ThreatConnect® Upgrade Guide

Software Version 7.4

Technical Guide

January 17, 2024

10021-22 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

This guide provides instructions for upgrading to the latest version of ThreatConnect. Please read all steps in their entirety before proceeding with the upgrade.

Important: These instructions assume that the application has been installed to the directory **/opt/threatconnect**. If the application has been installed to a different directory, please adjust the commands to reflect the location of the install directory.

Pre-Upgrade Steps

Python 3.11 Installation

As of ThreatConnect version 7.1.0, you must install both Python® 3.6.*x* and Python 3.11.*x*.

Install Python 3.11

Note: The instructions in this section compile and install Python from source code, which is one of the multiple ways to install Python on the application server. If the operating system to be used already has Python 3.11 installed, skip the compile steps of the Python installation.

CentOS 7 and RHEL 7

Run one of the following commands to install additional software collections (SCLs):

- **CentOS 7**

```
yum install centos-release-scl -y
```

- **RHEL 7**

```
subscription-manager repos --enable rhel-server-devtools-7-rpms
subscription-manager repos --enable rhel-server-rhscl-7-rpms
```

Install the GNU Complier Collection (GCC) and set it as the default:

```
yum install devtoolset-7
echo "source scl_source enable devtoolset-7" >> ~/.bash_profile
. ~/.bash_profile
```

Install dependencies specific to Python 3.11:

```
yum install -y epel-release --enablerepo=extras
yum install -y openssl11 openssl11-libs openssl11-devel lcms2-devel
```

It is required to have developer packages installed to compile Python from source:

```
yum install -y yum-utils \
    make gcc \
    openssl openssl-devel \
    postgresql-devel \
    libtiff-devel libjpeg-devel libzip-devel freetype-devel \
    libwebp-devel tcl-devel libxslt-devel libxml2-devel \
    bzip2-devel \
    gdbm-devel \
    libffi-devel \
    sqlite-devel \
    ncurses-devel \
    readline-devel \
    tk-devel \
    xz-devel \
    zlib-devel \
    wget ;\
    yum clean all
```

Download, build, and install Python 3.11:

```
mkdir /tmp/python3.11.1-build && \
    cd /tmp/python3.11.1-build && \
    curl .python.org/ftp/python/3.11.1/Python-3.11.1.tgz > python-3.11.1.tgz && \
    tar xzf python-3.11.1.tgz && \
    cd Python-3.11.1 && \
    mkdir -p /opt/python3.11.1/lib && \
    export CFLAGS="$CFLAGS $(pkg-config --cflags openssl11)" && \
    export LDFLAGS="$LDFLAGS $(pkg-config --libs openssl11)" && \
    ./configure --prefix=/opt/python3.11.1 \
        --with-ensurepip=install \
        --enable-optimizations \
```

```
        --enable-shared LDFLAGS="$LDFLAGS -Wl,-rpath /opt/python3.11.1/lib" && \
    make -j$(nproc)
```

Begin the compile process to ensure there are no errors:

```
cd /opt/python3.11.1/lib/python3.11
make install
```

Set up a symbolic link:

```
ln -s /opt/python3.11.1/bin/python3.11 /opt/python3.11.1/bin/python
```

RHEL 8

It is required to have developer packages installed to compile Python from source:

```
yum install -y yum-utils \
    make gcc \
    openssl openssl-devel \
    postgresql-devel \
    libtiff-devel libjpeg-devel libzip-devel freetype-devel \
    libwebp-devel tcl-devel libxslt-devel libxml2-devel \
    bzip2-devel \
    gdbm-devel \
    libffi-devel \
    sqlite-devel \
    ncurses-devel \
    readline-devel \
    tk-devel \
    xz-devel \
    zlib-devel \
    wget ;\
    yum clean all
```

Download, build, and install Python 3.11:

```
mkdir /tmp/python3.11.1-build && \
    cd /tmp/python3.11.1-build && \
    curl .python.org/ftp/python/3.11.1/Python-3.11.1.tgz > python-3.11.1.tgz && \
    tar xzf python-3.11.1.tgz && \
    cd Python-3.11.1 && \
    mkdir -p /opt/python3.11.1/lib && \
    export CFLAGS="$CFLAGS $(pkg-config --cflags openssl11)" && \
```

```
    export LDFLAGS="$LDFLAGS $(pkg-config --libs openssl11)" && \
    ./configure --prefix=/opt/python3.11.1 \
        --with-ensurepip=install \
        --enable-optimizations \
        --enable-shared LDFLAGS="$LDFLAGS -Wl,-rpath /opt/python3.11.1/lib" && \
    make -j$(nproc)
```

