>
<td>Users affected by the alerts in the incident. If more than one user is affected, click on + &lt;n&gt; more to see the list of all users in the incident.</td>
</tr>
</tbody>
</table>

From the **Incidents** page, you can right-click an incident to view the incident, and investigate the related assets, artifacts, and alerts. For more information see **Investigate Incidents**.
Monitor Administrative Activity

From Management Auditing, you can track the status of all administrative and investigative actions. Cortex XDR stores audit logs for 180 days. Use the page filters to narrow the results or Manage Columns and Rows to add or remove fields as needed.

You can also forward your Management Audit log to an email distribution list and Syslog servers.

The following table describes the default and optional additional fields that you can add in alphabetical order.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Email</td>
<td>Email address of the administrative user</td>
</tr>
<tr>
<td>Description</td>
<td>Descriptive summary of the administrative action</td>
</tr>
<tr>
<td>Host Name</td>
<td>Name of any relevant affected hosts</td>
</tr>
<tr>
<td>ID</td>
<td>Unique ID for the action</td>
</tr>
<tr>
<td>Result</td>
<td>Result of the administrative action: Success, Partial, or Fail.</td>
</tr>
<tr>
<td>Subtype</td>
<td>Sub category of action</td>
</tr>
<tr>
<td>Timestamp</td>
<td>Time the action took place</td>
</tr>
<tr>
<td>Type</td>
<td>Type of activity logged, one of the following:</td>
</tr>
<tr>
<td></td>
<td>• Live Terminal—Remote terminal sessions created and actions taken in the</td>
</tr>
<tr>
<td></td>
<td>file manager or task manager, a complete history of commands issued,</td>
</tr>
<tr>
<td></td>
<td>their success, and the response.</td>
</tr>
<tr>
<td></td>
<td>• Response—Remedial actions taken, for example to isolate a host and undo</td>
</tr>
<tr>
<td></td>
<td>isolate host, or blacklist a file hash signature, or undo a hash blacklist</td>
</tr>
<tr>
<td></td>
<td>• Result—Whether the action taken was successful or failed, and the result</td>
</tr>
<tr>
<td></td>
<td>reason when available.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Authentication</td>
<td>User sessions started, along with the user name that started the session.</td>
</tr>
<tr>
<td>Incident Management</td>
<td>Actions taken on incidents and on the assets, alerts, and artifacts in incidents.</td>
</tr>
<tr>
<td>Public API</td>
<td>Authentication activity using an associated Cortex XDR API key.</td>
</tr>
<tr>
<td>User Name</td>
<td>User who performed the action</td>
</tr>
</tbody>
</table>

Forward Your Management Audit Log

To ensure you and your colleagues are informed and updated, Cortex XDR allows you to forward your Management Audit log to an email and Syslog server.

To define a forwarding configuration:

**STEP 1** | Navigate to 
\[\text{Settings} > \text{Log Forwarding}.\]

**STEP 2** | + Add Notification Configuration.

**STEP 3** | Define the configuration Name and Description.

**STEP 4** | In the Log Type field, select Management Audit Logs.

**STEP 5** | In the Scope page, Filter your management audit table to define which type of event you want included in the forwarded log. For example, set a filter \(\text{TYPE} = \text{Authentication}, \text{SUBTYPE} = \text{Create}\). Events matching this filter are forwarded to your defined emails and Syslog servers.

*The filter you define is applied only to future events.*

**STEP 6** | In the Summary page, enter the details of how you want to forward your Management Audit log.

- Email Notification
  - In the Email Distribution field, add all the email addresses.
  - In the Email Grouping Time Frame field, set in minutes how often you to send the logs.

*By setting the time frame to zero, the log is sent as soon as the event is created.*

- Syslog Server
  - Select a Syslog server.

*Ensure you have defined a Syslog server in the Cortex XDR app.*

**STEP 7** | (Optional) In the Forwarding Configurations table, right-click a row to Edit, Disable, or Delete a configuration.
Monitor Agent Activity

Viewing agent audit logs requires either a Cortex XDR Prevent or Cortex XDR Pro per Endpoint license.

Cortex XDR agent logs include entries for events that are monitored by the Cortex XDR agent and are reported back to Cortex XDR hourly. All agent logs are classified with a system record type. To view all Cortex XDR agent logs, select > Agent Auditing.

The Cortex XDR agent logs these endpoint events using one of the following categories:

- **Audit**—Successful changes to the agent indicating correct behavior.
- **Monitoring**—Unsuccessful changes to the agent that may require administrator intervention.
- **Status**—Indication of the agent status.

You can also forward your Agent Audit log to an email distribution list and Syslog servers.

The following table describes the endpoint log types and sub-types that you can view in the Cortex XDR Agents Audit Table:

<table>
<thead>
<tr>
<th>Log Type</th>
<th>Log Sub-Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installation</td>
<td>• Install&lt;br&gt;• Uninstall&lt;br&gt;• Upgrade</td>
</tr>
<tr>
<td>Policy change</td>
<td>• Local Configuration Change&lt;br&gt;• Content Update&lt;br&gt;• Policy Update&lt;br&gt;• Process Exception&lt;br&gt;• Hash Exception</td>
</tr>
<tr>
<td>Agent service</td>
<td>• Service start&lt;br&gt;• Service stopped</td>
</tr>
<tr>
<td>Agent modules</td>
<td>• Module initialization&lt;br&gt;• Local analysis module&lt;br&gt;• Local analysis feature extraction</td>
</tr>
<tr>
<td>Agent status</td>
<td>• Fully protected&lt;br&gt;• OS incompatible&lt;br&gt;• Software incompatible</td>
</tr>
<tr>
<td>Log Type</td>
<td>Log Sub-Type</td>
</tr>
<tr>
<td>----------</td>
<td>--------------</td>
</tr>
</tbody>
</table>
|          | • Kernel driver initialization  
|          | • Kernel extension initialization  
|          | • Proxy communication  
|          | • Quota exceeded  
|          | • Minimal content  
| Action   | • Scan  
|          | • File retrieval  
|          | • Terminate process  
|          | • Isolate  
|          | • Cancel isolation  
|          | • Payload execution  
|          | • Quarantine  
|          | • Restore  

In addition to labeling all logs according to **Category**, **Type**, and **Sub-Type**, Cortex XDR stores for 180 days the following data for each log:

- Date and time on which the action occurred.
- Date and time on which the action was reported from the agent back to Cortex XDR.
- Endpoint identification data: endpoint ID, name, and domain.
- XDR agent version running on the endpoint.
- The result of the action (Success or fail).
- Reason for failed actions.
- If an incident has been auto-resolved because all of the alerts associated with the incident have been excluded.
- Full description of the action, including data such as number of affected endpoints, number of processes, reason for failure or success status.

You can customize your view of the logs by adding or removing fields to the **Agent Audits Table**. You can also filter the page result to narrow down your search.

**Forward Your Agent Audit Log**

To ensure you and your colleagues are informed and updated, Cortex XDR allows you to forward your Agent Audit log to an email and Syslog server.

To define a forwarding configuration:

1. Navigate to **Settings** > **Log Forwarding**.
2. + **Add Notification Configuration**.
3. Define the configuration **Name** and **Description**.
4. In the **Log Type** field, select **Agent Audit Logs**.
5. In the **Scope** page, **Filter** your management audit table to define which type of event you want included in the forwarded log. For example, set a filter **TYPE = Authentication, SUBTYPE = Create**. Events matching this filter are forwarded to your defined emails and Syslog servers.

   *The filter you define is applied only to future events.*

6. In the **Summary** page, enter the details of how you want to forward your Agent Audit log.
• Email Notification
  • In the Email Distribution field, add all the email addresses.
  • In the Email Grouping Time Frame field, set in minutes how often you to send the logs.
    
    By setting the time frame to zero, the log is sent as soon as the event is created.

• Syslog Server
  • Select a Syslog server.
    
    Ensure you have #unique_322 in the Cortex XDR app.

7. (Optional) In the Forwarding Configurations table, right-click a row to Edit, Disable, or Delete a configuration.
Log Forwarding

To help you stay informed and updated, you can easily forward Cortex XDR™ alerts and reports to an external syslog receiver, a Slack channel, or to email accounts.

> Log Forwarding Data Types
> Integrate Slack for Outbound Notifications
> Integrate a Syslog Receiver
> Configure Notification Forwarding
> Cortex XDR Log Notification Formats
Log Forwarding Data Types

To ensure you and your colleagues are informed and updated about events in your Cortex XDR deployment, you can Configure Notification Forwarding to Email, Slack, or a syslog receiver. The following table displays the data types supported by each notification receiver.

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Email</th>
<th>Slack</th>
<th>Syslog</th>
<th>Cortex XSOAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alerts</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Agent Audit Log</td>
<td>✓</td>
<td>—</td>
<td>✓</td>
<td>—</td>
</tr>
<tr>
<td>Cortex XDR Prevent or Cortex XDR Pro per Endpoint</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Management Audit Log</td>
<td>—</td>
<td>—</td>
<td>✓</td>
<td>—</td>
</tr>
<tr>
<td>Reports</td>
<td>✓</td>
<td>✓</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>
Integrate Slack for Outbound Notifications

Integrate Cortex XDR app with your Slack workspace to better manage and highlight your Cortex XDR alerts and reports. By creating a Cortex XDR Slack channel, you ensure that defined Cortex XDR alerts are exposed on laptop and mobile devices using the Slack interface. Unlike email notifications, Slack channels are dedicated to spaces that you can use to contact specific members regarding your Cortex XD alerts.

To configure a Slack notification, you must first install and configure the Cortex XDR app on Slack.

**STEP 1** | From Cortex XDR, select Settings > Integrations > External Applications.

**STEP 2** | Select the provided link to install Cortex XDR on your Slack workspace.

You are directed to the Slack browser to install the Cortex XDR app. You can only use this link to install Cortex XDR on Slack. Attempting to install from Slack marketplace will redirect you to Cortex XDR documentation.
STEP 3 | Click Submit.
Upon successful installation, Cortex XDR displays the workspace to which you connected.

STEP 4 | Configure Notification Forwarding.
After you integrate with your Slack workspace, you can configure your forwarding settings.
Integrate a Syslog Receiver

To receive Cortex XDR notifications using your Syslog server, you need to define the settings for the Syslog receiver from which you want to send notifications.

**STEP 1** | Before you define the Syslog settings, enable access to the following Cortex XDR IP addresses for your deployment region in your firewall configurations:

- US—35.232.87.9 and 35.224.66.220
- EU—34.90.202.186 and 34.90.105.250

**STEP 2** | Navigate to Settings > Integrations > External Applications.

**STEP 3** | In Syslog Servers, add a + New Server.

**STEP 4** | Define the Syslog server parameters:

- **Name**—Unique name for the server profile.
- **Destination**—IP address or fully qualified domain name (FQDN) of the Syslog server.
- **Port**—The port number on which to send Syslog messages.
- **Facility**—Choose one of the Syslog standard values. The value maps to how your Syslog server uses the facility field to manage messages. For details on the facility field, see RFC 5424.
- **Protocol**—Select a method of communication with the Syslog server:
  - **TCP**—No validation is made on the connection with the Syslog server. However, if an error occurred with the domain used to make the connection, the Test connection will fail.
  - **UDP**—Cortex XDR runs a validation to ensure connection was made with the syslog server.
  - **TCP + SSL**—Cortex XDR validates the syslog server certificate and uses the certificate signature and public key to encrypt the data sent over the connection.
- **Certificate**—The communication between Cortex XDR and the Syslog destination can use TLS. In this case, upon connection, Cortex XDR validates that the Syslog receiver has a certificate signed by either a trusted root CA or a self signed certificate.

  If your syslog receiver uses a self signed CA, Browse and upload your Self Signed Syslog Receiver CA.

  *Make sure the self signed CA includes your public key.*

  If you only use a trusted root CA leave the Certificate field empty.

- **Ignore Certificate Error**—Cortex XDR does not recommend, but you can choose to select this option to ignore certificate errors if they occur. This will forward alerts and logs even if the certificate contains errors.

**STEP 5** | Test the parameters to ensure a valid connection and Create when ready.

You can define up to five Syslog servers. Upon success, the table displays the Syslog servers and their status.
STEP 6 | (Optional) Manage your Syslog server connection.

   In the Syslog Servers table

   • Locate your Syslog server and right-click to Send text message to test the connection.
     Cortex XDR sends a message to the defined Syslog server which you can check to see if the test
     message indeed arrived.
   • Locate the Status field.
     The Status field displays a Valid or Invalid TCP connection. Cortex XDR tests connection with the
     Syslog server every 10min. If no connection is found after 1 hour, Cortex XDR send a notice to the
     Notification Center.

STEP 7 | Configure Notification Forwarding.

   After you integrate with your Syslog receiver, you can configure your forwarding settings.
Configure Notification Forwarding

With Cortex XDR you can choose to receive notifications to keep up with the alerts and events that matter to your teams. To forward notifications, you create a forwarding configuration that specifies the log type you want to forward. You can also add filters to your configuration to send notifications that match specific criteria.

*Cortex XDR applies the filter only to future alerts and events.*

Use this workflow to configure notifications for alerts, agent audit logs, and management audit logs. To receive notifications about reports, see Create a Report from Scratch.

**STEP 1 |** Navigate to ☐ > Settings > Notifications.

**STEP 2 |** + Add Forwarding Configuration.

**STEP 3 |** Define the configuration Name and Description.

**STEP 4 |** Select the Log Type you want to forward, one of the following:

- Alerts—Send notifications for specific alert types (for example, XDR Agent or BIOC).
- Agent Audit Logs—Send notifications for audit logs reported by your Cortex XDR agents.
- Management Audit Logs—Send notifications for audit logs about events related to your Cortex XDR management console.

**STEP 5 |** In the Configuration Scope, Filter the type of information you want included in a notification. For example, set a filter Severity = Medium, Alert Source = XDR Agent. Cortex XDR sends the alerts or events matching this filter as a notification.

**STEP 6 |** Define your Email Configuration.

1. In Email Distribution, add the email addresses to which you want to send email notifications.
2. Define the Email Grouping Time Frame, in minutes, to specify how often Cortex XDR sends notifications. Every 30 alerts or 30 events aggregated within this time frame are sent together in one notification, sorted according to the severity. To send a notification when one alert or event is generated, set the time frame to 0.
3. Choose whether you want Cortex XDR to provide an auto-generated subject.
4. If you previously used the Log Forwarding app and want to continue forwarding logs in the same format, you can Use Legacy Log Format. See Cortex XDR Log Notification Formats.

**STEP 7 |** Configure additional forwarding options:

Depending on the notification integrations supported by the Log Type, configure the desired notification settings.

- Slack notification—Select a Slack channel.

> *Before you can select a Slack channel, you must integrate Cortex XDR with Slack.*

- Syslog receiver—Select a Syslog receiver.
Before you can select a Syslog server, you must Integrate a Syslog Receiver in Cortex XDR app.

STEP 8 | (Optional) To later modify a saved forwarding configuration, right-click the configuration, and Edit, Disable, or Delete it.
Cortex XDR Log Notification Formats

When Cortex XDR alerts and audit logs are forwarded to an external data source, notifications are sent in the following formats. If you prefer Cortex XDR to forward logs in legacy format, you can choose the legacy option in your log forwarding configuration.

- Alert Notification Format
- Agent Audit Log Notification Format
- Management Audit Log Notification Format
- Legacy—Cortex XDR Log Format for IOC and BIOC Alerts
- Legacy—Cortex XDR (formerly Traps) Log Formats

Alert Notification Format

Cortex XDR Agent, BIOC, IOC, Analytics and third-party alerts are forwarded to external data resources according to the following formats.

**Email Account**

Alert notifications are sent to email accounts according to the settings you configured when you Configure Notification Forwarding. If only one alert exists in the queue, a single alert email format is sent. If more than one alert was grouped in the time frame, all the alerts in the queue are forwarded together in a grouped email format. Emails also include an alert code snippet of the fields of the alerts according to the columns in the Alert table.

**Single Alert Email**

```
Email Subject: Alert: <alert_name>
Email Body: Alert Name: Suspicious Process Creation
Alert ID: 2411
Description: Suspicious process creation detected
Severity: High
Source: XDR Agent
Category: Malware
Action: Detected
Host: WIN-RN4A1D7IM6L
Starred: Yes
Alert: https://xdr20apac.xdr.eu.paloaltonetworks.com/alerts/5463 (causality view)
Incident: https://xdr20apac.xdr.eu.paloaltonetworks.com/incident-view/31 (if doesn’t exist - null)
```

**Grouped Alert Email**

```
Email Subject: Alerts: <first_highest_severity_alert> + x others
Email Body: Alert Name: Suspicious Process Creation
Alert ID: 2411
Description: Suspicious process creation detected
Severity: High
Source: XDR Agent
Category: Malware
Action: Detected
Host: WIN-RN4A1D7IM6L
Starred: Yes
Alert: <link to Cortex XDR app alert view>
Incident: <link to Cortex XDR app incident view>
Alert Name: Behavioral Threat Protection
Alert ID: 2412
Description: A really cool detection
Severity: Medium
Source: XDR Agent
Category: Exploit
Action: Prevented
Host: WIN-RN4A1D7IM6L
Starred: Yes
Alert: <link to Cortex XDR app alert view>
Incident: <link to Cortex XDR app incident view>
Notification Name: “My notification policy 2”
Notification Description: “Starred alerts with medium severity”
```
Slack Channel

You can send alert notifications to a single Slack contact or a Slack channel. Notifications are similar to the email format.

