Prefiltering and Prefilter Policies

- About Prefiltering, on page 1
- Best Practices for Prefiltering, on page 4
- Encapsulated Traffic Handling Best Practices, on page 5
- Requirements and Prerequisites for Prefilter Policies, on page 6
- Configure Prefiltering, on page 6
- Tunnel Zones and Prefiltering, on page 11
- Prefilter Policy Hit Counts, on page 14
- Large Flow Offloads, on page 14

About Prefiltering

Prefiltering is the first phase of access control, before the system performs more resource-intensive evaluation. Prefiltering is simple, fast, and early. Prefiltering uses limited outer-header criteria to quickly handle traffic. Compare this to subsequent evaluation, which uses inner headers and has more robust inspection capabilities.

Configure prefiltering to:

- Improve performance— The sooner you exclude traffic that does not require inspection, the better. You can fastpath or block certain types of plaintext, passthrough tunnels based on their outer encapsulation headers, without inspecting their encapsulated connections. You can also fastpath or block any other connections that benefit from early handling.
- Tailor deep inspection to encapsulated traffic—You can rezone certain types of tunnels, so that you can later handle their encapsulated connections using the same inspection criteria. Rezoning is necessary because after prefiltering, access control uses inner headers.

Prefiltering vs Access Control

Prefilter and access control policies both allow you to block and trust traffic, though the prefiltering "trust" functionality is called "fastpathing" because it skips more inspection. The following table explains this and other differences between prefiltering and access control, to help you decide whether to configure custom prefiltering.

If you do not configure custom prefiltering, you can only approximate—not replicate—prefilter functionality with early-placed Block and Trust rules in the access control policy.

Characteristic	Prefiltering	Access Control	For more information, see...
Primary function	Quickly fastpath or block certain types of plaintext, passthrough tunnels (see Encapsulation Conditions), or tailor subsequent inspection to their encapsulated traffic. Fastpath or block any other connections that benefit from early handling.	Inspect and control all network traffic, using simple or complex criteria, including contextual information and deep inspection results.	About Prefiltering, on page 1
Implementation	Prefilter policy. The prefilter policy is invoked by the access control policy.	Access control policy. The access control policy is a main configuration. In addition to invoking subpolicies, access control policies have their own rules.	About Prefilter Policies, on page 7 Associating Other Policies with Access Control
Sequence within access control	First. The system matches traffic to prefilter criteria before all other access control configurations.	—	—
Rule actions	Fewer. You can stop further inspection (Fastpath and Block) or allow further analysis with the rest of access control (Analyze).	More. Access control rules have a larger variety of actions, including monitoring, deep inspection, block with reset, and interactive blocking.	Tunnel and Prefilter Rule Components, on page 9 Access Control Rule Actions
Bypass capability	Fastpath rule action. Fastpathing traffic in the prefilter stage bypasses all further inspection and handling, including: • Security Intelligence • authentication requirements imposed by an identity policy • SSL decryption • access control rules • deep inspection of packet payloads • discovery • rate limiting	Trust rule action. Traffic trusted by access control rules is only exempt from deep inspection and discovery.	Introduction to Access Control Rules

Characteristic	Prefiltering	Access Control	For more information, see...
Rule criteria	Limited. Rules in the prefilter policy use simple network criteria: IP address, VLAN tag, port, and protocol. For tunnels, tunnel endpoint conditions specify the IP address of the routed interfaces of the network devices on either side of the tunnel.	Robust. Access control rules use network criteria, but also user, application, requested URL, and other contextual information available in packet payloads. Network conditions specify the IP address of source and destination hosts.	Tunnel vs Prefilter Rules, on page 8 Rule Condition Types
IP headers used (tunnel handling)	Outermost. Using outer headers allows you to handle entire plaintext, passthrough tunnels. For nonencapsulated traffic, prefiltering still uses "outer" headers—which in this case are the only headers.	Innermost possible. For a nonencrypted tunnel, access control acts on its individual encapsulated connections, not the tunnel as a whole.	Passthrough Tunnels and Access Control, on page 3
Rezone encapsulated connections for further analysis	Rezones tunneled traffic. Tunnel zones allow you to tailor subsequent inspection to prefiltered, encapsulated traffic.	Uses tunnel zones. Access control uses the tunnel zones you assign during prefiltering.	Tunnel Zones and Prefiltering, on page 11
Connection logging	Fastpathed and blocked traffic only. Allowed connections may still be logged by other configurations.	Any connection.	Other Connections You Can Log
Supported devices	Firepower Threat Defense only.	All.	Best Practices for Prefiltering, on page 4

