Pidiendo y instalando un certificado global en el CSS11500

Contenido

Introducción

Si usted no tiene las claves preexistentes y Certificados para el Content Services Switch (CSS), usted puede generarlos en el CSS. El CSS incluye una serie de utilidades de administración del certificado y de la clave privada para simplificar el proceso de generar las claves privadas, los pedidos de firma de certificado (CSR), y los Certificados temporales uno mismo-firmados. Este documento describe el proceso para obtener un nuevo certificado de un Certificate Authority (CA) y instalarlo al CSS.

prerrequisitos

Requisitos

No hay requisitos específicos para este documento.

Componentes Utilizados

Este documento no tiene restricciones específicas en cuanto a versiones de software y de hardware.

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 la red está funcionando, asegúrese de haber comprendido el impacto que puede tener cualquier comando.

Convenciones

Para obtener más información sobre las convenciones del documento, consulte Convenciones de Consejos Técnicos de Cisco.

Configurar

En esta sección encontrará la información para configurar las funciones descritas en este documento.

Nota: Para encontrar la información adicional en los comandos usados en este documento, use la Command Lookup Tool (clientes registrados solamente).

Configuraciones

En este documento, se utilizan estas configuraciones:

- Genere el Rivest, el Shamir, y el par clave del Adelman (RSA)
- Asocie el archivo del par clave RSA
- Genere el CSR
- Obtenga el certificado del intermedio de Verisign
- Importe el archivo de certificado encadenado
- Asocie el archivo de certificado
- Configure la lista del proxy SSL
- Configure el servicio y las reglas de contenido del Secure Socket Layer (SSL)

Genere el Rivest, el Shamir, y el par clave del Adelman (RSA)

Publique el **comando ssl genrsa** de generar un soldado/el par clave público RSA para la encriptación asimétrica. El CSS salva el par clave generado RSA como archivo en el CSS. Por ejemplo, para generar el par clave myrsakey.pem RSA, teclee el siguiente:

```
CSS11500(config) # ssl genrsa myrsakey.pem 1024 "passwd123" Please be patient this could take a
few minutes
```

Asociación del archivo del par clave RSA

Publique el **comando ssl associate rsakey** de asociar el nombre del par clave RSA al par clave generado RSA. Por ejemplo, para asociar el nombre de la clave myrsakey1 RSA al archivo generado myrsakey.pem del par clave RSA, teclee el siguiente:

```
CSS11500(config) # ssl associate rsakey myrsakey1 myrsakey.pem
```

Genere el CSR

Publique el **comando ssl gencsr rsakey** de generar un archivo CSR para un archivo asociado del par clave RSA. Este CSR será enviado a CA para firmar. Por ejemplo, para generar un CSR basado en el par clave myrsakey1 RSA, teclee el siguiente:

```
CSS11503(config)# ssl gencsr myrsakey1 You are about to be asked to enter information that will
be incorporated into your certificate request. What you are about to enter is what is called a
```

```
Distinguished Name or a DN. For some fields there will be a default value, If you enter '.', the
field will be left blank. Country Name (2 letter code) [US] US State or Province (full name)
[SomeState] CA Locality Name (city) [SomeCity] San Jose Organization Name (company name) [Acme
Inc]Cisco Systems, Inc. Organizational Unit Name (section) [Web Administration] Web Admin Common
Name (your domain name) [www.acme.com] www.cisco.com Email address [webadmin@acme.com]
webadmin@cisco.com
```

