SAML™ Installation and Configuration Guide

Document Version 8.0

Technical Guide

April 5, 2024

10017-08 EN Rev. A

ThreatConnect, Inc.
3865 Wilson Blvd., Suite 550, Arlington, VA 22203

P: 1.800.965.2708 | F: 703.229.4489
www.ThreatConnect.com

Table of Contents

Overview

Security Assertion Markup Language™ (SAML) is used in a Single Sign-On (SSO) environment. It is a widely accepted standard for transporting authentication and authorization data between multiple systems. The central system is the Identity Provider (IDP), which contains the identity profile of users within an organization. The profile data can consist of a job title, an organizational role, an email address, etc. This information is shared when requested by other systems within the enterprise. These requesters, or clients, are applications called Service Providers (SP).

When a user attempts to log into an SP by providing his or her credentials, the SP issues a request to the IDP for the user's profile data. If the IDP confirms the user's credentials and discovers the associated profile, then the IDP will fulfill the request by responding with a SAML assertion containing the user's organizational information. Once the SAML assertion is received, the SP will interpret the assertion and grant the user access to the application with the affiliated role.

Multiple vendors implement and sell the IDP system. These vendors include CA Single Sign-On (aka SiteMinder), PingFederate®, and others. These products use the SAML standard to exchange authentication and authorization data with an SP.

ThreatConnect can be configured as an SP. This guide instructs users how to install the platform with SSO configurations to allow ThreatConnect to participate in an SSO enterprise as an SP requesting identity data from the designated IDP.

The ThreatConnect SSO feature has been tested with CA Single Sign-On and has been proven to work with other vendors who utilize the SAML standard. Thus, any IDP vendor will work with ThreatConnect if that vendor issues SAML assertions and strictly follows the SAML standard.

The SSO feature currently supports signed assertions, but it does not support encrypted assertions. For signature support, carefully follow the certificate installation instructions.

> **Note**: User authentication is supported by any integration that supports SAML 2.0.

Installation Prerequisites

SSL Certificates

ThreatConnect 7.4.1 or Older

Install and configure the Secure Sockets Layer (SSL) certificates on the ThreatConnect instance **before** implementing Security Assertion Markup Language (SAML). Refer to *ThreatConnect Installation Guide_Linux® Operating System* for detailed operating system instructions to install SSL certificates.

Note: When importing a customer's IDP public certificate into ThreatConnect's local keystore, use the IDP hostname as the alias for the certificate.

ThreatConnect 7.5.0 or Newer

If you have an IDP public certificate in your keystore, run the following commands to export it from `keystore.jks` to the `certs` folder. Before running the commands, replace the following placeholder values:

- `<alias>`: The alias that the IDP certificate is listed under in the keystore
- `<path to keystore>`: The file path to `keystore.jks`
- `<password>`: The password for the keystore

```
cd trusted
keytool -exportcert \
 -rfc \
 -alias <alias> \
 -file <alias>.pem \
 -keystore <path to keystore>/keystore.jks \
 -storepass <password> \
 -v
```

Required Information

Fill out the SAML-IDP Specification Chart **before** the scheduled SAML installation, and return it to the Deployment Engineer involved in the SAML installation and configuration so that they can review it and ensure all needed data parts have been entered.

Installation

ThreatConnect 7.4.1 or Older

Enabling SAML

Unless all prerequisites have been accomplished before the initial configuration of ThreatConnect, some options will need to be adjusted in order to access the SAML configuration. These changes will reset all ThreatConnect installations when running the SAML configuration.

To enable the SAML configuration, follow these steps:

1. Stop the ThreatConnect service and edit the `/<install location for ThreatConnect>/threatconnect/config/install.properties` file:

   ```
   # service threatconnect stop
   # vim /opt/threatconnect/config/install.properties
   ```

2. Change the value of `setupcompleted` from `true` to `false`.
3. Save the file and quit. Then log in as the `threatconnect` user and run `setup.sh`.

   ```
   # su - threatconnect
   # cd /<install location for ThreatConnect>/threatconnect/app
   # ./setup.sh
   ```

4. Run through the initial setup configuration for ThreatConnect again. Enter `true` when asked about SAML configuration. Input the items pulled from the completed SAML spreadsheet into the following fields:
   a. Enter IDP URL < *value of IDP URL location*>.
   b. Enter Service Provider (SP) URL < *value of SP URL location*>.

c. Enter SP assertion consumer URL.

d. Enter IDP entity ID.

e. Enter IDP SLO URL ‹ *value of IDP Single Logout URL*›.

f. Set IDP post binding (true or false).

g. Enter SP binding type (POST I REDIRECT) ‹ *value of SP binding method*›.

