

Configure Multiple RAVPN Profiles with SAML Authentication on FDM

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[Background Information](#)

[Configure](#)

[Step 1: Create a Self-Signed Certificate and PKCS#12 File using OpenSSL](#)

[Step 2: Upload the PKCS#12 File on Azure and FDM](#)

[Step 2.1. Upload the Certificate to Azure](#)

[Step 2.2. Upload the Certificate to the FDM](#)

[Verify](#)

Introduction

This document describes how to configure SAML authentication for Multiple Connection Profiles of Remote Access VPN using Azure as IdP on CSF via FDM.

Prerequisites

Requirements

Cisco recommends that you have basic knowledge of these topics:

- Secure Socket Layer (SSL) Certificates
- OpenSSL
- Remote Access Virtual Private Network (RAVPN)
- Cisco Secure Firewall Device Manager (FDM)
- Security Assertion Markup Language (SAML)
- Microsoft Azure

Components Used

The information in this document is based on these software versions:

- OpenSSL
- Cisco Secure Firewall (CSF) Version 7.4.1
- Cisco Secure Firewall Device Manager Version 7.4.1

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Background Information

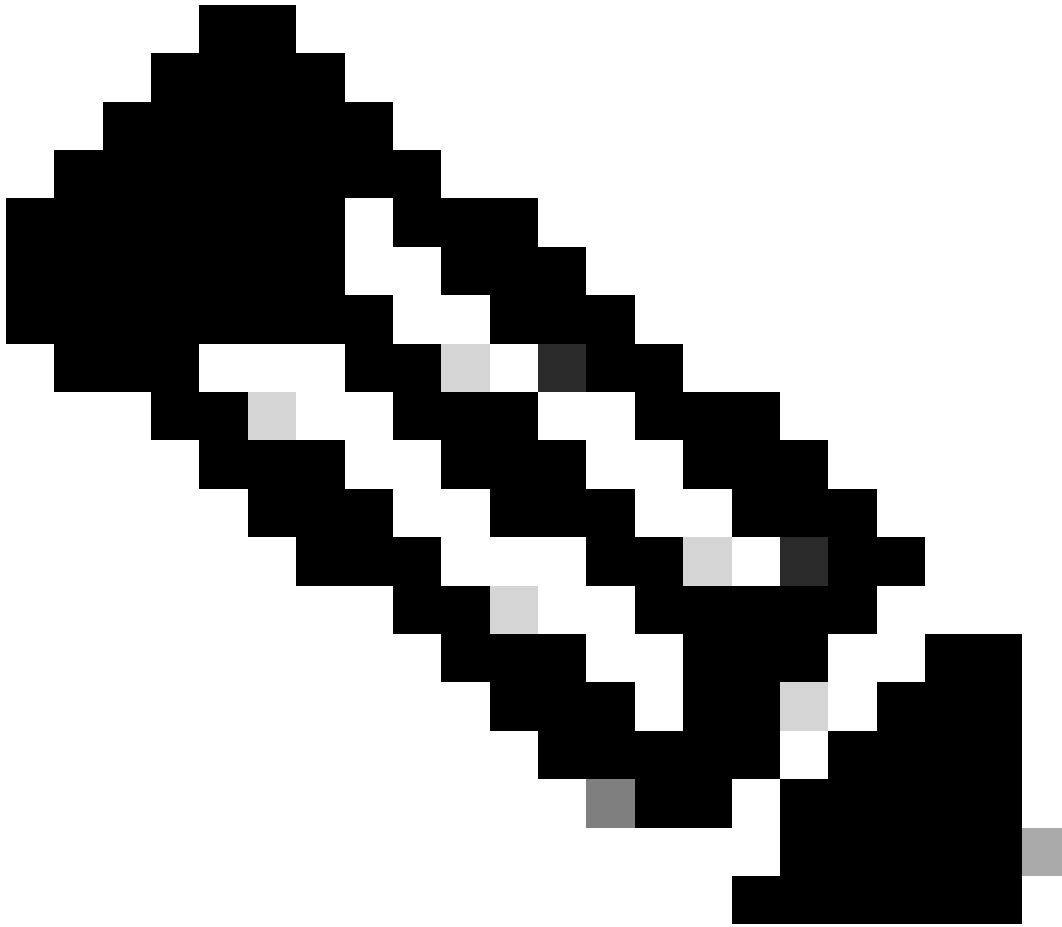
SAML, or Security Assertion Markup Language, is an open standard for exchanging authentication and authorization information between parties, specifically an Identity Provider (IdP) and a Service Provider (SP). The use of SAML authentication for Remote Access VPN (RAVPN) connections and various other applications has become increasingly popular due to its numerous advantages. On the Firepower Management Center (FMC), multiple Connection Profiles can be configured to use different IdP-protected applications because of the Override Identity Provider Certificate option available in the Connection Profile configuration menu. This feature allows administrators to override the primary IdP certificate in the Single Sign-On (SSO) Server object with a specific IdP certificate for each connection profile. However, this functionality is limited on the Firepower Device Manager (FDM) as it does not provide a similar option. If a second SAML object is configured, attempting to connect to the first Connection Profile results in an authentication failure, displaying the error message: **"Authentication failed due to a problem retrieving the single sign-on cookie."** To work around this limitation, a custom Self-Signed certificate can be created and imported into Azure for use across all applications. By doing so, only one certificate needs to be installed in the FDM, enabling seamless SAML authentication for multiple applications.

