Descripción General de Simple Certificate Enrollment Protocol

Contenido

Introducción

Este documento describe el Protocolo simple de inscripción de certificados (SCEP), que es un protocolo utilizado para las operaciones de inscripción y otras infraestructuras de clave pública (PKI).

Antecedentes

SCEP fue desarrollado originalmente por Cisco y está documentado en un borrador de IETF (Grupo de Trabajo de Ingeniería de Internet).

Sus principales características son:

- Modelo de solicitud/respuesta basado en HTTP (método GET; soporte opcional para el método POST)
- Solo admite criptografía basada en RSA
- Utiliza PKCS#10 como formato de solicitud de certificado
- Utiliza PKCS#7 para transmitir mensajes cifrados/firmados criptográficamente
- Admite la concesión asíncrona por parte del servidor, con sondeo regular por parte del solicitante
- Tiene compatibilidad limitada con la recuperación de la lista de revocación de certificados (CRL) (el método preferido es a través de una consulta de punto de distribución (CDP) de CRL por razones de escalabilidad).
- No admite la revocación de certificados en línea (se debe realizar sin conexión por otros medios)
- Requiere el uso de un campo de **contraseña de impugnación** en la Solicitud de firma de certificados (CSR), que sólo debe compartirse entre el servidor y el solicitante

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

1. Obtenga una copia del certificado de la autoridad certificadora (CA) y valide el certificado.
2. Genere un CSR y envíelo de forma segura a la CA.
3. Sondee el servidor SCEP para verificar si el certificado fue firmado.
4. Vuelva a inscribirse según sea necesario para obtener un nuevo certificado antes de la expiración del certificado actual.
5. Recupere la CRL según sea necesario.

Autenticación de CA

SCEP utiliza el certificado CA para asegurar el intercambio de mensajes para el CSR. Por consiguiente, es necesario obtener una copia del certificado de la CA. Se utiliza la operación **GetCACert**.

Petición

La solicitud se envía como una solicitud GET HTTP. Una captura de paquetes para la solicitud es similar a esta:

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

Respuesta

La respuesta es simplemente el certificado de CA con codificación binaria (X.509). El cliente necesita validar que el certificado de CA es de confianza mediante un examen de la huella dactilar/hash. Esto debe hacerse mediante un método fuera de banda (una llamada telefónica a un administrador del sistema o una configuración previa de la huella digital dentro del punto de confianza).