Syslog Server

Alert notification forwarded to a Syslog server are sent in a CEF format RF 5425.

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syslog Header</td>
<td>&lt;9&gt;: PRI (considered a priority field)1: version number2020-03-22T07:55:07.964311Z: timestamp of when alert/log was sentcortexxdr: host name</td>
</tr>
<tr>
<td>CEF Header</td>
<td>HEADER/Vendor=&quot;Palo Alto Networks&quot; (as a constant string) HEADER/Device Product=&quot;Cortex XDR&quot; (as a constant string) HEADER/Product Version= Cortex XDR version (2.0/2.1....) HEADER/Severity=severity (informational/low/medium/high) HEADER/Device Event Class ID=alert source HEADER/name =alert name</td>
</tr>
</tbody>
</table>
### CEF Body

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>end=timestamp</td>
<td>end=timestamp</td>
</tr>
<tr>
<td>host=user</td>
<td>host=3D4WRQ2</td>
</tr>
<tr>
<td>user=deviceFacility</td>
<td>user=deviceFacility</td>
</tr>
<tr>
<td>name=category msg=description externalId=alert request=alert</td>
<td>name=Restrictions msg=<a href="https://test.xdr.us.paloaltonetworks.com/alerts/11111">https://test.xdr.us.paloaltonetworks.com/alerts/11111</a></td>
</tr>
<tr>
<td>init=flexString1=starred flexString1Label=&quot;Initiated by&quot;</td>
<td>init=example.exe flexString1Label=&quot;Initiated by&quot;</td>
</tr>
<tr>
<td>flexString2=excluded flexString2Label=&quot;Excluded&quot;</td>
<td>flexString2=example.exe flexString2Label=&quot;Initiator CMD&quot;</td>
</tr>
<tr>
<td>initiated by=cs1Label=&quot;Initiated by&quot;</td>
<td>initiated by=Microsoft CorporationSIGNATURE_SIGNED-</td>
</tr>
<tr>
<td>initiator=cs2Label=&quot;Initiator CMD&quot;</td>
<td>initiator=cs5Label=&quot;CGO CMD&quot;</td>
</tr>
<tr>
<td>cmd=cs3Label=&quot;Signature&quot;</td>
<td>cmd=string.concat(target process sig, target process signer, &quot;-&quot; )</td>
</tr>
<tr>
<td>signature=cs4Label=&quot;CGO name&quot;</td>
<td>signature=Microsoft CorporationSIGNATURE_SIGNED-</td>
</tr>
<tr>
<td>CGO name=cs5</td>
<td>CGO name=targetprocesssignature=</td>
</tr>
<tr>
<td>cmd=cs6Label=&quot;CGO Signature&quot;</td>
<td>cmd=targetprocesssha256=</td>
</tr>
<tr>
<td>CGO Signature=tenantname=</td>
<td>CGO Signature=tenantCDLid=</td>
</tr>
<tr>
<td>tenantname=CSPaccountname=</td>
<td>tenantCDLid=CSPaccountname=</td>
</tr>
<tr>
<td>CSPaccountname=fileHash=</td>
<td>CSPaccountname=filePath=</td>
</tr>
<tr>
<td>fileHash=filePath=</td>
<td>filePath=file path</td>
</tr>
</tbody>
</table>

### Example

3/18/206:22:53.000 PMCEF:0|Palo Alto Networks|Cortex XDR|Cortex XDR x.x |XDR Agent|Example Cortex XDR Alert|5|end=1581471661000 shost=3D4WRQ2 suser=acme\user deviceFacility=None cat=Restrictions externalId=11148 request=https://test.xdr.us.paloaltonetworks.com/alerts/11111 csl=example.exe cs1Label=Initiated by cs2=example.exe cs2Label=Initiator CMD cs3=Microsoft CorporationSIGNATURE_SIGNED- cs3Label=Signature cs4=cmd.exe cs4Label=CGO name cs5=C:\this\is\example.exe /c ""\\host1\files\example.bat" " cs5Label=CGO CMD cs6=Microsoft CorporationSIGNATURE_SIGNED- cs6Label=CGO Signature targetprocesssignature=tenantname=E2ETest3 tenantCDLid=1399816473 CSPaccountname=Palo Alto Networks - PANW-XDR-BETA10 act=Detected (Reported)
Agent Audit Log Notification Format

To forward agent audit logs, you must have either a Cortex XDR Prevent or Cortex XDR Pro per Endpoint license.

Cortex XDR forwards the agent audit log to external data resources according to the following formats.

Email Account

Cortex XDR can forward agent audit log notifications to email accounts.

Syslog Server

Agent audit logs forwarded to a Syslog server are sent in a CEF format RFC 5425 according to the following mapping.

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syslog Header</td>
<td><code>&lt;9&gt;: PRI (considered a priority field)1: version number2020-03-22T07:55:07.964311Z: timestamp of when alert/log was sentcortexxdr: host name</code></td>
</tr>
<tr>
<td>CEF Header</td>
<td><code>HEADER/Vendor=&quot;Palo Alto Networks&quot; (as a constant string)HEADER/Device Product=&quot;Cortex XDR Agent&quot; (as a constant string)HEADER/Device Version=Cortex XDR Agent version (7.0/7.1....)HEADER/Severity=informationalHEADER/Device Event Class ID=&quot;Agent Audit Logs&quot; (as a constant string)HEADER/name = type</code></td>
</tr>
<tr>
<td>CEF Body</td>
<td><code>end=timestampprt=received timecat=categorymsg=descriptiondeviceHostName = domainexternalId = endpoint idshost = endpoint namecs1=xdr agent versioncs1Label=&quot;agentversion&quot; (as a constant string)cs2=subtypescs2Label=&quot;subtype&quot; (as a constant string)cs3=resultcs3Label=&quot;result&quot; (as</code></td>
</tr>
</tbody>
</table>
Management Audit Log Notification Format

Cortex XDR forwards the management audit log to external data sources according to the following formats.

Email Account

Management audit log notifications are forward to email accounts.

Syslog Server

Management Audit logs forwarded to a Syslog server are sent in a CEF format RF 5425 according to the following mapping:

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syslog Header</td>
<td>&lt;9&gt;: PRI (considered a priority field)1: version number2020-03-22T07:55:07.964311Z: timestamp of when alert/log was sent cortexxdr: host name</td>
</tr>
<tr>
<td>CEF Header</td>
<td>HEADER/Vendor=&quot;Palo Alto Networks&quot; (as a constant string) HEADER/Device Product=&quot;Cortex XDR&quot; (as a constant string) HEADER/Device Version= Cortex XDR version</td>
</tr>
<tr>
<td>Section</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>(2.0/2.1....)HEADER/Severity=informational</td>
<td>Event Class ID=&quot;Management Audit Logs&quot; (as a constant string) HEADER/name = type</td>
</tr>
</tbody>
</table>

**CEF Body**

end=timestamps user=user
namecat=categorymsg=description device HostName = host
name externalId = idcs1=email
Label"email" (as a constant string) cs2=subtype cs2Label="subtype" (as a constant string)
cs3=result cs3Label="result" (as a constant string) cs4=reason cs4Label="reason" (as a constant string)

**Example**

3/18/2012:05:17.567 PM<14>1 2020-03-18T12:05:17.567590Z cortexxdr -
-- CEF:0|Palo Alto Networks|Cortex XDR|Cortex XDR x.x |Management Audit Logs|REPORTING|5|user=test end=1584533117501 externalId=5820

cs1Label=email cs1=test@paloaltonetworks.com cs2Label=subtype cs2=Slack
Report cs3Label=result cs3=SUCCESS cs4Label=reason cs4=None msg=Slack report
'scheduled_1584533112442' ID 00 to ['CUXM741BK', 'C01022YU00L', 'CV51Y1E2X',
'CRK3VASN9'] tenantname=test tenantCDLid=11111 CSPaccountname=00000

**Cortex XDR Log Format for IOC and BIOC Alerts**

Cortex XDR™ logs its IOC and BIOC alerts to the Cortex Data Lake. If you configure Cortex XDR to forward logs in legacy format, when alert logs are forwarded from Cortex Data Lake, each log record has the following format:

**Syslog format:**

"/edrData/action_country","/edrData/action_download","/edrData/
action_external_hostname","/edrData/action_external_port","/
edrData/action_file_extension","/edrData/action_file_md5","/
edrData/action_file_name","/edrData/action_file_path","/
edrData/action_file_previous_file_extension","/edrData/
action_file_previous_file_name","/edrData/action_file_previous_file_path","/
edrData/action_file_sha256","/edrData/action_file_size","/edrData/
action_is_injected_thread","/edrData/action_local_ip","/edrData/
action_local_port","/edrData/action_module_base_address","/edrData/
action_module_image_size","/edrData/action_module_is_remote","/
edrData/action_module_is_replay","/edrData/action_module_path","/
edrData/action_module_process_causality_id","/
edrData/action_module_process_image_command_line","/
edrData/action_module_process_image_extension","/
edrData/action_module_process_image_md5","/edrData/
action_module_process_image_name","/edrData/
action_module_process_image_path","/edrData/
action_module_process_instance_id","/edrData/
action_module_process_is_causality_root","/edrData/
action_module_process_os_pid","/edrData/
action_module_process_signature_product","/edrData/
action_module_process_signature_status","/edrData/
action_module_process_signature_vendor","/edrData/
action_network_connection_id","/edrData/action_network_creation_time","/
edrData/action_network_is_ipv6","/edrData/action_process_causality_id","/
edrData/action_process_image_command_line","/edrData/
action_process_image_extension","/edrData/action_process_image_md5","/edrData/
action_process_image_name","/edrData/action_process_image_path","/edrData/
action_process_image_sha256","/edrData/action_process_instance_id","/edrData/
action_process_integrity_level","/edrData/action_process_is_causality_root","/
edrData/action_process_is_replay","/edrData/action_process_is_special","/
edrData/action_process_os_pid","/edrData/action_process_signature_product","/
edrData/action_process_signature_status","/edrData/
action_process_signature_vendor","/edrData/action_proxy","/edrData/
action_registry_data","/edrData/action_registry_file_path","/edrData/
action_registry_key_name","/edrData/action_registry_value_name","/
edrData/action_registry_value_type","/edrData/action_remote_ip","/edrData/
action_remote_port","/edrData/action_remote_process_causality_id","/
edrData/action_remote_process_image_command_line","/
action_remote_process_image_extension","/
edrData/action_remote_process_image_md5","/edrData/
action_remote_process_image_name","/edrData/
action_remote_process_image_path","/edrData/
action_remote_process_image_sha256","/edrData/
action_remote_process_is_causality_root","/edrData/
action_remote_process_signature_product","/edrData/
action_remote_process_signature_status","/edrData/
action_remote_process_signature_vendor","/edrData/
action_remote_process_signature_status","/edrData/
action_remote_process_signature_vendor","/edrData/
action_remote_process_signature_status","/edrData/
action_remote_process_signature_vendor","/edrData/
action_remote_process_signature_status","/edrData/
action_remote_process_signature_vendor","/edrData/
action_remote_process_signature_status","/edrData/
action_remote_process_signature_vendor","/edrData/
action_remote_process_signature_status","/edrData/
action_remote_process_signature_vendor","/edrData/
action_remote_process_signature_status","/edrData/
action_remote_process_signature_vendor","/edrData/
action_remote_process_signature_status","/edrData/
action_remote_process_signature_vendor","/edrData/
action_remote_process_signature_status","/edrData/
action_remote_process_signature_vendor","/edrData/
action_remote_process_signature_status","/edrData/
action_remote_process_signature_vendor","/edrData/
action_remote_process_signature_status","/edrData/
action_remote_process_signature_vendor","/edrData/
action_remote_process_signature_status","/edrData/
action_remote_process_signature_vendor","/edrData/
action_remote_process_signature_status","/edrData/
action_remote_process_signature_vendor","/edrData/
action_remote_process_signature_status","/edrData/
action_remote_process_signature_vendor","/edrData/
action_remote_process_signature_status","/edrData/
action_remote_process_signature_vendor","/edrData/
action_remote_process_signature_status","/edrData/
action_remote_process_signature_vendor","/edrData/
action_remote_process_signature_status","/edrData/
action_remote_process_signature_vendor","/edrData/
action_remote_process_signature_status","/edrData/
action_remote_process_signature_vendor","/edrData/
action_remote_process_signature_status","/edrData/
action_remote_process_signature_vendor","/edrData/
action_remote_process_signature_status","/edrData/
action_remote_process_signature_vendor","/edrData/
action_remote_process_signature_status","/edrData/
action_remote_process_signature_vendor","/edrData/
action_remote_process_signature_status","/edrData/
action_remote_process_signature_vendor","/edrData/
action_remote_process_signature_status","/edrData/
action_remote_process_signature_vendor","/edrData/
action_remote_process_signature_status","/edrData/
action_remote_process_signature_vendor","/edrData/
action_remote_process_signature_status","/edrData/
action_remote_process_signature_vendor","/edrData/
action_remote_process_signature_status","/edrData/
action_remote_process_signature_vendor","/edrData/
action_remote_process_signature_status","/edrData/
action_remote_process_signature_vendor","/edrData/
action_remote_process_signature_status","/edrData/
action_remote_process_signature_vendor","/edrData/
action_remote_process_signature_status","/edrData/
action_remote_process_signature_vendor","/edrData/
action_remote_process_signature_status","/edrData/
action_remote_process_signature_vendor","/edrData/
action_remote_process_signature_status",}
When alert logs are forwarded by email, each field is labeled, one line per field:

**Email body format example:**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>edrData/action_country</td>
<td></td>
</tr>
<tr>
<td>edrData/action_download</td>
<td></td>
</tr>
<tr>
<td>edrData/action_externalHostname</td>
<td></td>
</tr>
<tr>
<td>edrData/action_external_port</td>
<td></td>
</tr>
<tr>
<td>edrData/action_file_extension</td>
<td>pdf</td>
</tr>
<tr>
<td>edrData/action_file_md5</td>
<td>null</td>
</tr>
<tr>
<td>edrData/action_file_name</td>
<td>XORXOR2614081980.pdf</td>
</tr>
<tr>
<td>edrData/action_file_path</td>
<td>C:\ProgramData\Cyvera\Ransomware\1606798769637268494\XORXOR2614081980.pdf</td>
</tr>
<tr>
<td>edrData/action_file_previous_file_extension</td>
<td>null</td>
</tr>
<tr>
<td>edrData/action_file_previous_file_name</td>
<td>null</td>
</tr>
<tr>
<td>edrData/action_file_previous_file_path</td>
<td>null</td>
</tr>
<tr>
<td>edrData/action_file_sha256</td>
<td>null</td>
</tr>
<tr>
<td>edrData/action_file_size</td>
<td>0</td>
</tr>
<tr>
<td>edrData/action_file_remote_ip</td>
<td>null</td>
</tr>
<tr>
<td>edrData/action_file_remote_port</td>
<td>null</td>
</tr>
<tr>
<td>edrData/action_is_injected_thread</td>
<td></td>
</tr>
<tr>
<td>edrData/action_local_ip</td>
<td></td>
</tr>
<tr>
<td>edrData/action_local_port</td>
<td></td>
</tr>
<tr>
<td>edrData/action_module_base_address</td>
<td></td>
</tr>
<tr>
<td>edrData/action_module_image_size</td>
<td></td>
</tr>
<tr>
<td>edrData/action_module_is_remote</td>
<td></td>
</tr>
<tr>
<td>edrData/action_module_is_replay</td>
<td></td>
</tr>
<tr>
<td>edrData/action_module_path</td>
<td></td>
</tr>
<tr>
<td>edrData/action_module_process_causality_id</td>
<td></td>
</tr>
<tr>
<td>edrData/action_module_process_image_command_line</td>
<td></td>
</tr>
<tr>
<td>edrData/action_module_process_image_extension</td>
<td></td>
</tr>
<tr>
<td>edrData/action_module_process_image_md5</td>
<td></td>
</tr>
<tr>
<td>edrData/action_module_process_image_name</td>
<td></td>
</tr>
<tr>
<td>edrData/action_module_process_image_path</td>
<td></td>
</tr>
<tr>
<td>edrData/action_module_process_image_sha256</td>
<td></td>
</tr>
<tr>
<td>edrData/action_module_process_instance_id</td>
<td></td>
</tr>
<tr>
<td>edrData/action_module_process_is_causality_root</td>
<td></td>
</tr>
<tr>
<td>edrData/action_module_process_os_pid</td>
<td></td>
</tr>
<tr>
<td>edrData/action_module_process_signature_product</td>
<td></td>
</tr>
<tr>
<td>edrData/action_module_process_signature_status</td>
<td></td>
</tr>
<tr>
<td>edrData/action_module_process_signature_vendor</td>
<td></td>
</tr>
<tr>
<td>edrData/action_module_process_signature_vendor</td>
<td></td>
</tr>
<tr>
<td>edrData/action_network_connection_id</td>
<td></td>
</tr>
<tr>
<td>edrData/action_network_creation_time</td>
<td></td>
</tr>
</tbody>
</table>
edrData/action_network_is_ipv6:
edrData/action_process_causality_id:
edrData/action_process_image_command_line:
edrData/action_process_image_extension:
edrData/action_process_image_md5:
edrData/action_process_image_name:
edrData/action_process_image_path:
edrData/action_process_image_sha256:
edrData/action_process_instance_id:
edrData/action_process_integrity_level:
edrData/action_process_is_causality_root:
edrData/action_process_is_replay:
edrData/action_process_is_special:
edrData/action_process_os_pid:
edrData/action_process_signature_product:
edrData/action_process_signature_status:
edrData/action_process_signature_vendor:
edrData/action_process_signature:
edrData/action_proxy:
edrData/action_registry_data:
edrData/action_registry_file_path:
edrData/action_registry_key_name:
edrData/action_registry_value_name:
edrData/action_registry_value_type:
edrData/action_remote_ip:
edrData/action_remote_port:
edrData/action_remote_process_causality_id:
edrData/action_remote_process_image_command_line:
edrData/action_remote_process_image_extension:
edrData/action_remote_process_image_md5:
edrData/action_remote_process_image_name:
edrData/action_remote_process_image_path:
edrData/action_remote_process_image_sha256:
edrData/action_remote_process_is_causality_root:
edrData/action_remote_process_os_pid:
edrData/action_remote_process_signature_product:
edrData/action_remote_process_signature_status:
edrData/action_remote_process_signature_vendor:
edrData/action_remote_process_thread_id:
edrData/action_remote_process_thread_start_address:
edrData/action_thread_thread_id:
edrData/action_total_download:
edrData/action_total_upload:
edrData/action_upload:
edrData/action_user_status:
edrData/action_username:
edrData/actor_causality_id: AdUcamNT99kAAAAEAAAAAA==
edrData/actor_effective_user_sid: S-1-5-18
edrData/actor_effective_username: NT AUTHORITY\SYSTEM
edrData/actor_is_injected_thread: false
edrData/actor_primary_user_sid: S-1-5-18
edrData/actor_primary_username: NT AUTHORITY\SYSTEM
edrData/actor_process_causality_id: AdUcamNT99kAAAAEAAAAAA==
edrData/actor_process_command_line:
edrData/actor_process_execution_time: 1559827133585
edrData/actor_process_image_command_line:
edrData/actor_process_image_extension:
edrData/actor_process_image_md5:
edrData/actor_process_image_name: System
edrData/actor_process_image_path: System
edrData/actor_process_image_sha256:
edrData/actor_process_instance_id: AdUcamNT99kAAAAEAAAAAA==
edrData/actor_process_integrity_level: 16384
edrData/actor_process_is_special: 1
edrData/actor_process_os_pid: 4
edrData/actor_process_signature_product: Microsoft Windows
edrData/actor_process_signature_status: 1
edrData/actor_process_signature_vendor: Microsoft Corporation
edrData/actor_thread_thread_id: 64
edrData/agent_content_version: 58-9124
edrData/agent_host_boot_time: 1559827133585
edrData/agent_hostname: padme-7
edrData/agent_id: a832f35013f16a06fc2495843674a3e9
edrData/agent_ip_addresses: ["10.196.172.74"]
edrData/agent_is_vdi: false
edrData/agent_os_sub_type: Windows 7 [6.1 (Build 7601: Service Pack 1)]
edrData/agent_os_type: 1
edrData/agent_session_start_time: 1559827592661
edrData/agent_version: 6.1.0.13895
edrData/causality_actor_causality_id: AdUcamNT99kAAAAEAAAAA==
edrData/causality_actor_effective_user_sid:
edrData/causality_actor_effective_username:
edrData/causality_actor_primary_user_sid: S-1-5-18
edrData/causality_actor_primary_username: NT AUTHORITY\SYSTEM
edrData/causality_actor_process_causality_id:
edrData/causality_actor_process_command_line:
edrData/causality_actor_process_execution_time: 1559827133585
edrData/causality_actor_process_image_command_line:
edrData/causality_actor_process_image_extension:
edrData/causality_actor_process_image_md5:
edrData/causality_actor_process_image_name: System
edrData/causality_actor_process_image_path: System
edrData/causality_actor_process_image_sha256:
edrData/causality_actor_process_instance_id: AdUcamNT99kAAAAEAAAAA==
edrData/causality_actor_process_integrity_level: 16384
edrData/causality_actor_process_is_special: 1
edrData/causality_actor_process_os_pid: 4
edrData/causality_actor_process_signature_product: Microsoft Windows
edrData/causality_actor_process_signature_status: 1
edrData/causality_actor_process_signature_vendor: Microsoft Corporation
edrData/event_id: AAABa13u2PQsqXnCAB1qjw==
edrData/event_is_simulated: false
edrData/event_sub_type: 1
edrData/event_timestamp: 1560649063308
edrData/event_type: 3
edrData/event_utc_diff_minutes: 120
edrData/event_version: 20
edrData/host_metadata_hostname:
edrData/missing_action_remote_process_instance_id:
facility:
generatedTime: 2019-06-16T01:37:43
recordType: alert
recesize: trapsId:
uuid:
xdr_unique_id: ae65c92c6e704023df129c728eab3d3e
meta_internal_id: None
external_id: 318b7f91-ae74-4860-abd1-b463e8cd6deb
is_visible: null
is_secdo_event: null
severity: SEV_010_INFO
alert_source: BIOC
internal_id: None
matching_status: null
local_insert_ts: null
The following table summarizes the field prefixes and additional relevant fields available for BIOC and IOC alert logs.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>/edrData/action_file*</td>
<td>Fields that begin with this prefix describe attributes of a file for which Traps reported activity.</td>
</tr>
<tr>
<td>edrData/action_module*</td>
<td>Fields that begin with this prefix describe attributes of a module for which Traps reported module loading activity.</td>
</tr>
<tr>
<td>edrData/action_module_process*</td>
<td>Fields that begin with this prefix describe attributes and activity related to processes reported by Traps that load modules such as DLLs on the endpoint.</td>
</tr>
<tr>
<td>edrData/action_process_image*</td>
<td>Fields that begin with this prefix describe attributes of a process image for which Traps reported activity.</td>
</tr>
<tr>
<td>edrData/action_registry*</td>
<td>Fields that begin with this prefix describe registry activity and attributes such as key name, data, and previous value for which Traps reported activity.</td>
</tr>
<tr>
<td>edrData/action_network</td>
<td>Fields that begin with this prefix describe network attributes for which Traps reported activity.</td>
</tr>
</tbody>
</table>
### Field Name | Definition
--- | ---
edrData/action_remote_process* | Fields that begin with this prefix describe attributes of remote processes for which Traps reported activity.
edrData/actor* | Fields that begin with this prefix describe attributes about the acting user that initiated the activity on the endpoint.
edrData/agent* | Fields that begin with this prefix describe attributes about the Traps agent deployed on the endpoint.
edrData/causality_actor* | Fields that begin with this prefix describe attributes about the causality group owner.

**Additional useful fields:**

- **/severity**
  - Severity assigned to the alert:
    - SEV_010_INFO
    - SEV_020_LOW
    - SEV_030_MEDIUM
    - SEV_040_HIGH
    - SEV_090_UNKNOWN

- **/alert_source**
  - Source of the alert: BIOC or IOC

- **/local_insert_ts**
  - Date and time when Cortex XDR – Investigation and Response ingested the app.

- **/source_insert_ts**
  - Date and time the alert was reported by the alert source.

- **/alert_name**
  - If the alert was generated by Cortex XDR – Investigation and Response, the alert name will be the specific Cortex XDR rule that created the alert (BIOC or IOC rule name). If from an external system, it will carry the name assigned to it by Cortex XDR.

- **/alert_category**
  - Alert category based on the alert source:
    - BIOC alert categories:
      - OTHER
      - PERSISTENCE
      - EVASION
      - TAMPERING
      - FILE_TYPE_OBFUSCATION
      - PRIVILEGE_ESCALATION
      - CREDENTIAL_ACCESS
      - LATERAL_MOVEMENT
      - EXECUTION
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>/alert_description</td>
<td>Text summary of the event including the alert source, alert name, severity, and file path. For alerts triggered by BIOC and IOC rules, Cortex XDR displays detailed information about the rule.</td>
</tr>
<tr>
<td>/bioc_indicator</td>
<td>A JSON representation of the rule characteristics. For example:</td>
</tr>
<tr>
<td></td>
<td>```json</td>
</tr>
</tbody>
</table>
|                            | [{"pretty_name":"File","data_type":null,"render_type":"entity","entity_map":null},
|                            | {"pretty_name":"action type","data_type":null,"render_type":"attribute","entity_map":null},
|                            | {"pretty_name":"=","data_type":null,"render_type":"operator","entity_map":null},
|                            | {"pretty_name":"all","data_type":null,"render_type":"value","entity_map":null},
|                            | {"pretty_name":"AND","data_type":null,"render_type":"connector","entity_map":null},
|                            | {"pretty_name":"name","data_type":"TEXT","render_type":"attribute","entity_map":attributes}],
|                            | {"pretty_name":"=","data_type":null,"render_type":"operator","entity_map":attributes}],
<p>|                            | {&quot;pretty_name&quot;:&quot;*.pdf&quot;,&quot;data_type&quot;:null,&quot;render_type&quot;:&quot;value&quot;,&quot;entity_map&quot;:attributes}]``` |
| /bioc_category_enum_key    | Alert category based on the alert source. An example of a BIOC alert category is Evasion. An example of a Traps alert category is Exploit Modules. |</p>
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>/alert_action_status</td>
<td>Action taken by the alert sensor with action status displayed in parenthesis:</td>
</tr>
<tr>
<td></td>
<td>• Detected</td>
</tr>
<tr>
<td></td>
<td>• Detected (Download)</td>
</tr>
<tr>
<td></td>
<td>• Detected (Post Detected)</td>
</tr>
<tr>
<td></td>
<td>• Detected (Prompt Allow)</td>
</tr>
<tr>
<td></td>
<td>• Detected (Reported)</td>
</tr>
<tr>
<td></td>
<td>• Detected (Scanned)</td>
</tr>
<tr>
<td></td>
<td>• Prevented (Blocked)</td>
</tr>
<tr>
<td></td>
<td>• Prevented (Prompt Block)</td>
</tr>
<tr>
<td>/case_id</td>
<td>Unique identifier for the incident.</td>
</tr>
<tr>
<td>/global_content_version_id</td>
<td>Unique identifier for the content version in which a Palo Alto Networks global BIOC rule was released.</td>
</tr>
<tr>
<td>/global_rule_id</td>
<td>Unique identifier for an alert triggered by a Palo Alto Networks global BIOC rule.</td>
</tr>
<tr>
<td>/is_whitelisted</td>
<td>Boolean indicating whether the alert is excluded or not.</td>
</tr>
</tbody>
</table>

**Cortex XDR Analytics Log Format**

Cortex XDR™ Analytics logs its alerts to the Cortex Data Lake as analytics alert logs. If you configure Cortex XDR to forward logs in legacy format, each log record has the following format:

**Syslog format:**

sub_type, time_generated, id, version_info/document_version, version_info/magnifier_version, version_info/detection_version, alert/url, alert/category, alert/type, alert/name, alert/description/html, alert/description/text, alert/severity, alert/state, alert/is_whitelisted, alert/ports, alert/internal_destinations/single_destinations, alert/internal_destinations/ip_ranges, alert/external_destinations, alert/app_id, alert/schedule/activity_first_seen_at, alert/schedule/activity_last_seen_at, alert/schedule/first_detected_at, alert/schedule/last_detected_at, user/user_name, user/url, user/display_name, user/org_unit, device/id, device/url, device/mac, device/hostname, device/ip, device/ip_ranges, device/owner, device/org_unit, files

**Email body format example:**

When analytics alert logs are forwarded by email, each field is labeled, one line per field:

```
sub_type: Update
time_generated: 1547717480
id: 4
version_info/document_version: 1
version_info/magnifier_version: 1.8
version_info/detection_version: 2019.2.0rc1
alert/url: https://\/\/ddc1...
alert/category: Recon
```
The following table describes each field:

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>sub_type</td>
<td>Alert log subtype. Values are:</td>
</tr>
<tr>
<td></td>
<td>• New—First log record for the alert with this record id.</td>
</tr>
<tr>
<td></td>
<td>• Update—Log record identifies an update to a previously logged alert.</td>
</tr>
<tr>
<td></td>
<td>• StateOnlyUpdate—Alert state is updated. For internal use only.</td>
</tr>
<tr>
<td>time_generated</td>
<td>Time the log record was sent to the Cortex Data Lake. Value is a Unix Epoch timestamp.</td>
</tr>
<tr>
<td>id</td>
<td>Unique identifier for the alert. Any given alert can generate multiple log records—one when</td>
</tr>
<tr>
<td></td>
<td>the alert is initially raised, and then additional records every time the alert status changes.</td>
</tr>
<tr>
<td></td>
<td>This ID remains constant for all such alert records.</td>
</tr>
<tr>
<td></td>
<td>You can obtain the current status of the alert by looking for log records with this id and the</td>
</tr>
<tr>
<td></td>
<td>most recent alert/schedule/last_detected_at timestamp.</td>
</tr>
<tr>
<td>Field Name</td>
<td>Definition</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>version_info/document_version</td>
<td>Identifies the log schema version number used for this log record.</td>
</tr>
<tr>
<td>version_info/magnifier_version</td>
<td>The version number of the Cortex XDR – Analytics instance that wrote this log record.</td>
</tr>
<tr>
<td>version_info/detection_version</td>
<td>Identifies the version of the Cortex XDR – Analytics detection software used to raise the alert.</td>
</tr>
<tr>
<td>alert/url</td>
<td>Provides the full URL to the alert page in the Cortex XDR – Analytics user interface.</td>
</tr>
<tr>
<td>alert/category</td>
<td>Identifies the alert category, which is a reflection of the anomalous network activity location in the attack life cycle. Possible categories are:</td>
</tr>
<tr>
<td></td>
<td>• <strong>C&amp;C</strong>—The network activity is possibly the result of malware attempting to connect to its Command &amp; Control server.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Exfiltration</strong>—A large amount of data is being transferred to an endpoint that is external to the network.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Lateral</strong>—The network activity is indicative of an attacker who is attempting to move from one endpoint to another on the network.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Malware</strong>—A file has been discovered on an endpoint that is probably malware or riskware. Malware alerts can also be raised based on network activity that is indicative of automated malicious traffic generation.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Recon</strong>—The network activity is indicative of an attacker that is exploring the network for endpoints and other resources to attack.</td>
</tr>
<tr>
<td>alert/type</td>
<td>Identifies the categorization to which the alert belongs. For example <strong>Tunneling Process, Sandbox Detection, Malware</strong>, and so forth.</td>
</tr>
<tr>
<td>alert/name</td>
<td>The alert name as it appears in the Cortex XDR – Analytics user interface.</td>
</tr>
<tr>
<td>alert/description/html</td>
<td>The alert textual description in HTML formatting.</td>
</tr>
<tr>
<td>alert/description/text</td>
<td>The alert textual description in plain text.</td>
</tr>
<tr>
<td>alert/severity</td>
<td>Identifies the alert severity. These severities indicate the likelihood that the anomalous network activity is a real attack.</td>
</tr>
<tr>
<td></td>
<td>• <strong>High</strong>—The alert is confirmed to be a network attack.</td>
</tr>
<tr>
<td>Field Name</td>
<td>Definition</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>alert/state</td>
<td>Identifies the alert state.</td>
</tr>
<tr>
<td>• Medium—The alert is suspicious enough to</td>
<td>require additional investigation.</td>
</tr>
<tr>
<td>• Low—The alert is unverified. Whether the</td>
<td>alert is indicative of a network attack is unknown.</td>
</tr>
<tr>
<td>$alert/state$</td>
<td></td>
</tr>
<tr>
<td>alert/is_whitelisted</td>
<td>Indicates whether the alert is whitelisted. <strong>Whitelisting</strong> indicates that anomalous-appearing network activity is legitimate. If an alert is whitelisted, then it is not visible in the Cortex XDR Analytics user interface. Alerts can be dismissed or archived and still have a whitelist rule.</td>
</tr>
<tr>
<td>alert/ports</td>
<td>List of ports accessed by the network entity during its anomalous behavior.</td>
</tr>
<tr>
<td>alert/internal_destinations/single_destinations</td>
<td>Network destinations that the entity reached, or tried to reach, during the course of the network activity that caused Cortex XDR – Analytics to raise the alert. This field contains a sequence of JSON objects, each of which contains the following fields:</td>
</tr>
<tr>
<td>• ip—The destination IP address.</td>
<td></td>
</tr>
<tr>
<td>• name—The destination name (for example, a host name).</td>
<td></td>
</tr>
<tr>
<td>alert/internal_destinations/ip_ranges</td>
<td>IP address range subnets that the entity reached, or tried to reach, during the course of the network activity that caused Cortex XDR – Analytics to raise the alert. This field contains a sequence of JSON objects, each of which contains the following fields:</td>
</tr>
<tr>
<td>• max_ip—Last IP address in the subnet.</td>
<td></td>
</tr>
<tr>
<td>• min_ip—First IP address in the subnet.</td>
<td></td>
</tr>
<tr>
<td>• name—Subnet name.</td>
<td></td>
</tr>
<tr>
<td>Field Name</td>
<td>Definition</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>alert/external_destinations</td>
<td>Provides a list of destinations external to the monitored network that the entity tried to reach, or actually reached, during the activity that raised this alert. This list can contain IP addresses or fully qualified domain names.</td>
</tr>
<tr>
<td>alert/app_id</td>
<td>The App-ID associated with this alert.</td>
</tr>
<tr>
<td>alert/schedule/activity_first_seen_at</td>
<td>Time when Cortex XDR – Analytics first detected the network activity that caused it to raise the alert. Be aware that there is frequently a delay between this timestamp, and the time when Cortex XDR – Analytics raises an alert (see the alert/schedule/first_detected_at field).</td>
</tr>
<tr>
<td>alert/schedule/activity_last_seen_at</td>
<td>Time when Cortex XDR – Analytics last detected the network activity that caused it to raise the alert.</td>
</tr>
<tr>
<td>alert/schedule/first_detected_at</td>
<td>Time when Cortex XDR – Analytics first alerted on the network activity.</td>
</tr>
<tr>
<td>alert/schedule/last_detected_at</td>
<td>Time when Cortex XDR – Analytics last alerted on the network activity.</td>
</tr>
<tr>
<td>user/user_name</td>
<td>The name of the user associated with this alert. This name is obtained from Active Directory.</td>
</tr>
<tr>
<td>user/url</td>
<td>Provides the full URL to the user page in the Cortex XDR – Analytics user interface for the user who is associated with the alert.</td>
</tr>
<tr>
<td>user/display_name</td>
<td>The user name as retrieved from Active Directory. This is the user name displayed within the Cortex XDR – Analytics user interface for the user who is associated with this alert.</td>
</tr>
<tr>
<td>user/org_unit</td>
<td>The organizational unit of the user associated with this alert, as identified using Active Directory.</td>
</tr>
<tr>
<td>device/id</td>
<td>A unique ID assigned by Cortex XDR – Analytics to the device. All alerts raised due to activity occurring on this endpoint will share this ID.</td>
</tr>
<tr>
<td>device/url</td>
<td>Provides the full URL to the device page in the Cortex XDR – Analytics user interface.</td>
</tr>
<tr>
<td>device/mac</td>
<td>The MAC address of the network card in use on the device.</td>
</tr>
<tr>
<td>device/hostname</td>
<td>The device host name.</td>
</tr>
<tr>
<td>Field Name</td>
<td>Definition</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>device/ip</td>
<td>The device IP address.</td>
</tr>
</tbody>
</table>
| device/ip_ranges| Identifies the subnet or subnets that the device is on. This sequence can contain multiple inclusive subnets. Each element in this sequence is a JSON object with the following fields:  
  • asset—The asset name assigned to the device from within the Cortex XDR Analytics user interface.  
  • max_ip—Last IP address in the subnet.  
  • min_ip—First IP address in the subnet.  
  • name—Subnet name. |
| device/owner    | The user name of the person who owns the device.                                                                                             |
| device/org_unit | The organizational unit that owns the device, as identified by Active Directory.                                                            |
| files           | Identifies the files associated with the alert. Each element in this sequence is a JSON object with the following fields:  
  • full_path—The file full path (including the file name).  
  • md5—The file MD5 hash. |

**Cortex XDR Log Formats**

The following topics list the fields of each Cortex XDR log type that the Cortex Data Lake app can forward to an external server or email destination.

With log forwarding to a syslog receiver, the Cortex Data Lake sends logs in the IETF syslog message format defined in RFC 5425. To facilitate parsing, the delimiter is a comma and each field is a comma-separated value (CSV) string. The FUTURE_USE tag applies to fields that Cortex XDR does not currently implement.

With log forwarding to an email destination, the Cortex Data Lake sends an email with each field on a separate line in the email body.

