Descripción del protocolo simple certificate enrollment

Contenido

Introducción

Este documento describe el protocolo simple certificate enrollment (SCEP), que es un protocolo usado para la inscripción y otras operaciones del Public Key Infrastructure (PKI).

Antecedentes

El SCEP fue desarrollado originalmente por Cisco, y se documenta en un proyecto de la Fuerza

de tareas de ingeniería en Internet (IETF) (IETF).

Sus características principales son:

- Modelo de la petición/de la respuesta basado en HTTP (método GET; soporte opcional para el método del POSTE)
- Solamente los soportes RSA-basaron la criptografía
- Aplicaciones PKCS-10 como el formato del pedido de certificado
- Las aplicaciones PKCS-7 para transportar criptográficamente firmaron/los mensajes encriptados
- Soporta la concesión asíncrona por el servidor, con la interrogación regular del solicitante
- Ha limitado el soporte de la extracción del Listas de revocación de certificados (CRL) (el método preferido está con una interrogación del CRL Distribution Point (CDP), por los motivos de escalabilidad)
- No soporta la revocación de certificado en línea (debe ser hecho off-liné a través de los otros medios)
- Requiere el uso de un campo de **contraseña de impugnación** dentro del pedido de firma de certificado (CSR), que se debe compartir solamente entre el servidor y el solicitante

La inscripción y el uso del SCEP sigue generalmente este flujo de trabajo:

1. Obtenga una copia del certificado del Certificate Authority (CA) y valídela.
2. Genere un CSR y envíelo con seguridad a CA.
3. Sondee el servidor SCEP para marcar si el certificado fue firmado.
4. Re-aliste cuanto sea necesario para obtener un nuevo certificado antes de la expiración del certificado actual.
5. Extraiga el CRL cuanto sea necesario.

Autenticación de CA

El SCEP utiliza el certificado de CA para asegurar el intercambio del mensaje para el CSR. Como consecuencia, es necesario obtener una copia del certificado de CA. Se utiliza la operación de **GetCACert**.

Petición

La petición se envía como petición get HTTP. Una captura de paquetes para la petición parece similar a esto:

```
GET /cgi-bin/pkiclient.exe?operation=GetCACert
```

Respuesta

La respuesta es simplemente el certificado de CA binario-codificado (X.509). El cliente necesita validar que el certificado de CA esté confiado en a través de un examen de la huella dactilar/del hash. Esto tiene que ser hecha vía un método fuera de banda (llamadas telefónicas a un administrador de sistema o PRE-configuración de la huella dactilar dentro del trustpoint).