Configure

Step 1: Create a Self-Signed Certificate and PKCS#12 File using OpenSSL

This section describes how to create the Self-Signed certificate using OpenSSL

1. Log in to an endpoint which has the OpenSSL library installed.



Note: In this document, a Linux machine is used, so some commands are specific to a Linux environment. However, the OpenSSL commands are the same.

b. Create a **configuration file** using the `touch <config_name>.conf` command.

```
<#root>
root@host#
touch config.conf
```

c. Edit the **file** with a text editor. In this example, Vim is used and the `vim <config_name>.conf` command is run. You can use any other text editor.

```
<#root>
root@host#
```

```
vim config.conf
```

d. Enter the **information** to be included in the Self-Signed.

Ensure to replace the values between < > with the information of your organization.

```
[req]
distinguished_name = req_distinguished_name
prompt = no
```

```
[req_distinguished_name]
C = <Country Code>
ST = <State or Province>
L = <Locality Name>
O = <Organization Name>
OU = <Organizational Unit Name>
CN = <Common Name>
```

e. Using this command generates a new 2048-bit RSA private key and a self-signed certificate using the SHA-256 algorithm, valid for 3650 days, based on the configuration specified in the <config_name>.conf file. The private key is saved to <key_name>.pem and the Self-Signed certificate is saved to <self-signed_certificate>.crt.

```
<#root>
```

```
root@host#
```

```
openssl req -newkey rsa:2048 -nodes -keyout <key_name>.pem -x509 -sha256 -days 3650 -config <config_name>
```

```
root@host:~# openssl req -newkey rsa:2048 -nodes -keyout Azure_key.pem -x509 -sha256 -days 3650 -config config.conf -out Azure_SS0.crt
Generating a RSA private key
.....+++++
.....+++++
writing new private key to 'Azure_key.pem'
-----
root@host:~#
```

f. After creating the private key and the Self-Signed certificate, it exports them into a PKCS#12 file, which is a format that can include both the private key and the certificate.

```
<#root>
```

```
root@host#
```

