Práctica de laboratorio: Almacenes de entidades de certificación

Objetivos

Parte 1: Certificados de Confianza de nuestros Navegadores

Parte 2: Búsqueda de ataques Man-In-the-Middle

Aspectos Básicos / Escenario

Con la evolución de la web, también aumentó la necesidad de medidas de seguridad. Así fue como HTTPS (la 'S' signfica seguridad) junto con el concepto de Autoridad emisora de Certificados (CA, Certificate Authority) fueron presentados por Netscape en 1994 y siguen utilizándose actualmente. En esta práctica de laboratorio:

- Generarán una lista de todos los certificados en los que confían sus navegadores (lo harán en sus computadoras).
- Utilizarán hashing para detectar si su conexión a Internet está siendo interceptada (completado en la máquina virtual CyberOps Workstation)

Recursos necesarios

- Máquina virtual CyberOps Workstation
- Acceso a Internet

Instrucciones

Parte 1: Certificados de Confianza de nuestro Navegador

HTTPS depende de una entidad externa para la validación. Conocida como Autoridad de Certificación (CA, Certification Authority) esta entidad externa verifica si un nombre de dominio realmente pertenece a la organización que dice ser ser su propietario. Si la verificación es positiva, la CA crea un certificado con firma digital que contiene información sobre la organización, incluida su clave pública.

Todo el sistema se basa en el hecho de que los navegadores web y los sistemas operativos se envían con una lista de CAs de confianza. El navegador considerará como legítimo cualquier certificado firmado por cualquiera de las CA de la lista y confiarán en él automáticamente. Para fortalecer la seguridad y escalabilidad del sistema, las CA a menudo distribuyen la tarea de creación y firma de certificados en muchas CA secundarias. La CA principal se conoce como CA Raíz. Si un navegador confía en una CA Raíz, también confía todas sus CA secundarias.

Nota: Si bien los almacenes de certificados son similares en todos los navegadores, en esta práctica de laboratorio nos enfocamos en **Chrome 81** y **Firefox 75**. El menú y los gráficos pueden ser diferentes en otras versiones de los navegadores web.

Seguir los pasos para mostrar el almacén de CA en los navegadores:

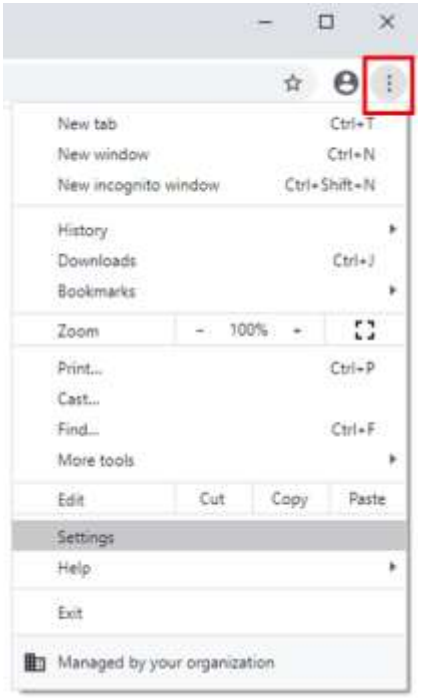
Paso 1: Mostrar los Certificados Raíz en Chrome

Puede realizar este paso en el equipo local o utilizar Firefox en la Máquina Virtual CyberOps Workstation. Si utilizamos Firefox, seguir con el Paso 2. Si no utilizamos Chrome ni Firefox, debemos buscar los pasos para mostrar los respectivos certificados raíz en Internet.

Nota: El menú y los gráficos pueden ser diferentes en otras versiones del navegador Chrome.

a. Abrir el navegador web Chrome en sus PC.