Passthrough Tunnels and Access Control

Plaintext (nonencrypted) tunnels can encapsulate multiple connections, often flowing between discontinuous networks. These tunnels are especially useful for routing custom protocols over IP networks, IPv6 traffic over IPv4 networks, and so on.

An outer *encapsulation header* specifies the source and destination IP addresses of the *tunnel endpoints*—the routed interfaces of the network devices on either side of the tunnel. Inner *payload headers* specify the source and destination IP addresses of the encapsulated connections' actual endpoints.

Often, network security devices handle plaintext tunnels as *passthrough* traffic. That is, the device is not one of the tunnel endpoints. Instead, it is deployed between the tunnel endpoints and monitors the traffic flowing between them.

Some network security devices, such as Cisco ASA firewalls running Cisco ASA Software (rather than Firepower Threat Defense), enforce security policies using outer IP headers. Even for plaintext tunnels, these devices have no control over or insight into individual encapsulated connections and their payloads.

By contrast, the Firepower System leverages access control as follows:

- Outer header evaluation—First, prefiltering uses outer headers to handle traffic. You can block or fastpath entire plaintext, passthrough tunnels at this stage.
- Inner header evaluation—Next, the rest of access control (and other features such as QoS) use the innermost detectable level of headers to ensure the most granular level of inspection and handling possible.

  If a passthrough tunnel is not encrypted, the system acts on its individual encapsulated connections at this stage. You must *rezone* a tunnel (see Tunnel Zones and Prefiltering, on page 11) to act on all its encapsulated connections.

Access control has no insight into encrypted passthrough tunnels. For example, access control rules see a passthrough VPN tunnel as one connection. The system handles the entire tunnel using only the information in its outer, encapsulation header.

Best Practices for Prefiltering

Consider the following guidelines and limitations for prefiltering:

Best Practices for Fastpath Prefiltering

When you use the fastpath action in a prefilter rule, the matching traffic bypasses inspection and is simply transmitted through the device. Use this action for traffic that you can trust and that would not benefit from any of the security features available.

The following types of traffic are ideal for fastpathing. For example, you could configure the rules to fastpath any traffic from or to the IP addresses of the endpoints or servers. You can further limit the rule based on ports used.

- Site-to-site VPN traffic that is going through the device. That is, the device is not an endpoint in the VPN topology.
- Scanner traffic. Scanner probes can create a lot of false-positive responses from intrusion policies.
- Voice/video.
- Backups.
- Management traffic (sftunnel) that traverses Firepower Threat Defense devices. Performing deep inspection on management traffic (using access control policies) can cause issues. You can prefilter based on port TCP/8305 between the management center and managed devices.

Model Requirements

Prefiltering is supported on Firepower Threat Defense devices only. Prefilter configurations have no effect on other devices.

Prefilter-like Capabilities on Non-FTD Devices

For Classic devices (7000/8000 series, ASA FirePOWER, NGIPSv):

- Use early-placed Trust and Block access control rules to approximate prefilter functionality, keeping in mind the differences between the two features. See Prefiltering vs Access Control, on page 1.
- Match entire GRE-encapsulated tunnels using access control rules, with some limitations. See Port and ICMP Code Conditions.
- In high availability, sessions in plaintext tunnels such as GRE v0 or IP4-in-IP have state replication if they match a tunnel allow rule.
- (8000 series only) Device-specific fastpath rules can bypass access control, but cannot block traffic. See Configure Fastpath Rules (8000 Series).

