Demande et installation d'un certificat global sur CSS11500

Contenu

Introduction

Si vous n'avez pas des clés préexistantes et des Certificats pour le Commutateur de services de contenu (CSS), vous pouvez les générer sur le CSS. Le CSS inclut une gamme de certificat et d'utilitaires privés de gestion des clés pour simplifier le processus de générer des clés privées, des demandes de signature de certificat (CSR), et des Certificats provisoires auto-signés. Ce document décrit le processus pour obtenir un nouveau certificat d'un Autorité de certification (CA) et l'installer sur le CSS.

Conditions préalables

Conditions requises

Aucune spécification déterminée n'est requise pour ce document.

Composants utilisés

Ce document n'est pas limité à des versions de matériel et de logiciel spécifiques.

Les informations contenues dans ce document ont été créées à partir des périphériques d'un environnement de laboratoire spécifique. Tous les périphériques utilisés dans ce document ont démarré avec une configuration effacée (par défaut). Si votre réseau est opérationnel, assurez-vous que vous comprenez l'effet potentiel de toute commande.

Conventions

Pour plus d'informations sur les conventions des documents, référez-vous aux Conventions utilisées pour les conseils techniques de Cisco.

Configurez

Cette section vous fournit des informations pour configurer les fonctionnalités décrites dans ce document.

Remarque: Pour obtenir plus d'informations sur les commandes utilisées dans ce document, utilisez l'Outil de recherche de commande (clients enregistrés uniquement).

Configurations

Ce document utilise les configurations suivantes :

- Générez Rivest, Shamir, et paire de clés d'Adelman (RSA)
- Associez le fichier de paire de clés RSA
- Générez le CSR
- Obtenez le certificat intermédiaire de Verisign
- Fichier du certificat enchaîné d'importation
- Associez le fichier du certificat
- Configurez la liste de proxy SSL
- Configurez le service et les règles de contenu de Protocole SSL (Secure Socket Layer)

Générez Rivest, Shamir, et paire de clés d'Adelman (RSA)

Émettez la commande de **genrsa SSL** de générer paire de clés privée/publique RSA pour le cryptage asymétrique. Le CSS enregistre la paire de clés RSA générée comme fichier sur le CSS. Par exemple, pour générer la paire de clés RSA myrsakey.pem, tapez ce qui suit :

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

Associer le fichier de paire de clés RSA

Émettez la commande de **rsakey d'associé SSL** d'associer le nom de paire de clés RSA à la paire de clés RSA générée. Par exemple, pour associer le nom de clé RSA myrsakey1 au fichier généré myrsakey.pem de paire de clés RSA, tapez ce qui suit :

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

Générez le CSR

Émettez la commande de **rsakey de gencsr SSL** de générer un fichier CSR pour un fichier associé de paire de clés RSA. Ce CSR sera envoyé au CA pour la signature. Par exemple, pour générer un CSR basé sur la paire de clés RSA myrsakey1, tapez ce qui suit :

```
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
```

La commande de **gencsr SSL** génère le CSR et le sort à l'écran. La plupart de commandant CAs ont des applications Web qui exigent de vous de couper-coller la demande de certificat à l'écran.

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

Le CA signe le CSR et le renvoie te, typiquement utilisant l'adresse e-mail fournie dans le CSR.

[Obtenez le certificat intermédiaire de Verisign]

Obtenez le certificat d'un CA

Après la soumission de votre CSR à un CA, il prend entre un et sept Business Day pour recevoir un certificat signé ; les temps varient en raison du CA. Une fois que le CA a signé et a fourni le certificat, il peut ajouter au CSS.

Si vous sollicitez un StepUp/SGC ou un certificat enchaîné, vous devez obtenir le certificat intermédiaire utilisé pour signer votre certificat. Vous pouvez obtenir le certificat intermédiaire de Verisign du lien suivant :

- Installer le certificat de CA intermédiaire

Sauvegardez le certificat intermédiaire à un fichier. Par exemple, intermediate.pem.