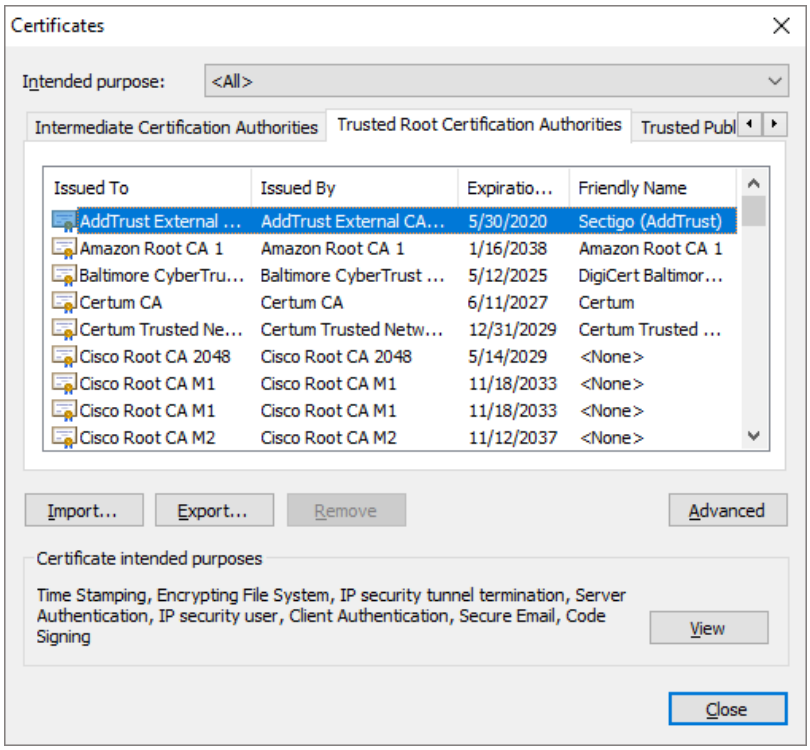
b. Hacer clic en el icono de tres puntos que se encuentra en el extremo derecho de la barra de búsqueda para mostrar las opciones de Chrome. Haga clic en **Configuración**.



c. Hacer clic en la pestaña **Privacidad y seguridad** y hacer clic en **Seguridad**.

d. Dirigase al final de la ventana y haga clic sobre **Gestionar Certificados.**

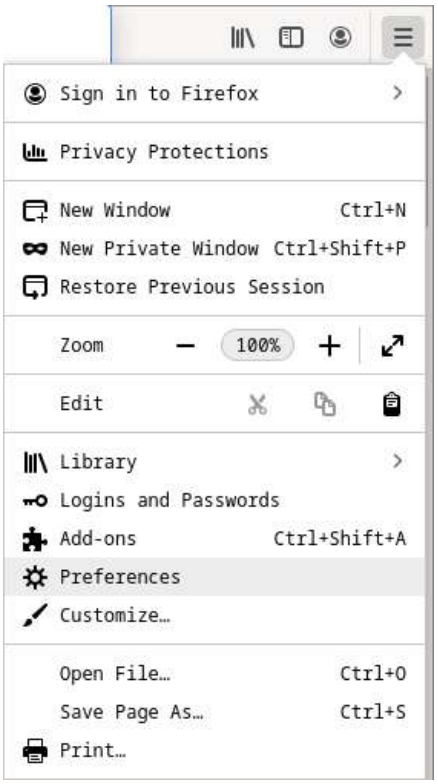
e. En la ventana Gestionar Certificados, selecciona la pestaña **Entitades de Certificación Raíz de Confianza** para mostrar todos los certificados y las CA de confianza de Chrome.



Paso 2: Mostrar los certificados en el Almacén de CA en Firefox

Nota: El menú y los gráficos pueden ser diferentes en otras versiones del navegador Firefox y en diferentes sistemas operativos. **En este paso se muestra Firefox 75** en la Máquina Virtual CyberOps Workstation.

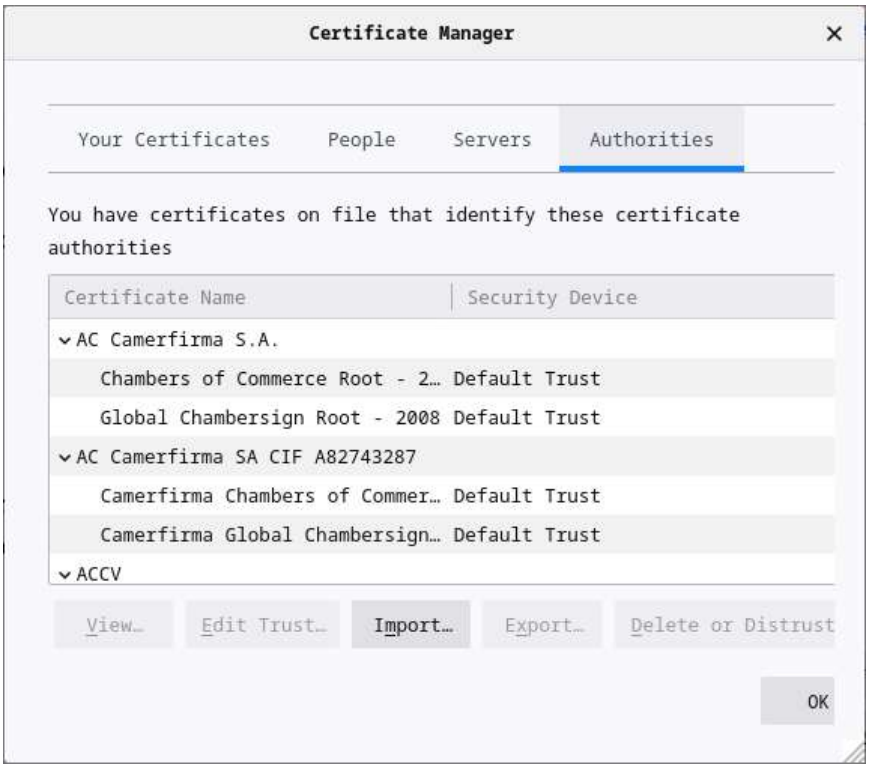
a. Abrir Firefox y hagan clic en el icono del Menú. El icono de **Menú** se encuentra en el extremo derecho de la ventana de Firefox, al lado de la barra de direcciones. Hacer clic en **Preferencias**.