- Threat Logs
- Config Logs
- Analytics Logs
- System Logs

**Threat Logs**

**Syslog format:** recordType, class, FUTURE_USE, eventType, generatedTime, serverTime, agentTime, tzOffset, FUTURE_USE, facility, customerId, trapsId, serverHost, serverComponentVersion, regionId, isEndpoint, agentId, osType, isVdi, osVersion, is64, agentIp, deviceName, deviceDomain, severity, trapsSeverity, agentVersion, contentVersion, protectionStatus, preventionKey, moduleId, profile, moduleStatusId, verdict, preventionMode, terminate, terminateTarget, quarantine, block, postDetected, eventParameters(Array), sourceProcessIdx(Array), targetProcessIdx(Array), fileIdx(Array), processes(Array), files(Array), users(Array), urls(Array), description(Array)
Email body format example:

```plaintext
recordType: threat
messageData/class: threat
messageData/subClass: 
eventType: AgentSecurityEvent
generatedTime: 2019-01-29T05:58:045-08:00
endPointHeader/agentTime: 2018-07-02T20:01:03Z
endPointHeader/tzOffset: 180
product:
  facility: TrapsAgent
customerId: 245143
trapsId: mac510a2monday-01
serverHost: coreop-gaauta-2606-0-112132729246-266
serverComponentVersion: 2.0.2
regionId: 70
isEndpoint: 1
agentId: dc3af3198f172048082c21ff09566b
endPointHeader/osType: 2
endPointHeader/isVdi: 0
endPointHeader/osVersion: 10.11.6
endPointHeader/is64: 1
endPointHeader/agentIp: 10.200.37.201
endPointHeader/deviceName: A1260700MC1011
endPointHeader/deviceDomain:
severity: emergency
messageData/trapsSeverity: medium
endPointHeader/agentVersion: 5.1.0.1401
endPointHeader/contentVersion: 26-3625
endPointHeader/protectionStatus: 0
messageData/preventionKey: 9a94965188d2455486dd8d60cf4b3849
messageData/moduleId: COMPONENT_EPM_J01
messageData/profile: ExploitModules
messageData/moduleStatusId: CYSTATUS_JIT_EXCEPTION
messageData/verdict:
messageData/preventionMode: blocked
messageData/terminate: 1
messageData/terminateTarget: quarantine:
messageData/block: 0
messageData/postDetected: 0
messageData/eventParameters: 
  
messageData/sourceProcessIdx: 0
messageData/targetProcessIdx: -1
messageData/fileIdx: 0
messageData/processes: 
  
messageData/files:
  
messageData/users: 
messageData/urls: []
messageData/description: Memory Corruption Exploit
```
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>recordType</td>
<td>Record type associated with the event and that you can use when managing logging quotas. In this case, the record type is threat which includes logs related to security events that occur on the endpoints.</td>
</tr>
<tr>
<td>class</td>
<td>Class of Cortex XDR agent log: config, policy, system, or agent_log.</td>
</tr>
<tr>
<td>generatedTime</td>
<td>Coordinated Universal Time (UTC) equivalent of the time at which an event was logged. For agent events, this represents the time on the endpoint. For policy, configuration, and system events, this represents the time on Cortex XDR in ISO-8601 string representation (for example, 2017-01-24T09:08:59Z).</td>
</tr>
<tr>
<td>serverTime</td>
<td>Coordinated Universal Time (UTC) equivalent of the time at which the server generated the log. If the log was generated on an endpoint, this field identifies the time the server received the log in ISO-8601 string representation (for example, 2017-01-24T09:08:59Z).</td>
</tr>
<tr>
<td>agentTime</td>
<td>Coordinated Universal Time (UTC) equivalent of the time at which an agent logged an event in ISO-8601 string representation.</td>
</tr>
<tr>
<td>tzOffset</td>
<td>Effective endpoint time zone offset from UTC, in minutes.</td>
</tr>
<tr>
<td>facility</td>
<td>The Cortex XDR system component that initiated the event, for example: TrapsAgent, TrapsServiceCore, TrapsServiceManagement, and TrapsServiceBackend.</td>
</tr>
<tr>
<td>customerId</td>
<td>The ID that uniquely identifies the Cortex Data Lake instance which received this log record.</td>
</tr>
<tr>
<td>trapsId</td>
<td>Tenant external ID.</td>
</tr>
<tr>
<td>serverHost</td>
<td>Hostname of Cortex XDR.</td>
</tr>
<tr>
<td>serverComponentVersion</td>
<td>Software version of Cortex XDR.</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>regionId</td>
<td>ID of Cortex XDR region:</td>
</tr>
<tr>
<td></td>
<td>• 10—Americas (N. Virginia)</td>
</tr>
<tr>
<td></td>
<td>• 70—EMEA (Frankfurt)</td>
</tr>
<tr>
<td>isEndpoint</td>
<td>Indicates whether the event occurred on an endpoint.</td>
</tr>
<tr>
<td></td>
<td>• 0—No, host is not an endpoint.</td>
</tr>
<tr>
<td></td>
<td>• 1—Yes, host is an endpoint.</td>
</tr>
<tr>
<td>agentId</td>
<td>Unique identifier for the Cortex XDR agent.</td>
</tr>
<tr>
<td>osType</td>
<td>Operating system of the endpoint:</td>
</tr>
<tr>
<td></td>
<td>• 1—Windows</td>
</tr>
<tr>
<td></td>
<td>• 2—OS X/macOS</td>
</tr>
<tr>
<td></td>
<td>• 3—Android</td>
</tr>
<tr>
<td></td>
<td>• 4—Linux</td>
</tr>
<tr>
<td>isVdi</td>
<td>Indicates whether the endpoint is a virtual desktop infrastructure (VDI):</td>
</tr>
<tr>
<td></td>
<td>• 0—The endpoint is not a VDI</td>
</tr>
<tr>
<td></td>
<td>• 1—The endpoint is a VDI</td>
</tr>
<tr>
<td>osVersion</td>
<td>Full version number of the operating system running on the endpoint. For</td>
</tr>
<tr>
<td></td>
<td>example, 6.1.7601.19135.</td>
</tr>
<tr>
<td>is64</td>
<td>Indicates whether the endpoint is running a 64-bit version of Windows:</td>
</tr>
<tr>
<td></td>
<td>• 0—The endpoint is not running x64 architecture</td>
</tr>
<tr>
<td></td>
<td>• 1—The endpoint is running x64 architecture</td>
</tr>
<tr>
<td>agentIp</td>
<td>IP address of the endpoint.</td>
</tr>
<tr>
<td>deviceName</td>
<td>Hostname of the endpoint on which the event was logged.</td>
</tr>
<tr>
<td>deviceDomain</td>
<td>Domain to which the endpoint belongs.</td>
</tr>
<tr>
<td>severity</td>
<td>Syslog severity level associated with the event.</td>
</tr>
<tr>
<td></td>
<td>• 2—Critical. Used for events that require immediate attention.</td>
</tr>
<tr>
<td></td>
<td>• 3—Error. Used for events that require special handling.</td>
</tr>
<tr>
<td></td>
<td>• 4—Warning. Used for events that sometimes require special handling.</td>
</tr>
<tr>
<td></td>
<td>• 5—Notice. Used for normal but significant events that can require attention.</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>trapsSeverity</td>
<td>Severity level associated with the event defined for Cortex XDR. Each of these severities corresponds to a syslog severity level:</td>
</tr>
<tr>
<td></td>
<td>• 0—Informational. Informational messages that do not require attention. Identical to the syslog 6 (Informational) severity level.</td>
</tr>
<tr>
<td></td>
<td>• 1—Low. Used for normal but significant events that can require attention. Corresponds to the syslog 5 (Notice) severity level.</td>
</tr>
<tr>
<td></td>
<td>• 2—Medium. Used for events that sometimes require special handling. Corresponds to the syslog 4 (Warning) severity level.</td>
</tr>
<tr>
<td></td>
<td>• 3—High. Used for events that require special handling. Corresponds to the syslog 3 (Error) severity level.</td>
</tr>
<tr>
<td></td>
<td>• 4—Critical. Used for events that require immediate attention. Corresponds to the syslog 2 (Critical) severity level.</td>
</tr>
<tr>
<td></td>
<td>See also the severity log field.</td>
</tr>
<tr>
<td>agentVersion</td>
<td>Version of the Cortex XDR agent.</td>
</tr>
<tr>
<td>contentVersion</td>
<td>Content version in the local security policy.</td>
</tr>
<tr>
<td>protectionStatus</td>
<td>Cortex XDR agent protection status:</td>
</tr>
<tr>
<td></td>
<td>• 0—Protected</td>
</tr>
<tr>
<td></td>
<td>• 1—OsVersionIncompatible</td>
</tr>
<tr>
<td></td>
<td>• 2—AgentIncompatible</td>
</tr>
<tr>
<td>preventionKey</td>
<td>Unique identifier for security events.</td>
</tr>
<tr>
<td>moduleld</td>
<td>Security module name.</td>
</tr>
<tr>
<td>profile</td>
<td>Name of the security profile that triggered the event.</td>
</tr>
<tr>
<td>moduleStatusId</td>
<td>Identifies the specific component of Cortex XDR modules.</td>
</tr>
<tr>
<td></td>
<td>• CYSTATUS_ABNORMAL_PROCESS_TERMINATION</td>
</tr>
<tr>
<td></td>
<td>• CYSTATUS_ALIGNED_HEAP_SPRAY_DETECTED</td>
</tr>
<tr>
<td></td>
<td>• CYSTATUS_CHILD_PROCESS_BLOCKED</td>
</tr>
<tr>
<td></td>
<td>• CYSTATUS_CORE_LIBRARY_LOADED</td>
</tr>
<tr>
<td></td>
<td>• CYSTATUS_CORE_LIBRARY_UNLOADING</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CYSTATUS_CPLPROT_BLACKLIST</td>
<td></td>
</tr>
<tr>
<td>CYSTATUS_CPLPROT_REMOTE_DRIVE</td>
<td></td>
</tr>
<tr>
<td>CYSTATUS_CPLPROT_REMOVABLE_DRIVE</td>
<td></td>
</tr>
<tr>
<td>CYSTATUS_CYINJCT_DISPATCH</td>
<td></td>
</tr>
<tr>
<td>CYSTATUS_CYINJCT_MAPPING</td>
<td></td>
</tr>
<tr>
<td>CYSTATUS_CYVERA_PREVENTION</td>
<td></td>
</tr>
<tr>
<td>CYSTATUS_DANGEROUS_SYSTEM_SERVICE_CALLED</td>
<td></td>
</tr>
<tr>
<td>CYSTATUS_DEMO_EVENT</td>
<td></td>
</tr>
<tr>
<td>CYSTATUS_DEP_SEH_INF_VIOLATION</td>
<td></td>
</tr>
<tr>
<td>CYSTATUS_DEP_SEH_VIOLATION</td>
<td></td>
</tr>
<tr>
<td>CYSTATUS_DEP_VIOLATION</td>
<td></td>
</tr>
<tr>
<td>CYSTATUS_DEP_VIOLATION_UNALLOCATED</td>
<td></td>
</tr>
<tr>
<td>CYSTATUS_DEVICE_BLOCKED</td>
<td></td>
</tr>
<tr>
<td>CYSTATUS_DLLPROT_BLACKLIST</td>
<td></td>
</tr>
<tr>
<td>CYSTATUS_DLLPROT_CURRENT_WORKING_DIRECTORY</td>
<td></td>
</tr>
<tr>
<td>CYSTATUS_DLLPROT_REMOTE_DRIVE</td>
<td></td>
</tr>
<tr>
<td>CYSTATUS_DLLPROT_REMVABLE_DRIVE</td>
<td></td>
</tr>
<tr>
<td>CYSTATUS_DOTNET_CRITICAL</td>
<td></td>
</tr>
<tr>
<td>CYSTATUS_DSE</td>
<td></td>
</tr>
<tr>
<td>CYSTATUS_EPM_INIT_FAILED</td>
<td></td>
</tr>
<tr>
<td>CYSTATUS_FAILED_CHECK_MEDIA</td>
<td></td>
</tr>
<tr>
<td>CYSTATUS_FILE_DELETION_BOOT_DONE</td>
<td></td>
</tr>
<tr>
<td>CYSTATUS_FILE_DELETION_FAILED</td>
<td></td>
</tr>
<tr>
<td>CYSTATUS_FILE_DELETION_SUCCEEDED</td>
<td></td>
</tr>
<tr>
<td>CYSTATUS_FINGERPRINTING_ATTEMPT</td>
<td></td>
</tr>
<tr>
<td>CYSTATUS_FONT_PROT_DUQU</td>
<td></td>
</tr>
<tr>
<td>CYSTATUS_FORBIDDEN_MEDIA</td>
<td></td>
</tr>
<tr>
<td>CYSTATUS_FORBIDDEN_OPTICAL_MEDIA</td>
<td></td>
</tr>
<tr>
<td>CYSTATUS_FORBIDDEN_REMOTE_MEDIA</td>
<td></td>
</tr>
<tr>
<td>CYSTATUS_FORBIDDEN_REMOVABLE_MEDIA</td>
<td></td>
</tr>
<tr>
<td>CYSTATUS_HS_COOKIE_CORRUPTED_COOKIE</td>
<td></td>
</tr>
<tr>
<td>CYSTATUS_GUARD_PAGE_VIOLATION</td>
<td></td>
</tr>
<tr>
<td>CYSTATUS_HASH_CONTROL</td>
<td></td>
</tr>
<tr>
<td>CYSTATUS_HEAP_CORRUPTION</td>
<td></td>
</tr>
<tr>
<td>CYSTATUS_HOOKING_ENTRY_POINT_FAILED</td>
<td></td>
</tr>
<tr>
<td>CYSTATUS_HOTPATCH_HIJACKING</td>
<td></td>
</tr>
<tr>
<td>CYSTATUS_ILLEGAL_EXECUTABLE</td>
<td></td>
</tr>
<tr>
<td>CYSTATUS_ILLEGAL_UNSIGNED_EXECUTABLE</td>
<td></td>
</tr>
<tr>
<td>CYSTATUS_INJ_APPCONTAINER_FAILURE</td>
<td></td>
</tr>
<tr>
<td>CYSTATUS_INJ_CTX_FAILURE</td>
<td></td>
</tr>
<tr>
<td>CYSTATUS_JAVA_FILE</td>
<td></td>
</tr>
<tr>
<td>CYSTATUS_JAVA_PROC</td>
<td></td>
</tr>
<tr>
<td>CYSTATUS_JAVA_REG</td>
<td></td>
</tr>
<tr>
<td>CYSTATUS_JIT_EXCEPTION</td>
<td></td>
</tr>
<tr>
<td>CYSTATUS_LINUX_BRUTEFORCE_PREVENTED</td>
<td></td>
</tr>
<tr>
<td>CYSTATUS_LINUX_ROOT_ESCALATION_PREVENTED</td>
<td></td>
</tr>
<tr>
<td>CYSTATUS_LINUX_SHELLCODE_PREVENTED</td>
<td></td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>CYSTATUS_LINUX_SOCKET_SHELL_PREVENTED</td>
<td>CYSTATUS_LOCAL_ANALYSIS</td>
</tr>
<tr>
<td>CYSTATUS_MACOS_DLPROT_CWD_HIJACK</td>
<td>CYSTATUS_MACOS_DLPROT_DUPLICATE_PATH_CHECK</td>
</tr>
<tr>
<td>CYSTATUS_MACOS_GO2_BLOCK_ALL</td>
<td>CYSTATUS_MACOS_GO2_SIGNER_NAME_MISMATCH</td>
</tr>
<tr>
<td>CYSTATUS_MACOS_GO2_SIGN_LEVEL_BELOW_MIN</td>
<td>CYSTATUS_MACOS_GO2_SIGN_LEVEL_BELOW_PARENT</td>
</tr>
<tr>
<td>CYSTATUS_MACOS_MALICIOUS_PYLIB</td>
<td>CYSTATUS_MACOS_ROOT_ESCALATION_PREVENTED</td>
</tr>
<tr>
<td>CYSTATUS_MACOS_MALICIOUS_APK</td>
<td>CYSTATUS_MACOS_MALICIOUS_DLL</td>
</tr>
<tr>
<td>CYSTATUS_MACOS_MALICIOUS_EXE</td>
<td>CYSTATUS_MACOS_MALICIOUS_EXE_ASYNC</td>
</tr>
<tr>
<td>CYSTATUS_MACOS_MALICIOUS_MACRO</td>
<td>CYSTATUS_MEMORY_USAGE_LIMIT_EXCEEDED</td>
</tr>
<tr>
<td>CYSTATUS_MACOS_MALICIOUS_STRING_detected</td>
<td>CYSTATUS_NOP_SLED_DETECTED</td>
</tr>
<tr>
<td>CYSTATUS_MALICIOUS_APK</td>
<td>CYSTATUS_MALICIOUS_DLL</td>
</tr>
<tr>
<td>CYSTATUS_MALICIOUS_EXE</td>
<td>CYSTATUS_MALICIOUS_EXE_ASYNC</td>
</tr>
<tr>
<td>CYSTATUS_MALICIOUS_MACRO</td>
<td>CYSTATUS_MEMORY_USAGE_LIMIT_EXCEEDED</td>
</tr>
<tr>
<td>CYSTATUS_MALICIOUS_STRING_detected</td>
<td>CYSTATUS_NOP_SLED_DETECTED</td>
</tr>
<tr>
<td>CYSTATUS_MEMORY_USAGE_LIMIT_EXCEEDED</td>
<td>CYSTATUS_NO_MEMORY</td>
</tr>
<tr>
<td>CYSTATUS_NO_MEMORY</td>
<td>CYSTATUS_NO_REGISTER_CORRECTED</td>
</tr>
<tr>
<td>CYSTATUS_PREALLOCATED_ADDR_ACCESSSED</td>
<td>CYSTATUS_PROCESS_CREATION_VIOLATION</td>
</tr>
<tr>
<td>CYSTATUS_PROCESS_CREATION_VIOLATION</td>
<td>CYSTATUS_QUARANTINE_FAILED</td>
</tr>
<tr>
<td>CYSTATUS_QUARANTINE_FAILED</td>
<td>CYSTATUS_QUARANTINE_SUCCEEDED</td>
</tr>
<tr>
<td>CYSTATUS_QUARANTINE_SUCCEEDED</td>
<td>CYSTATUS_RANSOMWARE</td>
</tr>
<tr>
<td>CYSTATUS_RANSOMWARE</td>
<td>CYSTATUS_RESTORE_FAILED</td>
</tr>
<tr>
<td>CYSTATUS_RESTORE_FAILED</td>
<td>CYSTATUS_RESTORE_SUCCEEDED</td>
</tr>
<tr>
<td>CYSTATUS_RESTORE_SUCCEEDED</td>
<td>CYSTATUS_ROP MITIGATION</td>
</tr>
<tr>
<td>CYSTATUS_ROP MITIGATION</td>
<td>CYSTATUS_SEH_CRITICAL</td>
</tr>
<tr>
<td>CYSTATUS_SEH_CRITICAL</td>
<td>CYSTATUS_SEH_INF_CRITICAL</td>
</tr>
<tr>
<td>CYSTATUS_SEH_INF_CRITICAL</td>
<td>CYSTATUS_SHELL_CODE_TRAP_CALLED</td>
</tr>
<tr>
<td>CYSTATUS_SHELL_CODE_TRAP_CALLED</td>
<td>CYSTATUS_STACK_OVERFLOW</td>
</tr>
<tr>
<td>CYSTATUS_STACK_OVERFLOW</td>
<td>CYSTATUS_SUSPENDED_PROCESS_BLOCKED</td>
</tr>
<tr>
<td>CYSTATUS_SUSPENDED_PROCESS_BLOCKED</td>
<td>CYSTATUS_SUSPICIOUS_APC</td>
</tr>
<tr>
<td>CYSTATUS_SUSPICIOUS_APC</td>
<td>CYSTATUS_SUSPICIOUS_LINK_FILE</td>
</tr>
<tr>
<td>CYSTATUS_SUSPICIOUS_LINK_FILE</td>
<td>CYSTATUS_SYSTEM_SCAN_FINISHED</td>
</tr>
<tr>
<td>CYSTATUS_SYSTEM_SCAN_FINISHED</td>
<td>CYSTATUS_SYSTEM_SCAN_STARTED</td>
</tr>
<tr>
<td>CYSTATUS_SYSTEM_SCAN_STARTED</td>
<td>CYSTATUS_THREAD_INJECTION</td>
</tr>
<tr>
<td>CYSTATUS_THREAD_INJECTION</td>
<td>CYSTATUS_TLA_MODEL_NOT_LOADED</td>
</tr>
<tr>
<td>CYSTATUS_TLA_MODEL_NOT_LOADED</td>
<td>CYSTATUS_TOKEN_THEFT_FILE_OPERATION</td>
</tr>
<tr>
<td>CYSTATUS_TOKEN_THEFT_FILE_OPERATION</td>
<td>CYSTATUS_TOKEN_THEFT_PROCESS_CREATED</td>
</tr>
<tr>
<td>CYSTATUS_TOKEN_THEFT_PROCESS_CREATED</td>
<td>CYSTATUS_TOKEN_THEFT_REGISTRY_OPERATION</td>
</tr>
<tr>
<td>CYSTATUS_TOKEN_THEFT_REGISTRY_OPERATION</td>
<td>CYSTATUS_TOKEN_THEFT_THREAD_CREATED</td>
</tr>
<tr>
<td>CYSTATUS_TOKEN_THEFT_THREAD_CREATED</td>
<td>CYSTATUS_TOKEN_THEFT_THREAD_INJECTED</td>
</tr>
<tr>
<td>CYSTATUS_TOKEN_THEFT_THREAD_INJECTED</td>
<td>CYSTATUS_TOKEN_THEFT_THREAD_STARTED</td>
</tr>
<tr>
<td>CYSTATUS_TOKEN_THEFT_THREAD_STARTED</td>
<td>CYSTATUS_UASLR_CRITICAL</td>
</tr>
<tr>
<td>CYSTATUS_UASLR_CRITICAL</td>
<td>CYSTATUS_UNALLOWED_CODE_SEGMENT</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>CYSTATUS_UNAUTHORIZED_CALL_TO_SYSTEM_SERVICE</td>
<td>Action carried out by the Cortex XDR agent (block or notify). The prevention mode is specified in the rule configuration.</td>
</tr>
<tr>
<td>CYSTATUS_UNSIGNED_CHILD_PROCESS_BLOCKED</td>
<td>Termination action taken on the file.</td>
</tr>
<tr>
<td>CYSTATUS_WILDFIRE_GRAYWARE</td>
<td>Termination action taken on the target file (relevant for some child process execution events where we terminate the child process but not the parent process):</td>
</tr>
<tr>
<td>CYSTATUS_WILDFIRE_MALWARE</td>
<td>Quarantine action taken on the file:</td>
</tr>
<tr>
<td>CYSTATUS_WILDFIRE_UNKNOWN</td>
<td>Block action taken on the file:</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>verdict</td>
<td>Verdict for the file:</td>
</tr>
<tr>
<td>preventMode</td>
<td>Termination action taken on the file.</td>
</tr>
<tr>
<td>terminateTarget</td>
<td>Termination action taken on the target file (relevant for some child process execution events where we terminate the child process but not the parent process):</td>
</tr>
<tr>
<td>terminate</td>
<td>Termination action taken on the file.</td>
</tr>
<tr>
<td>terminateTarget</td>
<td>Termination action taken on the target file (relevant for some child process execution events where we terminate the child process but not the parent process):</td>
</tr>
<tr>
<td>quarantine</td>
<td>Quarantine action taken on the file:</td>
</tr>
<tr>
<td>block</td>
<td>Block action taken on the file:</td>
</tr>
<tr>
<td>postDetected</td>
<td>Post detection status of the file:</td>
</tr>
<tr>
<td>eventParameters</td>
<td>Parameters associated with the type of event. For example, username, endpoint hostname, and filename.</td>
</tr>
<tr>
<td>sourceProcessIdx</td>
<td>The prevention source process index in the processes array.</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>--------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>targetProcessIdx(Array)</td>
<td>Target process index in the processes array. A missing or negative value means there is no target process.</td>
</tr>
<tr>
<td>fileIdx(Array)</td>
<td>Index of target files for specific security events such as: Scanning, Malicious DLL, Malicious Macro events.</td>
</tr>
<tr>
<td>processes(Array)</td>
<td>All related details for the process file that triggered an event:</td>
</tr>
<tr>
<td></td>
<td>• 1—System process ID</td>
</tr>
<tr>
<td></td>
<td>• 2—Parent process ID</td>
</tr>
<tr>
<td></td>
<td>• 3—File object corresponding to the process executable file</td>
</tr>
<tr>
<td></td>
<td>• 4—Command line arguments (if any)</td>
</tr>
<tr>
<td></td>
<td>• 5—Description field of the VERSIONINFO resource</td>
</tr>
<tr>
<td></td>
<td>• 6—File version field of the VERSIONINFO resource</td>
</tr>
<tr>
<td>files(Array)</td>
<td>File object includes:</td>
</tr>
<tr>
<td></td>
<td>• 1—SHA256 hash value of the file</td>
</tr>
<tr>
<td></td>
<td>• 2—SHA256 hash value of the macro</td>
</tr>
<tr>
<td></td>
<td>• 3—Raw full filepath</td>
</tr>
<tr>
<td></td>
<td>• 4—A predefined drive type: local, network mapped drive, UNC path host, removable media, etc.</td>
</tr>
<tr>
<td></td>
<td>• 5—File name (with no extension), such as AdapterTroubleshooter</td>
</tr>
<tr>
<td></td>
<td>• 6—File extension (for example, EXE or DLL)</td>
</tr>
<tr>
<td></td>
<td>• 7—File type defined by the Cortex XDR agent</td>
</tr>
<tr>
<td></td>
<td>• 8—UTC file creation time</td>
</tr>
<tr>
<td></td>
<td>• 9—UTC file modification time</td>
</tr>
<tr>
<td></td>
<td>• 10—UTC file access time</td>
</tr>
<tr>
<td></td>
<td>• 11—File attributes bitmask</td>
</tr>
<tr>
<td></td>
<td>• 12—File size in bytes</td>
</tr>
<tr>
<td></td>
<td>• 13—Signer field of the code signing certificate</td>
</tr>
<tr>
<td>users(Array)</td>
<td>Details about the active user on the endpoint when the event occurred:</td>
</tr>
<tr>
<td></td>
<td>• 1—Username of the active user on the endpoint.</td>
</tr>
<tr>
<td></td>
<td>• 2—Domain to which the user account belongs.</td>
</tr>
<tr>
<td>urls(Array)</td>
<td>Additional details related to a URL:</td>
</tr>
<tr>
<td></td>
<td>• 1—Raw URL</td>
</tr>
<tr>
<td></td>
<td>• 2—URL schema; For example: HTTP, HTTPS, FTP, LDAP</td>
</tr>
<tr>
<td></td>
<td>• 3—Hostname in punycode</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>• 4—Host port</td>
<td></td>
</tr>
<tr>
<td>• 5—Canonicalized URL path part according to schema requirements</td>
<td></td>
</tr>
<tr>
<td>• 6—Query parameters (for http's only)</td>
<td></td>
</tr>
<tr>
<td>• 7—Fragment parameters (for http's only)</td>
<td></td>
</tr>
<tr>
<td>description(Array)</td>
<td>(Mac only) Description of components related to Cortex XDR. For example, the description of the ROP, JIT, Dylib hijacking modules for Mac endpoints is Memory Corruption Exploit.</td>
</tr>
</tbody>
</table>

**Config Logs**

**Syslog format**: recordType, class, FUTURE_USE, subClassId, eventType, eventCategory, generatedTime, serverTime, FUTURE_USE, facility, customerId, trapsId, serverHost, serverComponentVersion, regionId, isEndpoint, severity, trapsSeverity, messageCode, friendlyName, FUTURE_USE, msgTextEn, userName, userRole, userDomain, additionalData(Array), messageCode, errorText, errorData, resultData