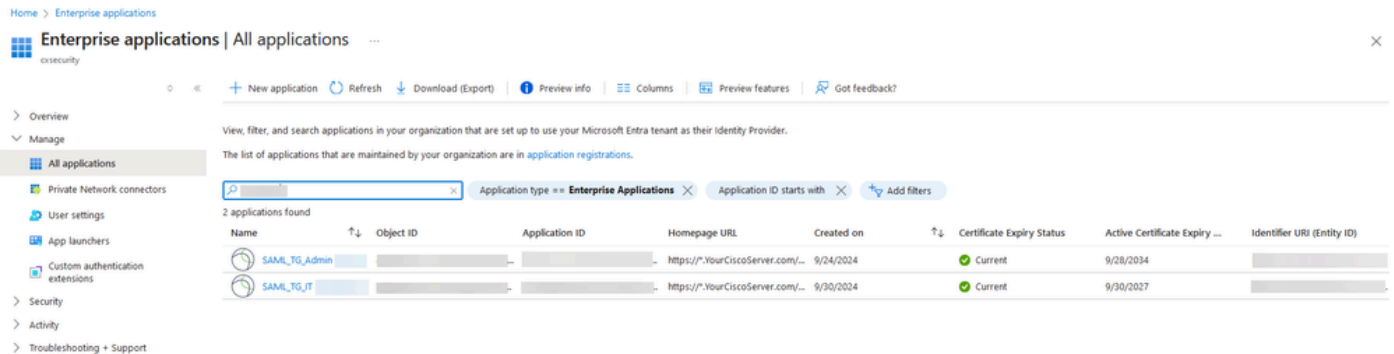
```
openssl pkcs12 -export -inkey <key_name>.pem -in <self-signed_certificate>.crt -name <Alias> -out <pkcs12_name>
```

```
root@host:~# openssl pkcs12 -export -inkey Azure_key.pem -in Azure_SS0.crt -out Azure_SS0.pfx
Enter Export Password:
Verifying - Enter Export Password:
root@host:~#
root@host:~# ls
Azure_SS0.crt Azure_SS0.pfx Azure_key.pem config.conf
```

Take note of the password.

Step 2: Upload the PKCS#12 File on Azure and FDM

Ensure to create an application on Azure for each Connection Profile that is using SAML authentication on the FDM.



The screenshot shows the Azure Enterprise Applications management console. The page title is "Enterprise applications | All applications". The left sidebar contains navigation options: Overview, Manage, All applications, Private Network connectors, User settings, App launchers, Custom authentication extensions, Security, Activity, and Troubleshooting + Support. The main content area displays a table of applications with the following columns: Name, Object ID, Application ID, Homepage URL, Created on, Certificate Expiry Status, Active Certificate Expiry, and Identifier URI (Entity ID). Two applications are listed:

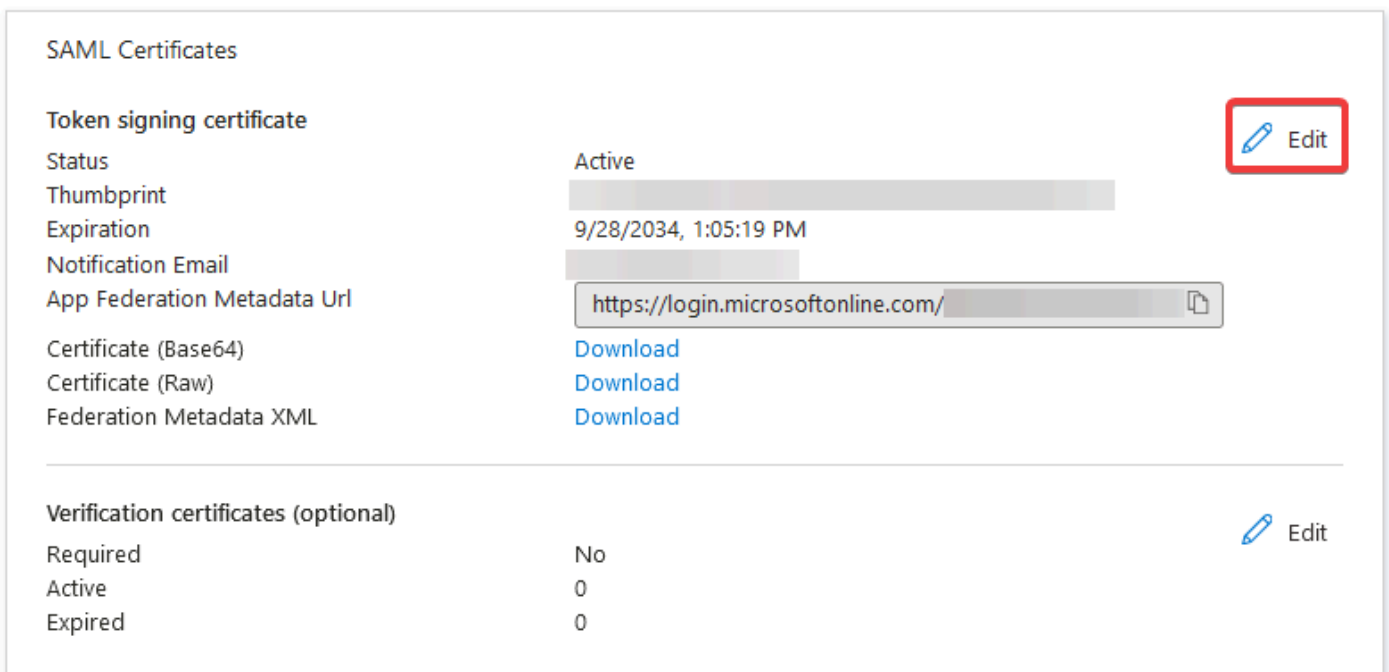
Name	Object ID	Application ID	Homepage URL	Created on	Certificate Expiry Status	Active Certificate Expiry	Identifier URI (Entity ID)
SAML_TQ_Admin			https://*.YourCiscoServer.com/...	9/24/2024	Current	9/28/2034	
SAML_TQ_JT			https://*.YourCiscoServer.com/...	9/30/2024	Current	9/30/2027	

