

Realización de llamadas manuales de API XML a CIMC

Contenido

[Introducción](#)

[Componentes Utilizados](#)

[Antecedentes](#)

[Realizar llamadas API XML a CIMC](#)

[Paso 1](#)

[Paso 2](#)

[Paso 3](#)

[Paso 4](#)

[Troubleshoot](#)

Introducción

Este documento describe cómo realizar llamadas manuales de API XML a Cisco Integrated Management Controller (CIMC).

Componentes Utilizados

- Una máquina Linux (cualquier distribución).
- Conectividad de red entre la máquina Linux y Cisco CIMC (una prueba de ping exitosa es suficiente).



Nota: Los nombres de archivo (`main.sh`, `login.xml`, `get_summary.xml`) utilizados en esta guía son arbitrarios. Puede utilizar sus propias convenciones de nomenclatura, siempre que se haga referencia a ellas correctamente durante el proceso.

La información que contiene este documento se creó a partir de los dispositivos en un ambiente de laboratorio específico. Todos los dispositivos que se utilizan en este documento se pusieron en funcionamiento con una configuración verificada (predeterminada). Si tiene una red en vivo, asegúrese de entender el posible impacto de cualquier comando.

Antecedentes

En esta demostración, se utiliza una secuencia de comandos de Bash denominada "`main.sh`" para ejecutar llamadas de API en el servidor.

Tenga en cuenta que los métodos alternativos, como CURL (directamente desde CLI) o Python, también pueden utilizarse para lograr este mismo resultado.

Realizar llamadas de API XML a CIMC

La API XML de CIMC requiere un paso de autenticación inicial mediante una llamada a la API "aaaLogin". Durante este proceso, se recupera una cookie de sesión que, a continuación, se utiliza para autenticar todas las llamadas de API posteriores.

Paso 1

Comience creando un archivo llamado main.sh en la máquina Linux remota. Este archivo contiene un script bash utilizado para enviar las llamadas API.

La secuencia de comandos main.sh se ejecuta varias veces a lo largo de este ejercicio: primero para la autenticación y luego para la recuperación de información del CIMC en llamadas subsiguientes.

Aquí está el contenido del archivo main.sh:

```
#!/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
```

La secuencia de comandos bash define los parámetros de inicio de sesión CIMC y también ejecuta una llamada API XML, especificada en otro archivo denominado login.xml identificado por una variable denominada XML_PAYLOAD_FILE.

La secuencia de comandos también comprueba si el archivo login.xml, definido por la variable XML_PAYLOAD_FILE existe y es un archivo normal.

Si el archivo definido por XML_PAYLOAD_FILE no existe, la secuencia de comandos imprime un error y sale.

Guarde el archivo y conviértalo en ejecutable ejecutando este comando en la CLI:

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

Paso 2

A continuación, cree el archivo denominado login.xml en el mismo directorio linux que el archivo main.sh.

Este archivo de inicio de sesión contiene la llamada API realLoginXML que se envía al CIMC para recuperar una cookie de sesión. La cookie recuperada se utiliza para las llamadas de API posteriores:

Recuerde reemplazar el nombre de usuario y la contraseña de CIMC con las credenciales apropiadas.

Ejecute la secuencia de comandos main.sh en CLI para recuperar una cookie:

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

Si la llamada a la API se realiza correctamente, la respuesta XML devuelta contiene una clave llamada outCookie cuyo valor es la cookie recuperada, como se muestra a continuación:

```
<?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>
```

En el resultado, localice el valor outCookie.

Guarde este valor de cookie.

Paso 3

Cree un nuevo archivo en el mismo directorio en el que se encuentra el archivo main.sh. Asigne al archivo el nombre get_summary.xml.

Para obtener una lista completa de las solicitudes de API que se pueden utilizar con Cisco IMC, consulte la [Guía del programador de la API XML CIMC de los servidores de montaje en bastidor Cisco UCS](#).

El nuevo archivo get_summary.xml se utiliza para recuperar la Información de resumen del

servidor y el Estado de alimentación del host. mediante el bloque de código XML de la documentación de referencia, pero reemplazando el valor de la clave cookie por la cookie de recuperación anterior.

Reemplace <cookie_value> por theoutCookivalue obtenido de la respuesta de inicio de sesión anterior.xmlresponse. La solicitud actualizada tiene el siguiente aspecto:

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

Asegúrese de reemplazar <cookie_value> por el valor real de la cookie recuperado durante el proceso de autenticación.

Paso 4

Una vez que la cookie se ha agregado al nuevo archivo get_summary.xml, actualice la secuencia de comandos main.sh para hacer referencia a la variable "get_summary.xml" como valor de la variable XML_PAYLOAD_FILE para esta solicitud.

```
#!/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}"
```

Ejecute de nuevo la secuencia de comandos main.sh para ejecutar la solicitud de API actualizada.

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

A continuación, recibirá una respuesta API con formato XML que devolverá el objeto solicitado de 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-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>
```

Troubleshoot

Es importante tener en cuenta que la llamada API XML aaaLogin devuelve una cookie de sesión con un valor outRefreshPeriod de aproximadamente 600 segundos. Esto significa que la cookie caduca después de diez (10) minutos si no se actualiza y que se necesita una nueva cookie para seguir realizando solicitudes de API.

Si intenta utilizar una cookie caducada (después de 600 segundos), se devuelve un bloque de respuesta de error XML 552 "Authorization required":

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

Acerca de esta traducción

Cisco ha traducido este documento combinando la traducción automática y los recursos humanos a fin de ofrecer a nuestros usuarios en todo el mundo contenido en su propio idioma.

Tenga en cuenta que incluso la mejor traducción automática podría no ser tan precisa como la proporcionada por un traductor profesional.

Cisco Systems, Inc. no asume ninguna responsabilidad por la precisión de estas traducciones y recomienda remitirse siempre al documento original escrito en inglés (insertar vínculo URL).