

Handmatige XML API-oproepen naar CIMC uitvoeren

Inhoud

[Inleiding](#)

[Gebruikte componenten](#)

[Achtergrondinformatie](#)

[XML API-aanroepen naar CIMC uitvoeren](#)

[Stap 1](#)

[Stap 2](#)

[Stap 3](#)

[Stap 4](#)

[Problemen oplossen](#)

Inleiding

In dit document wordt beschreven hoe u handmatige XML API-aanroepen naar Cisco Integrated Management Controller (CIMC) kunt uitvoeren.

Gebruikte componenten

- Een Linux-machine (elke distributie).
- Netwerkconnectiviteit tussen de Linux-machine en Cisco CIMC (een succesvolle ping-test is voldoende).



Opmerking: De bestandsnamen (main.sh, login.xml, get_summary.xml) die in deze handleiding worden gebruikt, zijn willekeurig. U bent vrij om uw eigen naamgevingsconventies te gebruiken, zolang ze gedurende het hele proces naar behoren worden verwezen.

De informatie in dit document is gebaseerd op de apparaten in een specifieke laboratoriumomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk live is, moet u zorgen dat u de potentiële impact van elke opdracht begrijpt.

Achtergrondinformatie

In deze demonstratie wordt een Bash-script met de naam "main.sh" gebruikt om API-oproepen naar de server uit te voeren.

Houd er rekening mee dat alternatieve methoden, zoals CURL (rechtstreeks van CLI) of Python, ook kunnen worden gebruikt om hetzelfde resultaat te bereiken.

XML API-aanroepen naar CIMC uitvoeren

De CIMC XML API vereist een eerste authenticatie stap met behulp van een "aaaLogin" API oproep. Tijdens dit proces wordt een sessiecookie opgehaald, die vervolgens wordt gebruikt om alle volgende API-oproepen te verifiëren.

Stap 1

Begin met het maken van een bestand met de naam main.sh in het externe Linux-systeem. Dit bestand bevat een bash-script dat wordt gebruikt om de API-oproepen te verzenden.

Het main.sh-script wordt gedurende deze oefening meerdere keren uitgevoerd: eerst voor authenticatie en vervolgens voor het ophalen van informatie van de CIMC bij volgende oproepen.

Hier is de inhoud van het main.sh bestand:

```
#!/bin/bash
# CONFIGURATION - Edit these variables
CIMC_IP="X.X.X.X" # <<<<<<< Customer CIMC IP address
USERNAME="admin"
PASSWORD="xxxxxxx" # <<<<<<< CIMC Password in plaintext
XML_PAYLOAD_FILE="login.xml" # <<<<<<< Referencing the login.xml file for authentication and cookie ret
CIMC_URL="https://${CIMC_IP}/nuova" # <<<<<<< Call is made to this URL

# Check if payload file exists
if [ ! -f "$XML_PAYLOAD_FILE" ]; then
echo "Error: XML payload file '$XML_PAYLOAD_FILE' not found."
exit 1
fi
# Send XML request using curl
curl -k -s -u "${USERNAME}:${PASSWORD}" -H "Content-Type: text/xml" -d @"${XML_PAYLOAD_FILE}" "${CIMC_U
```

Het bash-script definieert de CIMC-aanmeldparameters en voert ook een XML API-aanroep uit, die wordt opgegeven in een ander bestand genaamd login.xml, geïdentificeerd door een variabele genaamd XML_PAYLOAD_FILE.

Het script controleert ook of het bestand login.xml, gedefinieerd door de variabele XML_PAYLOAD_FILE bestaat en een regulier bestand is.

Als het bestand dat is gedefinieerd door XML_PAYLOAD_FILE niet bestaat, wordt een fout afgedrukt en wordt het script afgesloten.

Sla het bestand op en maak het uitvoerbaar door deze opdracht uit te voeren in de CLI:

```
$ sudo chmod +x main.sh
```

Stap 2

Maak vervolgens het bestand login.xml in dezelfde Linux-directory als het main.sh-bestand.

Dit aanmeldingsbestand bevat de eigenlijke API-aanroep aaaLoginXML die naar de CIMC wordt verzonden om een sessiecookie op te halen. De opgehaalde cookie wordt gebruikt voor volgende API-oproepen:

Vergeet niet om de CIMC-gebruikersnaam en het CIMC-wachtwoord te vervangen door de juiste referenties.

Voer het main.sh-script uit in CLI om een cookie op te halen:

```
$ sudo ./main.sh
```

Als de API-aanroep succesvol is, bevat het geretourneerde XML-antwoord een sleutel genaamd outCookie met als waarde de opgehaalde cookie zoals weergegeven:

```
<?xml version="1.0"?>
<aaaLogin cookie="" response="yes" outCookie="xxxxxxxx/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx" outRefreshPeriod="600" outPriv="admin" outSessionId="18"
outVersion="4.3(5.250001)"> </aaaLogin>
```

Zoek vanuit de uitvoer de waarde outCookie.

