



## Access Control Overview

---

- [About access control policies, on page 1](#)
- [Introduction to Rules, on page 2](#)
- [Access control policy default action, on page 4](#)
- [Deep inspection using file and intrusion policies, on page 6](#)
- [Access control policy inheritance, on page 9](#)
- [Best practices for application control, on page 10](#)
- [Best practices for access control rules, on page 14](#)

### About access control policies

Access control is a hierarchical policy that allows you to specify, inspect, and log network traffic. You can use the characteristics of each connection to allow, trust, block, or monitor the connection.

Each managed device can be assigned to one access control policy. The data that the policy's assigned (also known as targeted) devices collect about your network traffic can be used to filter and control that traffic based on:

- simple, easily determined transport and network layer characteristics: source and destination, port, protocol, and so on
- the latest contextual information on the traffic, including characteristics such as reputation, risk, business relevance, application used, or URL visited
- realm, user, user group, or ISE attribute
- custom Security Group Tag (SGT)
- characteristics of encrypted traffic; you can also decrypt this traffic for further analysis
- whether unencrypted or decrypted traffic contains a prohibited file, detected malware, or intrusion attempt
- time and day (on supported devices)

Each type of traffic inspection and control occurs where it makes the most sense for maximum flexibility and performance. For example, reputation-based blocking uses simple source and destination data, so it can block prohibited traffic early in the process. In contrast, detecting and blocking intrusions and exploits is a last-line of defense.

# Introduction to Rules

Rules in various policy types (access control, SSL, identity, and so on) exert granular control over network traffic. The system evaluates traffic against rules in the order that you specify, using a first-match algorithm.

Although these rules might include other configurations that are not consistent across policies, they share many basic characteristics and configuration mechanics, including:

- **Conditions:** Rule conditions specify the traffic that each rule handles. You can configure each rule with multiple conditions. Traffic must match all conditions to match the rule.
- **Action:** A rule's action determines how the system handles matching traffic. Note that even if a rule does not have an **Action** list you can choose from, the rule still has an associated action. For example, a custom network analysis rule uses a network analysis policy as its "action." As another example, QoS rules do not have an explicit action because all QoS rules do the same thing: rate limit traffic.
- **Position:** A rule's position determines its evaluation order. When using a policy to evaluate traffic, the system matches traffic to rules in the order you specify. Usually, the system handles traffic according to the first rule where all the rule's conditions match the traffic. (Monitor rules, which are designed to track and log, are an exception.) Proper rule order reduces the resources required to process network traffic, and prevents rule preemption.
- **Category:** To organize some rule types, you can create custom rule categories in each parent policy.
- **Logging:** For many rules, logging settings govern whether and how the system logs connections handled by the rule. Some rules (such as identity and network analysis rules) do not include logging settings because the rules neither determine the final disposition of connections, nor are they specifically designed to log connections. As another example, QoS rules do not include logging settings; you cannot log a connection simply because it was rate limited.
- **Comments:** For some rule types, each time you save changes, 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.



---

**Tip** A right-click menu in many policy editors provides shortcuts to many rule management options, including editing, deleting, moving, enabling, and disabling.

---

For more information, see the chapter that discusses the rules you're interested in (for example, access control rules).

## Related Topics

[Configuring Application Conditions and Filters](#)

## Filtering rules by device

Some policy editors allow you to filter your rule view by affected devices.

The system uses a rule's interface constraints to determine if the rule affects a device. If you constrain a rule by interface (security zone or interface group condition), the device where that interface is located is affected by that rule. Rules with no interface constraint apply to any interface, and therefore every device.

QoS rules are always constrained by interface.



**Note** The following procedure does not apply to the access control policy. To see the rules that apply to a specific device or set of devices in the access control policy, click the filter icon and select the devices.

### Procedure

- Step 1** In the policy editor, click **Rules**, then click **Filter by Device**.  
A list of targeted devices and device groups appears.
- Step 2** Check one or more check boxes to display only the rules that apply to those devices or groups. Or, check **All** to reset and display all of the rules.
- Tip**  
Hover your pointer over a rule criterion to see its value. If the criterion represents an object with device-specific overrides, the system displays the override value when you filter the rules list by only that device. If the criterion represents an object with domain-specific overrides, the system displays the override value when you filter the rules list by devices in that domain.
- Step 3** Click **OK**.

## Rule and other policy warnings

Policy and rule editors use icons to mark configurations that could adversely affect traffic analysis and flow. Depending on the issue, the system may warn you when you deploy or prevent you from deploying entirely.



**Tip** Hover your pointer over an icon to read the warning, error, or informational text.

**Table 1: Policy Error Icons**

Icon	Description	Example
Errors (✖)	If a rule or configuration has an error, you cannot deploy until you correct the issue, even if you disable any affected rules.	A rule that performs category and reputation-based URL filtering is valid until you assign a device that does not have a URL Filtering license. At that point, an error icon appears next to the rule, and you cannot deploy until you edit or delete the rule, unassign the device, or enable the license.

