Vertiv™ Avocent® ADX Ecosystem

Certificate Replacement Release Notes

VERSION 1.0.4, OCTOBER 2022

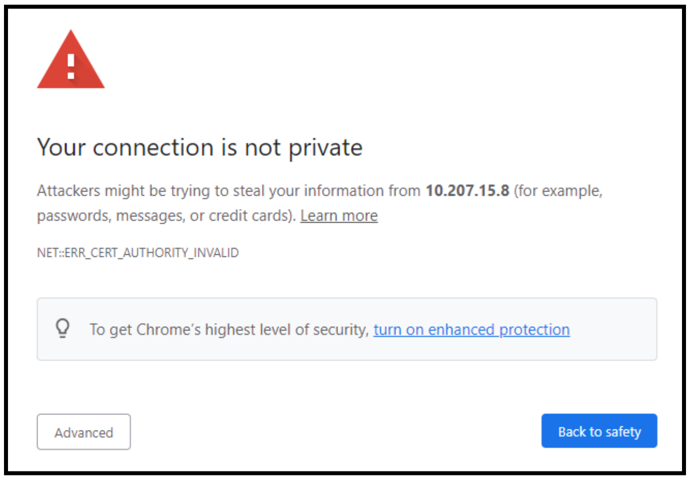
Release Notes Section Outline

1. Issue Overview

Vertiv™ Avocent® ADX Ecosystem devices use HTTPS protocols to communicate on web browser-based user interfaces (UIs) and a REST Application Programming Interface (API), which can be used for automation. HTTPS traffic is encrypted and secured using X.509 certificates. Since the default certificates on Vertiv™ Avocent® ADX Ecosystem devices are self-signed certificates, web browsers and other HTTPS clients will display warnings when connecting with Ecosystem devices. To eliminate these security warnings, the default self-signed certificates must be replaced with certificates signed by a Certificate Authority (CA). This document provides a sample Python script to assist you with the certificate replacement process.

NOTE: If you choose to generate certificates in-house, the in-house Certificate Authority must be added as a trusted CA for the browser and/or Operating System.

Example Warning



Your connection is not private

Attackers might be trying to steal your information from **10.207.15.8** (for example, passwords, messages, or credit cards). Learn more

NET::ERR_CERT_AUTHORITY_INVALID

To get Chrome's highest level of security, turn on enhanced protection

2. Certificate Replacement Script Overview

The sample adxcert.py Python script used in the following sections shows how to perform certificate replacement operations via the REST API on a Vertiv™ Avocent® ADX MP1000 Management Platform or Vertiv™ Avocent® ADX RM1048P Rack Manager appliance.

NOTE: If you need additional assistance with this process, please contact your Vertiv Technical Support representative.

By using the adxcert.py script, you will be able to do the following:

- Get a list of managed devices and their IDs
- Retrieve the current certificate from a device
- Create a Certificate Signing Request (CSR) for a device that can then be provided to a CA
- Install a CA-signed certificate matching the CSR

Prerequisites

- Ensure your computer or virtual machine is able to run Python 3.
  NOTE: Although the script was developed with Python v3.8.10, it should work with earlier versions as well.
  - Visit https://www.python.org for additional information on obtaining and installing Python.
  - Ensure your Operating System is compatible; this script was tested on Linux and Windows, but may run on other systems also.
    NOTE: If using Linux, Python 3 may already be included as a standard component.
- Ensure your computer or virtual machine is connected to the network where the Vertiv™ Avocent® ADX Ecosystem devices are installed.
- Ensure the script is copied to your computer or virtual machine.
- Ensure the commonly used "requests" package is installed.
  - Visit https://pypi.org/project/requests for additional information on obtaining and installing the requests package.
    NOTE: Version 2.28.1 of the requests package was used when the script was developed.

3. Running the Script

When your system meets the prerequisites, you are ready to run the script from your Operating System command prompt as shown in the following examples.

NOTE: Depending on where you choose to put the script, your prompts may differ from the examples shown.