h. Enter SP relay address ‹ *value of SP relay URL (must be encoded)*›.

i. Enter SAML2's skew milliseconds.

j. Please enter name ID format: ‹ *value should be* ***urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified****, unless directed by the authentication team to input something else*›

k. Verify SAML's signatures (true or false) ‹ *enable signature verification*›.

l. Enter IDP host for public certificate ‹ *value of IDP host*›.

m. Enter SP signing key alias ‹ *value of the signing key alias for SP*›.

n. Enter keystore absolute path ‹ *value of keystore location*›.

o. Enter keystore password.

p. Enter keystore password again.

q. Enter private key password.

r. Enter private key password again.

s. Enable SHA256 SAML signature algorithm (true or false). (False will use SHA1.)

t. Generate a SAML 2.0 SP metadata file (true or false). (True will generate a SAML metadata file that can be passed to an IDP administrator to aid in the SAML configuration within the IDP.)

u. Proceed with SAML changes in server configuration (yes or no).

> **Note**: When importing a customer's IDP public certificate into ThreatConnect's local keystore (step n.), use the value entered for "Enter IDP host for public certificate" (step l.) as the alias for the certificate.

5. Log out as the `threatconnect` user, add the admin account, and restart the `threatconnect` service to access the Web UI:

```
# logout
# service threatconnect start
```

6. For SAML to work on user accounts, the admin account will need to be set up before the user creation can occur. Connect to either the MySQL or PostgreSQL database that is being used by the ThreatConnect instance, and run the following MySQL or PostgreSQL strings, respectively:

Note: The value for `[UserName]` appears multiple times and needs to be entered in all places.

- **MySQL**

```
# mysql -u tcuser -p threatconnect

INSERT INTO UserTable (id_Organization, id_TimeZone, id_Role, userName,
password, salt, firstName, locked, resetRequired, failedAttempts, lastLogin,
summaryEmailHour, deleted, disabled, pseudonym, profileConfigured,
profileEditable, receiveReplyNotification, tracked, termsAccepted,
logoutInterval, mailBounceCount, lastPasswordChange) VALUES ('1', '24', '2',
'[UserName]', '', '', '[FirstName]', '0', '0', '0', NULL, '5', '0', '0',
'[UserPseudonym]', '1', '1', '0', '0', '0', '1800', '0', NOW());

INSERT INTO CommunityMembership (id_Organization, id_User, id_OwnerRole)
SELECT u.id_Organization, u.id, o.id FROM UserTable u, OwnerRole o WHERE
u.userName IN ('[UserName]') AND o.name = 'Organization Administrator' AND
NOT EXISTS (SELECT id FROM CommunityMembership c WHERE u.id_Organization  =
c.id_Organization AND u.id = c.id_User);

INSERT INTO UserNotificationSettings (id_User, severity,
id_NotificationType, actionImmediate, actionSummary, actionRealTime) SELECT
u.id, 'HIGH', NULL, 1, 0, 1 FROM User u where u.username = '[UserName]';

INSERT INTO UserNotificationSettings (id_User, severity,
id_notificationtype, actionImmediate, actionSummary, actionRealTime) SELECT
u.id, 'MEDIUM', NULL, 1, 0, 0 FROM User u where u.username = '[UserName]';

INSERT INTO UserNotificationSettings (id_User, severity,
id_notificationtype, actionImmediate, actionSummary, actionRealTime) SELECT
u.id, 'LOW', NULL, 0, 1, 0 FROM User u where u.username = '[UserName]';
```

- **PostgreSQL**

```
psql -U tcuser -d threatconnect


INSERT INTO UserTable (id_Organization, id_TimeZone, id_Role, userName,
password, salt, firstName, locked, resetRequired, failedAttempts, lastLogin,
summaryEmailHour, deleted, disabled, pseudonym, profileConfigured,
profileEditable, receiveReplyNotification, tracked, termsAccepted,
logoutInterval, mailBounceCount, lastPasswordChange) VALUES ('1', '24', '2',
'[UserName]', '', '', '[FirstName]', '0', '0', '0', NULL, '5', '0', '0',
'[UserPseudonym]', '1', '1', '0', '0', '0', '1800', '0', NOW());


INSERT INTO CommunityMembership (id_Organization, id_User, id_OwnerRole)
SELECT u.id_Organization, u.id, o.id FROM UserTable u, OwnerRole o WHERE
u.userName IN ('[UserName]') AND o.name = 'Organization Administrator' AND
NOT EXISTS (SELECT id FROM CommunityMembership c WHERE u.id_Organization  =
c.id_Organization AND u.id = c.id_User);


INSERT INTO UserNotificationSettings (id_User, severity,
id_NotificationType, actionImmediate, actionSummary, actionRealTime) SELECT
u.id, 'HIGH', NULL, 1, 0, 1 FROM UserTable u where u.username =
'[UserName]';


INSERT INTO UserNotificationSettings (id_User, severity,
id_notificationtype, actionImmediate, actionSummary, actionRealTime) SELECT
u.id, 'MEDIUM', NULL, 1, 0, 0 FROM UserTable u where u.username =
'[UserName]';


INSERT INTO UserNotificationSettings (id_User, severity,
id_notificationtype, actionImmediate, actionSummary, actionRealTime) SELECT
u.id, 'LOW', NULL, 0, 1, 0 FROM UserTable u where u.username = '[UserName]';
```

