Realizar chamadas manuais de API XML para o CIMC

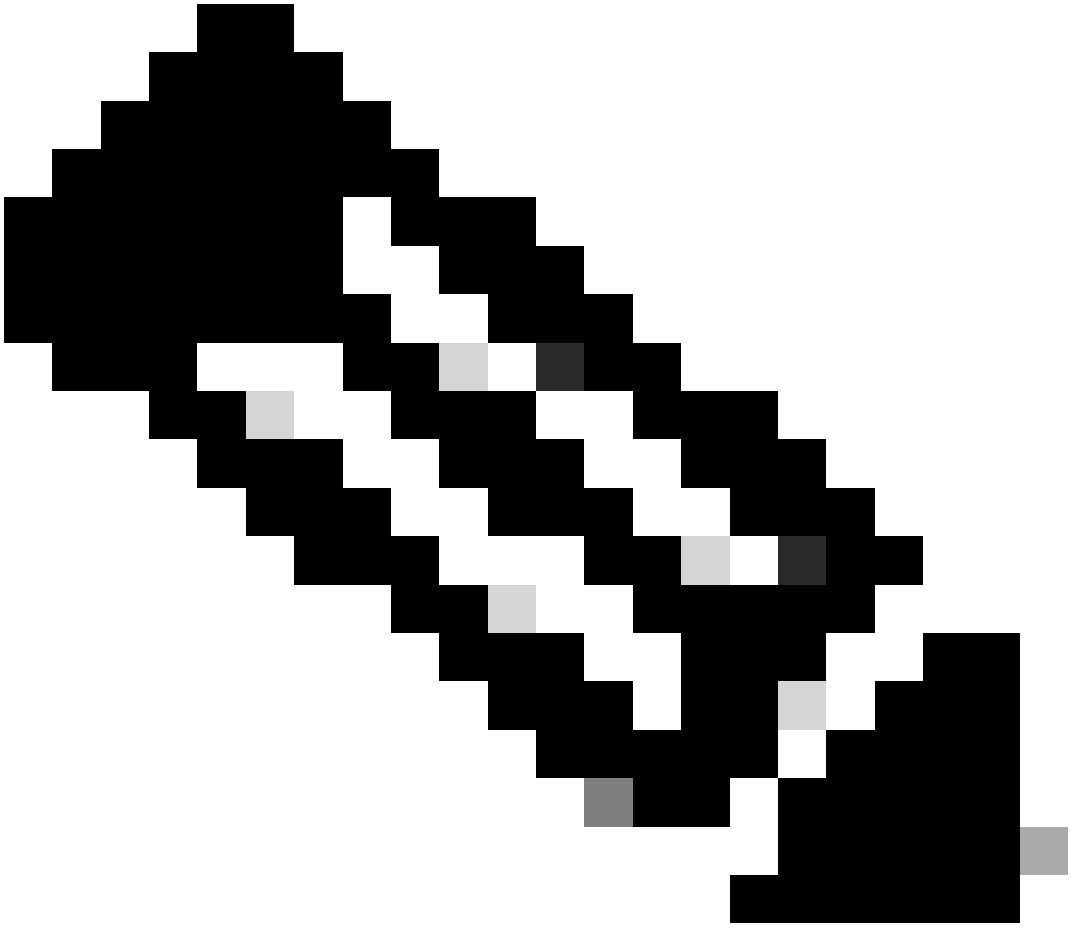
Contents

Introdução

Este documento descreve como realizar chamadas de API XML manuais para o Cisco Integrated Management Controller (CIMC).

Componentes Utilizados

- Uma máquina Linux (qualquer distribuição).
- Conectividade de rede entre a máquina Linux e o Cisco CIMC (um teste de ping bem-sucedido é suficiente).



Note: Os nomes de arquivo (main.sh, login.xml, get_summary.xml) usados neste guia são arbitrários. Você pode usar suas próprias convenções de nomenclatura, desde que elas sejam referenciadas adequadamente durante todo o processo.

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a rede estiver ativa, certifique-se de que você entenda o impacto potencial de qualquer comando.

Informações de Apoio

Nesta demonstração, um script Bash chamado "main.sh" é usado para executar chamadas de API para o servidor.

Observe que métodos alternativos, como CURL(diretamente do CLI) ou Python, também podem ser usados para atingir esse mesmo resultado.