Encapsulated Traffic Handling Best Practices

This topic discusses guidelines for the following types of encapsulated traffic:

- Generic Routing Encapsulation (GRE)
- Point-to-Point Protocol (PPTP)
- IPinIP
- IPv6inIP
- Teredo

GRE v1 and PPTP bypass outer flow processing

GRE v1 (sometimes referred to as *stateful GRE*) and PPTP traffic bypass outer flow processing.

Passenger flow processing is supported for GRE v0, IPinIP, IPv6inIP, and Teredo but the following limitations apply:

- Sessions are over a single tunnel that is not load-balanced
- There is no HA or clustering replication
- Primary and secondary flow relationships are not maintained
- Prefilter policy white and black lists are not supported

GRE v0 sequence number field must be optional

All endpoints sending traffic on the network must send GREv0 traffic with the sequence number field as optional; otherwise, the sequence number field is removed. RFC 1701 and RFC 2784 both specify the sequence field as optional.

How tunnels work with interfaces

Prefilter and access control policy rules are applied to all tunnel types on routed, transparent, inline-set, inline-tap, and passive interfaces.

References

For more information about the GRE and PPTP protocols, see the following:

- RFC 1701, RFC 2784, and RFC 2890 (GRE protocol v0)
- RFC 2637 (PPTP and GRE protocol v1)

Requirements and Prerequisites for Prefilter Policies

Model support

Firewall Threat Defense

Supported domains

Any

User roles

- Admin
- Access Admin
- Network Admin

Configure Prefiltering

To perform custom prefiltering, configure and deploy prefilter policies to managed devices, as a part of access control.

Only one person should edit a policy at a time, using a single browser window. If multiple users save the same policy, the last saved changes are retained. For your convenience, the system displays information on who (if anyone) is currently editing each policy. To protect the privacy of your session, a warning appears after 30 minutes of inactivity on the policy editor. After 60 minutes, the system discards your changes.

Procedure

Step 1 Choose **Policies** > **Access Control** > **Prefilter**.

Step 2 Click **New Policy** to create a custom prefilter policy.

A new prefilter policy has no rules and a default action of Analyze all tunnel traffic. It performs no logging or tunnel rezoning. You can also **Copy** ( ) or **Edit** ( ) an existing policy.

Step 3 Configure the prefilter policy's default action and its logging options.

- Default action—Choose a default action for supported plaintext, passthrough tunnels: **Analyze all tunnel traffic** (with access control) or **Block all tunnel traffic**.

- Default action logging—Click **Logging** ( ) next to the default action; see Logging Connections with a Policy Default Action. You can configure default action logging for blocked tunnels only.

Step 4 Configure tunnel and prefilter rules.

In a custom prefilter policy, you can use both kinds of rule, in any order. Create rules depending on the specific type of traffic you want to match and the actions or further analysis you want to perform; see Tunnel vs Prefilter Rules, on page 8.

Caution
Exercise caution when using tunnel rules to assign tunnel zones. Connections in rezoned tunnels may not match security zone constraints in later evaluation. For more information, see Tunnel Zones and Prefiltering, on page 11.

For detailed information on configuring rule components, see Tunnel and Prefilter Rule Components, on page 9 and Rule Management: Common Characteristics.

Step 5 Evaluate rule order. To move a rule, click and drag or use the right-click menu to cut and paste.

Properly creating and ordering rules is a complex task, but one that is essential to building an effective deployment. If you do not plan carefully, rules can preempt other rules or contain invalid configurations. For more information, see Best Practices for Access Control Rules.

Step 6 Save the prefilter policy.

Step 7 For configurations that support tunnel zone constraints, appropriately handle rezoned tunnels.

Match connections in rezoned tunnels by using tunnel zones as source zone constraints; see Configuring Interface Conditions.

Step 8 Associate the prefilter policy with the access control policy deployed to your managed devices.

