Contact Information

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

About this Guide

This guide describes how to use the new features introduced in PAN-OS 7.1. For additional information, refer to the following resources:

- For information on the additional capabilities and for instructions on configuring the features on the firewall, refer to https://docs.paloaltonetworks.com or search the documentation.
- For access to the knowledge base and community forums, refer to https://live.paloaltonetworks.com.
- For contacting support, for information on support programs, to manage your account or devices, or to open a support case, refer to https://www.paloaltonetworks.com/services/solution-assurance.
- For the most current PAN-OS and Panorama 7.1 release notes, go to https://docs.paloaltonetworks.com/pan-os/7-1/pan-os-release-notes.

To provide feedback on the documentation, please write to us at: documentation@paloaltonetworks.com.
Networking Features

- Failure Detection with BFD
- LACP and LLDP Pre-Negotiation on an HA Passive Firewall
- Binding a Floating IP Address to an HA Active-Primary Firewall
- Multicast Route Setup Buffering
- Per-VLAN Spanning Tree (PVST+) BPDU Rewrite
- Configurable MSS Adjustment Size
- DHCP Client Support on Management Interface
- PA-3000 Series and PA-500 Firewall Capacity Increases
- SSL/SSH Session End Reasons
- Fast Identification and Mitigation of Sessions That Overutilize the Packet Buffer
Failure Detection with BFD

The firewall now supports **Bidirectional Forwarding Detection (BFD)**, a protocol that recognizes a failure in the bidirectional path between two routing peers. BFD failure detection provides a faster failover than can be achieved by link monitoring or frequent dynamic routing health checks, such as Hello packets or heartbeats. Mission-critical data centers and networks that require high availability and extremely fast failover need the extremely fast failure detection that BFD provides. PAN-OS supports BFD on PA-3000 Series, PA-5000 Series, PA-7000 Series, and VM-Series firewalls starting with PAN-OS 7.1.

<table>
<thead>
<tr>
<th>Configure BFD</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
</tr>
</tbody>
</table>
LACP and LLDP Pre-Negotiation on an HA Passive Firewall

Firewalls in a high availability (HA) active/passive configuration must be able to fail over quickly. If the firewall is using LACP or LLDP, the LACP or LLDP negotiation upon failover prevents sub-second failover. To preempt this negotiation after a passive firewall becomes active, you can now enable an interface on a passive firewall to negotiate LACP and LLDP before a failover occurs. The firewall in passive or non-functional HA state communicates with neighboring devices using LACP or LLDP. Such pre-negotiation speeds up failover because LACP or LLDP is already negotiated before the failover occurs.

The PA-3000 Series, PA-5000 Series, and PA-7000 Series firewalls support a pre-negotiation configuration (Enable in HA Passive State) as shown in the following table. An HA passive firewall handles LACP and LLDP packets in one of two ways:

- **Active**—The firewall has LACP or LLDP configured on the interface and actively participates in LACP or LLDP pre-negotiation respectively.
- **Passive**—LACP or LLDP is not configured on the interface and the firewall does not participate in the protocol but allows the peers on either side of the firewall to pre-negotiate LACP or LLDP respectively.

Use the information in the following topics to determine where and how you can enable LACP and LLDP pre-negotiation on an HA passive firewall:

- LACP and LLDP Pre-Negotiation Support
- Enable LACP and LLDP Pre-Negotiation on an HA Passive Firewall

LACP and LLDP Pre-Negotiation Support

The following table displays what deployments are supported on Ethernet and Aggregate Ethernet (AE) interfaces.

