

CIMC에 대한 수동 XML API 호출 수행

목차

[소개](#)

[사용되는 구성 요소](#)

[배경 정보](#)

[CIMC에 대한 XML API 호출 수행](#)

[1단계](#)

[2단계](#)

[3단계](#)

[4단계](#)

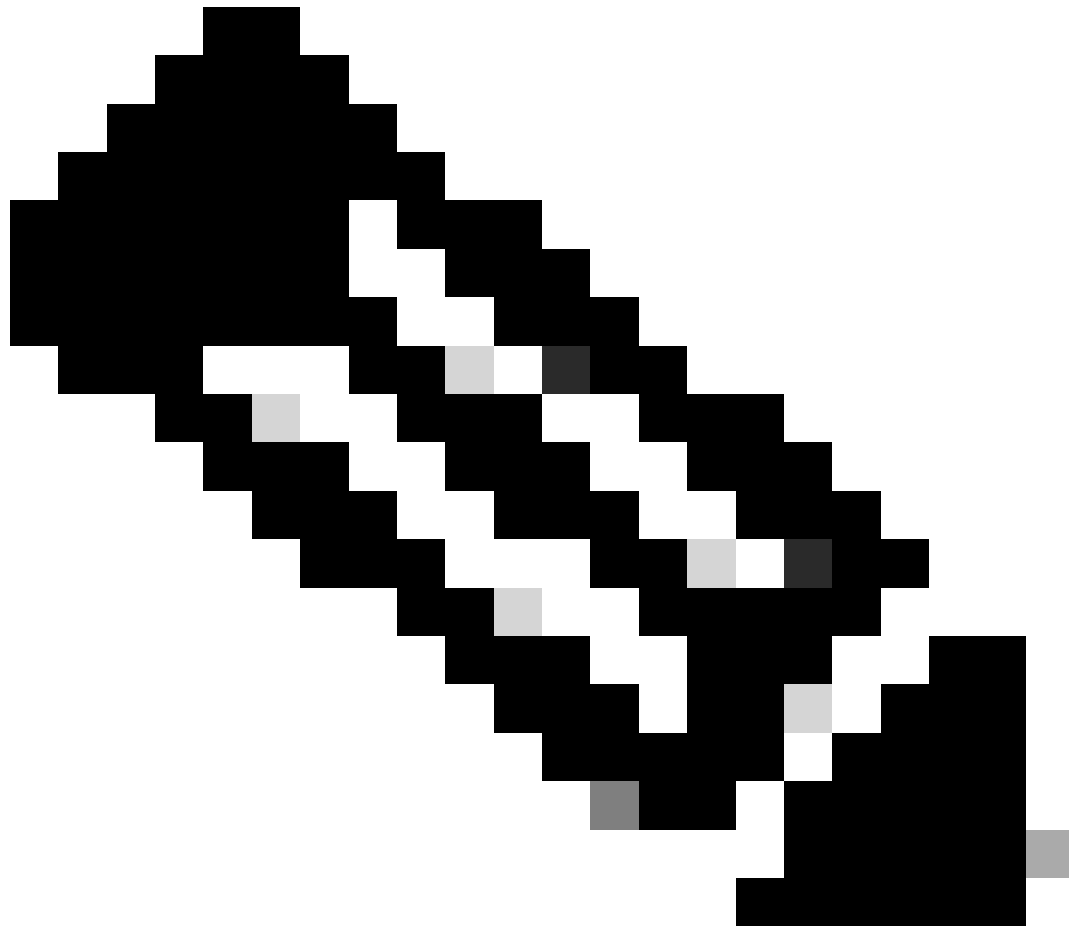
[문제 해결](#)

소개

이 문서에서는 CIMC(Cisco Integrated Management Controller)에 대한 수동 XML API 호출을 수행하는 방법에 대해 설명합니다.

사용되는 구성 요소

- Linux 시스템(모든 배포판).
- Linux 시스템과 Cisco CIMC 간의 네트워크 연결(ping 테스트를 성공하면 충분함)



참고: 이 설명서에서 사용되는 파일 이름(main.sh, login.xml, get_summary.xml)은 임의적입니다. 프로세스 전체에서 제대로 참조되는 경우 고유한 명명 규칙을 자유롭게 사용할 수 있습니다.

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 미리 숙지하시기 바랍니다.

배경 정보

이 데모에서는 "main.sh"라는 Bash 스크립트를 사용하여 서버에 대한 API 호출을 실행합니다.