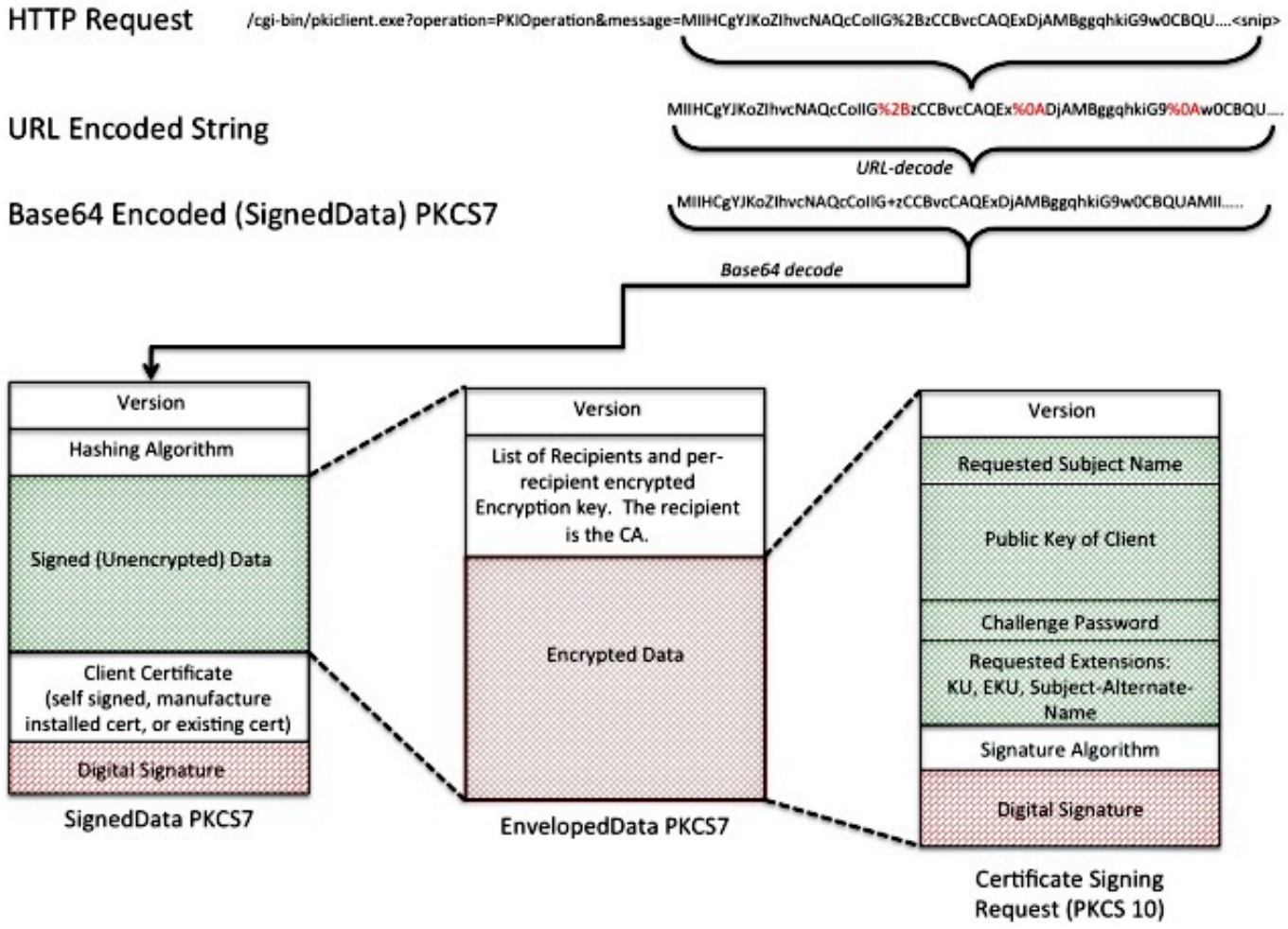
Inscripción del cliente

Petición

La petición de la inscripción se envía como petición get HTTP. Una captura de paquetes para la petición parece similar a esto:

```
/cgi-bin/pkiclient.exe?operation=PKIOperation&message=
MIIHCgYJKoZIhvcNAQcCoIIG%2BzCCBvcCAQExDjA.....<snip>
```