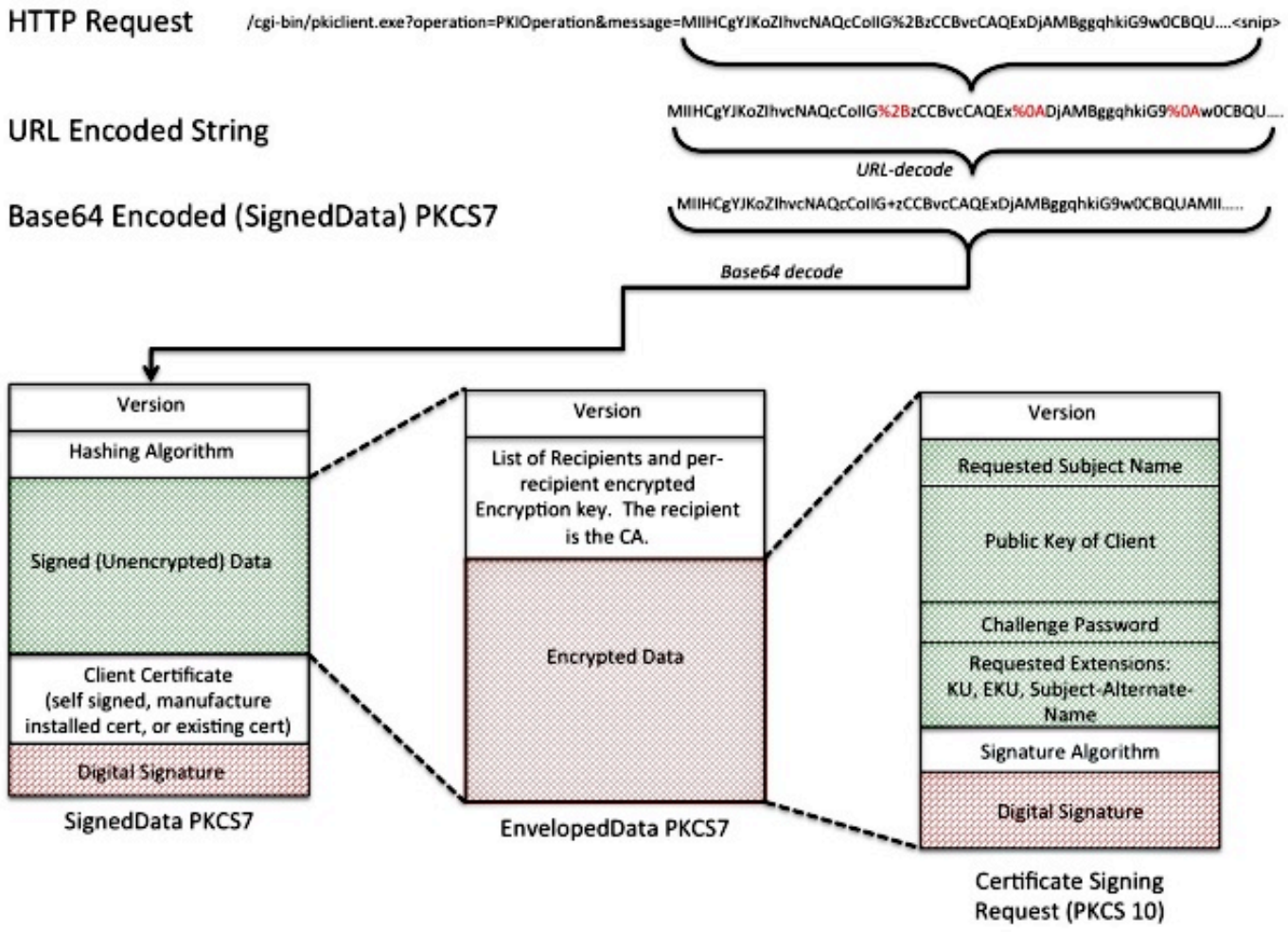
Inscripción de clientes

Petición

La solicitud de inscripción se envía como una solicitud GET HTTP. Una captura de paquetes para la solicitud es similar a esta:

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

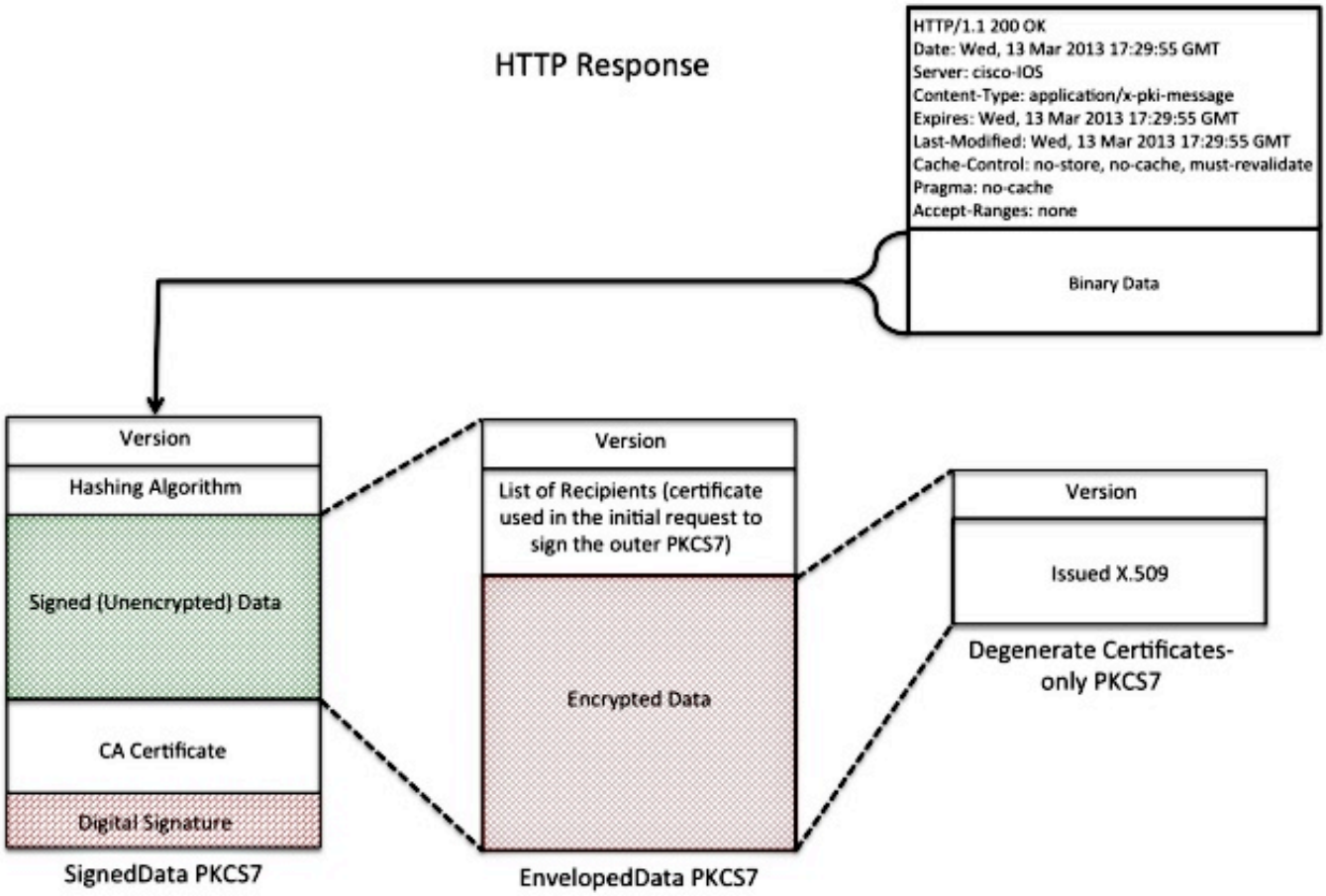
1. El texto que aparece después de "message=" es una URL Encoding String, que se extrae de la cadena de solicitud GET.
2. El texto se decodifica en una cadena de texto ASCII. Esa cadena de texto es un SignedData codificado en Base64 PKCS#7.
3. SignedData PKCS#7 está firmado por el cliente con uno de estos certificados; se utiliza para probar que el cliente lo envió y que no se ha modificado en tránsito:
   Certificado autofirmado (utilizado en la inscripción inicial)Certificado instalado por el fabricante (MIC)Una certificación actual que vence pronto (reinscripción)
4. La parte "Datos firmados" de los datos firmados PKCS#7 es un PKCS#7 de DatosEnvoldados.
5. Los datos envueltos PKCS#7 son un contenedor que contiene "Datos cifrados" y la "clave de descifrado". La clave de descifrado se cifra con la clave pública del destinatario. En este caso específico, el destinatario es la CA; como resultado. Sólo la CA puede descifrar realmente los "Datos cifrados".
6. La parte "Cifrado de datos" de PKCS#7 sobre envolvente es la CSR (PKCS#10).



Respuesta

La respuesta a la solicitud de inscripción SCEP es uno de los tres tipos siguientes:

- **Rechazar**: el administrador rechaza la solicitud por varios motivos, como:
  Tamaño de clave no válidoContraseña de desafío no válidaLa CA no pudo validar la solicitudLa solicitud solicitó atributos que la CA no autorizóLa solicitud fue firmada por una identidad en la que la CA no confía
- **Pendiente** - El administrador de CA aún no ha revisado la solicitud.
- **Éxito**: se acepta la solicitud y se incluye el certificado firmado. El certificado firmado se encuentra en un tipo especial de PKCS#7 denominado "Degenerar certificados sólo PKCS#7", que es un contenedor especial que puede contener uno o más X.509 o CRL, pero no contiene una carga de datos firmada o cifrada.