El comando ssl gencsr genera el CSR y lo hace salir a la pantalla. La mayoría de los CA importantes tienen aplicaciones basadas en Web que le requieran cortar y pegar el pedido de certificado a la pantalla.

```
-----BEGIN CERTIFICATE REQUEST-----
MIIBWDCCAQICAQAwgZwxCzAJBgNVBAYTAlVTMQswCQYDVQQIEwJNQTETMBEGA1UE
BxMKQm94Ym9yb3VnaDEcMBoGA1UEChMTQ2lzY28gU3lzdGVtcywgSW5jLjESMBAG
A1UECxMJV2ViIEFkbWluMRYwFAYDVQQDEw13d3cuY2lzY28uY29tMSEwHwYJKoZI
hvcNAQkBFhJra3JvZWJlckBjaXNjby5jb20wXDANBgkqhkiG9w0BAQEFAANLADBI
AkEAqHXjtQUVXvmo6tAWPiMpe6oYhZbJUDgTxbW4VMCygzGZn2wUJTgLrifDB6N3
v+1tKFndE686BhKqfyOidml3wQIDAQABoAAwDQYJKoZIhvcNAQEEBQADQQA94yC3
4SUJJ4UQEnO2OqRGLOZpAElc4+IV9aTWK6NmiZsM9Gt0vPhIkLx5jjhVRLlb27Ak
H6D5omXa0SPJan5x
-----END CERTIFICATE REQUEST-----
```

CA firma el CSR y le lo vuelve, típicamente usando la dirección de correo electrónico proporcionada dentro del CSR.

[Obtenga el certificado del intermedio de Verisign]

Obtenga el certificado de CA

Después de someter su CSR a CA, toma entre un y siete días hábiles para recibir un certificado firmado; los tiempos varían debido a CA. Una vez que CA ha firmado y ha entregado el certificado, puede ser agregado al CSS.

Si usted está solicitando un StepUp/SGC o un certificado encadenado, usted necesita obtener el certificado intermedio usado para firmar su certificado. Usted puede obtener el certificado intermedio de Verisign del siguiente enlace:

- Instalar el certificado de CA intermedio

Salve el certificado intermedio a un archivo. Por ejemplo, intermediate.pem.

