Panoramica del protocollo SCEP (Simple Certificate Enrollment Protocol)

Sommario

Introduzione

In questo documento viene descritto il protocollo SCEP (Simple Certificate Enrollment Protocol), utilizzato per la registrazione e altre operazioni PKI (Public Key Infrastructure).

Premesse

SCEP è stato originariamente sviluppato da Cisco ed è documentato in un progetto IETF (Internet

Engineering Task Force).

Le sue caratteristiche principali sono:

- Modello di richiesta/risposta basato su HTTP (metodo GET; supporto opzionale per il metodo POST)
- Supporta solo la crittografia basata su RSA
- Utilizza PKCS#10 come formato di richiesta del certificato
- Utilizza PKCS#7 per trasmettere messaggi crittografati/con firma crittografica
- Supporta la concessione asincrona da parte del server, con polling regolare da parte del richiedente
- Dispone di un supporto limitato per il recupero dell'elenco di revoche di certificati (CRL) (il metodo preferibile è una query del punto di distribuzione CRL (CDP) per motivi di scalabilità)
- Non supporta la revoca dei certificati online (deve essere eseguita offline con altri mezzi)
- Richiede l'utilizzo di un campo di **verifica password** all'interno della richiesta di firma del certificato (CSR), che deve essere condivisa solo tra il server e il richiedente

La registrazione e l'utilizzo di SCEP in genere seguono questo flusso di lavoro:

1. Ottenere una copia del certificato dell'Autorità di certificazione (CA) e convalidarlo.
2. Generare un CSR e inviarlo in modo sicuro alla CA.
3. Eseguire il polling del server SCEP per verificare se il certificato è stato firmato.
4. Eseguire nuovamente la registrazione, se necessario, per ottenere un nuovo certificato prima della scadenza del certificato corrente.
5. Recuperare il CRL, se necessario.

Autenticazione CA

SCEP utilizza il certificato CA per proteggere lo scambio di messaggi per il CSR. Di conseguenza, è necessario ottenere una copia del certificato CA. Viene utilizzata l'operazione **GetCACert**.

Richiesta

La richiesta viene inviata come richiesta HTTP GET. L'acquisizione di un pacchetto per la richiesta ha un aspetto simile al seguente:

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

Risposta

La risposta è semplicemente il certificato CA con codifica binaria (X.509). Il client deve verificare che il certificato CA sia attendibile tramite un esame dell'impronta digitale/hash. Questa operazione deve essere eseguita tramite un metodo fuori banda (una telefonata a un amministratore di sistema o la preconfigurazione dell'impronta digitale all'interno del trust point).