Once you have the PKCS#12 file from **Step 1: Create a Self-Signed Certificate and PKCS#12 file using OpenSSL**, it must be uploaded to Azure for multiple applications and configured in the FDM SSO configuration.

Step 2.1. Upload the Certificate to Azure

a. Log in to your Azure portal, navigate to the Enterprise application you want to protect with SAML authentication, and select **Single Sign-On**.

b. Scroll down to the **SAML Certificates** section and select the **More Options > Edit**.



The screenshot shows the "SAML Certificates" configuration page. It is divided into two sections: "Token signing certificate" and "Verification certificates (optional)".

Token signing certificate

- Status: Active
- Thumbprint: [Redacted]
- Expiration: 9/28/2034, 1:05:19 PM
- Notification Email: [Redacted]
- App Federation Metadata Url: [https://login.microsoftonline.com/\[Redacted\]](https://login.microsoftonline.com/[Redacted])
- Certificate (Base64): [Download](#)
- Certificate (Raw): [Download](#)
- Federation Metadata XML: [Download](#)

Verification certificates (optional)

- Required: No
- Active: 0
- Expired: 0

Both sections have an "Edit" button with a pencil icon.

c. Now, select the **Import certificate** option.

SAML Signing Certificate

Manage the certificate used by Microsoft Entra ID to sign SAML tokens issued to your app

Save + New Certificate **Import Certificate** Got feedback?

Status	Expiration Date	Thumbprint
Active	8/25/2029, 7:03:32 PM	[Redacted]


Signing Option: Sign SAML assertion

Signing Algorithm: SHA-256

d. Find the PKCS#12 file previously created and use the password you entered when you created the PKCS#12 file.

Import certificate

Upload a certificate with the private key and the pfx credentials, the type of this file should be .pfx and using RSA for the encryption algorithm

Certificate: "Azure_SSO.pfx" 

PFX Password: [Masked] ✓

Add

Cancel

e. Finally, select the **Make Certificate Active** option.

SAML Signing Certificate

Manage the certificate used by Microsoft Entra ID to sign SAML tokens issued to your app

Save + New Certificate **Import Certificate** Got feedback?