See Associating Other Policies with Access Control.

Step 9 Deploy configuration changes; see Deploy Configuration Changes.

Note
When you deploy a prefilter policy, its rules are not applied on the existing tunnel sessions. Hence, traffic on an existing connection is not bound by the new policy that is deployed. In addition, the policy hit count is incremented only for the first packet of a connection that matches a policy. Thus, the traffic of an existing connection that could match a policy is omitted from the hit count. To have the policy rules effectively applied, clear the existing tunnel sessions, and then deploy the policy.

About Prefilter Policies

Prefiltering is a policy-based feature. In the Firepower System, an access control policy is a main configuration that invokes subpolicies and other configurations, including a prefilter policy.

Policy Components: Rules and Default Action

In a prefilter policy, *tunnel rules*, *prefilter rules*, and a *default action* handle network traffic:

- Tunnel and prefilter rules—First, rules in a prefilter policy handle traffic in the order you specify. Tunnel rules match specific tunnels only and support rezoning. Prefilter rules have a wider range of constraints and do not support rezoning. For more information, see Tunnel vs Prefilter Rules, on page 8.

- Default action (tunnels only)—If a tunnel does not match any rules, the default action handles it. The default action can block these tunnels, or continue access control on their individual encapsulated connections. You cannot rezone tunnels with the default action.

  There is no default action for nonencapsulated traffic. If a nonencapsulated connection does not match any prefilter rules, the system continues with access control.

Connection Logging

You can log connections fastpathed and blocked by the prefilter policy; see Other Connections You Can Log.

Connection events contain information on whether and how logged connections—including entire tunnels—were prefiltered. You can view this information in event views (workflows), dashboards, and reports, and use it as correlation criteria. Keep in mind that because fastpathed and blocked connections are not subject to deep inspection, associated connection events contain limited information.

Default Prefilter Policy

Every access control policy has an associated prefilter policy.

The system uses a default policy if you do not configure custom prefiltering. Initially, this system-provided policy passes all traffic to the next phase of access control. You can change the policy's default action and configure its logging options, but you cannot add rules to it or delete it.

Prefilter Policy Inheritance and Multitenancy

Access control uses a hierarchical implementation that complements multitenancy. Along with other advanced settings, you can lock a prefilter policy association, enforcing that association in all descendant access control policies. For more information, see Access Control Policy Inheritance.

In a multidomain deployment, the system displays policies created in the current domain, which you can edit. It also displays policies created in ancestor domains, which you cannot edit. To view and edit policies created in a lower domain, switch to that domain. The default prefilter policy belongs to the Global domain.

Tunnel vs Prefilter Rules

Whether you configure a tunnel or prefilter rule depends on the specific type of traffic you want to match and the actions or further analysis you want to perform.

Characteristic	Tunnel Rules	Prefilter Rules
Primary function	Quickly fastpath, block, or rezone plaintext, passthough tunnels.	Quickly fastpath or block any other connection that benefits from early handling.
Encapsulation and port/protocol criteria	Encapsulation conditions match only plaintext tunnels over selected protocols, listed in Encapsulation Conditions.	Port conditions can use a wider range of port and protocol constraints than tunnel rules; see Port and ICMP Code Conditions.
Network criteria	Tunnel endpoint conditions constrain the endpoints of the tunnels you want to handle; see Tunnel Endpoint Conditions.	Network conditions constrain the source and destination hosts in each connection; see Network Conditions.

Characteristic	Tunnel Rules	Prefilter Rules
Direction	Bidirectional or unidirectional (configurable). Tunnel rules are bidirectional by default, so they can handle all traffic between tunnel endpoints.	Unidirectional only (nonconfigurable). Prefilter rules match source-to-destination traffic only.
Rezone sessions for further analysis	Supported, using tunnel zones; see Tunnel Zones and Prefiltering, on page 11.	Not supported.

Tunnel and Prefilter Rule Components

State (Enabled/Disabled)

By default, rules are enabled. If you disable a rule, the system does not use it and stops generating warnings and errors for that rule.

