TLS/SSL Rules Best Practices

TLS/SSL Rules Best Practices

This chapter provides an example SSL policy with TLS/SSL rules that illustrates our best practices and recommendations. First we'll discuss settings for the SSL and access control policies and then walk through all the rules and why we recommend they be ordered in a particular way.

Following is the SSL policy we'll discuss in this chapter.

SSL Policy Example

Enter Description

Rules | Trusted CA Certificates | Undecryptable Actions | Advanced Settings

+ Add Category   + Add Rule   🔍 Search Rules   ✕

#	Name	Source Zones	Dest Zones	Source Networks	Dest Networks	VLAN Tags	Users	Applicati...	Source Ports	Dest Ports	Categories	SSL	Action	
Administrator Rules														
This category is empty														
Standard Rules														
1	DND internal source network	any	any	Intranet	any	any	any	any	any	any	any	any	✔ Do not decrypt	✏ 🗑
2	Decrypt test site	any	any	any	any	any	any	any	any	any	Astrology (Any	any	→ Decrypt - Resign	✏ 🗑
3	ⓘ Do not decrypt low risk	any	any	any	any	any	any	Risks: Very Lov	any	any	any	any	✔ Do not decrypt	✏ 🗑
4	Do not decrypt applications	any	any	any	any	any	any	Facebook Facebook Mes Facebook Phot	any	any	any	any	✔ Do not decrypt	✏ 🗑
5	Decrypt all but trusted categ	any	any	any	any	any	any	any	any	any	Any (Except Ui	any	→ Decrypt - Resign	✏ 🗑
6	ⓘ Block bad cert status	any	any	any	any	any	any	any	any	any	any	1 Cert Status se	⊘ Block	✏ 🗑
7	ⓘ Block SSLv3. TLS 1.0, 1.1	any	any	any	any	any	any	any	any	any	any	3 Protocol Versi	⊘ Block	✏ 🗑
Root Rules														
This category is empty														
Default Action													Do not decrypt ▾	🗐

Bypass Inspection with Prefilter and Flow Offload

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.

If you have a Firepower 4100/9300 available, you can use *large flow offload*, a technique where trusted traffic can bypass the inspection engine for better performance. You can use it, for example, in a data center to transfer server backups.

Do Not Decrypt Best Practices

Log traffic

We recommend *against* creating **Do Not Decrypt** rules that do not log anything because these rules still take processing time on the managed device. If you set up any type of TLS/SSL rules, *enable logging* so you can see what traffic is being matched.

Guidelines for undecryptable traffic

We can determine that certain traffic is not decryptable either because the website itself is not decryptable or because the website uses SSL pinning, which effectively prevents users from accessing a decrypted site without errors in their browser.

We maintain the list of these sites as follows:

- A Distinguised Name (DN) group named **Cisco-Undecryptable-Sites**

- The **pinned certificate** application filter

If you are decrypting traffic and you do not want users to see errors in their browsers when going to these sites, we recommend you set up a **Do Not Decrypt** rule toward the bottom of your TLS/SSL rules.

An example of setting up a **pinned certificate** application filter follows.
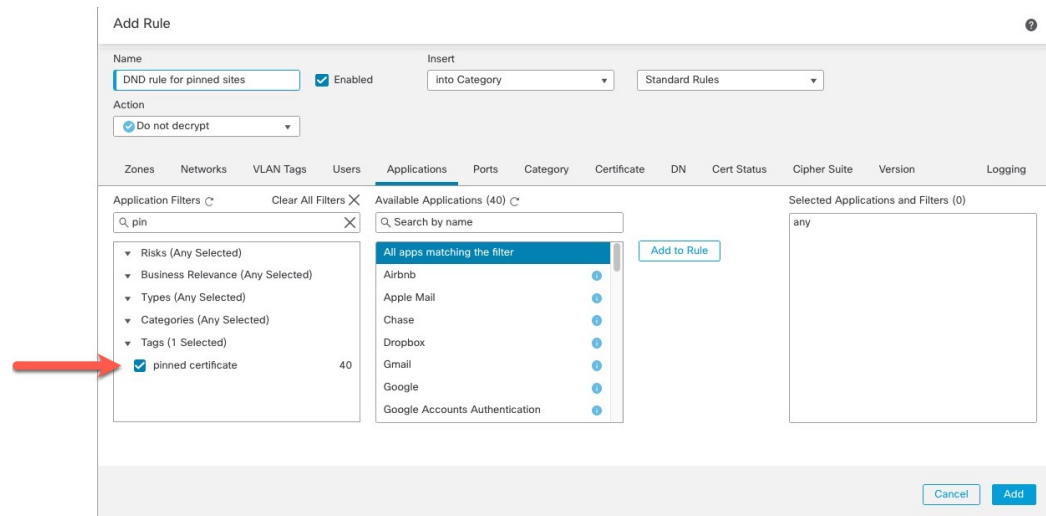
Decrypt - Resign and Decrypt - Known Key Best Practices

This topic discusses best practices for **Decrypt - Resign** and **Decrypt - Known Key** TLS/SSL rule.

Decrypt - Resign Best Practices With Certificate Pinning

Some applications use a technique referred to as *TLS/SSL pinning* or *certificate pinning*, which embeds the fingerprint of the original server certificate in the application itself. As a result, if you configured a TLS/SSL

rule with a **Decrypt - Resign** action, when the application receives a resigned certificate from a managed device, validation fails and the connection is aborted.

Because TLS/SSL pinning is used to avoid man-in-the-middle attacks, there is no way to prevent or work around it. You have the following options:

- Create a **Do Not Decrypt** for those applications rule ordered before **Decrypt - Resign** rules.

- Instruct users to access the applications using a web browser.

For more information about certificate pinning, see the section on SSL pinning in the Cisco Secure Firewall Management Center Device Configuration Guide.

Decrypt - Known Key Best Practices

Because a **Decrypt - Known Key** rule action is intended to be used for traffic going to an internal server, you should always add a destination network to these rules (**Networks** rule condition). That way the traffic goes directly to the network on which the server is located, thereby reducing traffic on the network.

TLS/SSL Rules to Put First

Put first any rules that can be matched by the first part of the packet; an example is a rule that references IP addresses (**Networks** rule condition).

TLS/SSL Rules to Put Last

Rules with the following rule conditions should be last because those rules require traffic to be examined for the longest amont of time by the system:

- Applications

- Category

- Certificate

- Distinguished Name (DN)

- Cert Status

- Cipher Suite

- Version