Concatene el servidor y los Certificados intermedios

Para utilizar encadenó los Certificados en el CSS, el certificado de servidor y el intermedio se debe concatenar junto. Esto permite que el CSS vuelva la Cadena de certificados entera al cliente sobre el contacto SSL inicial. Al crear el archivo de certificado encadenado para el CSS, esté seguro que los Certificados están en la orden apropiada. El certificado de servidor debe ser primer, después el certificado intermedio usado para firmar el certificado de servidor debe ser siguiente. Debe haber un solo newline entre el servidor y los Certificados intermedios. Por ejemplo, concatene el certificado de servidor servercert.pem y el intermediate.pem en un certificado encadenado llamado mychainedrsacert.pem. Las visualizaciones siguientes que el contenido entero del mychainedrsacert.pem clasifía.

```
-----BEGIN CERTIFICATE-----
MIICwTCCAioCAQUwDQYJKoZIhvcNAQEEBQAwgagxCzAJBgNVBAYTAlVTMRMwEQYD
 VQQIEwpDYWxpZm9ybmlhMREwDwYDVQQHEwhTYW4gSm9zZTEeMBwGA1UEChMVRXhh
```

```
 bXBsZSBTeXN0ZW1zLCBJbmMuMRIwEAYDVQQLEwlXZWIgQWRtaW4xGDAWBgNVBAMT
 D3d3dy5leGFtcGxlLmNvbTEjMCEGCSqGSIb3DQEJARYUd2ViYWRtaW5AZXhhbXBs
 ZS5jb20wHhcNMDQwMTA5MDgzMjI3WhcNMDQwMjA4MDgzMjI3WjCBqDELMAkGA1UE
 BhMCVVMxEzARBgNVBAgTCkNhbGlmb3JuaWExETAPBgNVBAcTCFNhbiBKb3NlMR4w
 HAYDVQQKExVFeGFtcGxlIFN5c3RlbXMsIEluYy4xEjAQBgNVBAsTCVdlYiBBZG1p
 bjEYMBYGA1UEAxMPd3d3LmV4YW1wbGUuY29tMSMwIQYJKoZIhvcNAQkBFhR3ZWJh
 ZG1pbkBleGFtcGxlLmNvbTCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEA2huF
 xhVeODHmoXJ4HulDqVQtCvX7eERyRarNI71p0ZV+q+qGYRtJdrlzUav/TbRn5dc0
 8IXjqrASAtTo2S4eWlTOJUnR2g0LH/lcPUaF8f+m+eODWoT8dCtNA5sgEnINAR2y
 HlS5j6dZNcyMY0nFOh68oRsZJJ58u0ZPJjl6eAsCAwEAATANBgkqhkiG9w0BAQQF
 AAOBgQADO/UTIIHnIq2Q0ICiqAQju9nz1vTiIYHBpBnUd8NkPhIHIOqNn9iZ5Q+a
 2zFjh+N2uEt5NxNOEZRbrTZH+HmZMsqJJfvfd62iq+636aPIcoo7X541DYotM05C
 OQjnehsjgwziKlp6UJtuiAwwaxtMIbP7lQXHGO6E9RnzQSvQGQ==
 -----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
 MIIDgzCCAuygAwIBAgIQJUuKhThCzONY+MXdriJupDANBgkqhkiG9w0BAQUFADBf
 MQswCQYDVQQGEwJVUzEXMBUGA1UEChMOVmVyaVNpZ24sIEluYy4xNzA1BgNVBAsT
 LkNsYXNzIDMgUHVibGljIFByaW1hcnkgQ2VydGlmaWNhdGlvbiBBdXRob3JpdHkw
 HhcNOTcwNDE3MDAwMDAwWhcNMTExMDI0MjM1OTU5WjCBujEfMB0GA1UEChMWVmVy
 aVNpZ24gVHJ1c3QgTmV0d29yazEXMBUGA1UECxMOVmVyaVNpZ24sIEluYy4xMzAx
 BgNVBAsTKlZlcmlTaWduIEludGVybmF0aW9uYWwgU2VydmVyIENBIC0gQ2xhc3Mg
 MzFJMEcGA1UECxNAd3d3LnZlcmlzaWduLmNvbS9DUFMgSW5jb3JwLmJ5IFJlZi4g
 TElBQklMSVRZIExURC4oYyk5NyBWZXJpU2lnbjCBnzANBgkqhkiG9w0BAQEFAAOB
 jQAwgYkCgYEA2IKA6NYZAn0fhRg5JaJlK+G/1AXTvOY2O6rwTGxbtueqPHNFVbLx
 veqXQu2aNAoV1Klc9UAl3dkHwTKydWzEyruj/lYncUOqY/UwPpMo5frxCTvzt01O
 OfdcSVq4wR3Tsor+cDCVQsv+K1GLWjw6+SJPkLICp1OcTzTnqwSye28CAwEAAaOB
 4zCB4DAPBgNVHRMECDAGAQH/AgEAMEQGA1UdIAQ9MDswOQYLYIZIAYb4RQEHAQEw
 KjAoBggrBgEFBQcCARYcaHR0cHM6Ly93d3cudmVyaXNpZ24uY29tL0NQUzA0BgNV
 HSUELTArBggrBgEFBQcDAQYIKwYBBQUHAwIGCWCGSAGG+EIEAQYKYIZIAYb4RQEI
 ATALBgNVHQ8EBAMCAQYwEQYJYIZIAYb4QgEBBAQDAgEGMDEGA1UdHwQqMCgwJqAk
 oCKGIGh0dHA6Ly9jcmwudmVyaXNpZ24uY29tL3BjYTMuY3JsMA0GCSqGSIb3DQEB
 BQUAA4GBAAgB7ORolANC8XPxI6I63unx2sZUxCM+hurPajozq+qcBBQHNgYL+Yhv
 1RPuKSvD5HKNRO3RrCAJLeH24RkFOLA9D59/+J4C3IYChmFOJl9en5IeDCSk9dBw
 E88mw0M9SR2egi5SX7w+xmYpAY5Okiy8RnUDgqxz6dl+C2fvVFIa
 -----END CERTIFICATE-----
```

