ThreatConnect® Environment Server Installation Guide

Software Version 2.2.0

Technical Guide

May 4, 2023

10028-14 EN Rev. A

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

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

Table of Contents

System Requirements

In order to install an instance of the ThreatConnect Environment Server, the requirements in the following sections must be met.

Hardware

The ThreatConnect Environment Server platform requires a server, virtual or physical, that meets the following minimum specifications:

- 4 CPU/vCPU Cores (2 GHz)
- 4 GB of memory
- 10 GB of storage

As the number or frequency of jobs increases, the need to increase system resources will likely occur. The listing in Table 1 highlights typical TC Exchange™ apps and their specific system-resource needs.

Table 1

App Name	Frequency	CPU Used	Memory Used
ArcSight™ EMS Extract	Daily	1.44	75
Tanium™ Extract v2.0	Daily	< 1	< 50
QRadar® Extract v2.0	Daily	<1	< 50
Palo Alto PAN-OS® Block List	Daily	.10	2.5

Software

The ThreatConnect Environment Server and its supporting packages require the following software environment in order to run properly:

- **Operating System**: Red Hat® Linux® variant—either Red Hat Enterprise Linux® (RHEL) 6, 7, or 8 or Community Enterprise Operating System (CentOS™) 6 or 7

> **Note**: This guide assumes that the user for the installation of ThreatConnect is named **threatconnect**.

- **Java® Development Kit (JDK)**: Access to a local installation of Java 11 (OpenJDK or Oracle Java version 11)
- **Python®**: Installation of Python 3.6.*x* and Python 3.11.*x* is required

Python 3.6 Installation

> **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 3.6 installed, skip the compile steps of the Python installation.

Install Python 3.6

Dependencies to compile Python from source:

```
yum install -y yum-utils epel-release --enablerepo=extras && \
    yum-builddep -y python python3 && \
    yum install -y make gcc \
    openssl openssl-devel \
    openssl11 openssl11-libs openssl11-devel \
    bzip2-devel \
    gdbm-devel \
    libffi-devel \
    sqlite-devel \
    ncurses-devel \
    readline-devel \
    tk-devel \
    xz-devel \
    zlib-devel \
    wget ;\
    yum clean all
```

Download the source code to **/tmp**:

```
cd /tmp
wget https://www.python.org/ftp/python/3.6.15/Python-3.6.15.tar.xz
```

Decompress the downloaded archive:

```
tar -xf Python-3.6.15.tar.xz
```

Navigate to the directory where the decompressed files reside:

```
cd Python-3.6.15
```

Run the following commands to configure Python:

```
mkdir -p /opt/python3.6.15/lib &&\
./configure --prefix=/opt/python3.6.15 \
--with-ensurepip=install --enable-optimizations \
--enable-shared LDFLAGS="$LDFLAGS -Wl,-rpath /opt/python3.6.15/lib"
```

Begin the compile process to ensure there are no errors:

```
make && make altinstall
```

Set up a symbolic link:

```
ln -s /opt/python3.6.15/bin/python3.6 /opt/python3.6.15/bin/python
```

Python 3.6 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/python3.6.15/lib/python3.6/site-packages
chmod -R 755 /opt/python3.6.15/lib/python3.6/lib2to3
```

To install TcEx using pip, execute the following command:

```
/opt/python3.6.15/bin/pip3.6 install tcex==2.0.29
```

Python 3.11 Installation

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 https://www.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:

```
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 https://www.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:

```
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==2.0.29
```

SMTP Server

TC Exchange requires an available Simple Mail Transfer Protocol (SMTP) server to send email alerts and to correspond with users. This server must be routable from the server running the platform, and if SMTP authorization is required, the ThreatConnect Environment Server will need access to a username and password in order to generate these emails.

Network Traffic Port Requirements

The ports and protocols listed in Table 2 must be opened when deploying the Environment Server inside a network. Appropriate firewall rules must be enabled for these ports from the machine running the Environment Server in order to allow connectivity to your ThreatConnect Dedicated Cloud instance.