**Email body format example:**

```plaintext
recordType: system
messageData/class: system
messageData/subClass: Provisioning
messageData/subClassId: 13
eventType: ServerLogPerTenant
messageData/eventCategory: tenant
generatedTime: 2019-01-31T18:15:19.000000+00:00
serverTime: 2019-01-31T18:15:19.000000+00:00
product:
facility: TrapsServerManagement
customerId: 004403511
trapsId: 18520498190303952
serverHost: 14917869646-201.proda.brz
serverComponentVersion: 2.0.9+624
regionId: isEndpoint: 0
agentId:
severity: notice
messageData/trapsSeverity: informational
messageData/messageCode: 19015
messageData/friendlyName: User Login
messageData/msgTextLoc: messageData/msgTextEn: User username@paloaltonetworks.com has logged in with role superadmin
endPointHeader/userFullName:
endPointHeader/username:
endPointHeader/userRole:
endPointHeader/userDomain:
endPointHeader/agentTime:
endPointHeader/tzOffset:
endPointHeader/osType:
endPointHeader/isVdi:
endPointHeader/osVersion:
endPointHeader/is64:
endPointHeader/agentIp:
endPointHeader/deviceName:
```
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>recordType</td>
<td>Record type associated with the event and that you can use when managing logging quotas. In this case, the record type is config which includes logs related to Cortex XDR administration and configuration changes.</td>
</tr>
<tr>
<td>class</td>
<td>Class of Cortex XDR log. System logs have a value of system.</td>
</tr>
<tr>
<td>subClass</td>
<td>Subclass of event. Used to categorize logs in Cortex XDR.</td>
</tr>
<tr>
<td>subClassId</td>
<td>Numeric representation of the subClass field for easy sorting and filtering.</td>
</tr>
<tr>
<td>eventType</td>
<td>Subtype of event.</td>
</tr>
<tr>
<td>eventCategory</td>
<td>Category of event, used internally for processing the flow of logs. Event categories vary by class:</td>
</tr>
<tr>
<td></td>
<td>• config—deviceManagement, distributionManagement, reportManagement, securityEventManagement, systemManagement</td>
</tr>
<tr>
<td></td>
<td>• policy—exceptionManagement, policyManagement, profileManagement, sam</td>
</tr>
<tr>
<td></td>
<td>• system—licensing, provisioning, tenant, userAuthentication, workerProcessing</td>
</tr>
<tr>
<td></td>
<td>• agent_log—agentFlow</td>
</tr>
<tr>
<td>generatedTime</td>
<td>Coordinated Universal Time (UTC) equivalent of the time at which an event was logged. For agent events, this represents the time on the endpoint. For policy, configuration, and system events, this represents the time on Cortex XDR in ISO-8601 string representation (for example, 2017-01-24T09:08:59Z).</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>serverTime</td>
<td>Coordinated Universal Time (UTC) equivalent of the time at which the server generated the log. If the log was generated on an endpoint, this field identifies the time the server received the log in ISO-8601 string representation (for example, 2017-01-24T09:08:59Z).</td>
</tr>
<tr>
<td>facility</td>
<td>The Cortex XDR system component that initiated the event, for example: TrapsAgent, TrapsServiceCore, TrapsServiceManagement, and TrapsServiceBackend.</td>
</tr>
<tr>
<td>customerId</td>
<td>The ID that uniquely identifies the Cortex Data Lake instance which received this log record.</td>
</tr>
<tr>
<td>trapsId</td>
<td>Tenant external ID.</td>
</tr>
<tr>
<td>serverHost</td>
<td>Hostname of Cortex XDR.</td>
</tr>
<tr>
<td>serverComponentVersion</td>
<td>Software version of Cortex XDR.</td>
</tr>
<tr>
<td>regionId</td>
<td>ID of Cortex XDR region:</td>
</tr>
<tr>
<td></td>
<td>• 10—Americas (N. Virginia)</td>
</tr>
<tr>
<td></td>
<td>• 70—EMEA (Frankfurt)</td>
</tr>
<tr>
<td>isEndpoint</td>
<td>Indicates whether the event occurred on an endpoint.</td>
</tr>
<tr>
<td></td>
<td>• 0—No, host is not an endpoint.</td>
</tr>
<tr>
<td></td>
<td>• 1—Yes, host is an endpoint.</td>
</tr>
<tr>
<td>agentId</td>
<td>Unique identifier for the Cortex XDR agent.</td>
</tr>
<tr>
<td>severity</td>
<td>Syslog severity level associated with the event.</td>
</tr>
<tr>
<td></td>
<td>• 2—Critical. Used for events that require immediate attention.</td>
</tr>
<tr>
<td></td>
<td>• 3—Error. Used for events that require special handling.</td>
</tr>
<tr>
<td></td>
<td>• 4—Warning. Used for events that sometimes require special handling.</td>
</tr>
<tr>
<td></td>
<td>• 5—Notice. Used for normal but significant events that can require attention.</td>
</tr>
<tr>
<td></td>
<td>• 6—Informational. Informational events that do not require attention.</td>
</tr>
<tr>
<td></td>
<td>Each event also has an associated Cortex XDR severity. See the <code>messageData.trapsSeverity</code> field for details.</td>
</tr>
<tr>
<td>trapsSeverity</td>
<td>Severity level associated with the event defined for Cortex XDR. Each of these severities corresponds to a syslog severity level:</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>messageCode</td>
<td>System-wide unique message code.</td>
</tr>
<tr>
<td>friendlyName</td>
<td>Descriptive log message name.</td>
</tr>
<tr>
<td>msgTextEn</td>
<td>Description of the event, in English.</td>
</tr>
<tr>
<td>userFullName</td>
<td>Full username of Cortex XDR user.</td>
</tr>
<tr>
<td>userName</td>
<td>Username associated with Cortex XDR user.</td>
</tr>
<tr>
<td>userRole</td>
<td>Role assigned to Cortex XDR user.</td>
</tr>
<tr>
<td>userDomain</td>
<td>Domain to which the user belongs.</td>
</tr>
<tr>
<td>agentTime</td>
<td>Coordinated Universal Time (UTC) equivalent of the time at which an agent logged an event in ISO-8601 string representation.</td>
</tr>
<tr>
<td>tzOffset</td>
<td>Effective endpoint time zone offset from UTC, in minutes.</td>
</tr>
<tr>
<td>osType</td>
<td>Operating system of the endpoint:</td>
</tr>
<tr>
<td></td>
<td>• 1—Windows</td>
</tr>
<tr>
<td></td>
<td>• 2—OS X/macOS</td>
</tr>
<tr>
<td></td>
<td>• 3—Android</td>
</tr>
<tr>
<td></td>
<td>• 4—Linux</td>
</tr>
<tr>
<td>isVdi</td>
<td>Indicates whether the endpoint is a virtual desktop infrastructure (VDI):</td>
</tr>
<tr>
<td></td>
<td>• 0—The endpoint is not a VDI</td>
</tr>
<tr>
<td></td>
<td>• 1—The endpoint is a VDI</td>
</tr>
</tbody>
</table>

- **0**—Informational. Informational messages that do not require attention. Identical to the syslog 6 (Informational) severity level.
- **1**—Low. Used for normal but significant events that can require attention. Corresponds to the syslog 5 (Notice) severity level.
- **2**—Medium. Used for events that sometimes require special handling. Corresponds to the syslog 4 (Warning) severity level.
- **3**—High. Used for events that require special handling. Corresponds to the syslog 3 (Error) severity level.
- **4**—Critical. Used for events that require immediate attention. Corresponds to the syslog 2 (Critical) severity level.

See also the severity log field.
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>osVersion</td>
<td>Full version number of the operating system running on the endpoint. For example, 6.1.7601.19135.</td>
</tr>
<tr>
<td>is64</td>
<td>Indicates whether the endpoint is running a 64-bit version of Windows:</td>
</tr>
<tr>
<td></td>
<td>• 0—The endpoint is not running x64 architecture</td>
</tr>
<tr>
<td></td>
<td>• 1—The endpoint is running x64 architecture</td>
</tr>
<tr>
<td>agentIp</td>
<td>IP address of the endpoint.</td>
</tr>
<tr>
<td>deviceName</td>
<td>Hostname of the endpoint on which the event was logged.</td>
</tr>
<tr>
<td>deviceDomain</td>
<td>Domain to which the endpoint belongs.</td>
</tr>
<tr>
<td>agentVersion</td>
<td>Version of the Cortex XDR agent.</td>
</tr>
<tr>
<td>contentVersion</td>
<td>Content version in the local security policy.</td>
</tr>
<tr>
<td>protectionStatus</td>
<td>Cortex XDRagent protection status:</td>
</tr>
<tr>
<td></td>
<td>• 0—Protected</td>
</tr>
<tr>
<td></td>
<td>• 1—OsVersionIncompatible</td>
</tr>
<tr>
<td></td>
<td>• 2—AgentIncompatible</td>
</tr>
<tr>
<td>userFullName</td>
<td>Full name of Cortex XDR user.</td>
</tr>
<tr>
<td>userName</td>
<td>Username associated with Cortex XDR user.</td>
</tr>
<tr>
<td>userRole</td>
<td>Role assigned to Cortex XDR user.</td>
</tr>
<tr>
<td>userDomain</td>
<td>Domain to which the user belongs.</td>
</tr>
<tr>
<td>messageName</td>
<td>Name of the message.</td>
</tr>
<tr>
<td>messageId</td>
<td>Unique numeric identifier of the message.</td>
</tr>
<tr>
<td>processStatus</td>
<td>State of the process related to the event.</td>
</tr>
<tr>
<td>errorText</td>
<td>If known, a description of the documented error.</td>
</tr>
<tr>
<td>errorData</td>
<td>Parameters related to an event error.</td>
</tr>
<tr>
<td>resultData</td>
<td>Parameters related to a successful event.</td>
</tr>
<tr>
<td>parameters</td>
<td>Parameters supplied in the log message.</td>
</tr>
<tr>
<td>additionalData</td>
<td>Additional information regarding event parameters.</td>
</tr>
<tr>
<td>loggedInUser</td>
<td>User that is logged in to the Cortex XDR.</td>
</tr>
</tbody>
</table>
Analytics Logs

Syslog format: recordType, class, FUTURE_USE, eventType, eventCategory, generatedTime, serverTime, agentTime, tzOffset, FUTURE_USE, facility, customerId, trapsId, serverHost, serverComponentVersion, regionId, isEndpoint, agentId, osType, isVdi, osVersion, is64, agentIp, deviceName, deviceDomain, severity, agentVersion, contentVersion, protectionStatus, sha256, type, parentSha256, lastSeen, fileName, filePath, fileSize, localAnalysisResult, reported, blocked, executionCount

Email body format example:

recordType: analytics
messageData/class: agent_data
messageData/subClass: 
eventType: AgentTimelineEvent
messageData/eventCategory: hash
generatedTime: 2019-01-31T18:00:43Z
endPointHeader/agentTime: 2019-01-31T18:00:43Z
endPointHeader/tzOffset: -480
product: 
facility: TrapsAgent
customerId: 110044035
trapsId: 18520039498190352
serverHost: coreop-f-proda-mnmauto03930348053-311.proda.brz
serverComponentVersion: 2.0.9+564
regionId: 10
isEndpoint: 1
agentId: 3bcf7e5ff56e2891c78b72e49
endPointHeader/osType: 2
endPointHeader/isVdi: 0
endPointHeader/osVersion: 10.12.6
endPointHeader/is64: 1
endPointHeader/agentIp: 192.168.0.21
endPointHeader/deviceName: Jeffreys-MacBook-Pro.local
endPointHeader/deviceDomain: 
severity: 
endPointHeader/agentVersion: 5.0.5.1193
endPointHeader/contentVersion: 42-6337
endPointHeader/protectionStatus: 0
messageData/sha256: 87e27ba9128d9c3b3d113c67623a06817a030b3bba42bd2871d1e6da9002206f26
messageData/type: macho
messageData/parentSha256: 
messageData/lastSeen: 2019-01-31T18:00:43Z
messageData/fileName: crashpad_handler
messageData/filePath: /users/username/library/google/googlesoftwareupdate/googleosftwareupdate.bundle/contents/macOS/
messageData/fileSize: 35680
messageData/localAnalysisResult: 
"""resultId"""":0,""""trustedId"""":0\"""
messageData/reported: 0
messageData/blocked: 0
messageData/executionCount: 4179
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>recordType</td>
<td>Record type associated with the event and that you can use when managing logging quotas. In this case, the record type is analytics which includes hash execution reports from the agent.</td>
</tr>
<tr>
<td>class</td>
<td>Class of Cortex XDR log: config, policy, system, and agent_log.</td>
</tr>
<tr>
<td>eventType</td>
<td>Subtype of event.</td>
</tr>
<tr>
<td>eventCategory</td>
<td>Category of event, used internally for processing the flow of logs. Event categories vary by class:</td>
</tr>
<tr>
<td></td>
<td>• config—deviceManagement, distributionManagement, securityEventManagement, systemManagement</td>
</tr>
<tr>
<td></td>
<td>• policy—exceptionManagement, policyManagement, profileManagement, sam</td>
</tr>
<tr>
<td></td>
<td>• system—licensing, provisioning, tenant, userAuthentication, workerProcessing</td>
</tr>
<tr>
<td></td>
<td>• agent_log—agentFlow</td>
</tr>
<tr>
<td>generatedTime</td>
<td>Coordinated Universal Time (UTC) equivalent of the time at which an event was logged. For agent events, this represents the time on the endpoint. For policy, configuration, and system events, this represents the time on Cortex XDR in ISO-8601 string representation (for example, 2017-01-24T09:08:59Z).</td>
</tr>
<tr>
<td>serverTime</td>
<td>Coordinated Universal Time (UTC) equivalent of the time at which the server generated the log. If the log was generated on an endpoint, this field identifies the time the server received the log in ISO-8601 string representation (for example, 2017-01-24T09:08:59Z).</td>
</tr>
<tr>
<td>agentTime</td>
<td>Coordinated Universal Time (UTC) equivalent of the time at which an agent logged an event in ISO-8601 string representation.</td>
</tr>
<tr>
<td>tzOffset</td>
<td>Effective endpoint time zone offset from UTC, in minutes.</td>
</tr>
<tr>
<td>facility</td>
<td>The Cortex XDR system component that initiated the event, for example: TrapsAgent, TrapsServiceCore, TrapsServiceManagement, and TrapsServiceBackend.</td>
</tr>
<tr>
<td>customerId</td>
<td>The ID that uniquely identifies the Cortex Data Lake instance which received this log record.</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>trapsId</td>
<td>Tenant external ID.</td>
</tr>
<tr>
<td>serverHost</td>
<td>Hostname of Cortex XDR.</td>
</tr>
<tr>
<td>serverComponentVersion</td>
<td>Software version of Cortex XDR.</td>
</tr>
<tr>
<td>regionId</td>
<td>ID of Cortex XDR region:</td>
</tr>
<tr>
<td></td>
<td>• 10—Americas (N. Virginia)</td>
</tr>
<tr>
<td></td>
<td>• 70—EMEA (Frankfurt)</td>
</tr>
<tr>
<td>isEndpoint</td>
<td>Indicates whether the event occurred on an endpoint.</td>
</tr>
<tr>
<td></td>
<td>• 0—No, host is not an endpoint.</td>
</tr>
<tr>
<td></td>
<td>• 1—Yes, host is an endpoint.</td>
</tr>
<tr>
<td>agentId</td>
<td>Unique identifier for the Cortex XDR agent.</td>
</tr>
<tr>
<td>osType</td>
<td>Operating system of the endpoint:</td>
</tr>
<tr>
<td></td>
<td>• 1—Windows</td>
</tr>
<tr>
<td></td>
<td>• 2—OS X/macOS</td>
</tr>
<tr>
<td></td>
<td>• 3—Android</td>
</tr>
<tr>
<td></td>
<td>• 4—Linux</td>
</tr>
<tr>
<td>isVdi</td>
<td>Indicates whether the endpoint is a virtual desktop infrastructure (VDI):</td>
</tr>
<tr>
<td></td>
<td>• 0—The endpoint is not a VDI</td>
</tr>
<tr>
<td></td>
<td>• 1—The endpoint is a VDI</td>
</tr>
<tr>
<td>osVersion</td>
<td>Full version number of the operating system running on the endpoint. For example, 6.1.7601.19135.</td>
</tr>
<tr>
<td>is64</td>
<td>Indicates whether the endpoint is running a 64-bit version of Windows:</td>
</tr>
<tr>
<td></td>
<td>• 0—The endpoint is not running x64 architecture</td>
</tr>
<tr>
<td></td>
<td>• 1—The endpoint is running x64 architecture</td>
</tr>
<tr>
<td>agentIp</td>
<td>IP address of the endpoint.</td>
</tr>
<tr>
<td>deviceName</td>
<td>Hostname of the endpoint on which the event was logged.</td>
</tr>
<tr>
<td>deviceDomain</td>
<td>Domain to which the endpoint belongs.</td>
</tr>
<tr>
<td>severity</td>
<td>Syslog severity level associated with the event.</td>
</tr>
</tbody>
</table>
|                 | • 2—Critical. Used for events that require immediate attention.
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>agentVersion</td>
<td>Version of the Cortex XDR agent.</td>
</tr>
<tr>
<td>contentVersion</td>
<td>Content version in the local security policy.</td>
</tr>
<tr>
<td>protectionStatus</td>
<td>Cortex XDR agent protection status:</td>
</tr>
<tr>
<td></td>
<td>• 0—Protected</td>
</tr>
<tr>
<td></td>
<td>• 1—OsVersionIncompatible</td>
</tr>
<tr>
<td></td>
<td>• 2—AgentIncompatible</td>
</tr>
<tr>
<td>sha256</td>
<td>Hash of the file using SHA256 encoding.</td>
</tr>
<tr>
<td>type</td>
<td>Type of file:</td>
</tr>
<tr>
<td></td>
<td>• 0—Unknown</td>
</tr>
<tr>
<td></td>
<td>• 1—PE</td>
</tr>
<tr>
<td></td>
<td>• 2—Mach-o</td>
</tr>
<tr>
<td></td>
<td>• 3—DLL</td>
</tr>
<tr>
<td></td>
<td>• 4—Office file (containing a macro)</td>
</tr>
<tr>
<td>parentSha256</td>
<td>Hash of the parent file using SHA256 encoding.</td>
</tr>
<tr>
<td>lastSeen</td>
<td>Coordinated Universal Time (UTC) equivalent of the time when the file last ran on an endpoint in ISO-8601 string representation (for example, 2017-01-24T09:08:59Z).</td>
</tr>
<tr>
<td>fileName</td>
<td>File name, without the path or the file type extension.</td>
</tr>
<tr>
<td>filePath</td>
<td>Full path, aligned to the OS format.</td>
</tr>
<tr>
<td>fileSize</td>
<td>Size of the file in bytes.</td>
</tr>
<tr>
<td>localAnalysisResult</td>
<td>This object includes the content version, local analysis module version, verdict result, file signer, and trusted signer result. The trusted signer result is an integer value:</td>
</tr>
<tr>
<td></td>
<td>• 0—Cortex XDR did not evaluate the signer of the file.</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| reported   | Reporting status of the file, in integer value:  
- 0—Cortex XDR did not report the security event.  
- 1—Cortex XDR reported the security event.  |
| blocked    | Blocking status of the file, in integer value:  
- 0—Cortex XDR did not block the process or file.  
- 1—Cortex XDR blocked the process or file.  |
| executionCount | The total number of times a file identified by a specific hash was executed.  |