7. Change these values based on the setup of the client's admin account:

```
[UserName] = admin@<client's email domain> (email address of admin account in
IDP LDAP)
[FirstName] - first name of user
[UserPseudonym] - pseudonym value for user
```

8. Once the admin is able to log in via SAML, the additional user account can be created within the ThreatConnect Web UI.

ThreatConnect 7.5.0 or Newer

Enabling SAML

In the `.env` file associated with the containerized deployment of ThreatConnect, update each variable in the "SAML Settings" section with the appropriate value. For descriptions of the values that you must provide in the `.env` file, reference the comments in the "SAML Settings" section of that file.

Additional Notes

Metadata Requested by the IDP Administrator

If your ThreatConnect instance has **version 7.4.1 or older installed** and you answered `true` to the "Generate a SAML 2.0 SP metadata file" question during the setup process, the `metadata.xml` file will have been created in the `config` folder.

If your ThreatConnect instance has **version 7.5.0 or newer installed** and you set the value of the `SAML_GENERATE_METADATA` variable in your `.env` file to `true`, run the following command to retrieve the `metadata.xml` from the Docker® container:

```
docker cp threatconnect-docker-tc-mon-1:/opt/threatconnect/config/metadata.xml
```

The client's SSO team may request the ThreatConnect instance metadata in order to set up their SAML configuration. In order to pull this data, the SSL certificates must be installed and operational **before** the full XML file can be given to the client's SSO team.

Template Metadata file:

```
<?xml version="1.0" encoding="UTF-8"?>
<md:EntityDescriptor xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
entityID="https://<FQDN-of-ThreatConnect-server>/auth/">
     <md:SPSSODescriptor AuthnRequestsSigned="true" WantAssertionsSigned="true"
protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
           <md:KeyDescriptor use="signing">
                 <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
                       <ds:X509Data>
<ds:X509Certificate>MIIC0TCCAbkCAQAwXDELMAkGA1UEBhMCSU4xCzAJBCgKCAQEAmEwfAFLjgDOBgN
VBAoTCXJicHJpdmF0ZTELMAkGA1UECxMCUUExETAPBgNVBAMTCHJhamt1bWFyMIIBIjANBgkqhkiG9w0BAQ
EFAAOCAQ8AMIIBCgKCAQEAmEwfAFLjgDOEZk1AYPhX7dbYMXqkk4rF3uyYZeoMnnXPLs463GzGvVPnRgjTd
Izm+1QOnkTx3BBu7kxlhtze2Sr7rtHLs1FYbzXREs5aVgIPnpkfuKdR9QNDaJJ1byxStnF+zI4feSYmHXsV
WfHm24+FK0kCk3tSnw2/noXyW5xc2UbrGLYqaezpPSlf5WJ3isKFlQr2k+HKXh4Rid4TUmEaoZXPAcB7Qtk
BYnIxzzmBoFCWSSsVldPRkaw=</ds:X509Certificate>
                       </ds:X509Data>
                 </ds:KeyInfo>
           </md:KeyDescriptor>
     <md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-
format:unspecified</md:NameIDFormat>
           <md:AssertionConsumerService
Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST" Location="https://<FQDN-
of-ThreatConnect-server>/auth/" index="0" isDefault="true" />
     </md:SPSSODescriptor>
</md:EntityDescriptor>
```

Adjust the following sections based on the setup:

- `SAML:2.0:bindings`: This item needs to be set to `HTTP-POST` or `HTTP-REDIRECT`, based on the spreadsheet documentation for the client.
- `ds:X509Certificate`: X509 data will be pulled from the ThreatConnect server certificate that is installed in the `keystore.jks` file on the ThreatConnect instance.
- `Location`: The value assigned to `Location` is `https://<FQDN-of-ThreatConnect-server>/auth/`.
- `entityID`: The value assigned to `entityID` is usually the same value as `Location`, but it can be defined by the IDP admin.

IDP Configuration

The Assertion Consumer Service URL must have `/saml` as a suffix that is appended to the end of the URL path (e.g., `https://www.example.com/auth/saml`).