Table 2

Network Port	Protocol	Traffic Direction	Description
443	HTTPS/TCP	Outbound to DC	This port connects to the ThreatConnect Dedicated Cloud API to download apps for execution. Traffic is limited to app installs and upgrades. App downloads are performed when an execution request is sent from the ThreatConnect Dedicated Cloud instance for the first time.
62000	TCP	Outbound to DC	This port is defined within the ThreatConnect System Settings. It enables the Environment Server to connect securely with the ThreatConnect Dedicated Cloud message broker to receive real-time commands in order to execute an app to fulfill orchestration requirements, as well as provide command-and-control capabilities. Traffic is lightweight and used primarily in a request/response model to direct app executions.

Preparing the Environment

Java JDK

The system needs access to the Java JDK as outlined in the "Software" section. In addition, the `JAVA_HOME` environment variable needs to be properly configured to point to that directory.

> **Note**: Some CentOS/Red Hat versions come with a pre-installed, though unsupported, version of OpenJDK.

Downloading Java

The ThreatConnect MEO (Multi-Environment Orchestration) Environment Server requires Java version 11.x to be installed and configured. The latest version of the ThreatConnect MEO software supports both OpenJDK and Oracle JDK version 11.

Installing and Configuring Java

> **Note**: The instructions in this section refer to a specific Oracle JDK version of Java 11. If using a different version, make sure to substitute the correct filename into the given commands.

Execute the following command:

```
rpm -ivh jdk-11.0.10_linux-x64_bin.rpm
```

Once installation is complete, execute the following command:

```
alternatives -- config java
```

This command will output the current location of the new Java installation.

Typically, an installation of this type will create a symbolic link to **/usr/java/latest**. The following command will confirm whether this location can be configured as the run location for Java:

```
cd /usr/java/latest
```

The next step is to create the **threatconnect** local OS user account:

```
adduser threatconnect
```

Use the following commands to log into and modify the **.bashrc** file for the **threatconnect** account:

```
su threatconnect
vi ~/.bashrc
```

Then add the following code as the second line of this file, where <*path to Java*> would typically look like **/user/java/latest**:

```
export JAVA_HOME=<path to Java>
```

Then reload the bash profile:

```
source ~/.bashrc
```

Installation

Getting Started

ThreatConnect clients can use this guide to configure and install their own Instance of the ThreatConnect Environment Server. This guide assumes a moderate level of systems-administration expertise and an operating environment that satisfies the requirements detailed previously. See *Playbook Environments* for more information about how to configure, administrate, and use Environments in ThreatConnect.

Downloading the Installer

The Environment Server installer **.zip** file is available for download on the **Environments** tab of the **Playbooks** screen in ThreatConnect (Figure 1).