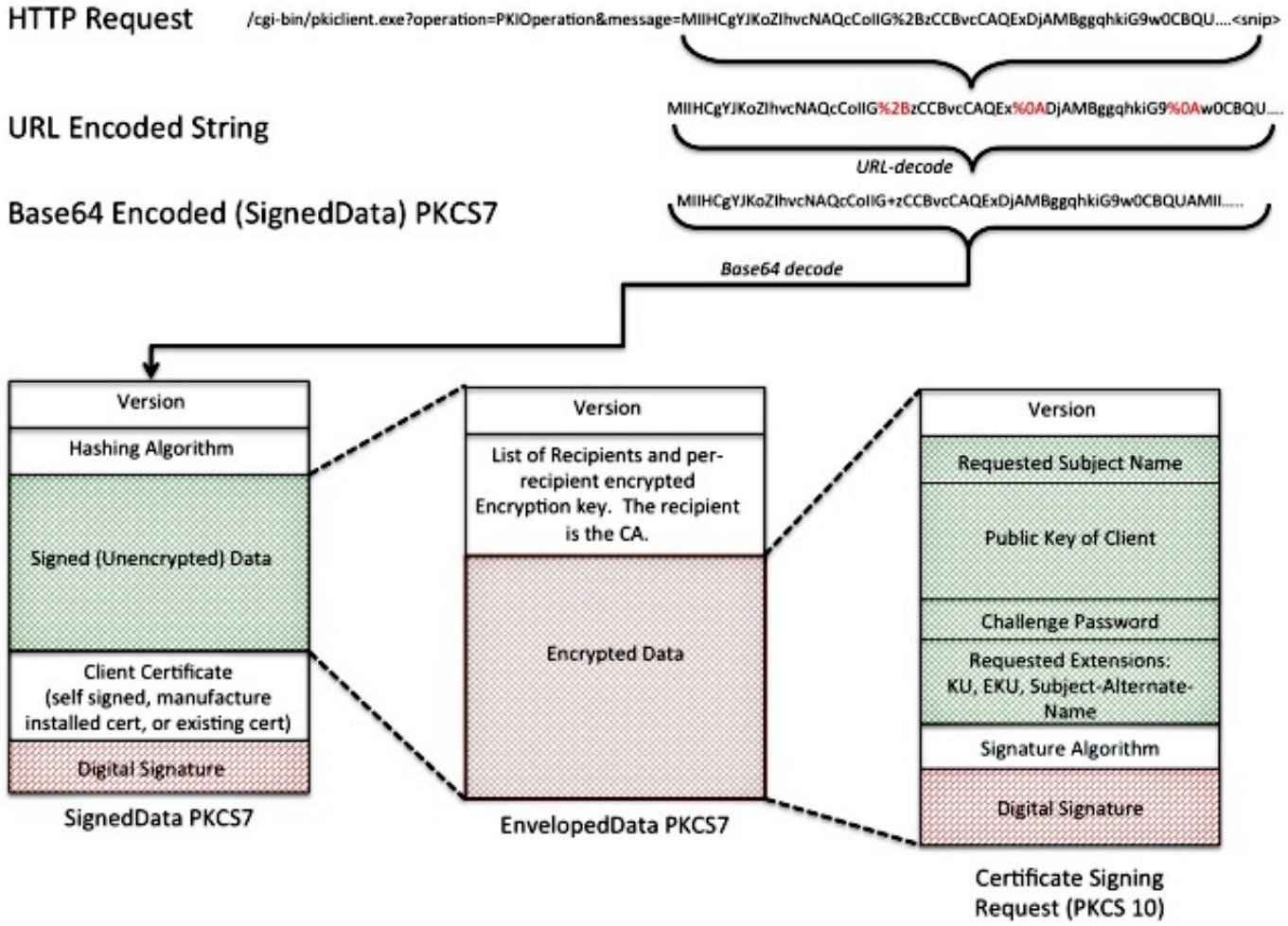
Registrazione client

Richiesta

La richiesta di registrazione viene inviata come richiesta HTTP GET. L'acquisizione di un pacchetto per la richiesta ha un aspetto simile al seguente:

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

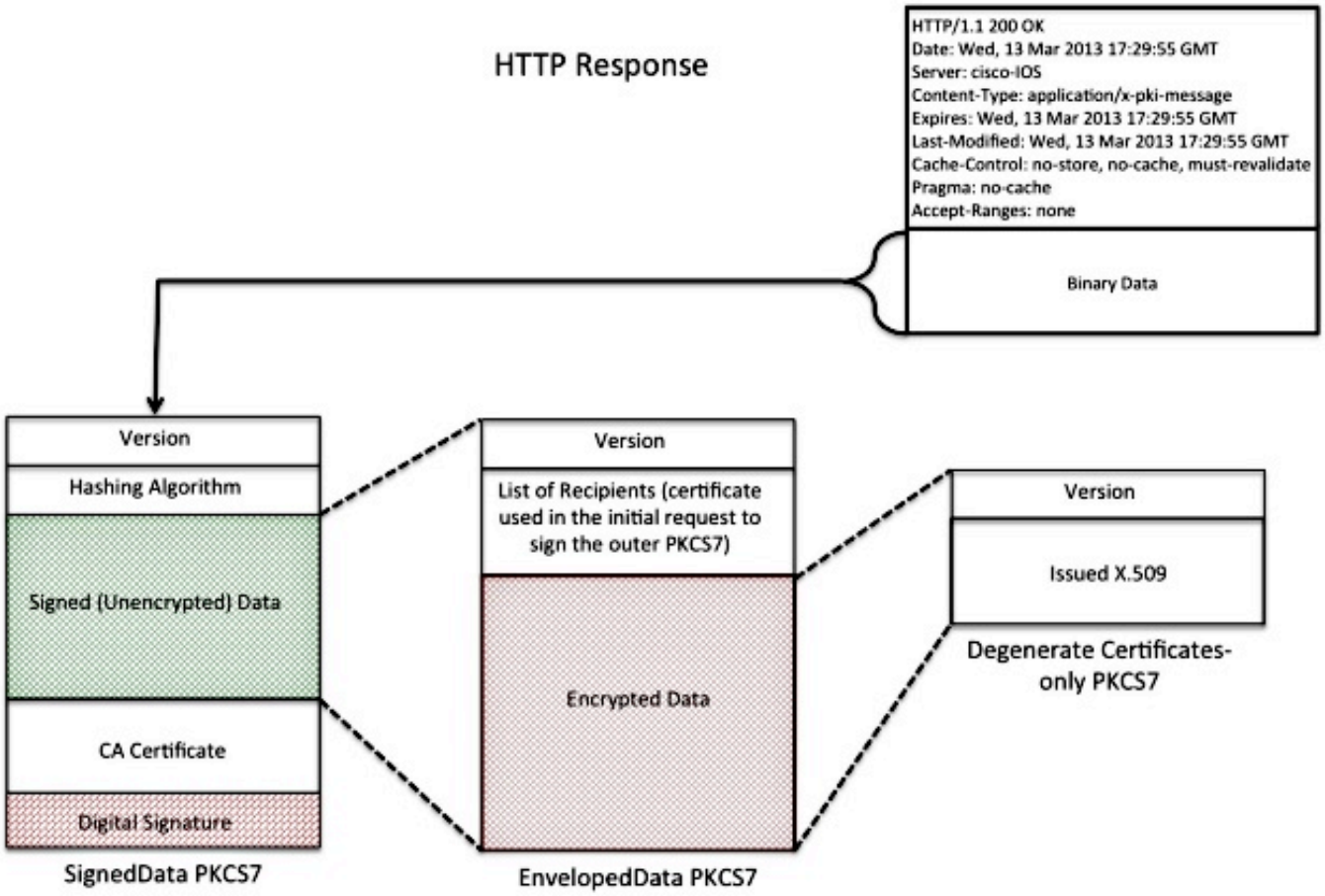
1. Il testo che segue "message=" è una stringa con codifica URL, estratta dalla stringa di richiesta GET.
2. Il testo viene quindi decodificato dall'URL in una stringa di testo ASCII. La stringa di testo è una SignedData PKCS#7 con codifica Base64.
3. SignedData PKCS#7 è firmato dal client con uno di questi certificati; è utilizzato per dimostrare che il cliente lo ha inviato e che non è stato alterato durante la trasmissione: Certificato autofirmato (utilizzato alla registrazione iniziale)Certificato di installazione produttore (MIC)Una certificazione corrente che scade a breve (nuova registrazione)
4. La sezione "Dati firmati" di SignedData PKCS#7 è una sezione di EnvelopedData PKCS#7.
5. EnvelopedData PKCS#7 è un contenitore che contiene "Dati crittografati" e la "chiave di decrittografia". La chiave di decrittografia è crittografata con la chiave pubblica del destinatario. In questo caso specifico, il destinatario è la CA; di conseguenza. Solo la CA può effettivamente decrittografare i "dati crittografati".
6. La parte "Dati crittografati" del PKCS#7 in busta digitale è il CSR (PKCS#10).