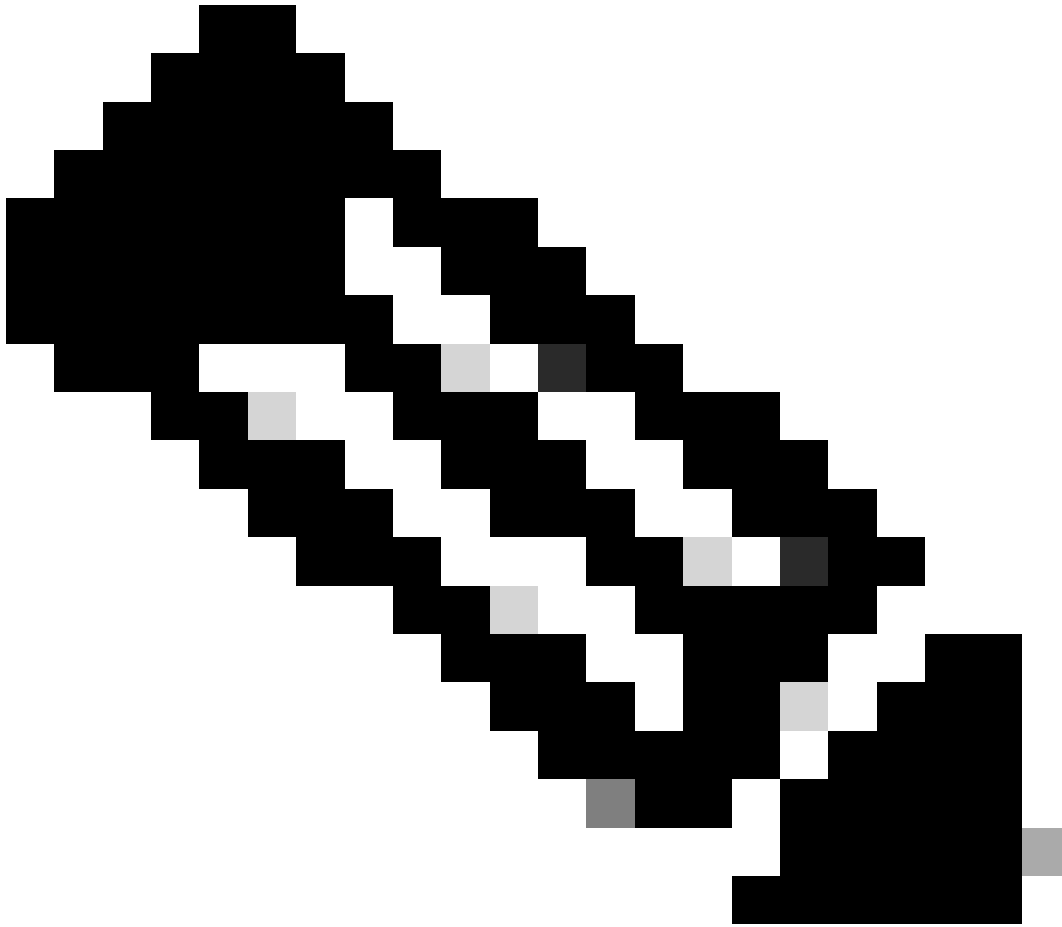
Status	Expiration Date	Thumbprint
Inactive	9/28/2034, 1:05:19 PM	[Redacted]
Active	9/27/2027, 5:51:21 PM	[Redacted]

Signing Option: Sign SAML assertion

Signing Algorithm: SHA-256

Notification Email Addresses: [Redacted]

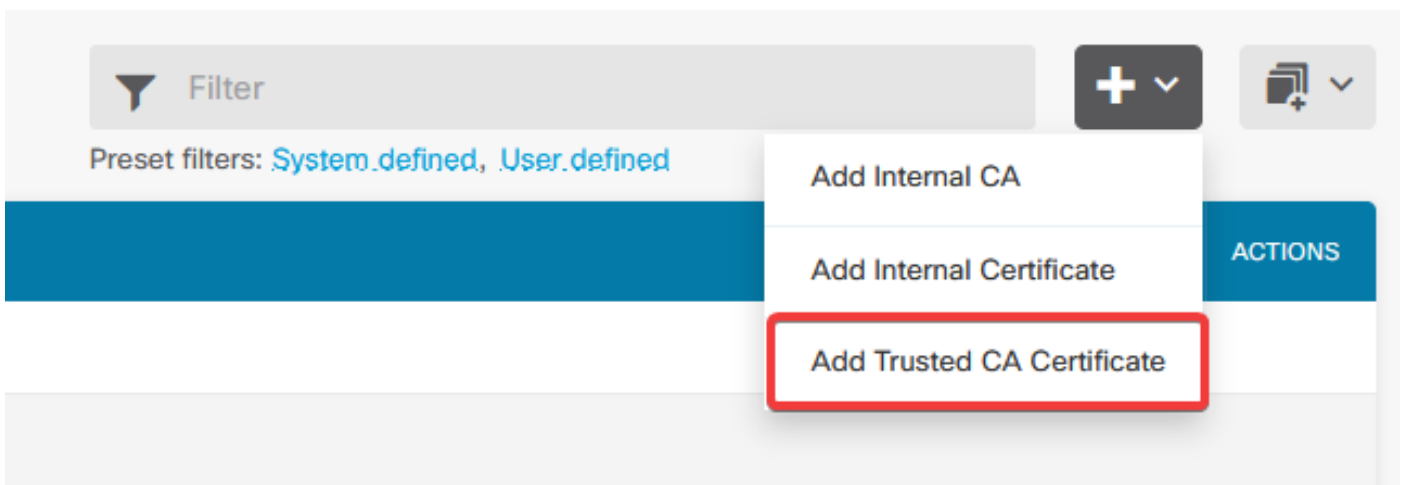
- Make certificate active**
- Base64 certificate download
- PEM certificate download
- Raw certificate download
- Download federated certificate XML
- Delete Certificate