Begin the compile process to ensure there are no errors:

```
cd /opt/python3.11.1/lib/python3.11
make install
```

Set up a symbolic link:

```
ln -s /opt/python3.11.1/bin/python3.11 /opt/python3.11.1/bin/python
```

Python 3.11 SDK: TcEx Installation

Note: The instructions in this section need to be run after Python has been installed on the application server.

Note: The instructions in this section are based on the Python installation method discussed in this guide. Adjust directories as needed if Python is installed in another location.

To ensure that no permissions issues arise from the use of Python packages, use the following commands to update permissions:

```
chmod -R 755 /opt/python3.11.1/lib/python3.11/site-packages
chmod -R 755 /opt/python3.11.1/lib/python3.11/lib2to3
```

To install TcEx using pip, execute the following command:

```
/opt/python3.11.1/bin/pip3 install --upgrade pip
/opt/python3.11.1/bin/pip3 install tcex-cli
```

Install Java 11

Java® 11 (OpenJDK or Oracle® version 11) is required for the latest version of ThreatConnect to function properly.

1. Back up the **cacerts** file under the current Java installation. Note that this procedure will vary between OpenJDK and Oracle installations.
2. Run the following command to install the Java **.rpm** file that is currently downloaded:

```
yum localinstall <downloaded-java-rpm-file>.rpm
```

3. Run the following command and select the Java 11 option (typically option **2**, but this option may vary depending on the version of Java that is currently installed). Press the **Enter** key to save:

```
sudo alternatives --config java #
```

4. Note the location of the Java 11 installation. Edit **bashrc** for the **threatconnect** user account, and update the existing location for **JAVA_HOME** to the new location for the Java 11 installation.

```
Export JAVA_HOME= <location>
```

Important: These instructions use the delete method to remove the current installation of Java and then install the OpenJDK version of Java 11. Testing has not been performed for an upgrade of Java from 8 to 11.

Install Redis 6.2.6

1. Navigate to **/usr/local/bin**, which is the default install path for Redis®.
2. From command line interface (CLI), run the following command to use the Redis prompt:

```
./redis-cli
```

3. From the Redis prompt, run the following command to stop Redis:

```
Enter SHUTDOWN SAVE
```

4. Type **quit**.
5. From the Redis prompt, run the following command to back up Redis files to **/tmp/redis5**:

```
cp -P /usr/local/bin/redis-* /tmp/redis5
```

6. From the CLI, run the following command to search for the **dump.rdb** file:

```
find / -iname dump.rdb
```

7. Delete the **dump.rdb** file within the Redis directory, which is typically located at **/var/lib/redis**.
8. Download the source code:

```
wget .redis.io/releases/redis-6.2.6.tar.gz
```

9. Run the following command within the folder containing the **redis-6.2.6.tar.gz** file:

```
tar -xvzf redis-6.2.6.tar.gz
```

10. Navigate to the location of the un-tarred Redis installation:

```
cd redis-6.2.6
```

11. Run the following command to copy Redis to **/usr/local/bin**:

```
cp redis-* /usr/local/bin
```

> **Note**: The default location for the Redis installation is **/usr/local/bin**. If a different location is to be used, adjust the file path accordingly in the command.

12. Select **y** to overwrite the files that are being prompted.
13. Run the following command to start the Redis service:

```
service redis_6379 start
```

14. Navigate to **/usr/local/bin** (or the location where the Redis installation is located), and execute **./redis-cli**.
15. Type **info**, and then scroll to the top of the information displayed and verify that the Redis version displayed is **6.2.6**.
16. Type **quit** to complete the installation of Redis.