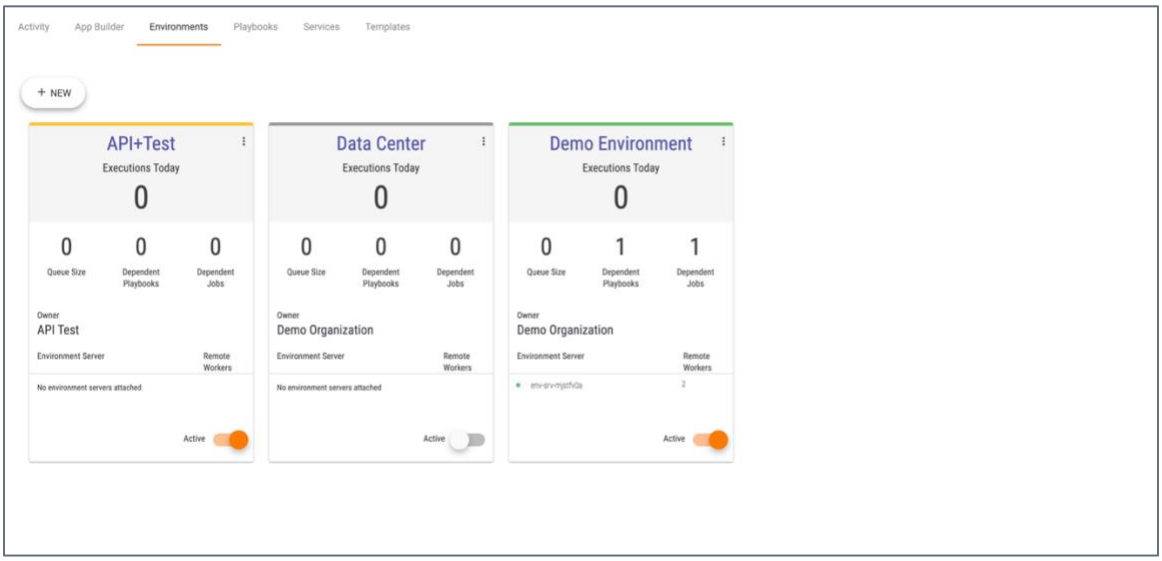

Figure 1

Click the vertical ellipsis at the upper-right corner of the desired Environment card and select **Download**. A window containing the download options for the Environment will be displayed (Figure 2).

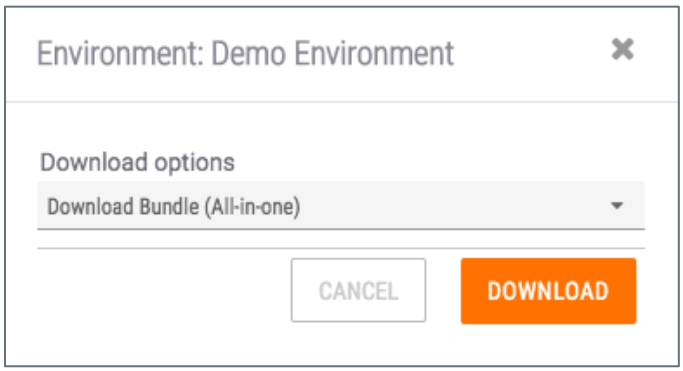

Figure 2

This window provides three options for download: **Download Bundle (All-in-one)**, **Environment Config Only**, and **Environment Server Only**. The **Download Bundle (All-in-one)** option includes both the Environment Config and the Environment Server as well as the KeyStore files required to make a secure connection to the host ThreatConnect instance. Select this option, and then click the **DOWNLOAD** button.

Note: Select **Environment Config Only** when an existing Environment Server needs to point to a new Environment. Select **Environment Server Only** when an existing Environment Server needs to be upgraded.

Opening the Installer

Unzipping the File

The ThreatConnect Environment Server .zip file serves as an archive of all the necessary files and folders needed to install the application. There are two ways to unzip this file:

1. Copy the **.zip** file to the desired directory on which the ThreatConnect Environment Server will be installed. By default, this directory is **/opt**, which will result in an installation directory of **/opt/threatconnect-envsvr**.
2. Unzip the file from the command-line interface with the following command:

```
unzip environment-server-bundle.zip.
```

3. Configure permissions within the operating system to ensure that the **threatconnect** user can access the ThreatConnect Environment Server files. In the following command, the default values of **threatconnect** and **/opt/threatconnect-envsvr**, respectively, are being used:

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

4. Run the following command to ensure that all **.sh** scripts are executable, which is a requirement for the MEO Environment Server:

```
chmod +x /opt/threatconnect-envsvr/*.sh
```

Installation Directory Structure

After the archive has been extracted, the new folder will contain the following directory structure:

```
threatconnect-envsvr/
    .tcenvsvr
    README.txt
    configure.sh
    run.sh
    shutdown.sh
    threatconnect-envsvr.init.sh
    threatconnect-envsvr.jar
```

- The **.tcenvsvr** directory contains the default keystore and broker connection settings.
- The **README.txt** file contains instructions on how to install the Environment Server.
- The **configure.sh** file is used to configure the Environment Server settings using a command-line interface.
- The **run.sh** file is used to run the Environment Server directly from the command line (not as a service).
- The **shutdown.sh** file is used to shut down the current Environment Server.
- The **threatconnect-envsvr.init.sh** file is the **init.d** service script.
- The **threatconnect-envsvr.jar** file is the file for the Environment Server only (i.e., it does not contain the configuration file).