Reinscripción del cliente

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

Renovación

A medida que se aproxima la fecha de vencimiento de un certificado de ID, un cliente SCEP puede querer obtener un nuevo certificado. El cliente genera una CSR y pasa por el proceso de inscripción (como se definió anteriormente). El certificado actual se utiliza para firmar el SignedData PKCS#7, que a su vez prueba la identidad a la CA. Tras recibir el nuevo certificado, el cliente elimina inmediatamente el certificado actual y lo reemplaza por el nuevo, cuya validez comienza inmediatamente.

Renovación

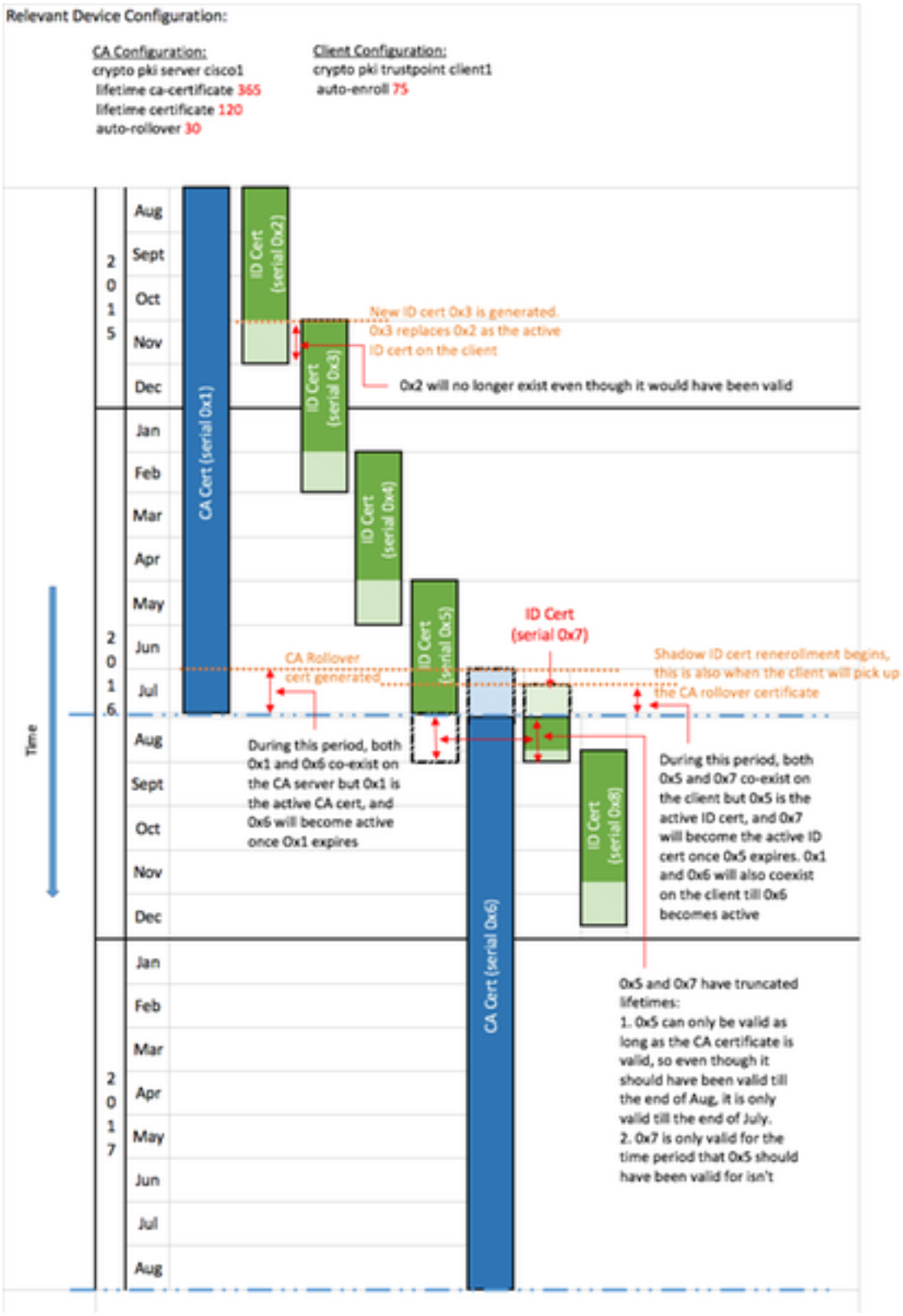
Rollover es un caso especial en el que el certificado de CA caduca y se genera un nuevo certificado de CA. La CA genera un nuevo certificado de CA que se vuelve válido una vez que caduca el certificado de CA actual. La CA generalmente genera este certificado "Shadow CA"