<table>
<thead>
<tr>
<th>Interface Deployment</th>
<th>AE Interface</th>
<th>Ethernet Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>LACP in Layer 2</td>
<td>Active</td>
<td>Not supported</td>
</tr>
<tr>
<td>LACP in Layer 3</td>
<td>Active</td>
<td>Not supported</td>
</tr>
<tr>
<td>LACP in Virtual Wire</td>
<td>Not supported</td>
<td>Passive</td>
</tr>
<tr>
<td>LLDP in Layer 2</td>
<td>Active</td>
<td>Active</td>
</tr>
<tr>
<td>LLDP in Layer 3</td>
<td>Active</td>
<td>Active</td>
</tr>
</tbody>
</table>
| LLDP in Virtual Wire | Active       | • Active if LLDP itself is configured.  
|                      |              | • Passive if LLDP itself is not configured. |

Pre-negotiation is not supported on subinterfaces or tunnel interfaces.
Enable LACP and LLDP Pre-Negotiation on an HA Passive Firewall

You should enable LACP and LLDP before you configure HA pre-negotiation if you want the firewall to actively participate in pre-negotiation respectively. Perform the following procedure on both firewalls to configure pre-negotiation.

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Configure active/passive HA.</th>
</tr>
</thead>
<tbody>
<tr>
<td>When you configure active/passive HA, set the link state to Auto.</td>
<td></td>
</tr>
<tr>
<td>2. Set the Passive Link State to Auto.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 2</th>
<th>Select an interface for LACP or LLDP pre-negotiation.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select Network &gt; Interfaces &gt; Ethernet and then enable appropriate pre-negotiation:</td>
<td></td>
</tr>
<tr>
<td>Enable LACP active pre-negotiation for an HA passive firewall</td>
<td></td>
</tr>
<tr>
<td>1. Select an Aggregate Ethernet (AE) interface in a Layer 2 or Layer 3 deployment.</td>
<td></td>
</tr>
<tr>
<td>2. On the LACP tab, select Enable in HA Passive State.</td>
<td></td>
</tr>
<tr>
<td>You cannot also select Same System MAC Address for Active-Passive HA because pre-negotiation requires unique interface MAC addresses on the active and passive firewalls.</td>
<td></td>
</tr>
<tr>
<td>3. Click OK.</td>
<td></td>
</tr>
<tr>
<td>Enable LACP passive pre-negotiation for an HA passive firewall</td>
<td></td>
</tr>
<tr>
<td>1. Select an Ethernet interface in a virtual wire deployment.</td>
<td></td>
</tr>
<tr>
<td>2. Select the Advanced tab.</td>
<td></td>
</tr>
<tr>
<td>3. On the LACP tab, select Enable in HA Passive State.</td>
<td></td>
</tr>
<tr>
<td>4. Click OK.</td>
<td></td>
</tr>
<tr>
<td>Enable LLDP active pre-negotiation for an HA passive firewall</td>
<td></td>
</tr>
<tr>
<td>If you want to allow LLDP passive pre-negotiation for a virtual wire deployment, perform this step but do not enable LLDP itself.</td>
<td></td>
</tr>
<tr>
<td>1. Select an Ethernet interface in Layer 2, Layer 3, or virtual wire deployment.</td>
<td></td>
</tr>
<tr>
<td>2. Select the Advanced tab.</td>
<td></td>
</tr>
<tr>
<td>3. On the LLDP tab, select Enable in HA Passive State.</td>
<td></td>
</tr>
<tr>
<td>4. Click OK.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 3</th>
<th>Save the configuration.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Click Commit.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 4</th>
<th>(Optional) Display pre-negotiation configuration and a summary of all interfaces on which pre-negotiation is enabled.</th>
</tr>
</thead>
<tbody>
<tr>
<td>In the web interface, select Dashboard &gt; Widgets &gt; System &gt; High Availability. The passive firewall indicates [Network pre-negotiation enabled]. Click that link to display a summary of all interfaces enabled with pre-negotiation.</td>
<td></td>
</tr>
<tr>
<td>In the CLI, use any of the following operational commands:</td>
<td></td>
</tr>
<tr>
<td>• show lacp aggregate-ethernet all</td>
<td></td>
</tr>
<tr>
<td>• show lldp config all</td>
<td></td>
</tr>
<tr>
<td>• show high-availability pre-negotiation summary</td>
<td></td>
</tr>
</tbody>
</table>
Networking Features

Binding a Floating IP Address to an HA Active-Primary Firewall

In a high availability (HA) active/active configuration, a Layer 3 deployment often uses floating IP addresses, which can move between HA firewalls to allow a persistent connection when a link or firewall fails. You can now bind a floating IP address to whichever firewall is in the active-primary state. Thus, on a failover, when the active-primary firewall (Peer A) goes down and the active-secondary firewall (Peer B) takes over as the active-primary peer, the floating IP address moves to Peer B. Peer B remains the active-primary firewall, even when Peer A recovers and becomes the active-secondary firewall. You control when Peer A becomes the active-primary firewall again.