b. Hacer clic en **Privacidad y seguridad** en el panel izquierdo.

c. Desplazarse hasta la sección Seguridad y hacer clic en **Ver certificados**.

d. Se abrirá una ventana con todos los certificados y las CA de confianza de Firefox.



Parte 2: Buscar ataques MITM (Man-In-The-Middle)

Esta parte se realiza con la Máquina CyberOps Workstation.

Una de las utilidades de hashing es verificar la integridad de los datos; pero también se pueden usar para detectar ataques MITM en HTTPS.

Para proteger los datos de los usuarios, cada vez más sitios web están adoptando el tráfico cifrado. Conocido como HTTPS, los sitios utilizan protocolos como TLS/SSL para cifrar el tráfico de los usuarios de extremo a extremo. Después de cifrar el tráfico correctamente es muy difícil que cualquier tercero que no sea el usuario o el sitio en cuestión vea el contenido del mensaje cifrado. Esto es bueno para los usuarios pero crea un problema para las organizaciones que quieren ver el contenido de ese tráfico. Las empresas y las organizaciones a menudo optan por espiar el tráfico generado por sus empleados para monitorearlos. Necesitaban poder ver el contenido del tráfico cifrado TLS/SSL. Esto se hace por medio de un proxy HTTPS.

Los navegadores web confían en la identidad de un sitio web que se visita si el certificado que presenta ese sitio web está firmado por una de las CA instaladas en el almacén de certificados del navegador. Para poder espiar el tráfico cifrado TLS/SSL de sus usuarios, una empresa u organización simplemente agrega otra CA a la lista de CA instaladas del navegador del usuario.

Consideren la siguiente situación hipotética: la Empresa X contrata a un empleado nuevo y le entrega una laptop nueva de la empresa. Antes de hacerlo, el departamento de TI de la empresa instala todo el software necesario para el trabajo. Entre el software y los paquetes que se instalan, el departamento de TI también incluye una CA adicional a la lista de CAs de confianza. Esta CA adicional apunta a una computadora controlada por la empresa conocida como el proxy HTTPS. Como la empresa controla los patrones de tráfico, el proxy HTTPS se puede ubicar en el medio de cualquier conexión. Funciona de la siguiente manera:

1. El usuario trata de establecer una conexión segura al sitio web HTTPS "H", alojado en Internet. "H" puede ser cualquier sitio HTTPS: un banco, una tienda en línea, un servidor de correo electrónico, entre otros.

2. Como la empresa controla los patrones de tráfico, lo hace de modo que todo el tráfico del usuario deba pasar por el proxy HTTPS. Entonces, el proxy HTTPS *se hace pasar por* el sitio web "H" y presenta un certificado firmado automáticamente para demostrar que es "H". Esencialmente, el proxy HTTPS dice: “Hola, soy un sitio HTTPS "H". Este es mi certificado. Fue firmado por… mí mismo”.
3. Como el certificado presentado está firmado por una de las CA incluidas en el almacén de CA de la laptop (recuerden que el departamento de TI la agregó), el navegador web cree erróneamente que de hecho se está comunicando con "H". Observen que, de no haberse agregado la CA adicional al almacén de CA, la laptop no confiaría en el certificado y se daría cuenta inmediatamente de que alguien más estaba tratando de *hacerse pasar por* "H".
4. La laptop confía en la conexión y establece un canal seguro con el proxy HTTPS, porque cree erróneamente que se está comunicando en forma segura con "H".
5. Entonces, el proxy HTTPS establece una segunda conexión a "H", el sitio web al que el usuario estaba tratando de acceder desde el comienzo.
6. Ahora, el proxy HTTPS es el punto extremo de dos conexiones seguras individuales; una establecida con el usuario y la otra con"H". Como el HTTPS es el punto extremo de ambas conexiones, ahora puede descifrar tráfico proveniente de las dos.
7. Ahora el proxy HTTPS puede recibir tráfico del usuario cifrado con TLS/SSL destinado a "H", descifrarlo, inspeccionarlo, volver a cifrarlo con TLS/SSL y enviarlo a "H". Cuando "H" responde, el proxy HTTPS invierte el proceso antes de reenviar el tráfico al usuario.