Bewaar deze cookie waarde.

Stap 3

Maak een nieuw bestand aan in dezelfde map als waar het main.sh bestand zich bevindt. Geef het bestand de naam get_summary.xml.

Voor een uitgebreide lijst met API-aanvragen die kunnen worden gebruikt met de Cisco IMC, raadpleegt u de [CIMC XML API-programmeerhandleiding voor CIMC XML API-servers voor Cisco UCS-rackmontage](#).

Het nieuwe bestand get_summary.xml wordt gebruikt om de serveroverzichtsgegevens en de energiestatus van de host op te halen. met behulp van het XML-codeblok in de referentiedocumentatie, maar door de cookie-sleutelwaarde te vervangen door de eerder

opgehaalde cookie.

Vervang <cookie_value> door de outCookiewaarde die is verkregen uit de eerdere login.xmlresponse. Het bijgewerkte verzoek ziet er als volgt uit:

```
1 | <configResolveClass cookie="xxxxxxxx/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx"
    | inHierarchical="false" classId="computeRackUnit"/>
```

Zorg ervoor dat u de <cookie_value> vervangt door uw werkelijke cookie-waarde die is opgehaald tijdens het authenticatieproces.

Stap 4

Zodra de cookie is toegevoegd aan het nieuwe bestand get_summary.xml, werkt u het main.sh-script bij om te verwijzen naar de "get_summary.xml" als waarde voor de XML_PAYLOAD_FILE variabele voor dit verzoek.

```
#!/bin/bash
# CONFIGURATION - Edit these variables
CIMC_IP="X.X.X.X"
USERNAME="admin"
PASSWORD="xxxxxxx"
XML_PAYLOAD_FILE="get_summary.xml" # <<<<<<< Referencing the get_summary.xml file for subsequent API call
CIMC_URL="https://${CIMC_IP}/nuova"
# Check if payload file exists
if [ ! -f "$XML_PAYLOAD_FILE" ]; then
echo "Error: XML payload file '$XML_PAYLOAD_FILE' not found."
exit 1
fi
# Send XML request using curl
curl -k -s -u "${USERNAME}:${PASSWORD}" -H "Content-Type: text/xml" -d @"${XML_PAYLOAD_FILE}" "${CIMC_URL}"
```

Voer het main.sh-script opnieuw uit om het bijgewerkte API-verzoek uit te voeren.

```
$ sudo ./main.sh
```

Vervolgens ontvangt u een API-antwoord in XML-indeling dat het aangevraagde object retourneert

van CIMC.

```
<?xml version="1.0"?>
<configResolveClass cookie="xxxxxxxx/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx" response="yes" classId="computeRackUnit">
  <outConfigs>
    <computeRackUnit dn="sys/rack-unit-1" adminPower="policy" availableMemory="524288" model="UCSC-C220-M6S" memorySpeed="3200" name="UCS C220
    M6S" numOfAdaptors="2" numOfCores="56" numOfCoresEnabled="56" numOfCpus="2" numOfEthHostIfs="0" numOfFcHostIfs="0" numOfThreads="112" operPower
    ="off" originalUuid="XXXXXXXX-XXXX-XXXX-XXXXXXXXXXXX" presence="equipped" serverId="1"
    serial="XXXXXXXX" totalMemory="524288" usrLbl="" uuid="XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX" vendor="Cisco Systems
    Inc" cimcResetReason="graceful-reboot
    " assetTag="Unknown" adaptorSecureUpdate="Enabled" resetComponents="components" storageResetStatus="NA" vicResetStatus="NA" bmcResetStatus="NA" sma
    rtUsbAccess="disabled" smartUsbStatus="Disabled" biosPostState="pending"/>
  </outConfigs>
</configResolveClass>
```

Problemen oplossen

Het is belangrijk op te merken dat de aaaLogin XML API call een sessie cookie retourneert met een outRefreshPeriod van ongeveer 600 seconden. Dit betekent dat de cookie na tien (10) minuten vervalst als deze niet wordt vernieuwd en dat een nieuwe cookie nodig is om API-verzoeken te blijven uitvoeren.

Als u probeert een verlopen cookie te gebruiken (na 600 seconden), wordt een 552 "Autorisatie vereist" XML-foutantwoordblok geretourneerd:

```
<?xml version="1.0"?>
<configResolveDn cookie="xxxxxxxx/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx" response="yes" errorCode="552" invocationResult="service-
unavailable" errorDescr="Authorization required"> </configResolveDn>
```

Over deze vertaling

Cisco heeft dit document vertaald via een combinatie van machine- en menselijke technologie om onze gebruikers wereldwijd ondersteuningscontent te bieden in hun eigen taal. Houd er rekening mee dat zelfs de beste machinevertaling niet net zo nauwkeurig is als die van een professionele vertaler. Cisco Systems, Inc. is niet aansprakelijk voor de nauwkeurigheid van deze vertalingen en raadt aan altijd het oorspronkelijke Engelstalige document (link) te raadplegen.