Icon	Description	Example
<b>Warning</b> (⚠)	You can deploy a policy that displays rule or other warnings. However, misconfigurations marked with warnings have no effect.  If you disable a rule with a warning, the warning icon disappears. It reappears if you enable the rule without correcting the underlying issue.	Preempted rules or rules that cannot match traffic due to misconfiguration have no effect. This includes conditions using empty object groups, application filters that match no applications, excluded LDAP users, invalid ports, and so on.  However, if a warning icon marks a licensing error or model mismatch, you cannot deploy until you correct the issue.
<b>Information</b> (i)	Information icons convey helpful information about configurations that may affect the flow of traffic. These issues do not prevent you from deploying.	The system might skip matching the first few packets of a connection against some rules, until the system identifies the application or web traffic in that connection. This allows connections to be established so that applications and HTTP requests can be identified.
<b>Rule Conflict</b> (⚡)	When you enable rule conflict analysis, this icon appears in the rule table for rules that have conflicts.	Conflicts include redundant rules, redundant objects, and shadowed rules. Redundant and shadowed rules do not match traffic because previous rules would already match the criteria. Redundant objects make your rules unnecessarily complex.

## Access control policy default action

A newly created access control policy directs its assigned devices to handle all traffic using its *default action*.

In a simple access control policy, the default action specifies how a device handles all traffic. In a more complex policy, the default action handles traffic that:

- is not fast pathed by the prefilter policy
- is not on a Security Intelligence block list
- is not blocked by the decryption policy (encrypted traffic only)
- matches none of the rules in the policy (except monitor rules, which match and log—but do not handle or inspect—traffic)

The access control policy default action can block or trust traffic without further inspection, or inspect traffic for intrusions and discovery data.



**Note** You cannot perform file or malware inspection on traffic handled by the default action. Logging for connections handled by the default action is initially disabled, though you can enable it.

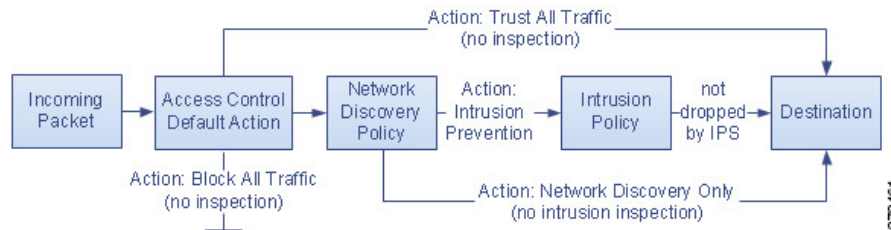
If you are using policy inheritance, the default action for the lowest-level descendant determines final traffic handling. Although an access control policy can inherit its default action from its base policy, you cannot enforce this inheritance.

The following table describes the types of inspection you can perform on traffic handled by each default action.

**Table 2: Access Control Policy Default Actions**

Default Action	Effect on Traffic	Inspection Type and Policy
Access Control: Block All Traffic	Block without further inspection.	None.
Access Control: Trust All Traffic	Trust (allow to its final destination without further inspection).	None.
Intrusion Prevention	Allow, as long as it is passed by the intrusion policy you specify.	Intrusion, using the specified intrusion policy and associated variable set.  Discovery, using the network discovery policy.
Network Discovery Only	Allow.	Discovery only, using the network discovery policy.
Inherit from base policy	Defined in base policy.	Defined in base policy.

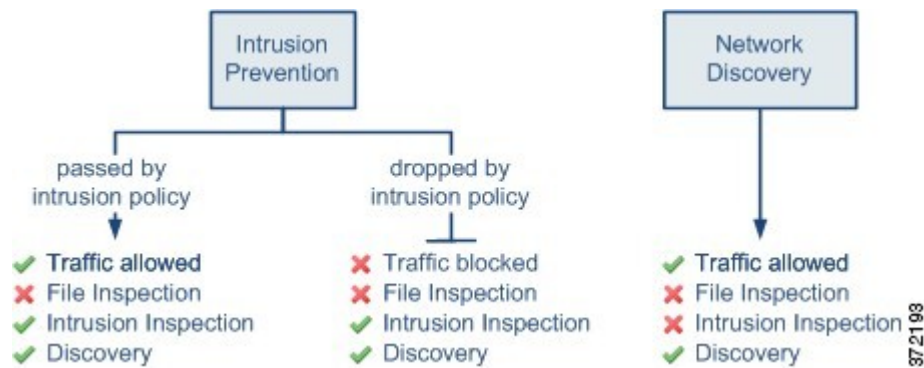
The following diagram illustrates the table.



The following diagrams illustrate the **Block All Traffic** and **Trust All Traffic** default actions.



The following diagrams illustrate the **Intrusion Prevention** and **Network Discovery Only** default actions.