Observemos que el proceso pasa desapercibido para el usuario, que ve la conexión como cifrada con TLS/SSL (remarcadas de color verde en el navegador). Si bien la conexión es segura (cifrada con TLS/SSL), se la estableció con un sitio web falso.

Incluso si su presencia pasa desapercibida para el usuario, los proxys TLS se pueden detectar fácilmente con la ayuda de hashes. Si consideramos el ejemplo anterior, como el proxy HTTPS no tiene acceso a las claves privadas de "H", el certificado que le presenta al usuario difiere del que presenta "H". En cada certificado se incluye un valor conocido como *huella digital*. En esencia, una huella digital es un hash calculado y firmado por el emisor del certificado que actúa como un resumen único de todo el contenido del certificado. Si se modifica al menos una de las letras del certificado, la huella digital generará un valor completamente diferente al calcularla. Debido a esta propiedad, las huellas digitales se utilizan para comparar certificados rápidamente. Si volvemos al ejemplo anterior, el usuario puede solicitar el certificado de "H" y comparar la huella digital que contiene con la proporcionada al establecer la conexión con el sitio web "H". Si las huellas digitales coinciden, la conexión realmente se estableció con "H". Si no coinciden, la conexión se estableció con algún otro punto extremo.

Seguir los pasos que se indican a continuación para determinar si hay un proxy HTTPS en sus conexiones.

Paso 1: Recopilar la huella digital del certificado correcta y no modificada

El primer paso es recopilar algunas huellas digitales de sitios. Estos es importante porque se las utilizará para compararlas más adelante. La siguiente tabla contiene las huellas digitales de los certificados de algunos sitios populares.

Nota: Es posible que las huellas digitales de SHA-1 que se muestran en la Tabla 1 ya no sean válidas debido a que las organizaciones renuevan sus certificados regularmente. La huella digital también se llama huella dactilar en máquinas basadas en Windows.

Tabla 1: Sitios populares y las huellas digitales de sus certificados "SHA-1"

Sitio	Dominios cubiertos por el certificado	Huella digital de certificado SHA-1 (a partir de Mayo 2020)
www.cisco.com	www.cisco.com	E2:BD:0B:58:C6:B4:FF:91:D6:23:AB:44:0D:8F:64:76:29:4E:30:0B
www.facebook.com	*.facebook.com	BB:E7:A0:97:C7:92:B2:2D:00:38:12:69:E4:64:E9:04:96:4B:C7:41
www.wikipedia.org	*.wikipedia.org	A8:F9:F7:79:BE:DB:3E:EB:59:F0:1D:A6:34:08:A1:64:5D:28:48:44
twitter.com	twitter.com	73:33:BB:96:1D:DB:9C:0C:4F:E5:1C:FF:68:26:CF:5E:3F:50:AB:96
www.linkedin.com	www.linkedin.com	04:BC:C5:09:DD:AE:99:40:7E:99:A5:65:32:68:EC:5D:2D:D7:5A:19

¿Qué son las huellas digitales? ¿Por qué son importantes?

¿Quién calcula las huellas digitales? ¿Cómo se las encuentra?

Paso 2: Recoger la huella digital del certificado que está utilizando la Máquina Virtual CyberOps Workstation

Ahora que tenemos las huellas digitales reales, es momento de obtener huellas de un host local y comparar los valores. Si las huellas digitales no coinciden, el certificado en uso NO pertenece al sitio HTTPS que se está verificando, lo que significa que hay un proxy HTTPS entre el servidor y el sitio HTTPS en cuestión. Si las huellas digitales coinciden, no hay ningún proxy HTTPS.