Concaténez les Certificats de serveur et d'intermédiaire

Afin d'utiliser a enchaîné des Certificats sur le CSS, le certificat de serveur et l'intermédiaire doit être concaténée ensemble. Ceci permet au CSS pour renvoyer la chaîne de certificat entière au client sur la prise de contact SSL d'initiale. En créant le fichier du certificat enchaîné pour le CSS, soyez sûr que les Certificats sont dans la commande appropriée. Le certificat de serveur doit être premier, puis le certificat intermédiaire utilisé pour signer le certificat de serveur doit être prochain. Il doit y a un saut de ligne simple entre le serveur et les Certificats d'intermédiaire. Par exemple, concaténez le certificat de serveur servercert.pem et l'intermediate.pem dans un certificat enchaîné appelé le mychainedrsacert.pem. Les affichages suivants que le contenu entier du mychainedrsacert.pem classe.

```
-----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-----
```

[Fichier du certificat enchaîné d'importation]

Une fois que le CSR a été signé par un CA, ce s'appelle maintenant un certificat. Le fichier du certificat doit être importé au CSS. Émettez la commande **SSL de copie** de faciliter l'importation ou l'exportation des Certificats et des clés privées ou derrière le CSS. Le CSS enregistre tous les fichiers importés dans un emplacement sécurisé sur le CSS. Cette commande est disponible seulement dans le mode de super utilisateur. Par exemple, pour importer le certificat mychainedrsacert.pem d'un serveur distant au CSS, tapez ce qui suit :

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

[Associez le fichier du certificat]

Émettez la commande de **CERT d'associé SSL** d'associer un nom de certificat au certificat importé. Par exemple, pour associer le nom mychainedrsacert1 de certificat au fichier du certificat importé mychainedrsacert.pem, tapez ce qui suit :

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

[Configurez la liste de proxy SSL]

Émettez la commande de SSL-proxy-**liste** de créer une liste de proxy SSL. Une liste de proxy SSL est un groupe de serveurs virtuels ou principaux relatifs SSL qui sont associés avec un service

SSL. La liste de proxy SSL contient toutes les informations de configuration pour chaque serveur virtuel SSL. Ceci inclut la création de serveur SSL, paire de clés SSL de Certificats et de correspondance, adresse virtuelle et port IP (VIP), chiffrements SSL pris en charge, et d'autres options SSL. Par exemple, pour créer la SSL-proxy-liste ssl_list1, tapez ce qui suit :

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

Une fois que vous créez une liste de proxy SSL, le CLI vous présente dans le mode de configuration de SSL-proxy-liste. Configurez votre serveur SSL comme affiché ci-dessous.

```
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
```

Configurez le service et les règles de contenu de Protocole SSL (Secure Socket Layer)

Une fois que la liste de proxy SSL est lancée, un besoin de service et de règle de contenu d'être configuré pour permettre au CSS pour envoyer le trafic SSL au module SSL. Cette table fournit un aperçu de l'étape nécessaire pour créer un service SSL pour un serveur virtuel SSL, y compris ajouter la liste de proxy SSL au service et créer une règle de contenu SSL.

Créez un service 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
```

Créez une règle de contenu 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
```

Créez une règle de contenu des textes clairs

```
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 ce moment, le trafic du client HTTPS peut être envoyé au CSS à 192.168.3.6:443. Le CSS déchiffre le trafic HTTPS, le convertissant en HTTP. Le CSS alors choisit un service et envoie le trafic http à un serveur Web de HTTP. Ce qui suit est une configuration fonctionnante CSS utilisant les exemples ci-dessus :

```
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
```

Vérifiez

Référez-vous à cette section pour vous assurer du bon fonctionnement de votre configuration.

Utilisez les commandes de **fichier de show ssl** et d'**associé de show ssl** de vérifier la configuration.

Vérifiez que tous les fichiers ont une taille plus grande que 0.

Vous pouvez retirer n'importe quel certificat ou clé à l'aide de la commande **claire de fichier SSL**.

Dépannez

Utilisez cette section pour dépanner votre configuration.

Si la négociation SSL échoue, utilisez la commande de **statistiques de show ssl** de visualiser les informations utiles au sujet de la négociation défectueuse SSL.

Par exemple, vérifiez ces champs :

```
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
```

Informations connexes

- Support matériel pour les commutateurs de services de contenu de la gamme CSS 11500
- Support matériel de Commutateurs de services satisfaits de gamme 11000 CSS
- Téléchargement logiciel de Cisco WebNS CSS11500 (clients enregistrés seulement)
- Téléchargement logiciel de Cisco WebNS CSS11000 (clients enregistrés seulement)
- Support et documentation techniques - Cisco Systems