In mission-critical data centers, you can benefit from binding the floating IP address to the active-primary firewall. You can have an HA active/active configuration for path monitoring out of both firewalls but have the firewalls function like an HA active/passive configuration because traffic directed to the floating IP address always goes to the active-primary firewall.

When you also disable preemption on both firewalls, you gain the following additional benefits:

- The floating IP address does not move back and forth between HA firewalls if the active-secondary firewall flaps up and down.
- You can run health checks on the recovered firewall before manually directing traffic to it again, which you can do during a convenient downtime.
- You have control over which firewall owns the floating IP address so that you keep all flows of new and existing sessions on the active-primary firewall, thereby minimizing traffic on the HA3 link.

### Bind a Floating IP Address to an HA Active-Primary Firewall

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Configure active/active HA with the floating IP address bound to the active-primary firewall.</th>
<th>The HA virtual address is a floating IP address that you bind to the active-primary firewall.</th>
</tr>
</thead>
</table>
| Step 2 | (Optional) Disable preemption. Disabling preemption gives you full control over when the recovered firewall becomes the active-primary firewall. | 1. Select Device > High Availability > General and edit the Election Settings.  
2. Clear Preemptive if checked.  
3. Click OK. |
| Step 3 | Enable link monitoring on the interface of the HA virtual address (floating IP address). | To enable link monitoring, see Step 1 in Define HA Failover Conditions. |
| Step 4 | Enable path monitoring. | To enable path monitoring, see Step 3 in Define HA Failover Conditions. |
| Step 5 | Next Steps... | Configure the HA peer firewall. |
Multicast Route Setup Buffering

You can now **enable buffering of the first packet in a multicast session** when the multicast route or forwarding information base (FIB) entry does not, yet, exist for the corresponding multicast group. By default, the firewall does not buffer the first multicast packet in a new session; instead, it uses the first packet to set up the multicast route. This is expected behavior for multicast traffic. You need to enable multicast route setup buffering only if your content servers are directly connected to the firewall and your custom application cannot withstand the first packet in the session being dropped.

### Enable Multicast Route Setup Buffering

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
</table>
| 1. **Enable buffering of the first packet in the multicast session.** | 1. Select **Device > Setup > Session** and edit the Session Settings.  
2. Select **Multicast Route Setup Buffering** to enable the feature.  
3. If you want to modify the buffer size, continue to the next step. Otherwise, click **OK** to save the setting. |
| 2. **(Optional) Tune the multicast route setup buffer.** | 1. In the Session Settings dialog, set the Buffer Size per flow to a value from 1-2,000.  
2. To save the settings, click **OK**. |
| 3. **(Optional) Tune the amount of time a multicast route stays in the routing table.** | 1. Select **Network > Virtual Router** and select the virtual router that handles the multicast traffic.  
2. Select **Multicast > Advanced.**  
3. Set the **Multicast Route Age Out Time (sec)** to a value from 210 (default) to 7,200.  
4. To save the settings, click **OK**. |
| 4. **Save the configuration.** | Click **Commit**. |
Networking Features

Per-VLAN Spanning Tree (PVST+) BPDU Rewrite

When an interface on the firewall is configured for a Layer 2 deployment, the firewall now rewrites the inbound Port VLAN ID (PVID) number in a Cisco per-VLAN spanning tree (PVST+) or Rapid PVST+ bridge protocol data unit (BPDU) to the proper outbound VLAN ID number and forwards the BPDU out. This default behavior beginning in PAN-OS 7.1 allows the firewall to correctly tag Cisco proprietary PVST+ and Rapid PVST+ frames between Cisco switches in VLANs on either side of the firewall so that spanning tree loop detection using Cisco PVST+ and Rapid PVST+ can function properly. The firewall is not participating in the Spanning Tree Protocol (STP) election process and there is no behavior change for other types of spanning tree.