a. Utilicen los siguientes tres comandos canalizados para obtener la huella digital correspondiente a Cisco.com. En la línea de abajo se utiliza OpenSSL para conectarse con cisco.com en el puerto 443 (HTTPS), procedemos a solicitar el certificado y almacenarlo en un archivo de texto de nombre **cisco.pem**. También se muestra la salida para ofrecer contexto.

```
[analyst@secOps ~]$ echo -n | openssl s_client -connect cisco.com:443 | sed
-ne '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p' > ./cisco.pem
depth=2 C = BM, O = QuoVadis Limited, CN = QuoVadis Root CA 2
verify return:1
depth=1 C = US, O = HydrantID (Avalanche Cloud Corporation), CN = HydrantID SSL ICA G2
verify return:1
depth=0 C = US, ST = CA, L = San Jose, O = "Cisco Systems, Inc.", CN = www.cisco.com
verify return:1
DONE
```

b. De manera opcional, utilicen el comando **cat** para generar una lista con el contenido del certificado obtenido y almacenarlo en el archivo de texto **cisco.pem**:

```
[analyst@secOps ~]$ cat cisco.pem
```

```
-----BEGIN CERTIFICATE-----
MIIG1zCCBL+gAwIBAgIUKBO9xTQoMemc9zFHNkdMW+SgFO4wDQYJKoZIhvcNAQEL
BQAwXjELMAkGA1UEBhMCVVMxMDAuBgNVBAoTJ0h5ZHJhbnRJRCAoQXZhbGFuY2hl
IENsb3VkIENvcnBvcmF0aW9uKTEdMBsGA1UEAxMUSHlkcmFudElEIFNTTCBJQ0Eg
RzIwHhcNMTcxMjA3MjIxODU1WhcNMTkxMjA3MjIyODAwWjBjMQswCQYDVQQGEwJV
UzELMAkGA1UECAwCQ0ExETAPBgNVBAcMCFNhbiBKb3NlMRwwGgYDVQQKDBNDaXNj
byBTeXN0ZW1zLCBJbmMuMRYwFAYDVQQDDA13d3cuY2lzY28uY29tMIIBIjANBgkq
yvo6dWpJdSircYy8HG0nz4+936+2waIVf1BBQXZUjNVuws74Z/eLIpl2c6tANmE0
q1i7fiWgItjDQ8rfjeX0oto6rvp8AXPjPY6X7PT1ulfhkLYnxqXHPETRwr8l5COO
MDEh95cRxATXNAlWAwLcBT7lDmrGron6rW6hDtuUPPG/rjZeZbNww5p/nT3EXX2L
Rh+m0R4j/tuvy/77YRWyp/VZhmSLrvZEYiVjM2MgCXBvqR+aQ9zWJkw+CAm5Z414
Eiv5RLctegYuBUMGTH1al9r5cuzfwEg2mNkxl4I/mtDro2kDAv7bcTm8T1LsZAO/
1bWvudsrTA8jksw+1WGAEd9bHi3ZpJPYedlL
-----END CERTIFICATE-----
[analyst@secOps ~]$
```

c. Ahora que el certificado está guardado en el archivo de texto **cisco.pem**, utilizar el siguiente comando para extraer la huella digital respectiva y mostrarla en la pantalla:

```
[analyst@secOps ~]$ openssl x509 -noout -in cisco.pem -fingerprint -sha1
SHA1 Huella digital = 64:19:CA:40:E2:1B:3F:92:29:21:A9:CE:60:7 D: C9:0 C: 39:B5:71:3E
[analyst@secOps ~]$
```

Nota: El valor de la huella digital puede ser diferente por dos motivos. En primer lugar, es posible que esté utilizando un sistema operativo diferente a la Máquina Virtual CyberOps Workstation. En segundo lugar, los certificados se actualizan con regularidad y cambian así el valor de la huella digital.

¿Qué algoritmo de hash utilizó OpenSSL para calcular la huella digital?

¿Por qué se eligió ese algoritmo específico? ¿Tiene alguna importancia?

Paso 3: Comparar las Huellas digitales

Procedamos a utilizar la Tabla 1 para comparar la huella digital del certificado que se obtuvo directamente desde el sitio HTTPS de Cisco con la que se obtuvo desde sus redes. Debemos recordar que las huellas digitales pueden cambiar con el tiempo.

¿Coinciden las huellas dactilares?

¿Qué significa eso?

¿Es este método 100 % infalible?

Parte 3: Desafíos (Opcionales)

a. Comprobar las huellas digitales correspondientes a los sitios que se ven en la Tabla 1, pero utilizando la GUI de nuestros propios navegadores web.

   Pistas: Busque una manera de mostrar la huella digital a través de la GUI del navegador. Debemos recordar que Google les resultará útil en este ejercicio, y Windows a menudo se refiere a la huella digital con el nombre de **Huella dactilar**.

b. Procedamos a utilizar OpenSSL (Parte 2, pasos 1 al 3) para comprobar todas las huellas digitales que están en la Tabla 1.

Pregunta de reflexión

¿Qué es necesario para que funcione el proxy HTTPS?