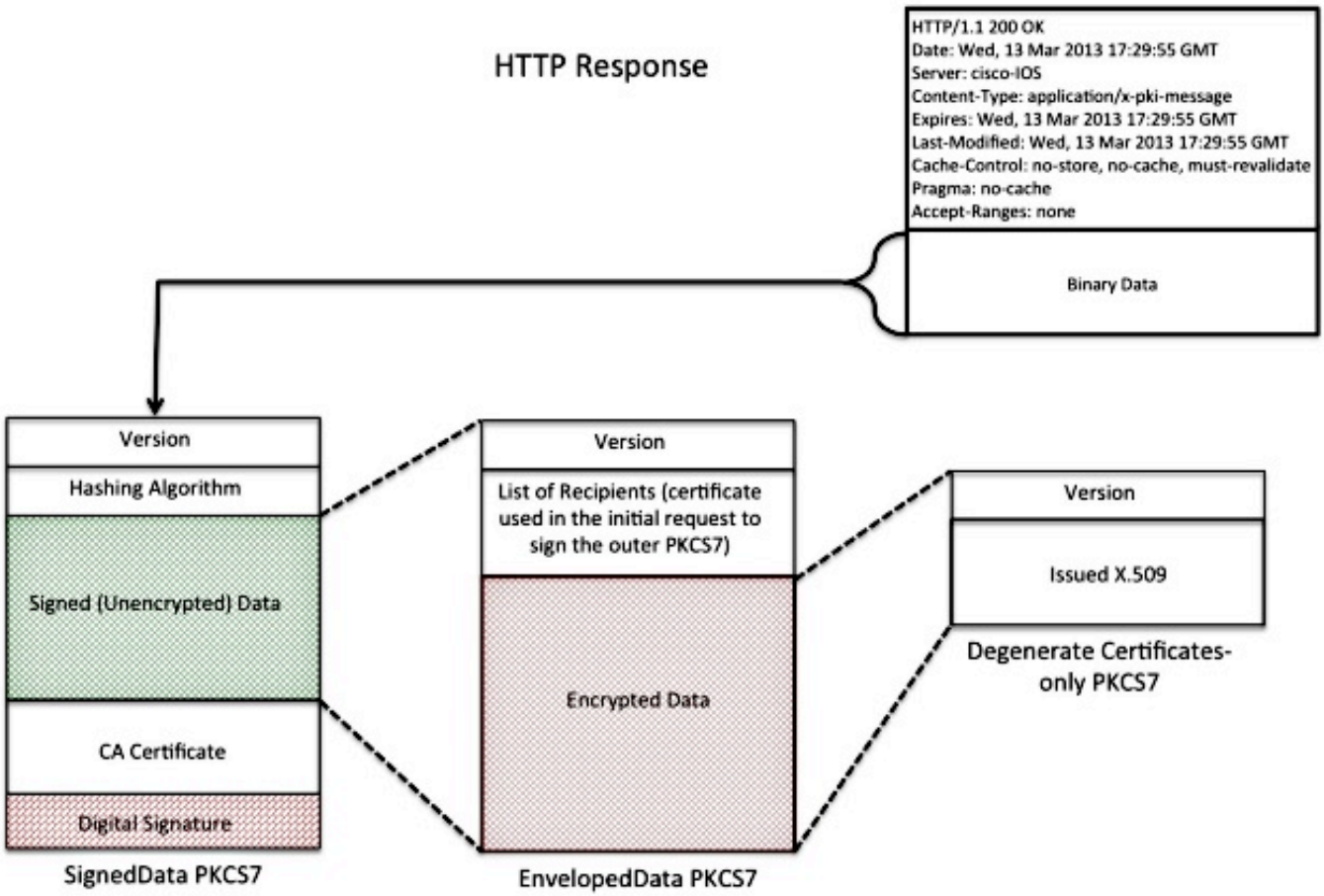
1. El texto después de que el “message=” sea una cadena codificada URL, que se extrae de la cadena de la petición get.
2. El texto es entonces URL decodificado en una cadena de texto ASCII. Esa cadena de texto es un SignedData codificado en base64 PKCS-7.
3. El SignedData PKCS-7 es firmado por el cliente con uno de estos Certificados; se utiliza para probar que el cliente lo envió y que no se ha alterado adentro transite:
   Un certificado autofirmado (usado sobre la inscripción inicial)Un certificado instalado fabricante (MIC)Una certificación actual que expira pronto (reinscripción)
4. La porción de los “datos firmados” del SignedData PKCS-7 es un EnvelopedData PKCS-7.
5. El EnvelopedData PKCS-7 es un envase que contiene los “datos encriptados” y la “clave del desciframiento.” La clave del desciframiento se cifra con la clave pública del beneficiario. En este caso específico, el beneficiario es CA; como consecuencia. Solamente CA puede desencriptar realmente los “datos encriptados.”
6. La porción de los “datos encriptados” de envuelta PKCS-7 es el CSR (PKCS-10).



Respuesta

La respuesta a la petición de la inscripción SCEP es uno de tres tipos:

- **Rechazo** - La petición es rechazada por el administrador para cualquier número de razones, por ejemplo:
  Tamaño de clave inválidoContraseña de impugnación inválidaCA no podía validar la peticiónLa petición pidió los atributos que CA no autorizóLa petición fue firmada por una identidad que CA no confía en
- **Pendiente** - El administrador de CA no ha revisado la petición todavía.
- **Éxito** - Se valida la petición y el certificado firmado es incluido. El certificado firmado se sostiene dentro de un tipo especial PKCS-7 llamado de los “Certificados-Solamente degenerados el PKCS#7," que sea un envase especial que puede sostener uno o más el X.509 o los CRL, pero no contenga un payload firmada o de los datos encriptados.