372193



**Tip** The purpose of **Network Discovery Only** is to improve performance in a discovery-only deployment. Different configurations can disable discovery if you are only interested in intrusion detection and prevention.

## Deep inspection using file and intrusion policies

Deep inspection uses intrusion and file policies as the last line of defense before traffic is allowed to its destination.

- *Intrusion policies* govern the system's intrusion prevention capabilities.

For more information, see [Network Analysis and Intrusion Policy Basics](#).

- *File policies* govern the system's file control and malware defense capabilities.

For more information, see [Network Malware Protection and File Policies](#).

Access control occurs before deep inspection; access control rules and the access control default action determine which traffic is inspected by intrusion and file policies.

By associating an intrusion or file policy with an access control rule, you are telling the system that before it passes traffic that matches the access control rule's conditions, you first want to inspect the traffic with an intrusion policy, a file policy, or both.

In an access control policy, you can associate one intrusion policy with each Allow and Interactive Block rule, as well as with the default action. Every unique **pair** of intrusion policy and variable set counts as one policy.



**Note** By default, the system disables intrusion and file inspection of encrypted payloads. This helps reduce false positives and improve performance when an encrypted connection matches an access control rule that has intrusion and file inspection configured.

To associate intrusion and file policies with an access control rule, see:

- [Access Control Rules: Intrusion Policy Selection](#)
- [Configuring an Access Control Rule to Perform Malware Protection](#)

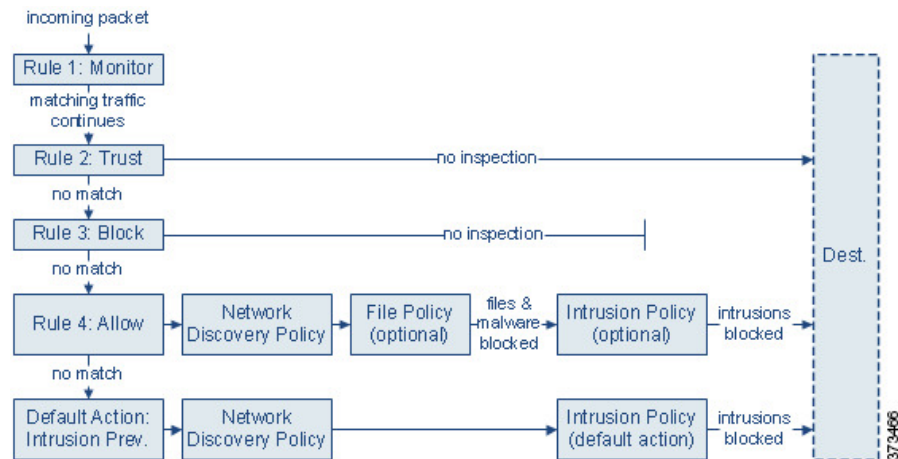
## Related Topics

[How Policies Examine Traffic For Intrusions](#)

[File Policies](#)

# Access control traffic handling with intrusion and file policies

The following diagram shows the flow of traffic in an inline intrusion prevention and malware defense deployment, as governed by an access control policy that contains four different types of access control rules and a default action.



In the scenario above, the first three access control rules in the policy—Monitor, Trust, and Block—cannot inspect matching traffic. Monitor rules track and log but do not inspect network traffic, so the system continues to match traffic against additional rules to determine whether to permit or deny it. (However, see an important exception and caveat at [Access Control Rule Monitor Action](#).) Trust and Block rules handle matching traffic without further inspection of any kind, while traffic that does not match continues to the next access control rule.

The fourth and final rule in the policy, an Allow rule, invokes various other policies to inspect and handle matching traffic, in the following order:

- Discovery: Network Discovery Policy**—First, the network discovery policy inspects traffic for discovery data. Discovery is passive analysis and does not affect the flow of traffic. Although you do not explicitly enable discovery, you can enhance or disable it. However, allowing traffic does not automatically guarantee discovery data collection. The system performs discovery only for connections involving IP addresses that are explicitly monitored by your network discovery policy.
- malware defense and File Control: File Policy**—After traffic is inspected by discovery, the system can inspect it for prohibited files and malware. malware defense detects and optionally blocks malware in many types of files, including PDFs, Microsoft Office documents, and others. If your organization wants to block not only the transmission of malware files, but all files of a specific type (regardless of whether the files contain malware), *file control* allows you to monitor network traffic for transmissions of specific file types, then either block or allow the file.
- Intrusion Prevention: Intrusion Policy**—After file inspection, the system can inspect traffic for intrusions and exploits. An intrusion policy examines decoded packets for attacks based on patterns, and can block or alter malicious traffic. Intrusion policies are paired with *variable sets*, which allow you to use named values to accurately reflect your network environment.

- **Destination**—Traffic that passes all the checks described above passes to its destination.