Risposta

La risposta alla richiesta di registrazione SCEP è di tre tipi:

- **Rifiuta**: la richiesta viene rifiutata dall'amministratore per una serie di motivi, ad esempio: Dimensioni della chiave non validePassword di richiesta non validaLa CA non può convalidare la richiestaLa richiesta ha richiesto attributi non autorizzati dalla CALa richiesta è stata firmata da un'identità non considerata attendibile dalla CA
- **In sospeso**: l'amministratore della CA non ha ancora esaminato la richiesta.
- **Operazione riuscita** - La richiesta viene accettata e il certificato firmato viene incluso. Il certificato firmato è contenuto in un tipo speciale di PKCS#7 denominato "Degenerate Certificates-Only PKCS#7", che è un contenitore speciale in grado di contenere uno o più X.509 o CRL, ma che non contiene un payload di dati firmato o crittografato.



Nuova registrazione client

Prima della scadenza del certificato, il client deve ottenere un nuovo certificato. C'è una leggera differenza di comportamento tra rinnovo e rollover. Il rinnovo si verifica quando il certificato ID del client si avvicina alla scadenza e la data di scadenza non corrisponde (precedente) alla data di scadenza del certificato CA. Il rollover si verifica quando il certificato ID si avvicina alla scadenza e la data di scadenza corrisponde alla data di scadenza del certificato della CA.

Rinnovo

Con l'avvicinarsi della data di scadenza di un certificato ID, è possibile che un client SCEP desideri ottenere un nuovo certificato. Il client genera un CSR ed esegue il processo di registrazione (come definito in precedenza). Il certificato corrente viene utilizzato per firmare

SignedData PKCS#7, che a sua volta fornisce l'identità alla CA. Alla ricezione del nuovo certificato, il client elimina immediatamente il certificato corrente e lo sostituisce con quello nuovo, la cui validità inizia immediatamente.