Reinscripción del cliente

Antes del vencimiento del certificado, el cliente necesita conseguir un nuevo certificado. Hay una diferencia del comportamiento leve entre la renovación y la renovación. La renovación sucede cuando el certificado ID del cliente se acerca a la expiración, y su fecha de vencimiento no es lo mismo (anterior que) que la fecha de vencimiento del certificado de CA. La renovación sucede cuando el certificado ID se acerca a la expiración, y su fecha de vencimiento es lo mismo que la fecha del vencimiento del certificado de CA.

Renovación

Pues la fecha de vencimiento de un certificado ID se acerca, un cliente SCEP pudo querer obtener un nuevo certificado. El cliente genera un CSR y pasa con el proceso de la inscripción (según lo definido previamente). El certificado actual se utiliza para firmar el SignedData PKCS-7, que a su vez prueba la identidad al CA tras el recibo del nuevo certificado, el cliente borra inmediatamente el certificado actual y lo substituye por el nuevo, cuyo comienzo de validez inmediatamente.

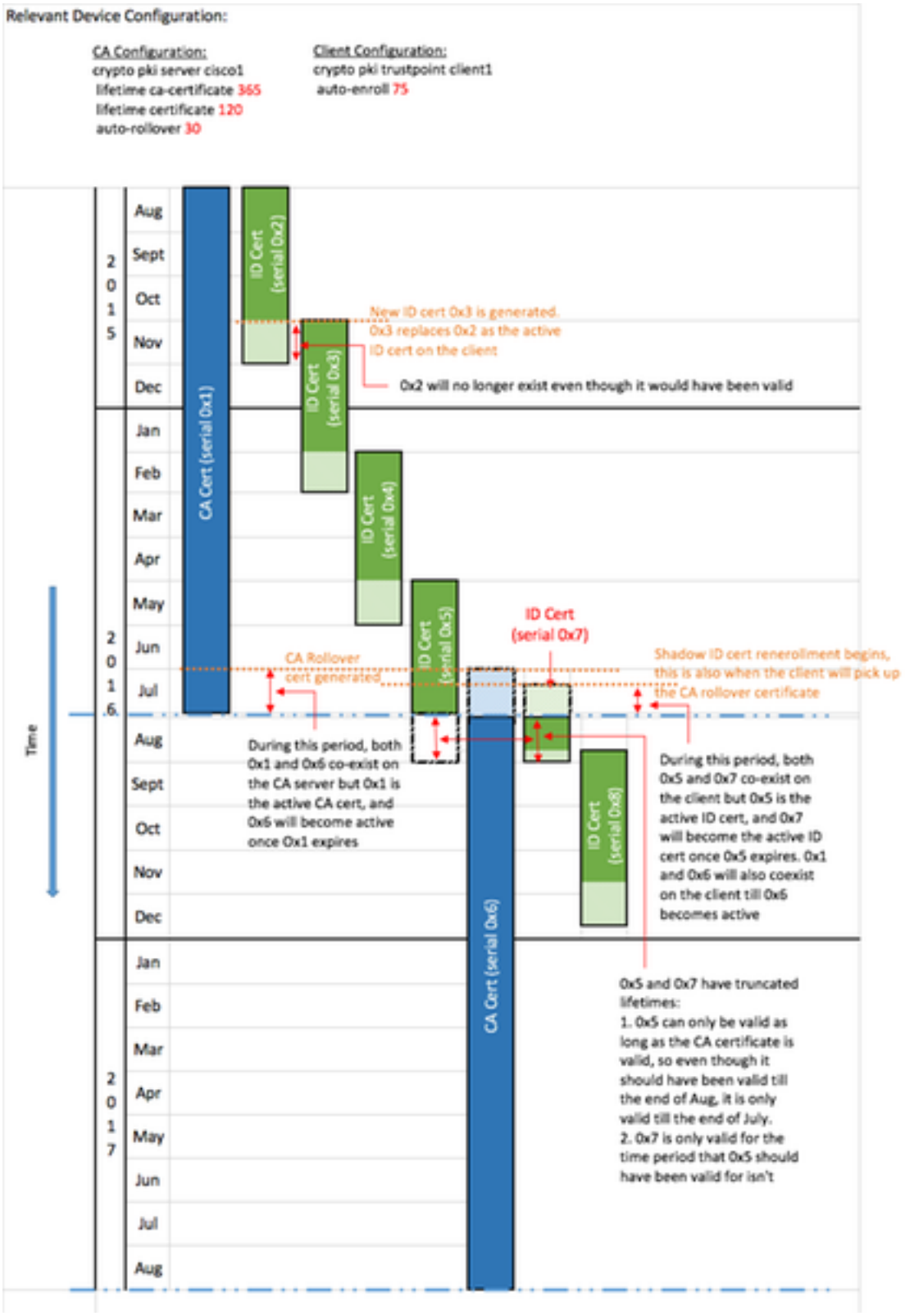
Renovación

La renovación es un caso especial donde expira el certificado de CA y se genera un nuevo certificado de CA. CA genera un nuevo certificado de CA que llega a ser válido el certificado de CA actual expira una vez. CA genera generalmente este “certificado de CA de la sombra” una cierta hora antes del tiempo de la renovación, porque es necesario para generar los Certificados de la “sombra ID” para los clientes.

Cuando el certificado ID del cliente SCEP se acerca a la expiración, las consultas del cliente SCEP CA para “el certificado de CA de la sombra”. Esto se hace con la operación de **GetNextCACert** como se muestra aquí:

```
GET /cgi-bin/pkiclient.exe?operation=GetNextCACert
```