The Cisco switch must have the loopguard disabled for the PVST+ or Rapid PVST+ BPDU rewrite to function properly on the firewall.

This feature is supported on Layer 2 Ethernet and Aggregated Ethernet (AE) interfaces only. The firewall supports a PVID range of 1-4,094 with a native VLAN ID of 1 to be compatible with the Cisco native VLAN implementation.

To support the PVST+ BPDU rewrite feature, PAN-OS now supports the concept of a PVST+ native VLAN. Frames sent to and received from a native VLAN are untagged with a PVID equal to the native VLAN. All switches and firewalls in the same Layer 2 deployment must have the same native VLAN for PVST+ to function properly. Although the Cisco native VLAN defaults to vlan1, the VLAN ID could be a number other than 1.

For example, the firewall is configured with a VLAN object (named VLAN_BRIDGE), which describes the interfaces and subinterfaces that belong to a switch or broadcast domain. In this example, the VLAN includes three subinterfaces: ethernet1/21.100 tagged with 100, ethernet1/22.1000 tagged with 1000, and ethernet1/23.1500 tagged with 1500.

The subinterfaces belonging to VLAN_BRIDGE look like this:

<table>
<thead>
<tr>
<th>Interface</th>
<th>Interface Type</th>
<th>Link State</th>
<th>Tag</th>
<th>VLAN / Virtual-Wire</th>
<th>Security Zone</th>
</tr>
</thead>
<tbody>
<tr>
<td>ethernet1/21</td>
<td>Layer2</td>
<td></td>
<td>Untagged</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>ethernet1/21.100</td>
<td>Layer2</td>
<td></td>
<td>100</td>
<td>VLAN_BRIDGE</td>
<td>Zone_Trust</td>
</tr>
<tr>
<td>ethernet1/22</td>
<td>Layer2</td>
<td></td>
<td>Untagged</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>ethernet1/22.1000</td>
<td>Layer2</td>
<td></td>
<td>1000</td>
<td>VLAN_BRIDGE</td>
<td>Zone_Untrust</td>
</tr>
<tr>
<td>ethernet1/23</td>
<td>Layer2</td>
<td></td>
<td>Untagged</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>ethernet1/23.1500</td>
<td>Layer2</td>
<td></td>
<td>1500</td>
<td>VLAN_BRIDGE</td>
<td>Zone_Management</td>
</tr>
</tbody>
</table>
The sequence in which the firewall automatically rewrites the PVST+ BPDU is shown in the following topology illustration:

- The Cisco switch port belonging to VLAN 100 sends a PVST+ BPDU—with the PVID and 802.1Q VLAN tag set to 100—to the firewall.
- The firewall interfaces and subinterfaces are configured as a Layer 2 interface type. The ingress subinterface on the firewall is tagged with VLAN 100, which matches the PVID and VLAN tag of the incoming BPDU so the firewall accepts the BPDU. The firewall floods the PVST+ BPDU to all other interfaces belonging to the same VLAN object (in this example, ethernet1/22.1000 and ethernet1/23.1500). If the VLAN tags did not match, the firewall would have, instead, dropped the BPDU.
- When the firewall floods the BPDU out through other interfaces (belonging to the same VLAN object), the firewall rewrites the PVID and any 802.1Q VLAN tags to match the VLAN tag of the egress interface. In this example, the firewall rewrites the BPDU PVID from 100 to 1000 for one subinterface and from 100 to 1500 for the second subinterface as the BPDU traverses the Layer 2 bridge on the firewall.
- Each Cisco switch receives the correct PVID and VLAN tag on the incoming BPDU and processes the PVST+ packet to detect possible loops in the network.

The following table describes different options you can use to manage PVST+ and Rapid PVST+ BPDUs.