An Interactive Block rule (not shown in the diagram) has the same inspection options as an Allow rule. This is so you can inspect traffic for malicious content when a user bypasses a blocked website by clicking through a warning page.

Traffic that does not match any access control rules in the policy with an action other than Monitor is handled by the default action. In this scenario, the default action is an Intrusion Prevention action, which allows traffic to its final destination as long as it is passed by the intrusion policy you specify. In a different deployment, you might have a default action that trusts or blocks all traffic without further inspection. Note that the system can inspect traffic allowed by the default action for discovery data and intrusions, but not prohibited files or malware. You **cannot** associate a file policy with the access control default action.




---

**Note** Sometimes, when a connection is analyzed by an access control policy, the system must process the first few packets in that connection, **allowing them to pass**, before it can decide which access control rule (if any) will handle the traffic. However, so these packets do not reach their destination uninspected, you can specify an intrusion policy (in the advanced settings for the access control policy) to inspect these packets and generate intrusion events.

---

## File and intrusion inspection order

In your access control policy, you can associate multiple Allow and Interactive Block rules with different intrusion and file policies to match inspection profiles to various types of traffic.

You do not have to perform both file and intrusion inspection in the same rule. For a connection matching an Allow or Interactive Block rule:

- without a file policy, traffic flow is determined by the intrusion policy
- without an intrusion policy, traffic flow is determined by the file policy
- without either, allowed traffic is inspected by network discovery only

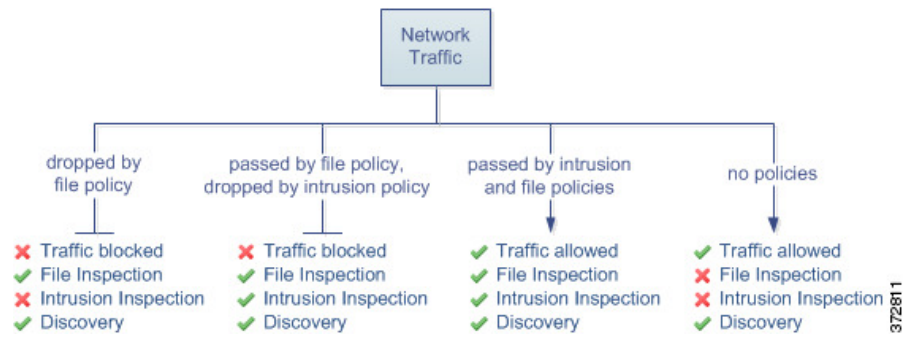



---

**Tip** The system does not perform any kind of inspection on trusted traffic. Although configuring an Allow rule with neither an intrusion nor file policy passes traffic like a Trust rule, Allow rules let you perform discovery on matching traffic.

---

The diagram below illustrates the types of inspection you can perform on traffic that meets the conditions of either an Allow or user-bypassed Interactive Block access control rule. For simplicity, the diagram displays traffic flow for situations where both (or neither) an intrusion and a file policy are associated with a single access control rule.



For any single connection handled by an access control rule, file inspection occurs before intrusion inspection. That is, the system does not inspect files blocked by a file policy for intrusions. Within file inspection, simple blocking by type takes precedence over malware inspection and blocking.

For example, consider a scenario where you normally want to allow certain network traffic as defined in an access control rule. However, as a precaution, you want to block the download of executable files, examine downloaded PDFs for malware and block any instances you find, and perform intrusion inspection on the traffic.

You create an access control policy with a rule that matches the characteristics of the traffic you want to provisionally allow, and associate it with both an intrusion policy and a file policy. The file policy blocks the download of all executables, and also inspects and blocks PDFs containing malware:

- First, the system blocks the download of all executables, based on simple type matching specified in the file policy. Because they are immediately blocked, these files are subject to neither malware nor intrusion inspection.
- Next, the system performs malware cloud lookups for PDFs downloaded to a host on your network. Any PDFs with a malware disposition are blocked, and are not subject to intrusion inspection.
- Finally, the system uses the intrusion policy associated with the access control rule to inspect any remaining traffic, including files not blocked by the file policy.



**Note** Until a file is detected and blocked in a session, packets from the session might be subject to intrusion inspection.

## Access control policy inheritance

You can nest access control policies, where each policy inherits the rules and settings from an ancestor (or *base*) policy. You can enforce this inheritance, or allow lower-level policies to override their ancestors.

Access control uses a hierarchical policy-based implementation. Just as you create a domain hierarchy, you can create a corresponding hierarchy of access control policies. A *descendant*, or *child*, access control policy inherits rules and settings from its direct *parent*, or *base*, policy. That base policy may have its own parent policy from which it inherits rules and settings, and so on.

An access control policy's rules are nested between its parent policy's Mandatory and Default rule sections. This implementation enforces Mandatory rules from ancestor policies, but allows the current policy to write rules that preempt Default rules from ancestor policies.

You can lock the following settings to enforce them in all descendant policies. Descendant policies can override unlocked settings.