Note: Users patching their ThreatConnect Environment Server Instance to a newer version should select the **Environment Server Only** option from an Environment instead of the **All-in-one Bundle**. (See the "Downloading the Installer" section for more information.) This download extracts to a **threatconnect-envsvr.jar** file. Replace the existing **.jar** file and restart to complete the upgrade.

The ThreatConnect Environment Server Setup

Creating **tc-job** User

Note: The information covered in this section is an optional security enhancement.

To provide additional security, it is recommended that a separate user on Linux systems be created to run TC Exchange jobs. It is also recommended that read and write groups be created to control the permissions to these files. Use the following code to perform these tasks:

```
useradd tc-job
echo "tc-job--pass123" | passwd tc-job --stdin
chgrp -R threatconnect /opt/threatconnect-envsvr/.tcenvsvr/exchange

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

# set new default ACLs
setfacl -Rdm u:tc-job:rx /opt/threatconnect-envsvr/.tcenvsvr/exchange/programs/
setfacl -Rdm u:tc-job:rwx /opt/threatconnect-envsvr/.tcenvsvr/exchange/jobs/
setfacl -Rdm u:threatconnect:rwx /opt/threatconnect-envsvr/.tcenvsvr/exchange/jobs/
```

User Privilege Configuration

Add the following lines to **/etc/pam.d/su** after the first `auth` command:

```
auth    sufficient pam_rootok.so
auth    [success=ignore default=1] pam_succeed_if.so user = tc-job
auth    sufficient pam_succeed_if.so use_uid user = threatconnect
```

Configuring sudoers

Create **/etc/sudoers.d/threatconnect** using the following command:

```
visudo -f /etc/sudoers.d/threatconnect
```

Then add the following lines:

```
Defaults:threatconnect !requiretty
threatconnect ALL=(tc-job) NOPASSWD: ALL
```

This configuration allows the **threatconnect** user to run the jobs as the **tc-job** user.

Starting the ThreatConnect Environment Server

After completing the installation and configuration procedures, it is time to start the ThreatConnect Environment Server.

Starting as a Linux Service

The options in the previous sections allow a user to run the ThreatConnect Environment Server in a single session. This approach presents a number of limitations: The platform will need to be started manually after each reboot, or a terminal window or Secure Shell (SSH) session may have to be left open. To address this problem, this section details the file configuration to run the ThreatConnect Environment Server as a service in Linux.

Open a terminal window and browse to the app directory within the ThreatConnect Environment Server directory. Run the run.sh file as follows:

```
su - threatconnect -c ./opt/threatconnect-envsvr/run.sh
```

Running this command will ensure proper connectivity to the ThreatConnect Dedicated Cloud instance for your organization. As long as logs indicating connectivity to your DC instance (FQDN:62000 with successful connection) are being generated, the MEO server will connect properly to the DC instance.

Run **CTRL-C** to force the process to close. Once the process is closed, execute the following command:

```
su - threatconnect -c ./opt/threatconnect-envsvr/configure.sh
```

A menu providing options for configuration of the MEO instance will be provided.

```
Please select an option:
     1: System Configuration
     2: Variables
     3: Exit
```

Select **1: System Configuration**. The System Configuration menu will be displayed.

```
System Configuration: Please select an option:
     1: List all System Config
     2: Edit System Config
     3: Export Configuration
     4: Go Back
```

Select **2: Edit System Config**. Then edit the Java and Python locations as they are configured within your current MEO server configuration. Typically, the options to select are **3: appsJavaHome** (for Java) and **5: appsPythonHome** and **6: appsPythonHome311** (for Python).

If implementing a proxy within the Environment Server, the fields in the following menu will need to be configured.

```
     13: proxyExternal = <empty>
     14: proxyHost = <empty>
     15: proxyPassword = <empty>
     16: proxyPort = <empty>
     17: proxyTC = <empty>
     18: proxyUsername = <empty>
```

- Set options 13 and 17 to **true**.
- Populate options 14, 15, 16, and 18 according to your organization's proxy configuration for where this server resides.