<table>
<thead>
<tr>
<th>Manage PVST+ and Rapid PVST+ BPDUs</th>
</tr>
</thead>
<tbody>
<tr>
<td>The PVST+ and Rapid PVST+ BPDU rewrite of the PVID is enabled by default; you can globally disable and re-enable it.</td>
</tr>
</tbody>
</table>
### Manage PVST+ and Rapid PVST+ BPDUs (Continued)

**Set the native VLAN ID for the firewall (range is 1-4,094; default is 1).**

If the native VLAN ID on your switch is a value other than 1, you must set the native VLAN ID on the firewall to that same number; otherwise, the firewall will drop packets with that VLAN ID. This applies to trunked and non-trunked interfaces.

Use the following CLI operational command:

```cli
class session pvst-native-vlan-id <vid>
```

**Drop all STP BPDU packets.**

Examples of why you might want to drop all STP BPDU packets:

- If there is only one switch on each side of the firewall and no other connections between the switches that can cause a loop, then STP is not required and can be disabled on the switch or blocked by the firewall.
- If there is a misbehaving STP switch inappropriately flooding BPDUs, you can drop the STP packets at the firewall to stop the BPDU flood.

Use the following CLI operational command:

```cli
class session drop-stp-packet yes|no
```

**Verify whether PVST+ BPDU rewrite is enabled, view the PVST native VLAN ID, and determine whether the firewall is dropping all STP BPDU packets.**

Use the following CLI operational command:

```cli
admin@pa-vm-30-20> show vlan all
```

```
[...]
pvst+ tag rewrite: disabled
pvst native vlan id: 5
drop stp: disabled
total vlans shown: 1

<table>
<thead>
<tr>
<th>name</th>
<th>interface</th>
<th>virtual interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>bridge</td>
<td>ethernet1/1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ethernet1/2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ethernet1/1.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ethernet1/2.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ethernet1/2.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ethernet1/1.5</td>
<td></td>
</tr>
</tbody>
</table>
```

**Troubleshoot PVST+ BPDU errors.**

Use the following CLI operational command:

```cli
admin@pa-vm-30-20> show counter global
```

Look at the `flow_pvid_inconsistent` counter, which counts the number of times the 802.1Q Tag and PVID fields inside a PVST+ BPDU packet do not match.
Configurable MSS Adjustment Size

The maximum transmission unit (MTU) is a value indicating the largest number of bytes that can be transmitted in a single TCP packet. The MTU includes the length of headers so, when you subtract the number of bytes in the headers from the MTU, the remaining value is the maximum segment size (MSS), which is the maximum number of data bytes that can be transmitted in a single packet.

Based on typical IP header and TCP header lengths of 20 bytes each, releases prior to PAN-OS 7.1 used a fixed length of 40 bytes as the IPv4 adjustment size to determine the MSS and used 60 bytes for the IPv6 adjustment size. Beginning with PAN-OS 7.1, the value of the MSS adjustment size is configurable. This flexibility allows the firewall to pass traffic with headers totaling more (or less) than 40 bytes as needed. Encapsulation adds length to headers so it is helpful to be able to configure the MSS adjustment size to allow bytes, for example, to accommodate an MPLS header or for tunneled traffic that includes a VLAN tag.

<table>
<thead>
<tr>
<th>Modify the MSS Adjustment Size</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> Change the MSS adjustment size settings for a Layer 3 interface if the default settings are not what you need.</td>
</tr>
<tr>
<td>1. Select <strong>Network &gt; Interfaces</strong>, select <strong>Ethernet, VLAN, or Loopback</strong>, and select a Layer 3 interface.</td>
</tr>
<tr>
<td>2. On the <strong>Advanced</strong> tab, select <strong>Other Info</strong>.</td>
</tr>
<tr>
<td>3. Select <strong>Adjust TCP MSS</strong> and enter a value for one or both of the following:</td>
</tr>
<tr>
<td>- <strong>IPv4 MSS Adjustment Size</strong> (range is 40-300 bytes; default is 40 bytes).</td>
</tr>
<tr>
<td>- <strong>IPv6 MSS Adjustment Size</strong> (range is 60-300 bytes; default is 60 bytes).</td>
</tr>
<tr>
<td>4. Click <strong>OK</strong>.</td>
</tr>
</tbody>
</table>

| **Step 2** Save the configuration. |
| Click **Commit**. |
DHCP Client Support on Management Interface

In cloud and multi-tenant service provider environments, one goal is to automatically deploy and provision firewalls without manually assigning an IP address to the management interface. The management interface on the firewall now supports the option to enable a DHCP client for IPv4 so that the management interface can receive its IPv4 address from a DHCP server. The management interface also supports DHCP Option 12 and Option 61; these options allow the firewall to send its host name and client identifier, respectively, to DHCP servers.