Linux Operating System Example 1

```
~$ python3 adxcert.py
Performs various ADX certificate management operations.
Try 'adxcert.py help' for more information
~$
```

Linux Operating System Example 2

```
~$ ./adxcert.py
Performs various ADX certificate management operations.
Try 'adxcert.py help' for more information
~$
```

Windows Operating System Example

```
C:\adx\> python3 adxcert.py
Performs various ADX certificate management operations.
Try 'adxcert.py help' for more information
C:\adx\>
```

For a listing of parameters and options, enter the **help** command.

Help Command Output Sample

```
$ ./adxcert.py help
Avocent(r) ADX certificate management tool, v1.0.4
Performs various certificate management operations.

Usage:
  adxcert.py COMMAND

COMMAND can be:
  gencsr PARAMS...      Generate and return a Certificate Signing Request.
  getcert PARAMS...     Get the current certificate.
  setcert PARAMS...     Install a new certificate.
  getdevices PARAMS...  Get a list of managed devices.

Get the current certificate and store to the specified file:
  getcert ip=IPADDRESS user=USERNAME pass=PASSWORD id=DEVICEID file=FILE
    IPADDRESS        IP address of the management appliance (MP1000 or a
standalone RM1048P).
    USERNAME         for access to the management appliance.
    PASSWORD         for access to the management appliance.
    DEVICEID         Device ID of the device from which to get the
certificate. Use "local" for the management appliance itself.
    FILE             Name of file to store the retrieved certificate (PEM
format). If not supplied, stdout is used.

Generate and get a CSR and store to the specified file:
  gencsr ip=IPADDRESS user=USERNAME pass=PASSWORD id=DEVICEID
subject=SUBJECT san=SAN file=FILE
    IPADDRESS        IP address of the management appliance (MP1000 or a
standalone RM1048P).
    USERNAME         username for access to the management appliance.
    PASSWORD         password for access to the management appliance.
    DEVICEID         Device ID of the device on which to generate the CSR.
Use "local" for the management appliance itself.
    SUBJECT          Subject line for CSR:

/emailAddress=EMAIL/C=COUNTRY/ST=STATE/L=CITY/O=ORG/OU=UNIT/CN=COMNAME
                         COUNTRY    2-character country code (such as US)
                         EMAIL      email address (optional)
                         STATE      state or province (optional)
                         CITY       city (optional)
                         ORG        organization name (optional)
                         UNIT       organization unit name (optional)
                         COMNAME    common name (such as www.example.com)
    SAN              Subject Alternative Name line for CSR:
                       DNS:COMNAME,DNS:ALTNAME,DNS:ALTNAME,IP:ADDR...
                         COMNAME    alternative name (typically the same as
common name (such as www.example.com)
                         ALTNAME    alternative name (such as
www2.example.com)
                         ADDR       IP address (optional)
    FILE             Name of file to store the generated CSR (PEM format).
If not supplied, stdout is used.
```

Help Command Output Sample (Continued)

```
Install the provided certificate (must correspond to the most recently generated
CSR):
  setcert ip=IPADDRESS user=USERNAME pass=PASSWORD id=DEVICEID file=FILE
    IPADDRESS         IP address of the management appliance (MP1000 or a
standalone RM1048P).
    USERNAME          username for access to the management appliance.
    PASSWORD          password for access to the management appliance.
    DEVICEID          Device ID of the device on which to install the certificate.
Use "local" for the management appliance itself.
    FILE              Name of file that contains the certificate (PEM format).

Get the list of devices managed by a management appliance (MP1000 or standalone
RM1048P):
  getdevices ip=IPADDRESS user=USERNAME pass=PASSWORD file=FILE
    IPADDRESS         IP address of the management appliance (MP1000 or a
standalone RM1048P).
    USERNAME          username for access to the management appliance.
    PASSWORD          password for access to the management appliance.
    FILE              Name of file where output should be written. If not
supplied, stdout is used.

Examples:

Get the current certificate and store to the specified file:
  adxcert.py getcert ip=10.207.15.8 user=admin pass=secret id=local
file=10.207.15.8-cert.pem

Generate and get a CSR and store to the specified file:
  adxcert.py gencsr ip=10.207.15.8 user=admin pass=secret id=local
subject="/emailAddress=me@someplace.com/C=US/ST=Oregon/L=Widget City/O=Advance
Widgets/OU=Engineering/CN=www.example.com" san=DNS:www.example.com,IP:10.207.15.8
file=10.207.15.8-csr.pem

Install the provided certificate (must correspond to the previously generated
CSR):
  adxcert.py setcert ip=10.207.15.8 user=admin pass=secret id=local
file=10.207.15.8-newcert.pem

Get the list of devices managed by an appliance:
  adxcert.py getdevices ip=10.207.15.8 user=admin pass=secret

Get the current certificate for a managed device, using its device ID:
  adxcert.py getcert ip=10.207.15.8 user=admin pass=secret id=563de19b-a5b2-4818-
8fc4-015876bdc7f5 file=563de19b-a5b2-4818-8fc4-015876bdc7f5-cert.pem

$
```