algún tiempo antes del tiempo de reversión, porque es necesario para generar certificados "Shadow ID" para los clientes.

Cuando el certificado de ID del cliente SCEP se acerca a la caducidad, el cliente SCEP consulta a la CA para el certificado "Shadow CA". Esto se hace con la operación **GetNextCACert** como se muestra aquí:

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

Una vez que el cliente SCEP tiene el certificado "Shadow CA", solicita un certificado "Shadow ID" después del procedimiento de inscripción normal. La CA firma el certificado "Shadow ID" con el certificado "Shadow CA". A diferencia de una solicitud de renovación normal, el certificado "Shadow ID" que se devuelve pasa a ser válido en el momento del vencimiento del certificado de CA (renovación). Como resultado, el cliente necesita guardar una copia de los certificados anteriores y posteriores a la renovación tanto para el certificado de CA como para el certificado de ID. En el momento del vencimiento de CA (renovación), el cliente SCEP elimina el certificado de CA y el certificado de ID actuales y los reemplaza por las copias "Shadow".



Bloques de creación

Esta estructura se utiliza como bloques de creación de SCEP.

Nota: PKCS#7 y PKCS#10 no son específicos de SCEP.

PKCS#7

PKCS#7 es un formato de datos definido que permite firmar o cifrar datos. El formato de datos incluye los datos originales y los metadatos asociados necesarios para realizar la operación criptográfica.

Sobre firmado (DatosFirmados)

El sobre firmado es un formato que transporta datos y confirma que los datos encapsulados no se modifican en tránsito a través de 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 versión: con SCEP, versión 1 utilizada.
- Lista de algoritmos de resumen utilizados: con SCEP, sólo hay un Signer y, por lo tanto, sólo un Algoritmo de hash.
- Datos reales firmados: con SCEP, se trata de un formato PKCS#7 de datos envueltos (sobre cifrado).
- Lista de certificados de los firmantes: con SCEP, se trata de un certificado autofirmado en la inscripción inicial o el certificado actual si se vuelve a inscribir.
- Lista de los firmantes y la huella dactilar generada por cada firmante: con SCEP, hay sólo un firmante.

Los datos encapsulados no están cifrados ni ofuscados. Este formato simplemente proporciona protección contra el mensaje que se modifica.

Datos sobre Sobres (EnvelopedData)

El formato de datos envueltos lleva los datos cifrados y sólo los destinatarios especificados pueden descifrarlos. Incluye esta información:

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

- Número de versión: con SCEP, se utiliza la versión 0.
- Lista de cada uno de los destinatarios y la clave de cifrado de datos cifrados relacionada. Con SCEP, solo hay un destinatario (para las solicitudes: el servidor CA; para obtener respuestas: el cliente).
- Datos cifrados: se cifra con una clave generada aleatoriamente (que se ha cifrado con la clave pública del destinatario).