The Service Script

Within the ThreatConnect Environment Server installation, there is a script used for running the ThreatConnect Environment Server as an initialized service:

```
/opt/threatconnect-envsvr/threatconnect-envsvr.init.sh
```

This script must be copied into the **/etc/init.d** directory for it to be recognized as a system service. Note that users may require privileges to copy to this directory:

```
cp /opt/threatconnect-envsvr/threatconnect-envsvr.init.sh
/etc/init.d/threatconnect-envsvr
```

Configuring Services

The service script requires proper permissions and paths to be set.

1. Specify the `TCENVSVR_HOME` variable within the script to point to the path where the ThreatConnect Environment Server installation exists. By default, this path is **/opt/threatconnect-envsvr**.
2. Specify the `USER` variable within the script to identify which user owns the files for the ThreatConnect Environment Server application. It is advisable, for security reasons, that the root user not be employed. By default, the username is assumed to be **threatconnect**.

Starting the ThreatConnect Environment Server as a Service

Once the services have been configured, the ThreatConnect Environment Server can be started as a service. To do so, enter one of the following commands while logged in as the root user:

```
service threatconnect-envsvr start
/etc/init.d/threatconnect-envsvr start
```

To stop the service, use either one of the following commands:

```
service threatconnect-envsvr stop
```

or

```
/etc/init.d/threatconnect-envsvr stop
```

To have the ThreatConnect Environment Server start on system startup, issue the following commands after the script is configured in the **/etc/init.d** directory:

For sysVinit systems:

```
chkconfig --add threatconnect-envsvr
chkconfig threatconnect-envsvr on
```

For systemd systems:

```
systemctl enable threatconnect-envsvr
```

The First Login

Setting Master Key for Keychain

The Keychain feature is required for the ThreatConnect Environment Server. When prompted, enter a Master Password. The Master Password is used to encrypt sensitive values and is required on every server restart.

System Settings Checklist

Users should review their Instance settings to ensure that they are configured according to their needs. Table 3 provides a description of each system configuration value.

Table 3

System Configuration Value	Description
apiURL	This setting should point to the URL for the API at port 8443 (e.g., https://api.threatconnect.com:8443).
appDeliveryToken	This setting is the token that is used to authenticate with the App Catalog Server.
appsJavaHome	This setting holds the path to the Java binary.
appsNumberofJobExecutors	This setting is the number of Job Executors that can run concurrently. It is a factor of the number of CPUs and the available memory on the server. It should not exceed available resources.
appsPythonHome	This setting holds the path to the Python 3.6 binary.
appsPythonHome311	This setting holds the path to the Python 3.11 binary.

appsSandboxUser	This setting represents the user account used to execute Jobs. It is pertinent only in Linux installs.
appsSessionDaystoKeep	This setting is placed at 5 in Cloud. It indicates the number of days that logs will be kept in the Jobs log directory: **%threatconnect%/exchange/jobs**.
brokerHost	This setting is the remote host name of the messaging server to which the Environment Server will connect.
brokerToken	This setting is the secure key used to authenticate a connection to the remote message broker.
proxyExternal	This setting is set to true when all external connections for apps should be routed through a proxy server.
proxyHost	This setting is the proxy host to use if a proxy server is required. Acceptable values are a valid IP address or host name for a proxy accessible by the ThreatConnect instance.
proxyPassword	This setting is the proxy password to use if a proxy server requires authentication.
proxyPort	This setting is the proxy port to use if a proxy server is required. Enter a valid proxy port number.
proxyTC	This setting is set to true when all connections to the ThreatConnect host server should be routed through a proxy server.
proxyUsername	This setting is the proxy username to use if a proxy server requires authentication.

queueTransport	This setting is empty by default and utilizes the raw TCP socket for messaging services. For deployments that require a proxy, set this value to **websocket**. This will change the transport to an HTTP-based transport protocol supported by secured proxy environments. All traffic will move through port 62000 over HTTP/S. If the proxy is defined, then the Environment Server will utilize this proxy for all messaging traffic.
relaySystemInfoPublishSeconds	The frequency at which to notify the remote ThreatConnect instance of the status of the Environment Server.
serverName	The name of the Environment Server to display on the ThreatConnect **Environments** screen and administration page.
serverXid	This setting is a static number that uniquely identifies the given Environment Server. Its value should not be changed.