

CIMCへの手動XML APIコールの実行

内容

[はじめに](#)

[使用するコンポーネント](#)

[背景説明](#)

[CIMCへのXML APIコールの実行](#)

[手順1](#)

[手順2](#)

[手順3](#)

[手順4](#)

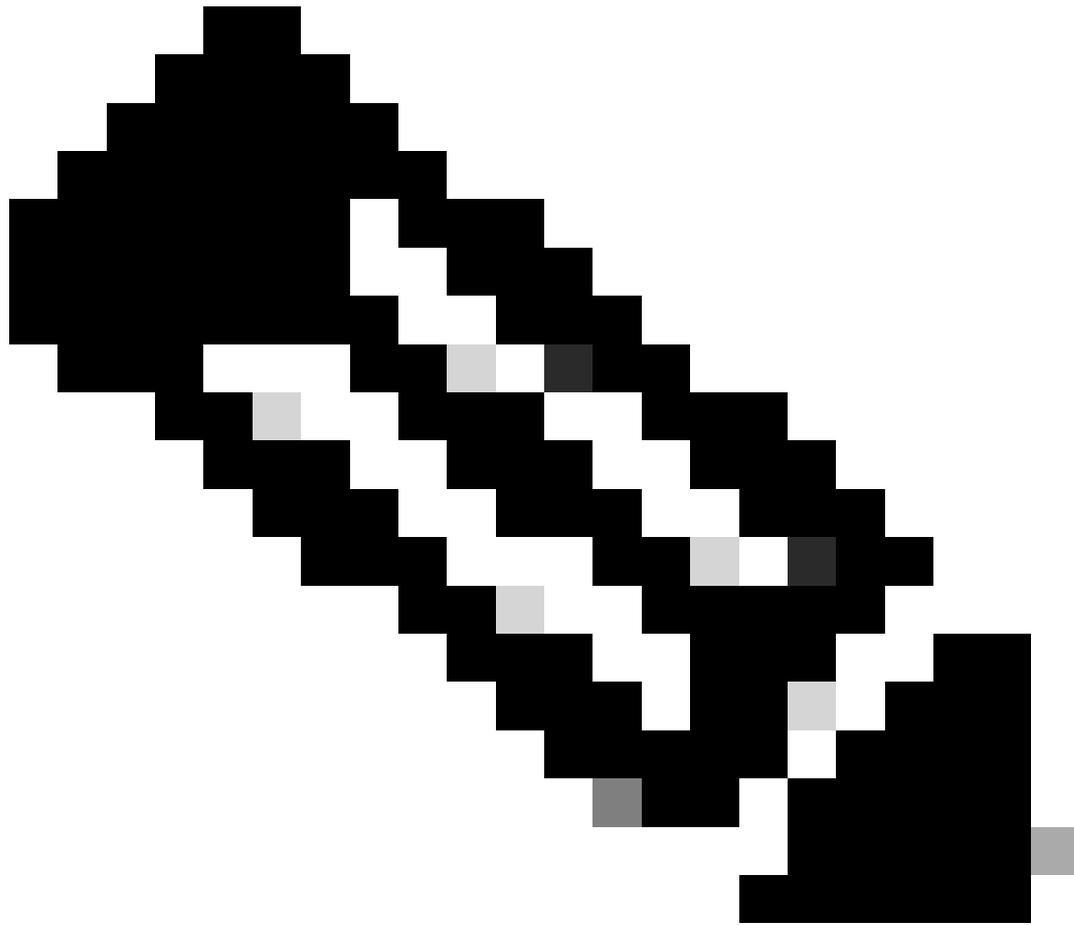
[トラブルシューティング](#)

はじめに

このドキュメントでは、Cisco Integrated Management Controller(CIMC)への手動XML APIコールを実行する方法について説明します。

使用するコンポーネント

- Linuxマシン (任意のディストリビューション) 。
- LinuxマシンとCisco CIMC間のネットワーク接続 (pingテストに成功すれば十分)



注：このガイドで使用されているファイル名(main.sh、login.xml、get_summary.xml)は任意です。プロセス全体を通して正しく参照されている限り、独自の命名規則を自由に使用できます。

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、クリアな（デフォルト）設定で作業を開始しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