- Security Intelligence—Connections that are allowed or blocked based on the latest reputation intelligence for IP addresses, URLs, and domain names.
- HTTP Response pages—Displaying a custom or system-provided response page when you block a user's website request.
- Advanced settings—Specifying associated subpolicies, network analysis settings, performance settings, and other general options.

When using policy inheritance, the default action for the lowest-level descendant determines final traffic handling. Although an access control policy can inherit its default action from an ancestor policy, you cannot enforce this inheritance.

## Best practices for application control

The following topics discuss our recommended best practices for controlling applications with access control rules.

### Recommendations for application control

Keep in mind the following guidelines and limitations for application control:

#### Ensure that adaptive profiling is enabled

If adaptive profiling is not enabled (its default state), access control rules cannot perform application control.

#### Enable application detectors automatically

If no detector is enabled for an application you want to detect, the system automatically enables all system-provided detectors for the application. If none exist, the system enables the most recently modified user-defined detector for the application.

#### Configure your policy to examine the packets that must pass before an application is identified

The system cannot perform application control before *both* of the following occur:

- A monitored connection is established between a client and server
- The system identifies the application in the session

This identification should occur in 3 to 5 packets, or after the server certificate exchange in the SSL handshake if the traffic is encrypted. If you configure the access control rule to use application default ports, the application rule can be enforced without allowing initial packets to pass.

If early traffic matches all other criteria but application identification is incomplete, the system allows the packet to pass and the connection to be established (or the SSL handshake to complete). After the system completes its identification, the system applies the appropriate action to the remaining session traffic.

To ensure that your system examines these initial packets, select an intrusion policy in the **Intrusion Policy used before Access Control rule is determined** options in the access control policy advanced settings.

### Understand limitations to identifying applications

A server must adhere to the protocol requirements of an application for the system to be able to recognize it. For example, if you have a server that sends a keep-alive packet rather than an ACK when an ACK is expected, that application might not be identified, and the connection will not match the application-based rule. Instead, it will be handled by another matching rule or the default action. This might mean that connections you want to allow can be denied instead. If you run into this problem, and you cannot fix the server to follow the protocol standards, you need to write a non-application-based rule to cover traffic for that server, for example, by matching the IP address and port number.

### Create separate rules for URL and application filtering

Create separate rules for URL and application filtering whenever possible, because combining application and URL criteria can lead to unexpected results, especially for encrypted traffic.

Rules that include both application and URL criteria should come after application-only or URL-only rules, unless the application+URL rule is acting as an exception to a more general application-only or URL-only rule.

### Place URL rules before application and other rules

For the most effective URL matching, place rules that include URL conditions before other rules, particularly if the URL rules are block rules and the other rules meet both of the following criteria:

- They include application conditions.
- The traffic to be inspected is encrypted.

### Application control for encrypted and decrypted traffic

The system can identify and filter encrypted and decrypted traffic:

- Encrypted traffic—The system can detect application traffic encrypted with StartTLS, including SMTPS, POPS, FTPS, TelnetS, and IMAPS. In addition, it can identify certain encrypted applications based on the Server Name Indication in the TLS ClientHello message, or the subject distinguished name value from the server certificate. These applications are tagged `SSL Protocol`; in a decryption policy rule, you can choose only these applications. Applications without this tag can only be detected in unencrypted or decrypted traffic.
- Decrypted traffic—The system assigns the `decrypted traffic` tag to applications that the system can detect in decrypted traffic only, not encrypted or unencrypted.

### TLS server identity discovery and application control

The latest version of the Transport Layer Security (TLS) protocol 1.3, defined by [RFC 8446](#), is the preferred protocol for many web servers to provide secure communications. Because the TLS 1.3 protocol encrypts the server's certificate for additional security, and the certificate is needed to match application and URL filtering criteria in access control rules, the Firepower System provides a way to extract the server certificate *without* decrypting the entire packet.

We strongly recommend enabling it for any traffic you want to match on application or URL criteria, especially if you want to perform deep inspection of that traffic. A decryption policy is not required because *traffic is not decrypted* in the process of extracting the server certificate.

For more information, see [Access control policy advanced settings](#).

### Exempting applications from active authorization

In an identity policy, you can exempt certain applications from active authentication, allowing traffic to continue to access control. These applications are tagged `User-Agent Exclusion`. In an identity rule, you can choose only these applications.

### Handling application traffic packets without payloads

When performing access control, the system applies the default policy action to packets that do not have a payload in a connection where an application is identified.

### Handling referred application traffic

To handle traffic referred by a web server, such as advertisement traffic, match the referred application rather than the referring application.

### Controlling application traffic that uses multiple protocols

Some applications use multiple protocols. To control their traffic, make sure your access control policy covers all relevant options. These applications typically include the application name in each aspect that you can control. Also see [Application-specific notes and limitations, on page 13](#).

### Controlling evasive application traffic

See [Application-specific notes and limitations, on page 13](#).