CURL(CLI에서 직접) 또는 Python과 같은 대체 방법도 이와 동일한 결과를 얻을 수 있습니다.

CIMC에 대한 XML API 호출 수행

CIMC XML API에서는 "aaaLogin" API 호출을 사용하여 초기 인증 단계를 수행해야 합니다. 이 프로세스 중에 세션 쿠키가 검색되며, 이 쿠키는 모든 후속 API 호출을 인증하는 데 사용됩니다.

1단계

원격 Linux 시스템에서 main.sh라는 파일을 만드는 것으로 시작합니다. 이 파일에는 API 호출을 전송하는 데 사용되는 bash 스크립트가 포함되어 있습니다.

main.sh 스크립트는 이 연습 동안 여러 번 실행됩니다. 먼저 인증을 수행한 다음 후속 통화에서 CIMC에서 정보를 검색합니다.

다음은 main.sh 파일의 내용입니다.

```
#!/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 스크립트는 CIMC 로그인 매개변수를 정의하고 login.xml이라는 또 다른 파일에 지정된 XML API 호출을 실행합니다. 이는 XML_PAYLOAD_FILE이라는 변수로 식별됩니다.

스크립트는 또한 변수 XML_PAYLOAD_FILE에 의해 정의된 login.xml 파일이 있으며 일반 파일인지 확인합니다.

XML_PAYLOAD_FILE에 의해 정의된 파일이 없으면 스크립트는 오류를 인쇄하고 종료합니다.

파일을 저장하고 CLI에서 다음 명령을 실행하여 실행할 수 있게 합니다.

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

2단계

그런 다음 main.sh 파일과 동일한 linux 디렉토리에 login.xml이라는 파일을 생성합니다.

이 로그인 파일에는 세션 쿠키를 검색하기 위해 CIMC로 전송되는 실제 aaaLoginXML API 호출이 포함되어 있습니다. 검색된 쿠키는 후속 API 호출에 사용됩니다.

CIMC 사용자 이름 및 비밀번호를 적절한 자격 증명으로 교체해야 합니다.

CLI에서 main.sh 스크립트를 실행하여 쿠키를 검색합니다.

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

API 호출이 성공하면 반환된 XML 응답에는 다음과 같이 검색된 쿠키와 함께 calledoutCookiekey가 포함됩니다.

```
<?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>
```

출력에서 outCookie 값을 찾습니다.

이 쿠키 값을 저장합니다.

3단계

main.sh 파일이 있는 디렉토리와 동일한 디렉토리에 새 파일을 생성합니다. 파일 이름을 get_summary.xml로 지정합니다.

Cisco IMC와 함께 사용할 수 있는 API 요청의 포괄적인 목록을 보려면 [Cisco UCS Rack-Mount Servers CIMC XML API Programmer's Guide](#)를 참조하십시오.

새 get_summary.xml 파일은 서버 요약 정보 및 호스트 전원 상태를 검색하는 데 사용됩니다. 참조 설명서의 XML 코드 블록을 사용하되 쿠키 키 값을 이전 검색 쿠키로 바꿉니다.

<cookie_value>를 이전 login.xmlresponse에서 가져온 outCookievalue로 교체합니다. 업데이트된

요청은 다음과 같습니다.

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

인증 프로세스 중에 검색된 실제 쿠키 값으로 <cookie_value>를 교체해야 합니다.

4단계

쿠키가 새 get_summary.xml 파일에 추가되었으면 main.sh 스크립트를 업데이트하여 이 요청에 대한 XML_PAYLOAD_FILE 변수 값으로 "get_summary.xml"을 참조합니다.

```
#!/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}"
```

업데이트된 API 요청을 실행하려면 main.sh 스크립트를 다시 실행하십시오.

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

그런 다음 CIMC에서 요청된 객체를 반환하는 XML 형식의 API 응답을 받습니다.

```
<?xml version="1.0"?>
<configResolveClass cookie="xxxxxxxx/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx" 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="XXXXXXXXXXXX" 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>
```

문제 해결

aaaLogin XML API 호출은 outRefreshPeriod가 약 600초로 설정된 세션 쿠키를 반환한다는 점에 유의해야 합니다. 즉, 쿠키를 새로 고치지 않으면 10분 후에 만료되며 API 요청을 계속 수행하려면 새 쿠키가 필요합니다.

만료된 쿠키를 사용하려고 할 경우(600초 후) 552 "Authorization required" XML 오류 응답 블록이 반환됩니다.

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

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.