背景説明

このデモンストレーションでは、「main.sh」という名前のBashスクリプトを使用して、サーバへのAPIコールを実行します。

CURL（直接CLIから）やPythonなどの別の方法でも、同じ結果を得ることができます。

CIMCへのXML APIコールの実行

CIMC XML APIでは、「aaaLogin」APIコールを使用した初期認証ステップが必要です。このプロセス中にセッションクッキーが取得され、後続のすべてのAPI呼び出しの認証に使用されます。

手順 1

まず、リモートLinuxマシンでmain.shという名前のファイルを作成します。このファイルには、APIコールの送信に使用されるbashスクリプトが含まれています。

main.shスクリプトは、この演習全体を通じて複数回実行されます。最初に認証を行い、次に後続のコールでCIMCから情報を取得します。

main.shファイルの内容は次のとおりです。

```
#!/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 retrieval
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_URL}"
```

bashスクリプトは、CIMCログインパラメータを定義し、XML APIコールも実行します。XML APIコールは、XML_PAYLOAD_FILEという変数によって識別されるlogin.xmlという別のファイルで指定されます。

このスクリプトは、変数XML_PAYLOAD_FILEで定義されたファイルlogin.xmlが存在するかどうか、および通常のファイルであるかどうかを確認します。

XML_PAYLOAD_FILEで定義されたファイルが存在しない場合、スクリプトはエラーを出力して終了します。

ファイルを保存し、CLIで次のコマンドを実行して実行可能にします。

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

手順 2

次に、main.shファイルと同じlinuxディレクトリに、login.xmlというファイルを作成します。

このログインファイルには、セッションCookieを取得するためにCIMCに送信される actualaaaLoginXML APIコールが含まれています。取得されたcookieは、後続のAPI呼び出しに使用されます。

CIMCのユーザ名とパスワードは適切なクレデンシャルに置き換えてください。

CLIでmain.shスクリプトを実行し、cookieを取得します。

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

API呼び出しが成功した場合、返されるXML応答には、次に示すように、取得されたcookieを値とするキーcalledoutCookieeが含まれます。

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

出力から、outCookie値を見つけます。

このCookie値を保存します。

手順 3

main.shファイルがあるディレクトリと同じディレクトリに新しいファイルを作成します。ファイルにget_summary.xmlという名前を付けます。

Cisco IMCで使用できるAPI要求の包括的なリストについては、『[Cisco UCSラックマウントサーバCIMC XML APIプログラマガイド](#)』を参照してください。

新しいget_summary.xmlファイルは、サーバの概要情報とホストの電力状態を取得するために使用されます。この場合、参照ドキュメント内のコードのXMLブロックを使用しますが、cookieキーの値は、以前に取得したcookieで置き換えます。

<cookie_value>の部分は、以前のlogin.xmlresponseから取得したtheoutCookievalueに置き換えてください。更新された要求は次のようになります。

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

<cookie_value>の部分は、認証プロセスで取得した実際のcookieの値に置き換えてください。

手順 4

cookieを新しいget_summary.xmlファイルに追加したら、この要求のXML_PAYLOAD_FILE変数の値として「get_summary.xml」を参照するようにmain.shスクリプトを更新します。

```
#!/bin/bash
# CONFIGURATION - Edit these variables
CIMC_IP="X.X.X.X"
USERNAME="admin"
PASSWORD="xxxxxxxx"
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}"
```

main.shスクリプトを再実行して、更新されたAPI要求を実行します。

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

次に、CIMCから要求されたオブジェクトを返すXML形式のAPI応答を受信します。

```
<?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-XXXX-XXXXXXXXXXXX" presence="equipped" serverId="1"
serial="XXXXXXXX" totalMemory="524288" usrLbl="" uid="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>
```

トラブルシュート

aaaLogin XML APIコールが、約600秒のoutRefreshPeriodを持つセッションクッキーを返すことに注意してください。つまり、更新されないとcookieは10分後に期限切れになり、API要求の実行を続行するには新しいcookieが必要になります。

期限切れのCookieを使用しようとする(600秒後)、552「Authorization required」XMLエラー応答ブロックが返されます。

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

翻訳について

シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。