Note: Ensure to perform **Step 2.1: Upload the Certificate to Azure** for each application.

Step 2.2. Upload the Certificate to the FDM

a. Navigate to **Objects > Certificates > Click Add Trusted CA certificate.**



Edit SAML Server



Name

AzureIDP

Description

Identity Provider (IDP) Entity ID URL

https://

Sign In URL

https://

Supported protocols: https, http

Sign Out URL

https://

Supported protocols: https, http

Service Provider Certificate

(Validation Us...

Identity Provider Certificate

Azure_SSO (Validation Usage: ...

Request Signature

None

Request Timeout

Range: 1 - 7200 (sec)

d. Set the **SAML object** on the different Connection Profiles that are using SAML as the authentication method and for which the application was created in Azure. Deploy the **changes**

Device Summary

Remote Access VPN Connection Profiles

2 connection profiles

Filter



#	NAME	AAA	GROUP POLICY	ACTIONS
1	SAML_TG_Admin	Authentication: SAML Authorization: None Accounting: None	SAML_GP_Admin	
2	SAML_TG_IT	Authentication: SAML Authorization: None Accounting: None	SAML_GP_IT	

Primary Identity Source

Authentication Type

SAML



SAML Login Experience

VPN client embedded browser

Default OS browser

Primary Identity Source for User Authentication

AzureIDP



Verify

Run the `show running-config webvpn` and `show running-config tunnel-group` commands to review the configuration and verify that the same IDP URL is configured on the different Connection Profiles.

```
<#root>
```

```
firepower#
```

```
show running-config webvpn
```

```
webvpn
```

```
enable outside
```

```
http-headers
```

```
hsts-server
```

```
enable
```

```
max-age 31536000
```

```
include-sub-domains
```

```
no preload
```

```
hsts-client
```

```
enable
```

```
x-content-type-options
```

```
x-xss-protection
```

```
content-security-policy
```

```
anyconnect image disk0:/anyconnpkgs/anyconnect-win-4.10.08029-webdeploy-k9.pkg 2
```

```
anyconnect profiles defaultClientProfile disk0:/anyconnprofs/defaultClientProfile.xml
```

```
anyconnect enable
```

```
saml idp https://saml.lab.local/af42bac0<omitted>/
```

```
url sign-in https://login.saml.lab.local/af42bac0<omitted>/saml2  
url sign-out https://login.saml.lab.local/af42bac0<omitted>/saml2  
base-url https://Server.cisco.com  
trustpoint idp
```

```
Azure_SSO
```

```
trustpoint sp FWCertificate  
no signature  
force re-authentication  
tunnel-group-list enable  
cache  
disable  
error-recovery disable  
firepower#
```

```
<#root>
```

```
firepower#
```

```
show running-config tunnel-group
```

```
tunnel-group SAML_TG_Admin type remote-access  
tunnel-group SAML_TG_Admin general-attributes  
address-pool Admin_Pool  
default-group-policy SAML_GP_Admin  
tunnel-group SAML_TG_Admin webvpn-attributes
```

```
authentication saml
```

```
group-alias SAML_TG_Admin enable
```

```
saml identity-provider https://saml.lab.local/af42bac0<omitted>/
```

```
tunnel-group SAML_TG_IT type remote-access  
tunnel-group SAML_TG_IT general-attributes  
address-pool IT_Pool  
default-group-policy SAML_GP_IT  
tunnel-group SAML_TG_IT webvpn-attributes
```

```
authentication saml
```

```
group-alias SAML_TG_IT enable
```

```
saml identity-provider https://saml.lab.local/af42bac0<omitted>/
```

```
firepower#
```