PKCS#10

PKCS#10 describe el formato de una CSR. Una CSR contiene la información que los clientes solicitan que se incluya en sus certificados:

- Nombre del asunto
- Copia de la clave pública
- Contraseña de desafío (opcional)
- Cualquier extensión de certificado solicitada, como:
  Uso de claves (KU)Uso de clave extendido (EKU)Nombre alternativo del asunto (SAN)Nombre principal universal (UPN)
- Una huella digital de la solicitud

Este es un ejemplo de una RSE:

```
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
- SCEP heredado mediante la Guía de Configuración de CLI
- Configuración del soporte SCEP para BYOD

Appendix

Solicitudes SCEP

Request Message Format

Las solicitudes se envían con un HTTP GET del formulario :

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

Where:

- **La ruta CGI** depende del servidor y apunta al programa Common Gateway Interface (CGI) que gestiona las solicitudes SCEP: La CA Cisco IOS® utiliza una cadena de trayectoria vacía.Microsoft CA utiliza **/certsrv/mscep/mscep.dll**, que señala al servicio IIS MSCEP/ Network Device Enrollment Service (NDES).
- **Operación** identifica la operación que se realiza.
- **El mensaje** lleva datos adicionales para esa operación (y puede estar vacío si no se requieren datos reales).

Con el método GET, la parte del **mensaje** es texto sin formato o PKCS#7 codificado por reglas de codificación distinguidas (DER) convertido en Base64. Si se admite el método POST, el contenido que se enviaría en codificación Base64 con GET podría enviarse en formato binario con POST en su lugar.

Vista esquemática

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

- **operación** = `operación PKIO`: **mensaje**es una estructura SCEP **pkiMessage**, basada en PKCS#7 y codificada con DER y Base64.la estructura **pkiMessage** puede ser de estos tipos: **PKCSReq**: CSR PKCS#10**GetCertInicial**: sondeo del estado de concesión de CSR**GetCert** o **GetCRL**: recuperación de certificado o CRL
- **operación** = **GetCACert**, **GetNextCACert** o (opcional) **GetCACaps**: **mensaje** se puede omitir o se puede establecer en un nombre que identifique la CA.

Respuestas de SCEP

Formato de mensaje de respuesta

Las respuestas SCEP se devuelven como contenido HTTP estándar, con un **tipo de contenido** que depende de la solicitud original y del tipo de datos devueltos. El contenido DER se devuelve como binario (no en Base64 como para la solicitud). El contenido PKCS#7 puede o no contener datos cifrados/firmados; si no lo hace (sólo contiene un conjunto de certificados), se denomina **degenerado** PKCS#7.

Tipos de contenido

Valores posibles para **Content-Type**:

application/x-pki-message:

- en respuesta a la operación **PKIOperation, con pkiMessage** de tipo: **PKCSReq, GetCertInicial, GetCert o GetCRL**
- response body es un **pkiMessage** de tipo: **CertRep**

application/x-x509-ca-cert:

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

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

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

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

- en respuesta a la operación **GetNextCACert**
- response body es una variación de un **pkiMessage** de tipo: **CertRep**

La estructura de pkiMessage

OID de SCEP

```
2.16.840.1.113733.1.9.2 scep-messageType
2.16.840.1.113733.1.9.3 scep-pkiStatus
2.16.840.1.113733.1.9.4 scep-failInfo
2.16.840.1.113733.1.9.5 scep-senderNonce
2.16.840.1.113733.1.9.6 scep-recipientNonce
2.16.840.1.113733.1.9.7 scep-transId
2.16.840.1.113733.1.9.8 scep-extensionReq
```

PkiMessage de SCEP

- **Datos firmados** PKCS#7
- PKCS#7 EnvelopedData (llamado **pkcsPKIEnvelope**; opcional, cifrado para el destinatario del mensaje)
  messageData (CSR, cert, CRL, ...)
- **SignerInfo** con **atributos autenticados**:
  TransactionID, **messageType**, **senderNoncepkiStatus**, **RecipientNonce** (sólo respuesta)**failInfo** (respuesta + fallo solamente)

Tipo de mensaje SCEP

- solicitud:
  PKCSReq (19): CSR PKCS#10**GetCertInicial** (20): sondeo de inscripción de certificados**GetCert** (21): recuperación de certificado**GetCRL** (22): recuperación de CRL
- respuesta:
  CertRep (3): respuesta al certificado o solicitud CRL

SCEP pkiStatus

- **ÉXITO** (0): solicitud concedida (respuesta en pkcsPKIEnvelope)
- **FALLO** (2): solicitud rechazada (detalles en el atributo failInfo)
- **PENDIENTE** (3): la solicitud espera aprobación manual