**System Logs**

**Syslog format:** recordType, class, FUTURE_USE, subClassId, eventType, eventCategory, generatedTime, serverTime, FUTURE_USE, facility, customerId, trapsId, serverHost, serverComponentVersion, regionId, isEndpoint, agentId, severity, trapsSeverity, messageCode, friendlyName, FUTURE_USE, msgTextEn, userFullName, username, userRole, userDomain, agentTime, tzOffset, osType, isVdi, osVersion, is64, agentIp, serviceName, deviceDomain, agentVersion, contentVersion, protectionStatus, userFullName, username, userRole, userDomain, messageName, messageId, processStatus, errorText, errorData, resultData, parameters, additionalData(Array)

**Email body format example:**

```
recordType: system
messageData/class: system
messageData/subClass: Provisioning
messageData/subClassId: 13
eventType: ServerLogPerTenant
messageData/eventCategory: tenant
generatedTime: 2019-01-31T18:15:19.000000+00:00
serverTime: 2019-01-31T18:15:19.000000+00:00
product:
facility: TrapsServerManagement
customerId: 004403511
trapsId: 18520498190303952
serverHost: 14917869646-201.proda.brz
serverComponentVersion: 2.0.9+624
regionId:
isEndpoint: 0
agentId:
severity: notice
messageData/trapsSeverity: informational
messageData/messageCode: 19015
messageData/friendlyName: User Login
messageData/msgTextLoc: User username@paloaltonetworks.com has logged in with role superadmin
messageData/msgTextEn: User username@paloaltonetworks.com has logged in with role superadmin
endPointHeader/userFullName:
endPointHeader/username:
endPointHeader/userRole:
endPointHeader/userDomain:
```
### Field Name | Description
--- | ---
recordType | Record type associated with the event and that you can use when managing logging quotas. In this case, the record type is system which includes logs related to automated system management and agent reporting events.
class | Class of Cortex XDR log. System logs have a value of system.
subClass | Subclass of event. Used to categorize logs in Cortex XDR user interface.
subClassId | Numeric representation of the subClass field for easy sorting and filtering.
eventType | Subtype of event.
eventCategory | Category of event, used internally for processing the flow of logs. Event categories vary by class:
- **config**—deviceManagement, distributionManagement, securityEventManagement, systemManagement
- **policy**—exceptionManagement, policyManagement, profileManagement, sam
- **system**—licensing, provisioning, tenant, userAuthentication, workerProcessing
- **agent_log**—agentFlow

---

endPointHeader/agentTime:
endPointHeader/tzOffset:
endPointHeader/osType:
endPointHeader/isVdi:
endPointHeader/osVersion:
endPointHeader/is64:
endPointHeader/agentIp:
endPointHeader/deviceName:
endPointHeader/deviceDomain:
endPointHeader/agentVersion:
endPointHeader/contentVersion:
endPointHeader/protectionStatus:
messageData/userFullName:
messageData/username:
messageData/userRole:
messageData/userDomain:
messageData/messageName:
messageData/messageId:
messageData/processStatus:
messageData/errorText:
messageData/errorData:
messageData/resultData:
messageData/parameters:
messageData/additionalData: {}
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>generatedTime</td>
<td>Coordinated Universal Time (UTC) equivalent of the time at which an event was logged. For agent events, this represents the time on the endpoint. For policy, configuration, and system events, this represents the time on Cortex XDR in ISO-8601 string representation (for example, 2017-01-24T09:08:59Z).</td>
</tr>
<tr>
<td>serverTime</td>
<td>Coordinated Universal Time (UTC) equivalent of the time at which the server generated the log. If the log was generated on an endpoint, this field identifies the time the server received the log in ISO-8601 string representation (for example, 2017-01-24T09:08:59Z).</td>
</tr>
<tr>
<td>facility</td>
<td>The Cortex XDR system component that initiated the event, for example: TrapsAgent, TrapsServiceCore, TrapsServiceManagement, and TrapsServiceBackend.</td>
</tr>
<tr>
<td>customerId</td>
<td>The ID that uniquely identifies the Cortex Data Lake instance which received this log record.</td>
</tr>
<tr>
<td>trapsId</td>
<td>Tenant external ID.</td>
</tr>
<tr>
<td>serverHost</td>
<td>Hostname of Cortex XDR.</td>
</tr>
<tr>
<td>serverComponentVersion</td>
<td>Software version of Cortex XDR.</td>
</tr>
<tr>
<td>regionId</td>
<td>ID of Cortex XDR region:</td>
</tr>
<tr>
<td></td>
<td>• 10—Americas (N. Virginia)</td>
</tr>
<tr>
<td></td>
<td>• 70—EMEA (Frankfurt)</td>
</tr>
<tr>
<td>isEndpoint</td>
<td>Indicates whether the event occurred on an endpoint.</td>
</tr>
<tr>
<td></td>
<td>• 0—No, host is not an endpoint.</td>
</tr>
<tr>
<td></td>
<td>• 1—Yes, host is an endpoint.</td>
</tr>
<tr>
<td>agentId</td>
<td>Unique identifier for the Cortex XDR agent.</td>
</tr>
<tr>
<td>severity</td>
<td>Syslog severity level associated with the event.</td>
</tr>
<tr>
<td></td>
<td>• 2—Critical. Used for events that require immediate attention.</td>
</tr>
<tr>
<td></td>
<td>• 3—Error. Used for events that require special handling.</td>
</tr>
<tr>
<td></td>
<td>• 4—Warning. Used for events that sometimes require special handling.</td>
</tr>
<tr>
<td></td>
<td>• 5—Notice. Used for normal but significant events that can require attention.</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>trapsSeverity</td>
<td>Severity level associated with the event defined for Cortex XDR. Each of these severities corresponds to a syslog severity level:</td>
</tr>
<tr>
<td></td>
<td>• 0—Informational. Informational messages that do not require attention. Identical to the syslog 6 (Informational) severity level.</td>
</tr>
<tr>
<td></td>
<td>• 1—Low. Used for normal but significant events that can require attention. Corresponds to the syslog 5 (Notice) severity level.</td>
</tr>
<tr>
<td></td>
<td>• 2—Medium. Used for events that sometimes require special handling. Corresponds to the syslog 4 (Warning) severity level.</td>
</tr>
<tr>
<td></td>
<td>• 3—High. Used for events that require special handling. Corresponds to the syslog 3 (Error) severity level.</td>
</tr>
<tr>
<td></td>
<td>• 4—Critical. Used for events that require immediate attention. Corresponds to the syslog 2 (Critical) severity level.</td>
</tr>
<tr>
<td></td>
<td>See also the severity log field.</td>
</tr>
<tr>
<td>messageCode</td>
<td>System-wide unique message code.</td>
</tr>
<tr>
<td>friendlyName</td>
<td>Descriptive log message name.</td>
</tr>
<tr>
<td>msgTextEn</td>
<td>Description of the event, in English.</td>
</tr>
<tr>
<td>userFullName</td>
<td>Full username of Cortex XDR user.</td>
</tr>
<tr>
<td>userName</td>
<td>Username associated with Cortex XDR user.</td>
</tr>
<tr>
<td>userRole</td>
<td>Role assigned to Cortex XDR user.</td>
</tr>
<tr>
<td>userDomain</td>
<td>Domain to which the user belongs.</td>
</tr>
<tr>
<td>agentTime</td>
<td>Coordinated Universal Time (UTC) equivalent of the time at which an agent logged an event in ISO-8601 string representation.</td>
</tr>
<tr>
<td>tzOffset</td>
<td>Effective endpoint time zone offset from UTC, in minutes.</td>
</tr>
<tr>
<td>osType</td>
<td>Operating system of the endpoint:</td>
</tr>
<tr>
<td></td>
<td>• 1—Windows</td>
</tr>
<tr>
<td></td>
<td>• 2—OS X/macOS</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
|            | • 3—Android  
|            | • 4—Linux    |
| isVdi      | Indicates whether the endpoint is a virtual desktop  
|            | infrastructure (VDI):   
|            | • 0—The endpoint is not a VDI  
|            | • 1—The endpoint is a VDI    |
| osVersion  | Full version number of the operating system running on the endpoint. For example,  
|            | 6.1.7601.19135. |
| is64       | Indicates whether the endpoint is running a 64-bit version of Windows:  
|            | • 0—The endpoint is not running x64 architecture  
|            | • 1—The endpoint is running x64 architecture |
| agentIp    | IP address of the endpoint. |
| deviceName | Hostname of the endpoint on which the event was logged. |
| deviceDomain | Domain to which the endpoint belongs. |
| agentVersion | Version of the Cortex XDR agent. |
| contentVersion | Content version in the local security policy. |
| protectionStatus | Cortex XDR agent protection status:  
|            | • 0—Protected  
|            | • 1—OsVersionIncompatible  
|            | • 2—AgentIncompatible |
| userFullName | Full name of Cortex XDR user. |
| userName  | Username associated with Cortex XDR user. |
| userRole  | Role assigned to Cortex XDR user. |
| userDomain | Domain to which the user belongs. |
| messageName | Name of the message. |
| messageId | Unique numeric identifier of the message. |
| processStatus | State of the process related to the event. |
| errorText | If known, a description of the documented error. |
| errorData | Parameters related to an event error. |
Field Name | Description
--- | ---
resultData | Parameters related to a successful event.
parameters | Parameters supplied in the log message.
additionalData(Array) | Additional information regarding event parameters.
loggedInUser | User that is logged in to the Cortex XDR.

**Analytics Logs**

**Format:** recordType, class, FUTURE_USE, eventType, category, generatedTime, serverTime, agentTime, tzoffset, FUTURE_USE, facility, customerId, trapsId, serverHost, serverComponentVersion, regionId, isEndpoint, agentId, osType, isVdi, osVersion, is64, agentIp, deviceName, deviceDomain, severity, agentVersion, contentVersion, protectionStatus, sha256, type, parentSha256, lastSeen, fileName, filePath, fileSize, localAnalysisResult, reported, blocked, executionCount

**Email body format example:**

```plaintext
recordType: analytics
messageData/class: agent_data
messageData/subClass:
eventType: AgentTimelineEvent
messageData/eventCategory: hash
generatedTime: 2019-01-31T18:00:43Z
d endPointHeader/agentTime: 2019-01-31T18:00:43Z
d endPointHeader/tzOffset: -480
product:
facility: TrapsAgent
customerId: 110044035
trapsId: 18520039498190352
serverHost: coreop-f-proda-mnmauto03930348053-311.proda.brz
serverComponentVersion: 2.0.9+564
regionId: 10
isEndpoint: 1
agentId: 3bcf7e5ff56e2891c78684a38b728e49
endPointHeader/osType: 2
endPointHeader/isVdi: 0
endPointHeader/osVersion: 10.12.6
endPointHeader/is64: 1
endPointHeader/agentIp: 192.168.0.21
endPointHeader/deviceName: Jeffreys-MacBook-Pro.local
endPointHeader/deviceDomain: severity:
endPointHeader/agentVersion: 5.0.5.1193
endPointHeader/contentVersion: 42-6337
endPointHeader/protectionStatus: 0
messageData/sha256: 87e27ba9128d9c3b3d113c67623a06817a030b3bbb4d2871d1e6da9002206f26
messageData/type: macho
messageData/parentSha256:
messageData/lastSeen: 2019-01-31T18:00:43Z
messageData/fileName: crashpad_handler
messageData/filePath: /users/username/library/google/googlesoftwareupdate/googlesoftwareupdate.bundle/contents/macos/
messageData/fileSize: 353680
```
messageData/localAnalysisResult:
{""contentVersion":"42-6337",""result":"Benign",""trusted":"None",""publishers":[""developer id application: google, inc. (eqhxz8m8av)""],""resultId":0,""trustedId":0}"
messageData/reported: 0
messageData/blocked: 0
messageData/executionCount: 4179