Configure Open File Limits

Standalone 3 Server ThreatConnect Server Instance

The following changes will need to be performed on the ThreatConnect server:

1. Add the following lines at the end of the **/etc/security/limits.conf** file:

```
threatconnect - nofile 150000
tc-job - nofile 100000
redis - nofile 100000
```

2. Add the following line at the end of the **/etc/sysctl.conf** file:

```
fs.file-max = 150000
```

All-in-One ThreatConnect Server

For an All-in-One server instance running ThreatConnect, the database server, and the OpenSearch server, the following changes need to be performed on the All-in-One ThreatConnect server:

1. Add the following lines at the end of the **/etc/security/limits.conf** file:

```
threatconnect - nofile 150000
tc-job - nofile 100000
redis - nofile 100000
```

2. Add the following line at the end of the **/etc/sysctl.conf** file:

```
fs.file-max = 150000
```

Create Database Backups

Before upgrading to the latest version of ThreatConnect, it is recommended that a backup of the database be performed.

MySQL

To perform a backup on the MySQL® server, run the following command:

```
mysqldump -u threatconnect user -p <database-name> > /<save-location>/TC_pre733.sql
```

Note: When naming the dump file, it is recommended to include the ThreatConnect version number from which you are upgrading in the file's name. For example, if you are upgrading from ThreatConnect version 7.3.3, name the dump file **TC_pre733.sql**.

Note: If a different MySQL socket is configured, execute the preceding command while specifying that particular socket.

If using MySQL 5.7 or upgrading from ThreatConnect version 6.7.2 or older, run the following commands to upgrade from MySQL 5.7 to 8.0:

```
yum repolist all | grep mysql
sudo yum-config-manager --disable mysql57-community
sudo yum-config-manager --enable mysql80-community
rpm --import .mysql.com/RPM-GPG-KEY-mysql-2022
yum update mysql-community-server
vi /etc/my.cnf
     # comment these two properties
     #innodb_large_prefix=on
     #innodb_file_format=barracuda
     # add this property
     max_connections = 300
systemctl start mysqld
tail -f /var/log/mysqld.log
```

PostgreSQL

To perform a backup on the PostgreSQL® server, run the following commands:

```
cd /tmp
sudo -u postgres pg_dumpall -p 5432 > TC_pre733.sql
```

Note: When naming the dump file, it is recommended to include the ThreatConnect version number from which you are upgrading in the file's name. For example, if you are upgrading from ThreatConnect version 7.3.3, name the dump file **TC_pre733.sql**.

If using PostgreSQL 11 or upgrading from ThreatConnect version 6.7.2 or older, perform the following steps as the Linux root user to upgrade PostgreSQL 11 to 14:

1. In the **/var/lib/pgsql/11/data/pg_hba.conf** and **/var/lib/pgsql/14/data/pg_hba.conf** files, locate the `local all all` entry and update the value in the `METHOD` column to `trust`.

   Important: Updating the value in the `METHOD` column for the `local all all` entry to `trust` is a temporary change that should be reverted after PostgreSQL 11 is upgraded to 14.

2. Restart PostgreSQL 11:

   ```
   service postgresql-11 restart
   ```

3. Create a backup:

   ```
   cd /tmp
   sudo -u postgres pg_dumpall -p 5432 > backup.sql
   ```

4. Download and install PostgreSQL 14 from https://www.postgresql.org/download/linux/redhat/.

5. Now that the repository has been installed and is active, utilize **yum** to install the PostgreSQL 14 server:

   ```
   yum install postgresql14 postgresql14-server -y
   ```

6. Initialize PostgreSQL 14:

   ```
   /usr/pgsql-14/bin/postgresql-14-setup initdb
   ```

7. Stop PostgreSQL 11:

   ```
   service postgresql-11 stop
   ```

8. Copy the **.conf** file from PostgreSQL 11 to PostgreSQL 14:

```
cp /var/lib/pgsql/11/data/*.conf /var/lib/pgsql/14/data/
```

> **Important**: The preceding command assumes the data folder is in the default location; however, your data folder may reside in **/opt/postgres**.