Position

Rules are numbered, starting at 1. The system matches traffic to rules in top-down order by ascending rule number. The first rule that traffic matches is the rule that handles that traffic, regardless of rule type (tunnel vs prefilter).

Action

A rule's action determines how the system handles and logs matching traffic:

- Fastpath—Exempts matching traffic from all futher inspection and control, including access control, identity requirements, and rate limiting. Fastpathing a tunnel fastpaths all encapsulated connections.
- Block—Blocks matching traffic without further inspection of any kind. Blocking a tunnel blocks all encapsulated connections.
- Analyze—Allows traffic to continue to be analyzed by the rest of access control, using inner headers. If passed by access control and any related deep inspection, this traffic may also be rate limited. For tunnel rules, enables rezoning with the Assign Tunnel Zone option.

Direction (Tunnel Rules Only)

A tunnel rule's direction determines how the system source and destination criteria:

- Match tunnels only from source (unidirectional)—Match source-to-destination traffic only. Matching traffic must originate from one of the specified source interfaces or tunnel endpoints, and leave through one of the destination interfaces or tunnel endpoints.
- Match tunnels from source and destination (bidirectional)—Match both source-to-destination traffic and destination-to-source traffic. The effect is identical to writing two unidirectional rules, one the mirror of the other.

Prefilter rules are always unidirectional.

Assign Tunnel Zone (Tunnel Rules Only)

In a tunnel rule, assigning a tunnel zone (whether existing or created on the fly), *rezones* matching tunnels. Rezoning requires the Analyze action.

Rezoning a tunnel allows other configurations—such as access control rules—to recognize all the tunnel's encapsulated connections as belonging together. By using a tunnel's assigned tunnel zone as an interface constraint, you can tailor inspection to its encapsulated connections. For more information, see Tunnel Zones and Prefiltering, on page 11.

Caution Exercise caution when assigning tunnel zones. Connections in rezoned tunnels may not match security zone constraints in later evaluation. See Using Tunnel Zones, on page 11 for a brief walkthrough of a tunnel zone implementation, and a discussion of the implications of rezoning without explicitly handling rezoned traffic.

Conditions

Conditions specify the specific traffic the rule handles. Traffic must match all a rule's conditions to match the rule. Each condition type has its own tab in the rule editor.

You can prefilter traffic using the following *outer-header* constraints:

- Interface—Interface Conditions
- Network—Tunnel Endpoint Conditions or Network Conditions
- Port—Encapsulation Conditions or Port and ICMP Code Conditions
- VLAN—VLAN Conditions

You must constrain tunnel rules by encapsulation protocol.

Logging

A rule's logging settings govern the records the system keeps of the traffic it handles.

In tunnel and prefilter rules, you can log fastpathed and blocked traffic (the Fastpath and Block actions). For traffic subject to further analysis (the Analyze action), logging in the prefilter policy is disabled, although matching connections may still be logged by other configurations. Logging is performed on inner flows, not on the encapsulating flow. For more information, see Logging Connections with Tunnel and Prefilter Rules.

Comments

Each time you save changes to a rule you can add comments. For example, you might summarize the overall configuration for the benefit of other users, or note when you change a rule and the reason for the change.

You cannot edit or delete these comments after you save the rule.

Related Topics

Best Practices for Access Control Rules

Tunnel Zones and Prefiltering

Tunnel zones allow you to use prefiltering to tailor subsequent traffic handling to encapsulated connections.

A special mechanism is required because usually, the system handles traffic using the innermost detectable level of headers. This ensures the most granular level of inspection possible. But it also means that if a passthrough tunnel is not encrypted, the system acts on its individual encapsulated connections; see Passthrough Tunnels and Access Control, on page 3.

Tunnel zones solve this problem. During the first phase of access control (prefiltering) you can use outer headers to identify certain types of plaintext, pasthrough tunnels. Then, you can *rezone* those tunnels by assigning a custom *tunnel zone*.