DHCP client functionality on the management interface is enabled by default on VM-Series firewalls in the AWS and Azure public cloud deployments and on VMware NSX edition firewalls. VM-Series firewalls in these environments require this automation and must use the management interface as a DHCP client to obtain the IP address (instead of using a static IP address). For VM-Series firewalls on all other hypervisors and on all hardware-based firewalls, DHCP client functionality on the management interface is disabled by default. You can enable the DHCP client using the web interface or the CLI on the firewall, or as a part of the process to Bootstrap the Firewall.

WildFire and Panorama do not support DHCP on the management interface—you must assign a static IP address, instead.

Configure the Management Interface as a DHCP Client

| Configure the Management interface as a DHCP client. | The Management interface can receive its IPv4 address, netmask, and default gateway from the DHCP server. Additionally, you have the option to configure your firewall to receive its hostname and domain from the server. You can also send the hostname and client identifier of the firewall to the server if the orchestration system you use accepts this information. |
PA-3000 Series and PA-500 Firewall Capacity Increases

PA-3000 Series and PA-500 firewalls support larger networking capacities in PAN-OS 7.1 than in earlier PAN-OS releases. The increase is a doubling of capacity unless you executed the `debug system arp-mac-capacity increase` operational command (available in PAN-OS 6.1 and 7.0.1 releases) on a PA-3020 or PA-3050 firewall, in which case you already doubled the capacity.

The following table lists the capacities in PAN-OS 7.1 and later releases.

<table>
<thead>
<tr>
<th>Capacity Description</th>
<th>PA-500</th>
<th>PA-3020</th>
<th>PA-3050</th>
<th>PA-3060</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of entries in an ARP table per broadcast domain</td>
<td>2,000</td>
<td>3,000</td>
<td>5,000</td>
<td>10,000</td>
</tr>
<tr>
<td>Number of entries in an ARP table per firewall</td>
<td>2,000</td>
<td>3,000</td>
<td>5,000</td>
<td>10,000</td>
</tr>
<tr>
<td>Number of addresses in a MAC table per broadcast domain</td>
<td>2,000</td>
<td>3,000</td>
<td>5,000</td>
<td>5,000</td>
</tr>
<tr>
<td>Number of addresses in a MAC table per firewall</td>
<td>2,000</td>
<td>3,000</td>
<td>5,000</td>
<td>5,000</td>
</tr>
<tr>
<td>Number of addresses in a Forwarding Information Base (FIB) table per virtual router</td>
<td>1,250</td>
<td>5,000</td>
<td>10,000</td>
<td>10,000</td>
</tr>
<tr>
<td>Number of routes in a Forwarding Information Base (FIB) table per virtual router</td>
<td>1,250</td>
<td>5,000</td>
<td>10,000</td>
<td>10,000</td>
</tr>
<tr>
<td>Number of addresses in a FIB table per firewall</td>
<td>1,250</td>
<td>5,000</td>
<td>10,000</td>
<td>10,000</td>
</tr>
<tr>
<td>Number of addresses in an IPv6 Neighbor table per firewall</td>
<td>2,000</td>
<td>3,000</td>
<td>5,000</td>
<td>10,000</td>
</tr>
</tbody>
</table>

If you decide to downgrade to an older PAN-OS release and you want to keep the larger capacities, see *Downgrade While Maintaining Enhanced Capacities on PA-3050 Firewalls and PA-3020 Firewalls.*
SSL/SSH Session End Reasons

The Session End Reason column in Traffic logs now includes additional end reasons pertaining to terminated SSL/SSH sessions. You can use this information to troubleshoot access issues for internal users requesting external services or for external users requesting internal services. If a session ends for multiple reasons, the field displays only the highest priority reason based on the following list, where the first reason in the list is the highest priority (the decrypt- prefix indicates an SSL/SSH session end reason): threat, policy-deny, decrypt-cert-validation, decrypt-unsupport-param, decrypt-error, tcp-rst-from-client, tcp-rst-from-server, resources-unavailable, tcp-fin, tcp-reuse, decoder, aged-out, and unknown.