Una vez que el cliente SCEP tiene “el certificado de CA de la sombra”, pide un certificado de la “sombra ID” después del procedimiento normal de la inscripción. CA firma el certificado de la “sombra ID” con “el certificado de CA de la sombra”. A diferencia de un pedido de renovación normal, el certificado de la “sombra ID” se devuelve que llega a ser válido a la hora de la expiración del certificado de CA (renovación). Como consecuencia, el cliente necesita guardar una copia del PRE y de los Certificados de la poste-renovación para CA y el certificado ID. A la hora de la expiración de CA (renovación), el cliente SCEP borra el certificado de CA actual y el certificado ID y los substituye por las copias de la “sombra”.



Bloques de construcción

Esta estructura se utiliza como los bloques de construcción de SCEP.

Nota: PKCS-7 y PKCS-10 no sea SCEP-específico.

PKCS-7

PKCS-7 está un formato de datos definido que permite que cifrans los datos sean firmados o. El formato de datos incluye las informaciones originales y los meta datos asociados necesarios para realizar la operación criptográfica.

Sobre firmado (SignedData)

El sobre firmado es un formato que lleva los datos y confirma que los datos encapsulados no están alterados adentro transitan vía las firmas digitales. Incluye esta información:

```
SignedData &colon;:= SEQUENCE {
version CMSVersion,
digestAlgorithms DigestAlgorithmIdentifiers,
encapContentInfo EncapsulatedContentInfo,
certificates [0] IMPLICIT CertificateSet OPTIONAL,
crls [1] IMPLICIT RevocationInfoChoices OPTIONAL,
signerInfos SignerInfos }
```

- Número de la versión - Con el SCEP, versión 1 usada.
- Lista de algoritmos de la publicación usados - Con el SCEP, hay solamente un firmante y así solamente un algoritmo de troceo.
- Datos reales se firman que - Con el SCEP, esto es PKCS-7 un formato Envolver-DATA (sobre cifrado).
- Lista de Certificados de los firmantes - Con el SCEP, éste es un certificado autofirmado en la inscripción inicial o el certificado actual si usted re-alista.
- Lista de los firmantes y de la huella dactilar generados por cada firmante - con el SCEP, hay solamente un firmante.

Los datos encapsulados no se cifran ni se ofuscan. Este formato proporciona simplemente la protección contra el mensaje se altera que.