Rezoning a tunnel allows other configurations—such as access control rules—to recognize all the tunnel's encapsulated connections as belonging together. By using a tunnel's assigned tunnel zone as an interface constraint, you can tailor inspection to its encapsulated connections.

Despite its name, a tunnel zone is not a security zone. A tunnel zone does not represent a set of interfaces. It is more accurate to think of a tunnel zone as a tag that, in some cases, replaces the security zone associated with an encapsulated connection.

Caution For configurations that support tunnel zone constraints, connections in rezoned tunnels do **not** match security zone constraints. For example, after you rezone a tunnel, access control rules can match its encapsulated connections with their newly assigned *tunnel* zone, but not with any original *security* zone.

See Using Tunnel Zones, on page 11 for a brief walkthough of a tunnel zone implementation, and a discussion of the implications of rezoning without explicitly handling rezoned traffic.

Configurations Supporting Tunnel Zone Constraints

Only access control rules support tunnel zone constraints.

No other configurations support tunnel zone constraints. For example, you cannot use QoS to rate limit a plaintext tunnel as a whole; you can only rate limit its individual encapsulated sessions.

Using Tunnel Zones

This example procedure summarizes how you might rezone GRE tunnels for further analysis, using tunnel zones. You can adapt the concepts described in this example to other scenarios where you need to tailor traffic inspection to connections encapsulated in plaintext, passthrough tunnels.

Consider a Firepower System deployment where your organization's internal traffic flows through the Trusted security zone. The Trusted security zone represents a set of sensing interfaces across multiple managed devices deployed in various locations. Your organization's security policy requires that you allow internal traffic after deep inspection for exploits and malware.

Internal traffic sometimes includes plaintext, passthrough, GRE tunnels between particular endpoints. Because the traffic profile of this encapsulated traffic is different from your "normal" interoffice activity—perhaps it is known and benign—you can limit inspection of certain encapsulated connections while still complying with your security policy.

In this example, after you deploy configuration changes:

- Plaintext, passthrough, GRE-encapsulated tunnels detected in the Trusted zone have their individual encapsulated connections evaluated by one set of intrusion and file policies.
- All other traffic in the Trusted zone is evaluated with a different set of intrusion and file policies.

You accomplish this task by *rezoning* GRE tunnels. Rezoning ensures that access control associates GRE-encapsulated connections with a custom *tunnel* zone, rather than their original Trusted *security* zone. Rezoning is required due to the way the Firepower System and access control handle encapsulated traffic; see Passthrough Tunnels and Access Control, on page 3 and Tunnel Zones and Prefiltering, on page 11.

Procedure

Step 1 Configure custom intrusion and file policies that tailor deep inspection to encapsulated traffic, and another set of intrusion and file policies tailored to nonencapsulated traffic.

Step 2 Configure custom prefiltering to rezone GRE tunnels flowing through the Trusted security zone.

Create a custom prefilter policy and associate it with access control. In that custom prefilter policy, create a tunnel rule (in this example, **`GRE_tunnel_rezone`**) and a corresponding tunnel zone (**`GRE_tunnel`**). For more information, see Configure Prefiltering, on page 6.

Table 1: GRE_tunnel_rezone Tunnel Rule

Rule Component	Description
Interface object condition	Match internal-only tunnels by using the Trusted security zone as both a Source Interface Object and Destination Interface Object constraint.
Tunnel endpoint condition	Specify the source and destination endpoints for the GRE tunnels used in your organization. Tunnel rules are bidirectional by default. If you do not change the **Match tunnels from...** option, it does not matter which endpoints you specify as source and which as destination.
Encapsulation condition	Match GRE traffic.
Assign Tunnel Zone	Create the **`GRE_tunnel`** tunnel zone, and assign it to tunnels that match the rule.
Action	Analyze (with the rest of access control).

Step 3 Configure access control to handle connections in rezoned tunnels.

In the access control policy deployed to your managed devices, configure a rule (in this example, **`GRE_inspection`**) that handles the traffic you rezoned. For more information, see Create and Edit Access Control Rules.