Archivo de certificado encadenado de la importación

Una vez que el CSR ha sido firmado por CA, ahora se llama un certificado. El archivo de certificado se debe importar al CSS. Publique el **comando copy ssl** de facilitar la importación o la exportación de los Certificados y de las claves privadas o al CSS. El CSS salva todos los archivos importados en una ubicación segura en el CSS. Este comando está disponible solamente en el modo de superusuario. Por ejemplo, para importar el certificado del mychainedrsacert.pem de un servidor remoto al CSS, teclee el siguiente:

```
CSS11500# copy ssl sftp ssl_record import mychainedrsacert.pem PEM “passwd123” Connecting
Completed successfully
```

Asocie el archivo de certificado

Publique el **comando ssl associate cert** de asociar un nombre del certificado al certificado importado. Por ejemplo, para asociar el nombre mychainedrsacert1 del certificado al mychainedrsacert.pem importado del archivo de certificado, teclee el siguiente:

```
CSS11500(config)# ssl associate cert mychainedrsacert1 mychainedrsacert.pem
```

Configure la lista del proxy SSL

Publique el **comando ssl-proxy-list** de crear una lista del proxy SSL. Una lista del proxy SSL es un

grupo de servidores SSL virtuales o backend relacionados que se asocien a un servicio SSL. La lista del proxy SSL contiene toda la información de la configuración para cada servidor SSL virtual. Esto incluye el par clave SSL de la creación de servidor SSL, de los Certificados y de la correspondencia, IP virtual el direccionamiento (VIP) y puerto, las cifras SSL soportadas, y otras opciones de SSL. Por ejemplo, para crear la lista SSL del proxy ssl_list1, teclee el siguiente:

```
CSS11500(config)# ssl-proxy-list ssl_list1 Create ssl-list <ssl_list1>, [y/n]: y
```

Una vez que usted crea una lista del proxy SSL, el CLI le ingresa en el modo de configuración de la lista SSL del proxy. Configure a su servidor SSL como se muestra abajo.

```
CSS11500(ssl-proxy-list[ssl_list1])# ssl-server 20 CSS11500(ssl-proxy-list[ssl_list1])# ssl-server 20 vip address 192.168.3.6 CSS11500(ssl-proxy-list[ssl_list1])# ssl-server 20 rsacert mychainedrsacert1 CSS11500(ssl-proxy-list[ssl_list1])# ssl-server 20 rsakey myrsakey1 CSS11500(ssl-proxy-list[ssl_list1])# ssl-server 20 cipher rsa-export-with-rc4-40-md5 192.168.11.2 80 5 CSS11500(ssl-proxy-list[ssl_list1])# active
```

[Configure el servicio y las reglas de contenido del Secure Socket Layer (SSL)]

Una vez que se activa la lista del proxy SSL, una necesidad del servicio y de la regla de contenido de ser configurado para permitir que el CSS envíe el tráfico SSL al módulo SSL. Esta tabla proporciona una descripción de los pasos requeridos para crear un servicio SSL para un servidor SSL virtual, incluyendo agregar la lista del proxy SSL al servicio y crear una regla de contenido SSL.

Cree un servicio SSL

```
CSS11500(config)# service ssl_serv1Create service <ssl_serv1>, [y/n]: y CSS11500(config-service[ssl_serv1])# type ssl-accel CSS11500(config-service[ssl_serv1])# slot 2 CSS11500(config-service[ssl_serv1])# keepalive type none CSS11500(config-service[ssl_serv1])# add ssl-proxy-list ssl_list1 CSS11500(config-service[ssl_serv1])# active
```

Cree una regla de contenido SSL

```
CSS11500(config)# owner ssl_owner Create owner <ssl_owner>, [y/n]: y CSS11500(config-owner[ssl_owner])# content ssl_rule1 Create content <ssl_rule1>, [y/n]: y CSS11500(config-owner-content[ssl-rule1]# vip address 192.168.3.6 CSS11500(config-owner-content[ssl-rule1]# port 443 CSS11500(config-owner-content[ssl_rule1])# add service ssl_serv1 CSS11500(config-owner-content[ssl_rule1])# active
```