## Choosing between application matching and port matching

In traditional firewalls, you can match traffic based on OSI layers 3 (protocol) and 4 (transport), such as IP or TCP/80. All traffic on a given port (or a whole protocol) is then either allowed or blocked based on that rule's action.

Application criteria, on the other hand, are OSI layer 7. Different applications can use the same TCP/UDP port. By using application criteria, you can selectively allow or block different applications on the same port without allowing or blocking all applications on that port.

Whether you use port-based or application-based criteria in your rule can impact rule performance. Because TCP/UDP ports are quickly identifiable in packets, the system can match the correct rule on the first packet. With application-layer criteria, it can take 3-5 packets to identify the specific application (if you do not also specify ports).

Consider the following recommendations.

- If you want to handle all of the traffic on a given TCP/UDP port in the same way for the specified interfaces and networks, use port-based matching. For example, to handle all SSH traffic the same way, select the SSH port (TCP/22) on the Ports tab.
- If you want to narrowly match a specific application that uses the same port as other applications, select the application on the Applications tab. This is how you can handle web applications, which all use TCP/80 or 443, so that you can selectively block or allow certain web applications without blocking or allowing all web applications.
- If you want to control application use by a user group, select the user group on the Users tab and the application on the Application tab. For example, you could block the Gaming applications category for members of the Contractor user group.

You can also allow or block protocol/port by user group if the rule should apply to all connections for a protocol/port.

- For rules that go from a less secure network (such as the internet) to a more secure network (such as an internal protected network), use the Ports tab whenever possible. For example, you can allow/block ICMP traffic from the internet to an internal network.
- Avoid mixing Port tab and Applications tab specifications in a single rule. Instead, when you select an application, leave the Port tab empty, and instead specify the port as **Application Default** on the Applications tab. This will target the rule to the right ports (per application) and enhance how quickly the right application can be matched to the rule.
- If you do have a mix of port-based and application-based rules, place the port-based rules higher in the rules list, so that connections can be matched to those rules first. Protocol and port can be more quickly identified than applications.

## Application-specific notes and limitations

- Office 365 Admin Portal—If the access policy has logging enabled at the beginning as well as at the end of a connection, the first packet will be detected as Office 365 and the end of connection will be detected as Office 365 Admin Portal. This should not affect blocking.
- Skype—To control Skype traffic, choose the **Skype** tag from the **Application Filters** list rather than selecting individual applications. This ensures that the system can detect and control all Skype traffic the same way.
- GoToMeeting—In order to fully detect GoToMeeting, your rule must include all of the following applications:
  - GoToMeeting
  - Citrix Online
  - Citrix GoToMeeting Platform
  - LogMeIn
  - STUN
- Zoho—To control Zoho mail, choose *both* **Zoho** and **Zoho mail** from the Available Application list.
- Evasive applications such as Bittorrent, Tor, Psiphon, and Ultrasurf—For evasive applications, only the highest-confidence scenarios are detected by default. If you need to take action on this traffic (such as block or implement QoS), it may be necessary to configure more aggressive detection with better effectiveness. To do this, contact TAC to review your configurations as these changes may result in false positives.
- WeChat—It is not possible to selectively block WeChat Media if you allow WeChat.
- RDP (Remote Desktop Protocol)—If allowing the RDP application does not allow file transfers, ensure that the rule for RDP includes both the TCP and UDP port 3389. RDP file transfer uses UDP.

# Best practices for access control rules

Properly configuring and ordering access control rules is essential to protecting your network. The following topics summarize best practices to maximize rule performance and efficacy.



---

**Note** When you deploy configuration changes, the system evaluates all rules together and creates an expanded set of criteria that assigned devices use to evaluate network traffic. If these criteria exceed the resources (physical memory, processors, and so on) of a device, you cannot deploy to that device.

---

## General best practices for access control

Review the following requirements and general best practices:

- Use a prefilter policy to provide early blocking for unwanted traffic, and to fastpath traffic that does not benefit from access control inspection. For more information, see [Best Practices for Fastpath Prefiltering](#).
- Although you can configure the system without licensing your deployment, many features require that you enable the appropriate licenses before you deploy.
- Access control rules are deployed as access control lists (ACL) on the device. To minimize the number of access control entries created per access control rule, and improve overall performance, enable object group search for each device. Object group search is a device setting, not an access control policy setting, so you must edit each device to ensure the feature is enabled. For more information, see [Configure Object Group Search](#).
- When you deploy an access control policy, its rules are not applied to existing connections. 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 on an existing connection that could match a policy is omitted from the hit count. To have the policy rules effectively applied, clear the existing connection sessions, and then deploy the policy.
- Whenever possible, combine multiple network objects into a single object group. The system automatically creates an object group (during deployment) when you select more than one object (for source or destination separately). Selecting existing groups can avoid object group duplication and reduce the potential impact on CPU usage when there are a large number of duplicate objects.
- For the system to affect traffic, you must deploy relevant configurations to managed devices using routed, switched, or transparent interfaces, or inline interface pairs.