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>recordType</td>
<td>Record type associated with the event and that you can use when managing logging quotas:</td>
</tr>
<tr>
<td></td>
<td>• <strong>config</strong>—Cortex XDR administration and configuration changes.</td>
</tr>
<tr>
<td></td>
<td>• <strong>system</strong>—Automated system management and agent reporting events.</td>
</tr>
<tr>
<td></td>
<td>• <strong>analytics</strong>—Hourly hash execution report from the agent.</td>
</tr>
<tr>
<td></td>
<td>• <strong>threats</strong>—Security events that occur on the endpoints.</td>
</tr>
<tr>
<td>class</td>
<td>Class of Cortex XDR log: config, policy, system, and agent_log.</td>
</tr>
<tr>
<td>eventType</td>
<td>Subtype of event.</td>
</tr>
<tr>
<td>eventCategory</td>
<td>Category of event, used internally for processing the flow of logs. Event categories vary by class:</td>
</tr>
<tr>
<td></td>
<td>• <strong>config</strong>—deviceManagement, distributionManagement, securityEventManagement, systemManagement</td>
</tr>
<tr>
<td></td>
<td>• <strong>policy</strong>—exceptionManagement, policyManagement, profileManagement, sam</td>
</tr>
<tr>
<td></td>
<td>• <strong>system</strong>—licensing, provisioning, tenant, userAuthentication, workerProcessing</td>
</tr>
<tr>
<td></td>
<td>• <strong>agent_log</strong>—agentFlow</td>
</tr>
<tr>
<td>generatedTime</td>
<td>Coordinated Universal Time (UTC) equivalent of the time at which an event was logged. For agent events, this represents the time on the endpoint. For policy, configuration, and system events, this represents the time on Cortex XDR in ISO-8601 string representation (for example, 2017-01-24T09:08:59Z).</td>
</tr>
<tr>
<td>serverTime</td>
<td>Coordinated Universal Time (UTC) equivalent of the time at which the server generated the log. If the log was generated on an endpoint, this field identifies the time the server received the log in ISO-8601 string representation (for example, 2017-01-24T09:08:59Z).</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>agentTime</td>
<td>Coordinated Universal Time (UTC) equivalent of the time at which an agent logged an event in ISO-8601 string representation.</td>
</tr>
<tr>
<td>tzOffset</td>
<td>Effective endpoint time zone offset from UTC, in minutes.</td>
</tr>
<tr>
<td>facility</td>
<td>The Cortex XDR system component that initiated the event, for example: TrapsAgent, TrapsServiceCore, TrapsServiceManagement, and TrapsServiceBackend.</td>
</tr>
<tr>
<td>customerId</td>
<td>The ID that uniquely identifies the Cortex Data Lake instance which received this log record.</td>
</tr>
<tr>
<td>trapsId</td>
<td>Tenant external ID.</td>
</tr>
<tr>
<td>serverHost</td>
<td>Hostname of Cortex XDR.</td>
</tr>
<tr>
<td>serverComponentVersion</td>
<td>Software version of Cortex XDR.</td>
</tr>
<tr>
<td>regionId</td>
<td>ID of Cortex XDR region:</td>
</tr>
<tr>
<td></td>
<td>• 10—Americas (N. Virginia)</td>
</tr>
<tr>
<td></td>
<td>• 70—EMEA (Frankfurt)</td>
</tr>
<tr>
<td>isEndpoint</td>
<td>Indicates whether the event occurred on an endpoint.</td>
</tr>
<tr>
<td></td>
<td>• 0—No, host is not an endpoint.</td>
</tr>
<tr>
<td></td>
<td>• 1—Yes, host is an endpoint.</td>
</tr>
<tr>
<td>agentId</td>
<td>Unique identifier for the Cortex XDR agent.</td>
</tr>
<tr>
<td>osType</td>
<td>Operating system of the endpoint:</td>
</tr>
<tr>
<td></td>
<td>• 1—Windows</td>
</tr>
<tr>
<td></td>
<td>• 2—OS X/macOS</td>
</tr>
<tr>
<td></td>
<td>• 3—Android</td>
</tr>
<tr>
<td></td>
<td>• 4—Linux</td>
</tr>
<tr>
<td>isVdi</td>
<td>Indicates whether the endpoint is a virtual desktop infrastructure (VDI):</td>
</tr>
<tr>
<td></td>
<td>• 0—The endpoint is not a VDI</td>
</tr>
<tr>
<td></td>
<td>• 1—The endpoint is a VDI</td>
</tr>
<tr>
<td>osVersion</td>
<td>Full version number of the operating system running on the endpoint. For example, 6.1.7601.19135.</td>
</tr>
<tr>
<td>is64</td>
<td>Indicates whether the endpoint is running a 64-bit version of Windows:</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>agentIp</td>
<td>IP address of the endpoint.</td>
</tr>
<tr>
<td>deviceName</td>
<td>Hostname of the endpoint on which the event was logged.</td>
</tr>
<tr>
<td>deviceDomain</td>
<td>Domain to which the endpoint belongs.</td>
</tr>
<tr>
<td>severity</td>
<td>Syslog severity level associated with the event:</td>
</tr>
<tr>
<td></td>
<td>• 2—Critical. Used for events that require immediate attention.</td>
</tr>
<tr>
<td></td>
<td>• 3—Error. Used for events that require special handling.</td>
</tr>
<tr>
<td></td>
<td>• 4—Warning. Used for events that sometimes require special handling.</td>
</tr>
<tr>
<td></td>
<td>• 5—Notice. Used for normal but significant events that can require attention.</td>
</tr>
<tr>
<td></td>
<td>• 6—Informational. Informational events that do not require attention.</td>
</tr>
<tr>
<td></td>
<td>Each event also has an associated Cortex XDR severity. See the <code>messageData.trapsSeverity</code> field for details.</td>
</tr>
<tr>
<td>agentVersion</td>
<td>Version of the Cortex XDR agent.</td>
</tr>
<tr>
<td>contentVersion</td>
<td>Content version in the local security policy.</td>
</tr>
<tr>
<td>protectionStatus</td>
<td>Cortex XDR agent protection status:</td>
</tr>
<tr>
<td></td>
<td>• 0—Protected</td>
</tr>
<tr>
<td></td>
<td>• 1—OsVersionIncompatible</td>
</tr>
<tr>
<td></td>
<td>• 2—AgentIncompatible</td>
</tr>
<tr>
<td>sha256</td>
<td>Hash of the file using SHA256 encoding.</td>
</tr>
<tr>
<td>type</td>
<td>Type of file:</td>
</tr>
<tr>
<td></td>
<td>• 0—Unknown</td>
</tr>
<tr>
<td></td>
<td>• 1—PE</td>
</tr>
<tr>
<td></td>
<td>• 2—Mach-o</td>
</tr>
<tr>
<td></td>
<td>• 3—DLL</td>
</tr>
<tr>
<td></td>
<td>• 4—Office file (containing a macro)</td>
</tr>
<tr>
<td>parentSha256</td>
<td>Hash of the parent file using SHA256 encoding.</td>
</tr>
<tr>
<td>lastSeen</td>
<td>Coordinated Universal Time (UTC) equivalent of the time when the file last ran on an endpoint in ISO-8601 string representation (for example, 2017-01-24T09:08:59Z).</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>fileName</td>
<td>File name, without the path or the file type extension.</td>
</tr>
<tr>
<td>filePath</td>
<td>Full path, aligned to the OS format.</td>
</tr>
<tr>
<td>fileSize</td>
<td>Size of the file in bytes.</td>
</tr>
<tr>
<td>localAnalysisResult</td>
<td>This object includes the content version, local analysis module version,</td>
</tr>
<tr>
<td></td>
<td>verdict result, file signer, and trusted signer result. The trusted signer</td>
</tr>
<tr>
<td></td>
<td>result is an integer value:</td>
</tr>
<tr>
<td></td>
<td>• 0—Cortex XDR did not evaluate the signer of the file.</td>
</tr>
<tr>
<td></td>
<td>• 1—The signer is trusted.</td>
</tr>
<tr>
<td></td>
<td>• 2—The signer is not trusted.</td>
</tr>
<tr>
<td>reported</td>
<td>Reporting status of the file, in integer value:</td>
</tr>
<tr>
<td></td>
<td>• 0—Cortex XDR did not report the security event.</td>
</tr>
<tr>
<td></td>
<td>• 1—Cortex XDR reported the security event.</td>
</tr>
<tr>
<td>blocked</td>
<td>Blocking status of the file, in integer value:</td>
</tr>
<tr>
<td></td>
<td>• 0—Cortex XDR did not block the process or file.</td>
</tr>
<tr>
<td></td>
<td>• 1—Cortex XDR blocked the process or file.</td>
</tr>
<tr>
<td>executionCount</td>
<td>The total number of times a file identified by a specific hash was executed.</td>
</tr>
</tbody>
</table>
Managed Security

- About Managed Security
- Cortex XDR Managed Security Access Requirements
- Set up Managed Threat Hunting
- Pair a Parent Tenant with Child Tenant
- Manage a Child Tenant
About Managed Security

Cortex XDR supports pairing multiple Cortex XDR environments with a single interface enabling Managed Security Services Providers (MSSP) and Managed Detection and Response (MDR) providers to easily manage security on behalf of their clients.

Pairing an MSSP/MDR (parent) tenant with a client (child) tenant requires a separate Cortex XDR license for the parent tenant. To ensure tenant access is acceptable to the parent and child, alike, both need to approve the pairing from within the Cortex XDR app.
Cortex XDR Managed Security Access Requirements

To set up a managed security pairing, you and your child tenants must activate the Cortex XDR app, provide role permission, and define access configurations.

The following table describes what and where you and your child tenants need to define:

<table>
<thead>
<tr>
<th>Tenant</th>
<th>Application</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Child</td>
<td>Customer Support Portal (CSP) Account</td>
<td>Add the user name from the parent tenant who is initiating the parent-child pairing.</td>
</tr>
<tr>
<td></td>
<td>Hub</td>
<td>Provide the user name added in CSP with Admin role permissions to access the child Cortex XDR instance.</td>
</tr>
<tr>
<td>Parent</td>
<td>Hub</td>
<td>Ensure the user name added to the child tenant’s CSP account has Admin role permissions on the parent Cortex XDR instance.</td>
</tr>
</tbody>
</table>
Set up Managed Threat Hunting

Cortex XDR provides the Managed Threat Hunting service as an add-on security service. To use Cortex XDR Managed Threat Hunting, you must purchase a Managed Threat Hunting license and have a Cortex XDR Pro for Endpoint license with a minimum of 500 endpoints.

Managed Threat Hunting augments your security by providing 24/7, year-round monitoring by Palo Alto Networks threat researchers and Unit 42 experts. The Managed Threat Hunting teams proactively safeguard your organization and provide threat reports for critical security incidents and impact reports for emerging threats that provide an analysis of exposure in your organization. In addition, the Managed Threat Hunting team can identify incidents and provide in-depth review of related threat reports.

To get started with Managed Threat Hunting:

STEP 1 | Access the Cortex XDR app and approve the pairing request sent to your Cortex XDR tenant.

After the request is approved, Cortex XDR displays the Managed Threat Hunting label at the top of the screen.

STEP 2 | Configure notification emails for the impact reports and threat inquiries you want Cortex XDR to send.

1. Select > Settings > Managed Threat Hunting.
2. Enter one or more email addresses to which you want to send reports and inquires and ADD each one.

3. Save your changes.

STEP 3 | (Optional) If desired, forward Managed Threat Hunting alerts to external sources such as email or slack from the > Settings > Notifications page.

This will forward both the alert itself and the detailed report in a PDF format.
Pair a Parent Tenant with Child Tenant

After you and your child tenants have acquired the appropriate role permissions, you can pair your tenant to your child tenants.

Pairing a Parent and Child Tenant

**STEP 1** | From your Cortex XDR app, select 📡 > Settings > Tenant Management.

The Tenant Management table displays the:

- **Tenant Name**—Name of the child tenant
- **Pairing Status**—State of a pairing request; Paired, Pending, Failed, Rejected
- **Account Name**—CSP account to which the child tenant is associated with
- **Last Sync**—Timestamp of when parent tenant last made contact with child tenant
- Managed Security Actions - a column for each security action with a status; configuration name or Unmanaged. Unmanaged status means that a configuration for the security action has not yet been selected.

**STEP 2** | + Pair Tenant.

**STEP 3** | In the Pair Tenant window, select the child tenant you want to pair. The drop-down only displays child tenants your are allowed to pair with.

Child tenants are grouped according to:

- **Unpaired**—Children that have not yet been paired and are available. If another parent has requested to pair with the child but the child has not yet agreed, the tenant will appear.
- **Paired**—Children that have already been paired to this parent.
- **Paired with others**—Children that have been paired with other parents.
- **Pending**—Children with a pending pairing request.

**STEP 4** | Pair the tenant.

Cortex XDR sends a Request for Pairing to the specified child tenant.

**STEP 5** | In the child tenant Cortex XDR console, a child tenant user with Admin role permissions needs to approve the pairing by navigating to 📊, locate the Request for Pairing notification and select Approve.

**STEP 6** | Verify the parent-child pairing.

After pairing has been approved, in the child tenant's Cortex XDR app, when navigating to a page managed by a parent configuration, the child user is notified by a flag who is managing their security:

In the child tenant’s, pages managed by you appear with a read-only banner. Child tenant users cannot perform any actions from these pages, but can view the configurations you create on their behalf.
Unpairing a Parent and Child Tenant

When you want to discontinue the pairing with a child tenant, in the Tenant Management page, right-click the tenant row and select Request Unpairing. For the unpairing to take effect, the child tenant must approve the request.

When a child wants to unpair, the child user needs to navigate to and select Unpair.
Manage a Child Tenant

Pairing a child tenant allows you to view and investigate the child tenant Cortex XDR data, and initiate security actions on their behalf.

In your Cortex XDR console you have access to view the following pages:

- Incidents
- Alerts
- Query Builder
- Query Center and Results
- Causality View
- Timeline View

To initiate security actions on your child tenant, you need to create a Configuration. Security actions are managed by configurations you create in the Cortex XDR app and then assign to each of the child tenants. Each action requires its own configuration and allocation to a child tenant.

You can create configuration for the following actions:

- BIOC Rules
- Exclusions
- Starred Alerts
- Profiles

The following sections describe how to manage your child tenants.

- Track your Tenant Management
- View Child Tenant Data
- Create and Allocate Action Configurations
- Initiate a Security Managed Action

Track your Tenant Management

After successfully pairing your child tenant, navigate to > Settings > Tenant Management to view the child tenant details.

The Tenant Management page displays the following information about each of your child tenants:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status Indicator (•)</td>
<td>Identifies whether the child tenant is connected.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TENANT ID</td>
<td>The Cortex Data Lake tenant ID.</td>
</tr>
<tr>
<td>TENANT NAME</td>
<td>Name you defined during the pairing process.</td>
</tr>
<tr>
<td>ACCOUNT ID</td>
<td>The CSP account ID.</td>
</tr>
<tr>
<td>ACCOUNT NAME</td>
<td>Name of the parent tenant.</td>
</tr>
<tr>
<td>PAIRING STATUS</td>
<td>Status of the child paring process:</td>
</tr>
<tr>
<td></td>
<td>• Pending</td>
</tr>
<tr>
<td></td>
<td>• Paired</td>
</tr>
<tr>
<td></td>
<td>• Approved</td>
</tr>
<tr>
<td></td>
<td>• Declined</td>
</tr>
<tr>
<td></td>
<td>• Pending</td>
</tr>
<tr>
<td></td>
<td>• Paired to another</td>
</tr>
<tr>
<td></td>
<td>• Not Paired</td>
</tr>
<tr>
<td>LAST SYNC</td>
<td>Timestamp of the last security action sync initiated by the parent tenant.</td>
</tr>
<tr>
<td>BIOC RULES &amp; EXCEPTIONS</td>
<td>Name of the configuration managing the BIOC rules and exceptions actions.</td>
</tr>
<tr>
<td>STARRED INCIDENTS POLICY</td>
<td>Name of the configuration managing the starred incidents policy actions.</td>
</tr>
<tr>
<td>ALERT EXCLUSION</td>
<td>Name of the configuration managing the alert exclusion actions.</td>
</tr>
<tr>
<td>PROFILES</td>
<td>Name of the configuration managing the profile actions.</td>
</tr>
</tbody>
</table>

**View Child Tenant Data**

Cortex XDR managed security allows you access to view Cortex XDR child tenant data.

When accessing the following pages, by default, your tenant data is displayed. To display the child tenant data, select the tenant from the drop-down.

Access the following child tenant data:
- **Incidents**
  1. Navigate to Investigation > Incidents.
  2. Right-click to View Incident.

    Cortex XDR opens the Incident View.
  3. In the Incident View page, view the incident details and right-click the event table rows to investigate further.

- **Alerts**
  1. Navigate to Investigation > Incidents > Alerts.
  2. In the Alerts table, right-click and select Analyze.

    Cortex XDR open the Causality View.
  3. In the Causality View page, view the alert details and right-click the event table rows to investigate further.

- **Query Builder**
  1. Navigate to Investigation > Query Builder.
  2. Run a query based on the child tenant cortex XDR data.
  3. View the query results.

- **Query Center**
  1. Navigate to Investigation > Query Center.

    The Query Center displays queries you ran on your tenant and the child tenants.
  2. Apply a filter to the Tenant filed to view queries according to the specific tenants.

### Create and Allocate Configurations

To manage security actions on behalf of your child tenant, you need to first create and allocate an action configuration.

**STEP 1** | Navigate to each of the following Cortex XDR app pages and follow the steps below:

1. Rules > BIOC (Requires Cortex XDR Pro per TB License)
2. Rules > Rules Exceptions (Requires Cortex XDR Pro per TB License)
3. Investigation > Exclusions
4. Investigation > Starred Alerts

**STEP 2** | In the Configuration panel (1), select + Create New (2).
STEP 3 | In the **Create New Configuration** window, enter the configuration **Name** and **Description**.

STEP 4 | **Create.**

A new configuration (3) appears in the **Configuration** panel.

STEP 5 | **Navigate to Settings > Tenant Management.**

STEP 6 | **In the Tenant Management** table, right-click a child tenant row and select **Edit Configurations.**

STEP 7 | **Assign the configuration you want to manage each of the security actions.**

You can only configure Profiles as Managed or Unmanaged, all profiles you create are automatically cloned to your child tenants.

STEP 8 | **Update.**

The **Tenant Management** table is updated with your assigned configurations.

Create a Security Manged Action

After you’ve created and assigned a configuration for each of your child tenant’s security actions, you can define the specific rules and exceptions.

STEP 1 | **Navigate to each of the following Cortex XDR pages:**

1. **Rules > BIOC** (Requires Cortex XDR Pro per TB License)
2. **Rules > Rules Exceptions** (Requires Cortex XDR Pro per TB License)
3. **Investigation > Exclusions**
4. **Investigation > Starred Alerts**
STEP 2 | In the corresponding **Configuration** panel (2), select **Rules** or **Exceptions** from your child configuration.

  The corresponding security action Table displays the rules/exceptions managing the child tenant.

STEP 3 | Depending on the security action, click:

  1. **Add BIOC** to create a BIOC Rule (Requires Cortex XDR Pro per TB License).
  2. **New Exception** to create a BIOC Exception (Requires Cortex XDR Pro per TB License).
  3. **Add Exclusion** to create an Alert Exclusion.
  4. **Add Starring Configuration** to create a started alert inclusion.
  5. **New Profile** to create a new endpoint profile.

 Profiles you create are automatically cloned to your child tenants.

You can now create a rule/exception on behalf of your child tenants.