Datos envueltos (EnvelopedData)

El formato de datos envuelto lleva los datos que se cifran y se pueden desencriptar solamente por el receptor especificado. Incluye esta información:

```
EnvelopedData &colon;:= SEQUENCE {
version CMSVersion,
originatorInfo [0] IMPLICIT OriginatorInfo OPTIONAL,
recipientInfos RecipientInfos,
encryptedContentInfo EncryptedContentInfo,
unprotectedAttrs [1] IMPLICIT UnprotectedAttributes OPTIONAL }
```

- Número de la versión - Con el SCEP, se utiliza la versión 0.
- Lista de cada uno de los beneficiarios y del Data Encryption Key cifrado relacionado - con el SCEP, hay solamente un beneficiario (para las peticiones: el servidor de CA; para las respuestas: el cliente).
- Los datos encriptados - Esto se cifra con una clave aleatoriamente generada (que se ha cifrado con la clave pública del beneficiario).

PKCS-10

PKCS-10 describe el formato de un CSR. Un CSR contiene la información que los clientes piden sean incluidos dentro de sus Certificados:

- Asunto

- Una copia de la clave pública
- Una contraseña de impugnación (opcional)
- Cualquier Extensiones del certificado reqested, por ejemplo:
  Uso dominante (KU)Uso dominante extendido (EKU)Nombre alternativo sujeto (SAN)Nombre principal universal (UPN)
- Una huella dactilar de la petición

Aquí está un ejemplo de un CSR:

```
Certificate Request:
Data&colon;
Version: 0 (0x0)
Subject: CN=scepclient
Subject Public Key Info:

Public Key Algorithm: rsaEncryption Public-Key: (1024 bit)
Modulus:
00:cd:46:5b:e2:13:f9:bf:14:11:25:6d:ff:2f:43:
64:75:89:77:f6:8a:98:46:97:13:ca:50:83:bb:10:
cf:73:a4:bc:c1:b0:4b:5c:8b:58:25:38:d1:19:00:
a2:35:73:ef:9e:30:72:27:02:b1:64:41:f8:f6:94:
7b:90:c4:04:28:a1:02:c2:20:a2:14:da:b6:42:6f:
e6:cb:bb:33:c4:a3:64:de:4b:3a:7d:4c:a0:d4:e1:
b8:d8:71:cc:c7:59:89:88:43:24:f1:a4:56:66:3f:
10:25:41:69:af:e0:e2:b8:c8:a4:22:89:55:e1:cb:
00:95:31:3f:af:51:3f:53:ad
Exponent: 65537 (0x10001)
Attributes:
challengePassword :
Requested Extensions:
X509v3 Key Usage: critical
Digital Signature, Key Encipherment
X509v3 Subject Alternative Name:
DNS:webserver.example.com
Signature Algorithm: sha1WithRSAEncryption
8c:d6:4c:52:4e:c0:d0:28:ca:cf:dc:c1:67:93:aa:4a:93:d0:
d1:92:d9:66:d0:99:f5:ad:b4:79:a5:da:2d:6a:f0:39:63:8f:
e4:02:b9:bb:39:9d:a0:7a:6e:77:bf:d2:49:22:08:e2:dc:67:
ea:59:45:8f:77:45:60:62:67:64:1d:fe:c7:d6:a0:c3:06:85:
e8:f8:11:54:c5:94:9e:fd:42:69:be:e6:73:40:dc:11:a5:9a:
f5:18:a0:47:33:65:22:d3:45:9f:f0:fd:1d:f4:6f:38:75:c7:
a6:8b:3a:33:07:09:12:f3:f1:af:ba:b7:cf:a6:af:67:cf:47: 60:fc
```

Información Relacionada

- Borrador IETF SCEP
- Herencia SCEP usando la guía de configuración CLI
- Configurar el soporte SCEP para BYOD

Apéndice

Peticiones SCEP

Formato del mensaje request

Las peticiones se envían con un HTTP GET de la forma:

```
GET CGI-path/pkiclient.exe?operation=operation&message=message HTTP/version
```
Donde:

- **El trayecto CGI** es dependiente en el servidor y las puntas a la interfaz de gateway común (CGI) programan que las manijas SCEP piden: El ® CA del Cisco IOS utiliza una cadena vacía de la trayectoria.Microsoft CA utiliza **/certsrv/mscep/mscep.dll**, que señala al servicio del servicio de la inscripción del dispositivo de red MSCEP/(NDE) IIS.
- **La operación** identifica la operación se realiza que.
- **El mensaje** lleva los datos adicionales para esa operación (y ella puede estar vacío si no se requiere ningunos datos reales).