Cree una regla de contenido del texto claro

```
CSS11500(config-owner[ssl_owner])# content decrypted_www Create content <decrypted_www>, [y/n]: y CSS11500(config-owner-content[decrypted_www]# vip address 192.168.11.2 CSS11500(config-owner-content[decrypted_www]# port 80 CSS11500(config-owner-content[decrypted_www])# add service linux_http CSS11500(config-owner-content[decrypted_www])# add service win2k_http CSS11500(config-owner-content[decrypted_www])# active
```

En este momento, el tráfico del cliente HTTPS se puede enviar al CSS en 192.168.3.6:443. El CSS desencripta el tráfico HTTPS, convirtiéndolo al HTTP. El CSS después elige un servicio y envía el tráfico HTTP a un servidor Web HTTP. Lo que sigue es una configuración CSS de trabajo usando los ejemplos anteriores:

```
CSS11501# show run configure !************************** GLOBAL ************************** ssl associate rsakey myrsakey1 myrsakey.pem ssl associate cert mychainedrsacert1 mychainedrsacert.pem ip route 0.0.0.0 0.0.0.0 192.168.3.1 1 ftp-record conf 192.168.11.101 admin des-password 4f2bxansrcehjgka /tftpboot !************************ INTERFACE
```

```
************************ interface 1/1 bridge vlan 10 description "Client Side" interface 1/2
bridge vlan 20 description "Server Side" !************************* CIRCUIT
************************* circuit VLAN10 description "Client Segment" ip address 192.168.3.254
255.255.255.0 circuit VLAN20 description "Server Segment" ip address 192.168.11.1 255.255.255.0
!********************** SSL PROXY LIST ********************** ssl-proxy-list ssl_list1 ssl-
server 20 ssl-server 20 vip address 192.168.3.6 ssl-server 20 rsakey myrsakey1 ssl-server 20
rsacert mycertcert1 ssl-server 20 cipher rsa-with-rc4-128-md5 192.168.11.2 80 active
!************************* SERVICE ************************* service linux-http ip address
192.168.11.101 port 80 active service win2k-http ip address 192.168.11.102 port 80 active
service ssl_serv1 type ssl-accel slot 2 keepalive type none add ssl-proxy-list ssl_list1 active
!************************** OWNER ************************** owner ssl_owner content ssl_rule1
vip address 192.168.3.6 protocol tcp port 443 add service ssl_serv1 active content decrypted_www
vip address 192.168.11.2 add service linux-http add service win2k-http protocol tcp port 80
active
```

Verificación

Use esta sección para confirmar que su configuración funciona correctamente.

Utilice los **comandos show ssl file y show ssl associate** de verificar la configuración.

Verifique que todos los archivos tengan un tamaño más grande de 0.

Usted puede quitar cualquier certificado o clave usando el **comando clear ssl file**.

Troubleshooting

Use esta sección para resolver problemas de configuración.

Si la negociación SSL falla, utilice el **comando show ssl statistics** de ver la información útil sobre la negociación SSL fallada.

Por ejemplo, marque estos campos:

```
0 Unknown issuer certificates
0 Failed signatures decryptions
0 Invalid issuer keys
0 Not yet valid certificates
0 Expired Client certificates
0 Revoked certificates
0 CRLs not obtained from host
0 CRLs with bad HTTP return codes
0 CRLs not loaded because of low memory
0 CRLs obtained but failed to load
0 CRLs with invalid signatures
0 CRLs successfully loaded
0 Successful server authentications
0 Server authentications failed
0 Expired Server certificates
```

Información Relacionada

- Soporte del hardware de los CSS 11500 Series Content Services Switch
- Soporte del hardware de los CSS 11000 Series Content Services Switch
- Descarga del software de Cisco WebNS CSS11500 (clientes registrados solamente)

- Descarga del software de Cisco WebNS CSS11000 (clientes registrados solamente)
- Soporte Técnico y Documentación - Cisco Systems