Table 2: GRE_inspection Access Control Rule

Rule Component	Description
Security zone condition	Match rezoned tunnels by using the GRE_tunnel security zone as a Source Zone constraint; see Interface Conditions.

Rule Component	Description
Action	Allow, with deep inspection enabled. Choose the file and intrusion policies tailored to inspect encapsulated internal traffic.

Caution
If you skip this step, the rezoned connections may match **any** access control rule not constrained by security zone. If the rezoned connections do not match any access control rules, they are handled by the access control policy default action. Make sure this is your intent.

Step 4 Configure access control to handle nonencapsulated connections flowing through the Trusted security zone.

In the same access control policy, configure a rule (in this example, **`internal_default_inspection`**) that handles non-rezoned traffic in the Trusted security zone.

Table 3: internal_default_inspection Access Control Rule

Rule Component	Description
Security zone condition	Match non-rezoned internal-only traffic by using the Trusted security zone as both a Source Zone and Destination Zone constraint.
Action	Allow, with deep inspection enabled. Choose the file and intrusion policies tailored to inspect nonencapsulated internal traffic.

Step 5 Evaluate the position of the new access control rules relative to preexisting rules. Change rule order if necessary.

If you place the two new access control rules next to each other, it does not matter which you place first. Because you rezoned GRE tunnels, the two rules cannot preempt each other.

Step 6 Save all changed configurations.

What to do next

- Deploy configuration changes; see Deploy Configuration Changes.

Creating Tunnel Zones

Procedure

Step 1 Choose **Objects** > **Object Management**.

Step 2 Chose **Tunnel Zone** from the list of object types.

Step 3 Click **Add Tunnel Zone**.

Step 4 Enter a **Name** and, optionally, a **Description**.

Step 5 Click **Save**.

What to do next

- Assign tunnel zones to plaintext, passthrough tunnels as part of custom prefiltering; see Configure Prefiltering, on page 6.

Prefilter Policy Hit Counts

Hit count indicates the number of times a policy rule has triggered for a matching connection.

For complete information on viewing prefilter policy hit counts, see Viewing Policy Hit Counts.

Large Flow Offloads

On devices that run FXOS (such as Firepower 4100/9300 chassis), certain traffic that you configure to be fastpathed by a prefilter policy is handled by the hardware (specifically, in the NIC), not by your Firepower Threat Defense software. Offloading these connection flows results in higher throughput and lower latency, especially for data-intensive applications such as large file transfers. This feature is especially useful for data centers. This is called *static flow offload*.

In addition, by default, Firepower Threat Defense devices offload flows based on other criteria, including trust. This is called *dynamic flow offload*.

Offloaded flows continue to receive limited stateful inspection, such as basic TCP flag and option checking. The system can selectively escalate packets to the firewall system for further processing if necessary.

Examples of applications that can benefit from offloading large flows are:

- High Performance Computing (HPC) Research sites, where the Firepower Threat Defense device is deployed between storage and high compute stations. When one research site backs up using FTP file transfer or file sync over NFS, the large amount of data traffic affects all connections. Offloading FTP file transfer and file sync over NFS reduces the impact on other traffic.
- High Frequency Trading (HFT), where the Firepower Threat Defense device is deployed between workstations and the Exchange, mainly for compliance purposes. Security is usually not a concern, but latency is a major concern.

The following flows can be offloaded:

- (Static flow offload only.) Connections that are fastpathed by the prefilter policy.
- Standard or 802.1Q tagged Ethernet frames only.
- (Dynamic flow offload only):
  - Inspected flows that the inspection engine decides no longer need inspection. These flows include:
    - Flows handled by access control rules that apply the Trust action and are based on security zone, source and destination network and port matching only.
    - TLS/SSL flows that are not selected for decryption using an SSL policy.