▲ Diagnose SSL/SSH Session Terminations
▲ Monitor SSL/SSH Session Termination Events

Diagnose SSL/SSH Session Terminations

The SSL/SSH session end reasons indicate that a session ended because you configured a firewall decryption rule with a Decryption Profile that blocks SSL forward proxy decryption or SSL inbound inspection when one (or more) of the following conditions occurs:

<table>
<thead>
<tr>
<th>SSL/SSH Session End Reason</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>decrypt-cert-validation</td>
<td>The session used client authentication or used a server certificate with one or more of the following conditions: expired, untrusted issuer, unknown status, or status verification time-out. This session end reason is also displayed when the server certificate produced a fatal error alert of type bad_certificate, unsupported_certificate, certificate_revoked, access_denied, or no_certificate_RESERVED (SSLv3 only).</td>
</tr>
<tr>
<td>decrypt-unsupport-param</td>
<td>The session used an unsupported protocol version, cipher, or SSH algorithm. This session end reason is also displayed when the session produced a fatal error alert of type unsupported_extension, unexpected_message, or handshake_failure.</td>
</tr>
<tr>
<td>decrypt-error</td>
<td>Firewall resources or the hardware security module (HSM) were unavailable for the session. This session end reason is also displayed when you configured the firewall to block SSL traffic that has SSH errors or that produced any fatal error alert other than those listed for the decrypt-cert-validation and decrypt-unsupport-param end reasons.</td>
</tr>
</tbody>
</table>
Monitor SSL/SSH Session Termination Events

To see session end reasons, select Monitor > Logs > Traffic and check the Session End Reason column. To configure a custom report that lists SSL/SSH session termination events, perform the following steps:

<table>
<thead>
<tr>
<th>Monitor SSL/SSH Session Termination Events</th>
</tr>
</thead>
</table>
| **Step 1** Configure the report parameters. | 1. Select Monitor > Manage Custom Reports and Add a new report.  
2. Enter a Name for the report.  
3. In the Database drop-down, under Detailed Logs, select Traffic Log.  
4. Select Scheduled to automatically run the report each night.  
5. In the Available Columns list, double-click Session_end_reason and any other columns you want the report to include. |
| **Step 2** Configure queries if you want to filter the report. This example shows how to configure the report to display only SSL/SSH session termination events as shown in the following screen capture. | Perform the following steps for each SSL/SSH session end reason:  
1. Set the Connector to or.  
2. Set the Attribute to Session End Reason.  
3. Set the Operator to equal.  
4. Set the Value to an SSL/SSH session end reason.  
5. Add the query to the Query Builder field. |
| **Step 3** Test and save the report. | 1. Run Now to test the report settings; a new tab within the dialog displays the report.  
2. Modify the settings as needed and then click OK and Commit. |
Fast Identification and Mitigation of Sessions That Overutilize the Packet Buffer

When a firewall exhibits signs of resource depletion, it might be experiencing an attack that is sending an overwhelming number of packets. In such events, the firewall starts buffering inbound packets. Now you can quickly identify the sessions that are using an excessive percentage of the packet buffer and mitigate their impact by discarding them.

- Identify Sessions That Use an Excessive Percentage of the Packet Buffer
- Discard a Session Without a Commit

Identify Sessions That Use an Excessive Percentage of the Packet Buffer

Perform the following task on any hardware-based firewall platform (not a VM-Series firewall) to identify the packet buffer percentage used, the top five sessions using more than two percent of the packet buffer, and the source IP addresses associated with those sessions; this information allows you to take appropriate actions.