9. Use one of the following methods to upgrade PostgreSQL 11 to PostgreSQL 14:

   - Upgrade using the `pg_upgrade` command (**recommended**):

```
sudo -u postgres /usr/pgsql-14/bin/pg_upgrade \
--old-datadir=/var/lib/pgsql/11/data \
--new-datadir=/var/lib/pgsql/14/data \
--old-bindir=/usr/pgsql-11/bin \
--new-bindir=/usr/pgsql-14/bin \
--old-options '-c config_file=/var/lib/pgsql/11/data/postgresql.conf' \
--new-options '-c config_file=/var/lib/pgsql/14/data/postgresql.conf'
```

   - Upgrade from the **backup.sql** file:

```
service postgresql-14 start
sudo -u postgres psql < /tmp/backup.sql
```

10. Add the following property to **/var/lib/pgsql/14/data/postgresql.conf**:

```
max_connections = 300
```

11. Start PostgreSQL 14 (if it is not already started):

```
service postgresql-14 start
```

12. Remove PostgreSQL 11 (**optional**):

```
yum remove postgresql11*
rm -rf /var/lib/pgsql/11
```

Upgrade Steps

1. Stop the ThreatConnect service from running by using either of the following commands:

```
service threatconnect stop
```

```
systemctl stop threatconnect
```

2. Run the following commands to flush all Redis databases:

```
$ redis-cli
$ flushall
```

3. Run the following commands to back up the **threatconnect.xml** file, which is typically located at **/opt/threatconnect/config**:

```
cp /opt/threatconnect/config/threatconnect.xml
/opt/threatconnect/config/threatconnect.xml.bak
```

> **Note**: If an installation location other than **/opt/threatconnect/config** is set up within the organization's ThreatConnect installation, then this file will be located within the **/config/** folder for the organization.

4. Download all upgrade files and extract them in the **/tmp** directory.

> **Note**: A link will be supplied ahead of time for the latest ThreatConnect installer **.zip** file.

5. Run the following command to copy the new **version.txt** file, which is typically located at **/opt/threatconnect/config**:

```
cp /tmp/threatconnect/config/version.txt /opt/threatconnect/config/version.txt
```

> **Note**: If an installation location other than **/opt/threatconnect/config** is set up within the organization's ThreatConnect installation, then this file will be located within the **/config/** folder for the organization.

6. Rename the existing **app** directory in the current ThreatConnect installation, and then replace it with the **app** directory from the downloaded **.zip** file. The proper way to do this is to create a recursive copy of the new **app** folder and place it in the current installation directory, as demonstrated in the following example:

```
cp -R /tmp/threatconnect/app /opt/threatconnect
```

Important: Change the origin and destination directories according to the organization's current configuration, as the preceding command is just an example.

7. Confirm that the **threatconnect** user has the appropriate permissions for the newly copied **app** folder. Permission changes can be accomplished by running the following command:

```
chown -R threatconnect:threatconnect /opt/threatconnect/app
```

8. Confirm that Server Name Indication (SNI) has been enabled by ensuring that the following property is inside the **<system-properties/>** tag of the **threatconnect.xml** file:

```
<property name="jsse.enableSNIExtension" value="true"/>
```

9. Run the following commands to install dependencies for the ThreatConnect reporting feature:

```
yum install fontconfig libXrender libXext -y
curl -L https://dl.google.com/linux/direct/google-chrome-
stable_current_x86_64.rpm -o google-chrome.rpm && \
yum install ./google-chrome.rpm -y && \
rm -f google-chrome.rpm
echo "export CHROME_BIN=/usr/bin/google-chrome" >> /home/threatconnect/.bashrc
```

10. Run the following commands to apply secure permissions for App integrations:

```
# get rid of old group permissions
chgrp -R threatconnect /opt/threatconnect/exchange/

# get rid of old groups
groupdel tc-job-read
groupdel tc-job-write

# correct octal permissions
find /opt/threatconnect/exchange/ -type f -exec chmod 644 -- {} +
find /opt/threatconnect/exchange/ -type d -exec chmod 755 -- {} +

# set exiting ACLs
setfacl -Rm u:tc-job:rx /opt/threatconnect/exchange/programs/
setfacl -Rm u:tc-job:rwx /opt/threatconnect/exchange/jobs/
setfacl -Rm u:threatconnect:rwx /opt/threatconnect/exchange/jobs/

# set new default ACLs
```

```
setfacl -Rdm u:tc-job:rx /opt/threatconnect/exchange/programs/
setfacl -Rdm u:tc-job:rwx /opt/threatconnect/exchange/jobs/
setfacl -Rdm u:threatconnect:rwx /opt/threatconnect/exchange/jobs/
```

11. Run the following command to change to the **threatconnect** user account within the organization:

```
su - threatconnect
```

12. Navigate to the ThreatConnect installation folder that contains the **app** folder, which is typically **/opt/threatconnect/app**.
13. Execute the **setup.sh** script as the **threatconnect** user account.
14. Select option **6** to apply the upgrade.