Sometimes, the system prevents you from deploying inline configurations to passively deployed devices, including inline devices in tap mode.

In other cases, the policy might deploy successfully, but attempting to block or alter traffic using passively deployed devices can have unexpected results. For example, the system might report multiple beginning-of-connection events for each blocked connection, because blocked connections are not blocked in passive deployments.

- Certain features, including URL filtering, application detection, and rate limiting, must allow some packets to pass in order for the system to identify the traffic.

- Some features are only available on certain device models. Warning icons and confirmation dialog boxes designate unsupported features.
- If you will use syslog or store events externally, avoid special characters in object names such as policy and rule names. Object names should not contain special characters, such as commas, that the receiving application may use as separators.
- Best practices for creating, ordering, and implementing access control rules are detailed in [Best practices for access control rules, on page 14](#) and subtopics.

## Best practices for ordering rules

General guidelines:

- In general, place top-priority rules that must apply to all traffic near the top of the policy.
- Specific rules should come before general rules, especially when the specific rules are exceptions to general rules.  
Otherwise, traffic will match the general rule first and never hit the applicable specific rule.
- Rules that drop traffic based on layer-3/4 criteria only (such as IP address, security zone, and port number) should come as early as possible. Rules based on these criteria do not require inspection to identify matching connections.
- Whenever possible, put specific drop rules near the top of the policy. This ensures the earliest possible decision on undesirable traffic.
- URL filtering, application-based, and geolocation-based rules and others that require inspection should come after rules that drop traffic based on layer-3/4 criteria only (such as IP address, security zone, and port number), but before rules that specify file and intrusion policies.
- For the most effective URL matching, place rules that include URL conditions before other rules, particularly if the URL rules are block rules and the other rules meet both of the following criteria:
  - They include application conditions.
  - The traffic to be inspected is encrypted.
- Put URL filtering rules above application rules, and follow application rules with micro-application rules and Common Industrial Protocol (CIP) sub-classification application filtering rules.
- Rules that specify file policies and intrusion policies should come at the bottom of the rule order. These rules require resource-intensive deep inspection, and you should eliminate as many threats as possible using less-intensive methods first, for performance reasons, in order to minimize the number of potential threats that require deep inspection.
- Always order rules to suit your organization's needs.

Exceptions and additions to the above guidelines are noted in the following sections.

### Application rule order

In general, rules with application conditions should be lower in your access control list because it takes longer for those rules to match than rules based on IP address, for example.

Access control rules that use *specific* conditions (such as networks and IP addresses) should be ordered *before* rules that use *general* conditions (such as applications). If you're familiar with the Open Systems Interconnect (OSI) model, use similar numbering in concept. Rules with conditions for layers 1, 2, and 3 (physical, data link, and network) should be ordered first in your access control rules. Conditions for layers 5, 6, and 7 (session, presentation, and application) should be ordered later in your access control rules. For more information about the OSI model, see this [Wikipedia article](#).

For more information, see [Choosing between application matching and port matching, on page 12](#) and [Recommendations for application control, on page 10](#).

## Rule preemption

Rule preemption occurs when a rule will never match traffic because a rule earlier in the evaluation order matches the traffic first. A rule's conditions govern whether it preempts other rules. In the following example, the second rule cannot block Admin traffic because the first rule allows it:

Access control rule 1: allow Admin users

Access control rule 2: block Admin users

Any type of rule condition can preempt a subsequent rule. The VLAN range in the first rule includes the VLAN in the second rule, so the first rule preempts the second. To block VLAN 27, you must move that rule above the one that allows VLAN 22-33.

Access control rule 1: allow Source Network VLAN 22-33

Access control rule 2: block Source Network VLAN 27

In the following example, Rule 1 matches any VLAN because no VLANs are configured, so Rule 1 preempts Rule 2, which attempts to match VLAN 2:

Access control rule 1: allow Source Network 10.4.0.0/16

Access control rule 2: allow Source Network 10.4.0.0/16, VLAN 2

A rule also preempts an identical subsequent rule where all configured conditions are the same:

Access control rule 1: allow Source Network 10.10.10.0/24 to URL www.netflix.com

Access control rule 2: allow Source Network 10.10.10.0/24 to URL www.netflix.com

A subsequent rule would not be preempted if any condition is different:

Access control rule 1: allow Source Network 10.10.10.0/24 to URL www.netflix.com

Access control rule 2: allow Source Network 10.10.11.0/24 to URL www.netflix.com

### Example: Ordering decryption rules to avoid preemption

Consider a scenario where a trusted CA (Good CA) mistakenly issued a CA certificate to a malicious entity (Bad CA), but has not yet revoked that certificate. You want to use an SSL policy to block traffic encrypted with certificates issued by the untrusted CA, but otherwise allow traffic within the trusted CA's chain of trust. After you upload the CA certificates and all intermediate CA certificates, configure a decryption policy with rules in the following order:

Rule 1: Block issuer CN=www.badca.com

Rule 2: Do not decrypt issuer CN=www.goodca.com

If you reverse the rules, you first match all traffic trusted by Good CA, including traffic trusted by Bad CA. Because no traffic ever matches the subsequent Bad CA rule, malicious traffic might be allowed instead of blocked.