### View Firewall Resource Usage, Top Sessions, and Session Details

**Step 1** View firewall resource usage, top sessions, and session details. Execute the following operational command in the CLI (sample output from the command follows):

```plaintext
admin@PA-7050> show running resource-monitor ingress-backlogs
-- SLOT:s1, DP:dp1 --
USAGE - ATOMIC: 92% TOTAL: 93%
TOP SESSIONS:
SESS-ID  PCT  GRP-ID  COUNT
6        92%  1       156
7        1732

SESSION DETAILS
SESS-ID  PROTO  SZONE  SRC  SPORT  DST  DPORT  IGR-IF  EGR-IF  APP
6       6     trust  192.168.2.35  55653  10.1.8.89  80  ethernet1/21  ethernet1/22  undecided

The command displays a maximum of the top five sessions that each use 2% or more of the packet buffer. To restrict the display output:

- On a PA-7000 Series platform, you can limit output to a slot, a dataplane, or both. For example:
  ```plaintext
  admin@PA-7050> show running resource-monitor ingress-backlogs slot s1
  ```

- On a PA-5000 Series platform, you can limit output to a dataplane. For example:
  ```plaintext
  admin@PA-7050> show running resource-monitor ingress-backlogs dp dp1
  ```
Networking Features

**Step 2** Use the command output to determine whether the source at the source IP address using a high percentage of the packet buffer is sending legitimate or attack traffic.

In the sample output above, a single-session attack is likely occurring. A single session (Session ID 6) is using 92% of the packet buffer for Slot 1, DP 1, and the application at that point is undecided.

- If you determine a single user is sending an attack and the traffic is not offloaded, you can Use the CLI to End a Single Attacking Session. At a minimum, you can Configure DoS Protection Against Flooding of New Sessions.

- On a hardware platform that has a field-programmable gate array (FPGA), the firewall offloads traffic to the FPGA when possible to increase performance. However, if the traffic is offloaded to hardware, clearing the session does not help because then the software must handle the barrage of packets. You should instead Discard a Session Without a Commit.

To see whether a session is offloaded or not, use the `show session id <session-id>` operational command in the CLI as shown in the following example. The `layer7 processing value` indicates completed for sessions offloaded or enabled for sessions not offloaded.

```
admin@PA-5800> show session id 60000184
Session 60000184

C2S flow:
- source: 1.1.42.15 [trust]
- dest: 1.2.27.99
- proto: 6
- sport: 55993
dport: 6081
- state: ACTIVE
- type: FLOW
- src user: unknown
- dst user: unknown
- offload: Yes

S2C flow:
- source: 1.2.27.99 [untrust]
- dest: 1.1.42.15
- proto: 6
- sport: 6081
dport: 55993
- state: ACTIVE
- type: FLOW
- src user: unknown
- dst user: unknown
- offload: Yes

DP
- index(local): 979920
- start time: Tue Oct 27 14:20:09 2015
- timeout: 1200 sec
- time to live: 1187 sec
- total byte count(c2s): 278
- total byte count(s2c): 378
- layer7 packet count(c2s): 3
- layer7 packet count(s2c): 3
- vsys: vsys1
- application: bitTorrent
- rule: rule1
- session to be logged at end: True
- session in session age: True
- session updated by RA peer: False
- layer7 processing: completed
- url filtering enabled: False
- session via syn-cookies: False
- session terminated on host: False
- captive portal session: False
- impress interface: ethernet1/21
- egress interface: ethernet1/22
- session 605 rule: N/A (class 0)
- tracker stage 7 proc: ctd decoder bypass
- end re-encrypt: unknown
```
Discard a Session Without a Commit

Perform this task to permanently discard a session, such as a session that is overloading the packet buffer. No commit is required; the session is discarded immediately after executing the command. The commands apply to both offloaded and non-offloaded sessions.

**Discard a Session Without a Commit**

**Step 1**  In the CLI, execute the following operational command on any hardware platform.
```
admin@PA-7050> request session-discard [timeout <seconds>] [reason <reason-string>] id <session-id>
```

**Step 2**  Verify that sessions that have been discarded.
```
admin@PA-7050> show session all filter state discard
```