Realizar chamadas de API XML para o CIMC

A API XML do CIMC requer uma etapa de autenticação inicial usando uma chamada à API "aaaLogin". Durante esse processo, um cookie de sessão é recuperado, que é usado para autenticar todas as chamadas de API subsequentes.

Passo 1

Comece criando um arquivo chamado main.sh na máquina Linux remota. Este arquivo contém um script bash usado para enviar as chamadas de API.

O script main.sh é executado várias vezes durante este exercício: primeiro para autenticação e, em seguida, para recuperar informações do CIMC em chamadas subsequentes.

Aqui está o conteúdo do arquivo main.sh:

```
#!/bin/bash
# CONFIGURATION - Edit these variables
CIMC_IP="X.X.X.X" # <<<<<<<< Customer CIMC IP address
USERNAME="admin"
PASSWORD="xxxxxxxx" # <<<<<<< 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
```

O script bash define os parâmetros de login do CIMC e também executa uma chamada de API XML, especificada em outro arquivo chamado login.xml identificado por uma variável chamada XML_PAYLOAD_FILE.

O script também verifica se o arquivo login.xml, definido pela variável XML_PAYLOAD_FILE existe e é um arquivo regular.

Se o arquivo definido por XML_PAYLOAD_FILE não existir, o script imprimirá um erro e sairá.

Salve o arquivo e torne-o executável executando este comando na CLI:

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

Passo 2

Em seguida, crie o arquivo chamado login.xml no mesmo diretório linux do arquivo main.sh.

Este arquivo de login contém a chamada de API actualaaaLoginXML que é enviada ao CIMC para recuperar um cookie de sessão. O cookie recuperado é usado para chamadas API subsequentes:

Lembre-se de substituir o nome de usuário e a senha do CIMC pelas credenciais apropriadas.

Execute o script main.sh na CLI para recuperar um cookie:

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

Se a chamada de API for bem-sucedida, a resposta XML retornada conterá uma chave chamada outCookie cujo valor será o cookie recuperado conforme mostrado:

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

Na saída, localize o valor outCookie.

Salvar este valor de cookie.

Etapa 3

Crie um novo arquivo no mesmo diretório em que o arquivo main.sh está localizado. Nomeie o arquivo get_summary.xml.

Para obter uma lista abrangente de solicitações de API que podem ser usadas com o Cisco IMC, consulte o Guia do programador de API XML para servidores com montagem em rack Cisco UCS.

O novo arquivo get_summary.xml é usado para recuperar as Informações de Resumo do Servidor e o Estado de Potência do Host. usando o bloco XML de código na documentação de referência, mas substituindo o valor da chave cookie pelo cookie de recuperação anterior.

Substitua <cookie_value> por theoutCookievalue obtido do login anterior.xmlresponse. A solicitação atualizada é semelhante a esta:

```
<configResolveClass cookie="xxxxxxxxx/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
inHierarchical="false" classId="computeRackUnit"/>
```

Certifique-se de substituir o <cookie_value> pelo valor real do cookie recuperado durante o processo de autenticação.

Passo 4

Depois que o cookie tiver sido adicionado ao novo arquivo get_summary.xml, atualize o script main.sh para fazer referência ao "get_summary.xml" como valor para a variável XML_PAYLOAD_FILE para essa solicitação.

```
#!/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 ca
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_U
```

Execute o script main.sh novamente para executar a solicitação de API atualizada.

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

Em seguida, você recebe uma resposta de API em formato XML retornando o objeto solicitado do CIMC.

```
<?xml version="1.0"?>
<configResolveClass cookie="xxxxxxxxx/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="XXXXXXXXXX" 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>
```

Troubleshooting

É importante observar que a chamada de API XML aaaLogin retorna um cookie de sessão com um outRefreshPeriod de aproximadamente 600 segundos. Isso significa que o cookie expira após dez (10) minutos se não for atualizado, e um novo cookie é necessário para continuar a executar as solicitações de API.

Se você tentar usar um cookie expirado (após 600 segundos), um bloco de resposta de erro 552 "Authorization required" XML será retornado:

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

Sobre esta tradução
A Cisco traduziu este documento com a ajuda de tecnologias de tradução automática e humana para oferecer conteúdo de suporte aos seus usuários no seu próprio idioma, independentemente da localização.

Observe que mesmo a melhor tradução automática não será tão precisa quanto as realizadas por um tradutor profissional.

A Cisco Systems, Inc. não se responsabiliza pela precisão destas traduções e recomenda que o documento original em inglês (link fornecido) seja sempre consultado.