> **Note**: The option number listed in this step does change at times with ThreatConnect releases.

15. After selecting option **6**, the ThreatConnect installer will prompt you for super user credentials. You must provide a super user account that can execute Database Definition Language (DDL) SQL statements such as CREATE TABLE and ALTER TABLE, as well as grant privileges on the ThreatConnect database. After entering the super user credentials, the ThreatConnect installer will upgrade the ThreatConnect database to the latest schema.

> **Warning**: Before completing this section, verify that the ThreatConnect database schema has been upgraded to 6.6.2.

16. Once the upgrade has been applied, select option **0** to exit the **setup.sh** script.

> **Warning**: Running **setup.sh** as the root user may inadvertently break permissions to some of the necessary configuration files.

> **Important**: If you are using Java version 11.0.11 or above, you will need to modify the **threatconnect.xml** file afterwards and make the following changes to each **"jdbc"** database connection entry:

a. Search for "**jdbc**" in the **threatconnect.xml** file within your ThreatConnect installation folder/configuration.
b. In the two **jdbc** entries, add the following string to each jdbc property, at the end of each string:

```
&amp;useSSL=false
```

c. Save the changes, and quit your file editor. Then proceed with the remaining steps in this section.

17. Upgrade your search index to OpenSearch 2.6.0, as described in the "Upgrade OpenSearch" section.

18. Start the ThreatConnect service using either of the following commands, depending on your environment:

```
service threatconnect start
```

```
systemctl start threatconnect
```

ThreatConnect should now be upgraded to the latest version.

Upgrade OpenSearch

Important: Before proceeding with these steps, confirm that ThreatConnect is not running.

1. Create a backup folder and change its owner to the **opensearch** user:

```
mkdir -p /opt/opensearch-1.3.2-backups/snapshot
chown opensearch:opensearch -R /opt/opensearch-1.3.2-backups
```

2. Back up the **data** folder:

```
cp -r /opt/opensearch-data /opt/opensearch-1.3.2-backups/data
```

3. Back up important OpenSearch configuration files (**certificates**, **opensearch.yml**, **jvm.options**):

```
mkdir /opt/opensearch-1.3.2-backups/config
cp -r /etc/opensearch/certificates /etc/opensearch/opensearch.yml
/etc/opensearch/jvm.options /opt/opensearch-1.3.2-backups/config/
```

4. Add the **path.repo** property to **opensearch.yml** (if it is not already added) and set it to the path for the backup folder created in Step 1:

```
path.repo: /opt/opensearch-1.3.2-backups/snapshot
```

5. Run the OpenSearch pre-upgrade script (**opensearch-pre-upgrade.sh**). This script creates a snapshot, accepts a username and password, and deletes indices that do not need to be reindexed, such as empty indices and WHOIS cache indices.

Important: If you intend to create a snapshot, then the path assigned to the **path.repo** property in Step 4 must match the path for the `-s` parameter in the following command. Also, if you already backed up the data at the path assigned to the **path.repo** property, then you most likely do not need to create a snapshot, in which case you can set the `-s` parameter to `false`.

```
cd /opt/threatconnect/app/util
yum install jq
./opensearch-pre-upgrade.sh -h https://localhost:9200 -s /opt/opensearch-1.3.2-backups/snapshot -u admin:password -d
```

Note: Running this script with no arguments will return a description of each available argument.

6. Upgrade OpenSearch to version 2.6.0:

```
cd /opt
systemctl stop opensearch.service
curl -
L  "https://artifacts.opensearch.org/releases/bundle/opensearch/2.6.0/opensearch-2.6.0-linux-x64.rpm" -o opensearch-2.6.0-linux-x64.rpm
yum install opensearch-2.6.0-linux-x64.rpm
systemctl enable opensearch.service
chmod 755 /etc/init.d/opensearch
cd /usr/share/opensearch
bin/opensearch-plugin remove ingest-attachment
bin/opensearch-plugin install --batch ingest-attachment
systemctl start opensearch.service
```