## Rule actions and rule order

A rule's action determines how the system handles matching traffic. Improve performance by placing rules that do not perform or ensure further traffic handling before the resource-intensive rules that do. Then, the system can divert traffic that it might otherwise have inspected.

The following examples show how you might order rules in various policies, given a set of rules where none is more critical and preemption is not an issue.

If your rules include application conditions, also see [Choosing between application matching and port matching, on page 12](#).

### Optimum order for access control rules

Intrusion, file, and malware inspection requires resources, especially if you use multiple custom intrusion policies and variable sets. Place access control rules that invoke deep inspection last.

1. Monitor—Rules that log matching connections, but take no other action on traffic. (However, see the important exception and caveat at [Access Control Rule Monitor Action](#).)
2. Trust, Block, Block with reset—Rules that handle traffic without further inspection.
3. Allow, Interactive Block (no deep inspection)—Rules that do not inspect traffic further, but allow discovery.
4. Allow, Interactive Block (deep inspection)—Rules associated with file or intrusion policies that perform deep inspection for prohibited files, malware, and exploits.

### Optimum order for Decryption rules

Not only does decryption require resources, but so does further analysis of the decrypted traffic. Place rules that decrypt traffic last.

1. Monitor—Rules that log matching connections, but take no other action on traffic.
2. Block, Block with reset—Rules that block traffic without further inspection.
3. Do not decrypt—Rules that do not decrypt encrypted traffic, passing the encrypted session to access control rules. The payloads of these sessions are not subject to deep inspection.
4. Decrypt - Known Key—Rules that decrypt incoming traffic with a known private key.
5. Decrypt - Resign—Rules that decrypt outgoing traffic by re-signing the server certificate.

## Best practices for simplifying and focusing rules

### Simplify: do not over configure

Minimize individual rule criteria. Use as few individual elements in rule conditions as possible. For example, in network conditions use IP address blocks rather than individual IP addresses.

If one condition is enough to match the traffic you want to handle, do not use two. Using conditions that are redundant can greatly expand the deployed configuration, which can lead to problems in device performance, and unexpected device behavior in a cluster and high-availability unit re-join. For example:

- Use security zones that represent multiple interfaces carefully. If you specify source and destination networks as conditions, and these are enough to match the traffic you are targeting, then specifying a security zone is not required.
- If you want to match a set of internal interfaces to ANY destination on the Internet (for example), then simply use a source security zone that includes your internal interfaces. No network or destination interface criteria are needed.

Combining elements into objects does **not** improve performance. For example, using a network object that contains 50 individual IP addresses gives you only an organizational—not a performance—benefit over including those IP addresses in the condition individually.

For recommendations related to application detection, see [Choosing between application matching and port matching, on page 12](#).

### **Focus: narrowly constrain resource-intensive rules, especially by interface**

As much as possible, use rule conditions to narrowly define the traffic handled by resource-intensive rules. Focused rules are also important because rules with broad conditions can match many different types of traffic, and can preempt later, more specific rules. Examples of resource-intensive rules include:

- TLS/SSL rules that decrypt traffic—Not only the decryption, but further analysis of the decrypted traffic, requires resources. Narrow focus, and where possible, block or choose not to decrypt encrypted traffic.
- Access control rules that invoke deep inspection—Intrusion, file, and malware inspection requires resources, especially if you use multiple custom intrusion policies and variable sets. Make sure you only invoke deep inspection where required.
- If you specify security zones in a rule, the rule is deployed only to devices that have interfaces in the specified zone. So, if you want to focus a rule on only some devices that are assigned to the policy, ensure that you select a security zone that applies to the appropriate subset of devices. This ensures that unnecessary rules are not deployed to a device.

## Maximum number of access control rules and intrusion policies

The maximum number of access control rules or intrusion policies that are supported by a device depends on many factors, including policy complexity, physical memory, and the number of processors on the device.

If you exceed the maximum supported by a device, you cannot deploy the access control policy and you must reevaluate.

Guidelines for intrusion policies:

- In an access control policy, you can associate one intrusion policy with each Allow and Interactive Block rule, as well as with the default action. Every unique **pair** of intrusion policy and variable set counts as one policy.
- You may want to consolidate intrusion policies or variable sets so you can associate a single intrusion policy-variable set pair with multiple access control rules. On some devices you may find you can use only a single variable set for all your intrusion policies, or even a single intrusion policy-variable set pair for the whole device.