4 Script Example Scenarios in Linux

In each of the example scenarios:

- Commands are shown from a Linux terminal session, where $ represents the command prompt and is not part of the actual command.
- The IP address and username/password of the Vertiv™ Avocent® ADX MP1000 Management Platform appliance are shown with placeholder values.
- Commands with the required parameters are long, so they are shown wrapped to multiple lines as necessary; the \ character is used to fit them within the width of this document and make them easier to read.
- The pem file listings are also abbreviated.
- If you need additional information on the meaning of the various parameters used in the examples, refer to the previous Help Command Output Sample.

Scenario 1: Replacing the Certificate on a Vertiv™ Avocent® ADX MP1000 Management Platform Appliance

To replace the management platform appliance certificate:

1. To ensure your system is properly set up and can communicate with the Vertiv™ Avocent® ADX MP1000 Management Platform appliance, enter a **getcert** command to retrieve the current certificate.

   NOTE: The id parameter in this command is "local", which means you must retrieve the certificate from the management platform appliance itself rather than from a device managed by the management platform appliance.

```
$ ./adxcert.py getcert ip=192.168.1.123 user=admin pass=mypass id=local \
> file=original-cert.pem
$ cat original-cert.pem
-----BEGIN CERTIFICATE-----
MIIF1jCCA76gAwIBAgIUAKWGdeea1Ol/yoW+bZVN87MweWwwDQYJKoZIhvcNAQEL
BQAwaDELMAkGA1UEBhMCVVMxEDAOBgNVBAgMB0FsYWJhbWExEzARBgNVBAcMCkh1
...
cHYUyuI2KUDUeFYuVOYHKBv0fxxjXpErudmbzD0ZSdEsVy9J4MKq6Fth4ZiMDSbm
gJQj7st1GvdHdQ==
-----END CERTIFICATE-----
$
```

2. Enter a **gencsr** command to generate a CSR.

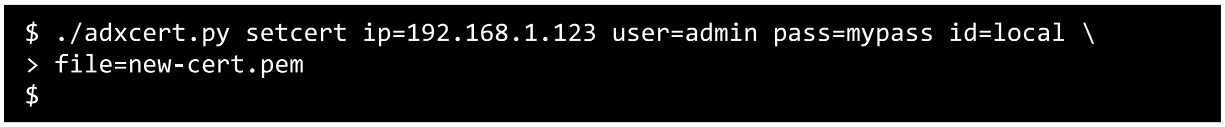
```
$ ./adxcert.py gencsr ip=192.168.1.123 user=admin pass=mypass id=local \
> file=csr.pem subject="/emailAddress=mp1000@example.com/C=US/ST=New \
> Jersey/L=Trenton/O=Widgets Inc/OU=Engineering/CN=mp1000.example.com" \
> san=DNS:mp1000.example.com,IP:192.168.1.123
$ cat csr.pem
-----BEGIN CERTIFICATE REQUEST-----
MIIE5jCCAs4CAQAwgaAxITAfBgkqhkiG9w0BCQEWEm1wMTAwMEBleGFtcGxlLmNv
bTELMAkGA1UEBhMCVVMxEzARBgNVBAgMCk5ldyBKZXJzZXkxEDAOBgNVBAcMB1Ry
...
1gCdIFqYczohfD3ZspR8HMGrcyMG7u/Mu56y4AoSM2fb5G3SuFnMsrdtcLYj/O7e
4eHxDl4bAo+6pQ==
-----END CERTIFICATE REQUEST-----
$
```

3. Submit the generated csr.pem file (this is the CSR) to a CA and ensure they send back the signed certificate.

   NOTE: The process of issuing/receiving the CA-signed certificate can vary depending on the CA used.