Con el método GET, la pieza de **mensaje** es sólo texto, o las reglas distinguidas de la codificación (DER) - codificado PKCS-7 convertido al base64. Si se soporta el método del POSTE, contente que sería enviado en el base64 que codifica con el GET se pudo enviar en el formato binario con el POSTE en lugar de otro.

Visión esquemática

Valores posibles para las **operaciones** y sus valores asociados del **mensaje**:

- **operación** = `PKIOperation`: **messageis una** estructura del **pkiMessage** SCEP, sobre la base de PKCS-7 y codificado con el DER y el base64.la estructura del **pkiMessage** puede ser de estos tipos: **PKCSReq**: PKCS-10 CSR**GetCertInitial**: el sondear para el CSR que concede el estatus**GetCert** o **GetCRL**: certificado o extracción CRL
- **operación** = **GetCACert**, **GetNextCACert**, o **GetCACaps** (opcional): **el mensaje** se puede omitir, o se puede fijar a un nombre que identifique CA.

Respuestas SCEP

Formato del mensaje de respuesta

Las respuestas SCEP se vuelven como contenido estándar HTTP, con un **tipo de contenido** que dependa de la solicitud original y del tipo de datos devueltos. El contenido DER se vuelve como binario (no en el base64 en cuanto a la petición). PKCS-7 el contenido pudo o no pudo contener los datos envueltos cifrados/firmados; si no lo hace (contiene solamente un conjunto de los Certificados), se refiere como **degenerado** PKCS-7.

Tipos de contenido

Valores posibles para el **tipo de contenido**:

application/x-pki-message:

- en respuesta a la operación de **PKIOperation**, con el **pkiMessage del** tipo: **PKCSReq, GetCertInitial, GetCert o GetCRL**
- el cuerpo de la respuesta es un **pkiMessage del** tipo: **CertRep**

application/x-x509-ca-cert:

- en respuesta a la operación de **GetCACert**
- el cuerpo de la respuesta es el certificado de CA DER-codificado X.509

application/x-x509-ca-ra-cert:

- en respuesta a la operación de **GetCACert**
- el cuerpo de la respuesta es un degenerado DER-codificado PKCS-7 que contiene los Certificados de CA y RA

application/x-x509-next-ca-cert:

- en respuesta a la operación de **GetNextCACert**
- el cuerpo de la respuesta es una variación de un **pkiMessage del** tipo: **CertRep**

La estructura del pkiMessage

SCEP OID

```
GET CGI-path/pkiclient.exe?operation=operation&message=message HTTP/version
```

PkiMessage SCEP

- PKCS-7 **SignedData**
- PKCS-7 EnvelopedData (llamado **pkcsPKIEnvelope**; opcional, cifrado al receptor del mensaje)
  messageData (CSR, CERT, CRL,…)
- **SignerInfo** con los **authenticatedAttributes**:
  transactionID, **messageType**, **senderNoncepkiStatus**, **recipientNonce** (respuesta solamente)**failInfo** (respuesta + error solamente)

MessageType SCEP

- petición:
  PKCSReq (19): PKCS-10 CSR**GetCertInitial** (20): interrogación de la inscripción del certificado**GetCert** (21): recuperación de certificados**GetCRL** (22): Extracción CRL
- respuesta:
  CertRep (3): respuesta a certificar o petición CRL

PkiStatus SCEP

- **ÉXITO** (0): petición concedida (respuesta en el pkcsPKIEnvelope)
- **ERROR** (2): petición rechazada (detalles en el atributo del failInfo)
- **PENDIENTE** (3): la petición aguarda la aprobación manual