- Flows that are trusted by the Intelligent Application Bypass (IAB) policy either explicitly or due to exceeding flow bypass thresholds.
- Flows that match file or intrusion policies that result in trusting the flow.
- Any allowed flow that no longer needs to be inspected.

- The following IPS preprocessor inspected flows:
  - SSH and SMTP.
  - FTP preprocessor secondary connections.
  - Session Initiation Protocol (SIP) preprocessor secondary connections.
- Intrusion rules that use keywords (also referred to as *options*)

Important For details, exceptions, and limitations to the above, see Restrictions for flow offload implementation, on page 15.

Use Static Flow Offload

To offload eligible traffic to hardware, create a prefilter policy rule that applies the **Fastpath** action. Use prefilter rules for TCP/UDP, and tunnel rules for GRE.

(Not recommended.) To disable static flow offload and as a by-product, dynamic flow-offload, use FlexConfig to run the **no flow-offload enable** command. After deployment, you will have to reload the device to implement the change. For information about this command, see the *Cisco ASA Series Command Reference*, available from https://www.cisco.com/c/en/us/support/security/asa-5500-series-next-generation-firewalls/products-command-reference-list.html.

Use Dynamic Flow Offload

Dynamic flow offload is enabled by default.

To disable dynamic offload:

```
> configure flow-offload dynamic whitelist disable
```

To re-enable dynamic offload:

```
> configure flow-offload dynamic whitelist enable
```

Note that dynamic offload occurs only if static flow offload is enabled, regardless of whether prefiltering is configured.

Restrictions for flow offload implementation

Device limitations

The feature is supported on the following devices:

- Firepower 9300

Flows that cannot be offloaded

Do not attempt to offload these flow types:

- Any flows that do not use IPv4 addressing, such as IPv6 addressing.
- Flows for any protocol other than TCP, UDP, and GRE.

Note PPTP GRE connections cannot be offloaded.

- Flows on interfaces configured in passive, inline, or inline tap mode. Routed and switched interfaces are the only types supported.
- Flows that require inspection by Snort or other inspection engines. In some cases, such as FTP, the secondary data channel can be offloaded although the control channel cannot be offloaded.
- IPsec and TLS/DTLS VPN connections that terminate on the device.

Note The Secure Firewall 3100 and Secure Firewall 4200 platforms are exceptions to this restriction. These platforms support flow-offload for IPsec and DTLS VPN connections terminating to the device. For detailed information on this feature and its implementation, please refer to the release notes specific to the Secure Firewall 3100 and Secure Firewall 4200 platforms.

- Multicast flows in routed mode. They are supported in transparent mode if there are only two member interfaces in a bridge group.
- TCP Intercept flows.
- TCP state bypass flows. You cannot configure flow offload and TCP state bypass on the same traffic.
- Flows tagged with security groups.
- Reverse flows that are forwarded from a different cluster node, in the case of asymmetric flows in a cluster.
- Centralized flows in a cluster, if the flow owner is not the control unit.
- Flows that include IP options cannot be dynamically offloaded.

Additional limitations

Consider these operational restrictions when implementing flow offload:

- Flow offload and Dead Connection Detection (DCD) are not compatible. Do not configure DCD on connections that can be offloaded.
- If more than one flow that matches flow offload conditions are queued to be offloaded at the same time to the same location on the hardware, only the first flow is offloaded. The other flows are processed normally. This is called a *collision*. Use the **show flow-offload flow** command in the CLI to display statistics for this situation.

- Dynamic flow offload disables all TCP normalizer checks.
- Although offloaded flows pass through FXOS interfaces, statistics for these flows do not appear on the logical device interface. Thus, logical device interface counters and packet rates do not reflect offloaded flows.

Conditions for reversing offload

After a flow is offloaded, packets within the flow are returned to the firewall operating system for further processing if they meet these conditions:

- They include TCP options other than Timestamp.
- They are fragmented.
- They are subject to Equal-Cost Multi-Path (ECMP) routing, and ingress packets move from one interface to another.
- A change to the routing table applies to the flow. A packet is returned to the device to determine the new route.