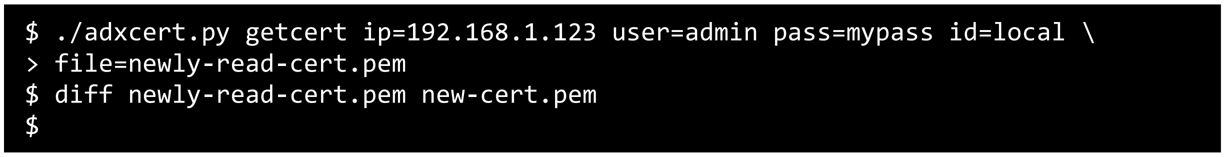
4. After receiving the signed certificate from the CA, enter a **setcert** command to install the signed certificate.

   NOTE: In the following screenshot, the certificate file is new-cert.pem.



```
$ ./adxcert.py setcert ip=192.168.1.123 user=admin pass=mypass id=local \
> file=new-cert.pem
$
```

5. Retrieve the new certificate and check it against the one received from the CA to confirm the certificate was installed as expected. If the two files match, there will be no output from the diff command (no differences, as shown in the following screenshot).



```
$ ./adxcert.py getcert ip=192.168.1.123 user=admin pass=mypass id=local \
> file=newly-read-cert.pem
$ diff newly-read-cert.pem new-cert.pem
$
```

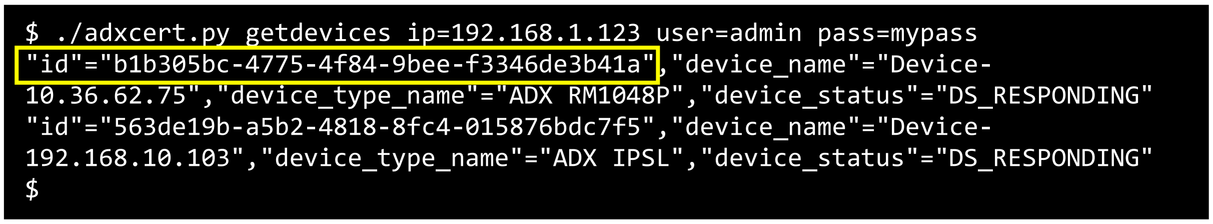
Scenario 2: Replacing the Certificate on a Managed Vertiv™ Avocent® ADX RM1048P Rack Manager Appliance

In this scenario, a Vertiv™ Avocent® ADX RM1048P Rack Manager appliance is being managed by a Vertiv™ Avocent® ADX MP1000 Management Platform appliance. The steps in this scenario also apply other devices managed by the management platform appliance, but the id value varies by device.

To replace the rack manager appliance certificate:

1. Enter a **getdevices** command to retrieve the list of managed devices and their corresponding id values.

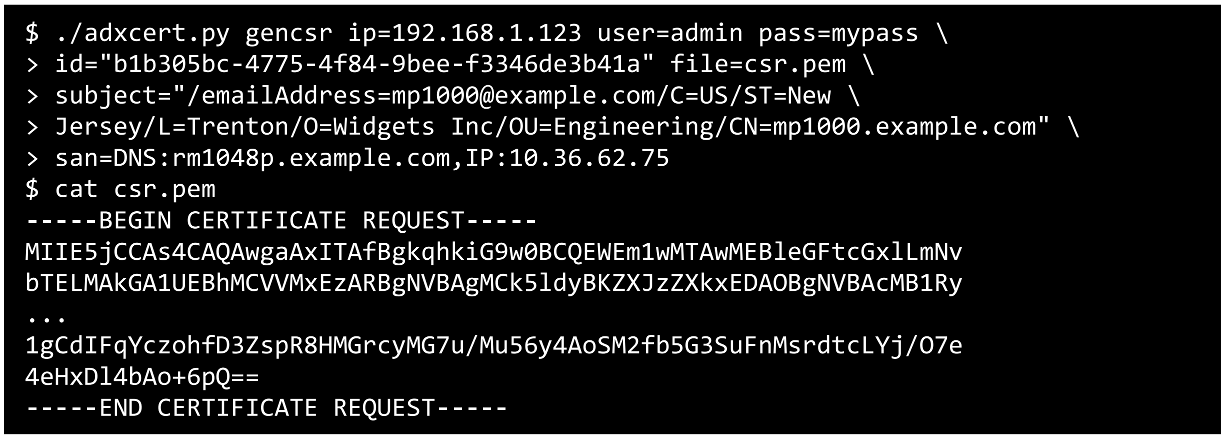
   NOTE: In the following screenshot, a list of two devices is shown and the output is wrapped to fit the terminal screen width. For this scenario, we are using the “ADX RM1048P” id value shown.



```
$ ./adxcert.py getdevices ip=192.168.1.123 user=admin pass=mypass
"id"="b1b305bc-4775-4f84-9bee-f3346de3b41a","device_name"="Device-
10.36.62.75","device_type_name"="ADX RM1048P","device_status"="DS_RESPONDING"
"id"="563de19b-a5b2-4818-8fc4-015876bdc7f5","device_name"="Device-
192.168.10.103","device_type_name"="ADX IPSL","device_status"="DS_RESPONDING"
$
```

2. Enter a **gencsr** command to generate a CSR.

   NOTE: In the following screenshot, the IP address used in the command is the address of the Vertiv™ Avocent® ADX MP1000 Management Platform appliance. The id value, however, is for the rack manager appliance (from the previous step).



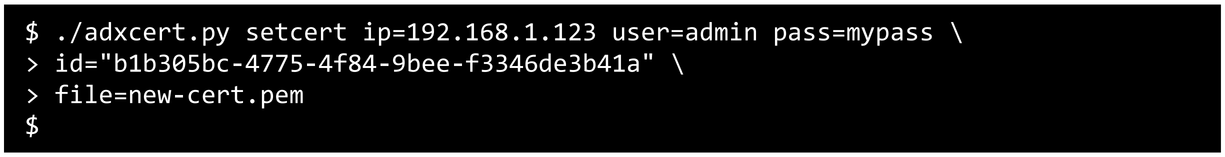
```
$ ./adxcert.py gencsr ip=192.168.1.123 user=admin pass=mypass \
> id="b1b305bc-4775-4f84-9bee-f3346de3b41a" file=csr.pem \
> subject="/emailAddress=mp1000@example.com/C=US/ST=New \
> Jersey/L=Trenton/O=Widgets Inc/OU=Engineering/CN=mp1000.example.com" \
> san=DNS:rm1048p.example.com,IP:10.36.62.75
$ cat csr.pem
-----BEGIN CERTIFICATE REQUEST-----
MIIE5jCCAs4CAQAwgaAxITAfBgkqhkiG9w0BCQEWEm1wMTAwMEBleGFtcGxlLmNv
bTELMAkGA1UEBhMCVVMxEzARBgNVBAgMCk5ldyBKZXJzZXkxEDAOBgNVBAcMB1Ry
...
1gCdIFqYczohfD3ZspR8HMGrcyMG7u/Mu56y4AoSM2fb5G3SuFnMsrdtcLYj/O7e
4eHxDl4bAo+6pQ==
-----END CERTIFICATE REQUEST-----
```

3. Submit the generated csr.pem file (this is the CSR) to a CA and ensure they send back the signed certificate.

   NOTE: The process of issuing/receiving the CA-signed certificate can vary depending on the CA used.

4. After receiving the signed certificate from the CA, enter a **setcert** command to install the signed certificate. For the device id value, continue using the rack manager appliance id.

   NOTE: In the following screenshot, the certificate file is new-cert.pem.



```
$ ./adxcert.py setcert ip=192.168.1.123 user=admin pass=mypass \
> id="b1b305bc-4775-4f84-9bee-f3346de3b41a" \
> file=new-cert.pem
$
```