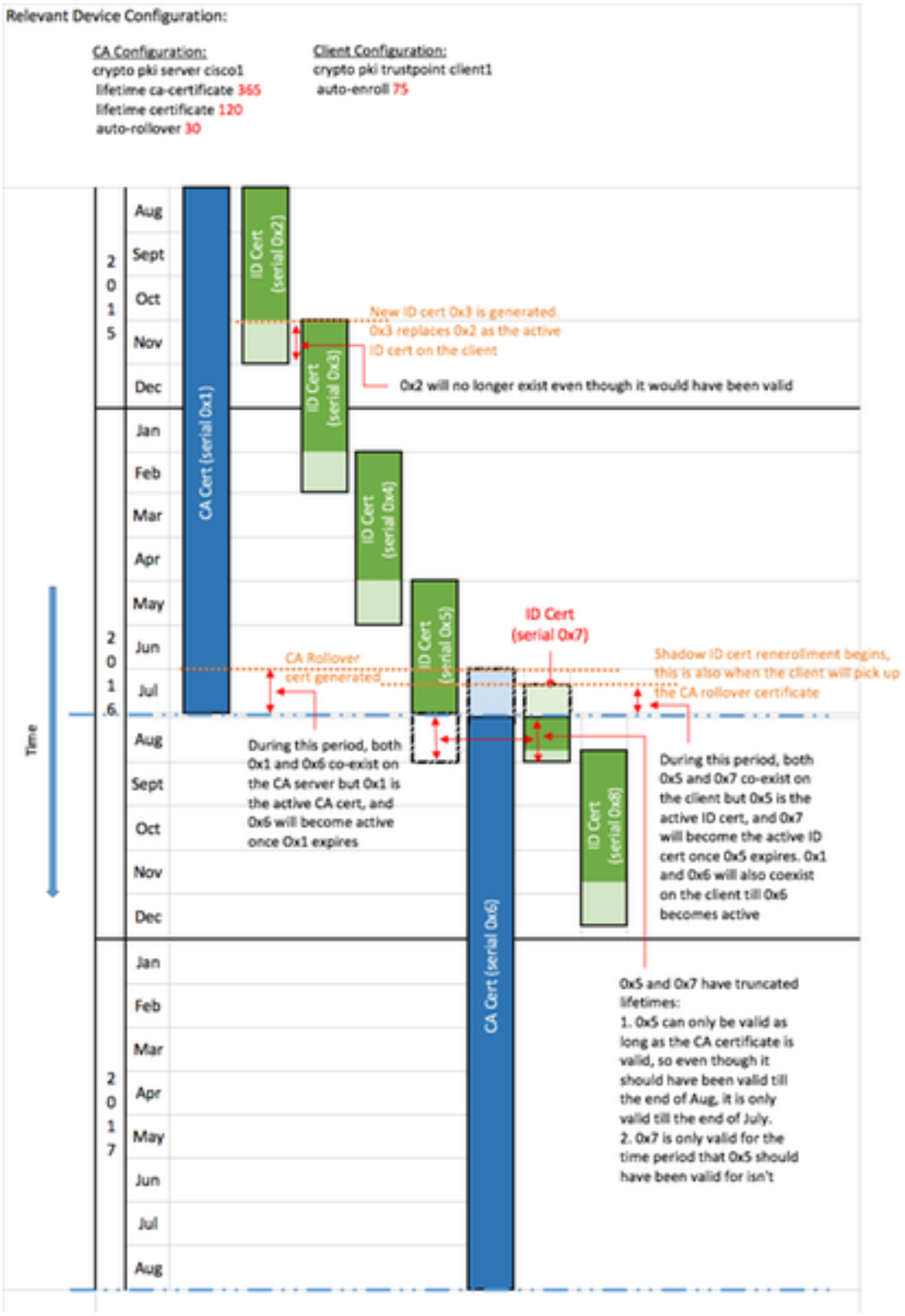
Rollover

Il rollover è un caso speciale in cui il certificato CA scade e viene generato un nuovo certificato CA. La CA genera un nuovo certificato CA che diventa valido alla scadenza del certificato CA corrente. La CA in genere genera questo certificato "CA shadow" un po' prima del tempo di rollover, perché è necessario per generare certificati "ID shadow" per i client.

Quando il certificato ID del client SCEP si avvicina alla scadenza, il client SCEP richiede alla CA il certificato "CA shadow". A tale scopo, eseguire l'operazione **GetNextCACert** come illustrato di seguito:

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

Quando il client SCEP dispone del certificato "CA shadow", richiede un certificato "ID shadow" dopo la normale procedura di registrazione. La CA firma il certificato "ID shadow" con il certificato "CA shadow". A differenza di una normale richiesta di rinnovo, il certificato "ID shadow" restituito diventa valido alla scadenza del certificato CA (rollover). Di conseguenza, il client deve conservare una copia dei certificati precedenti e successivi al rollover sia per la CA che per il certificato ID. Al momento della scadenza (rollover) della CA, il client SCEP elimina il certificato CA e il certificato ID correnti e li sostituisce con le copie "shadow".



Blocchi predefiniti

Questa struttura è utilizzata come elementi costitutivi di SCEP.

Nota: PKCS#7 e PKCS#10 non sono specifici di SCEP.

PKCS#7

PKCS#7 è un formato di dati definito che consente la firma o la crittografia dei dati. Il formato include i dati originali e i metadati associati necessari per eseguire l'operazione di crittografia.

Busta firmata (SignedData)

La busta firmata è un formato che trasporta i dati e conferma che i dati incapsulati non vengono alterati durante la trasmissione tramite firme digitali. Include le seguenti informazioni:

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

- Numero versione: con SCEP, viene utilizzata la versione 1.
- Elenco di algoritmi digest utilizzati: SCEP contiene un solo Signer e quindi un solo algoritmo hash.
- Dati effettivi firmati. Con SCEP, il formato dei dati in busta digitale PKCS#7 (Encrypted Envelope).
- Elenco di certificati dei firmatari - Con SCEP, si tratta di un certificato autofirmato alla registrazione iniziale o del certificato corrente se si esegue di nuovo la registrazione.
- Elenco dei firmatari e impronta digitale generata da ogni firmatario. Con SCEP, esiste un solo firmatario.

I dati incapsulati non vengono crittografati o offuscati. Questo formato fornisce semplicemente protezione contro il messaggio che viene alterato.

Dati in busta digitale (EnvelopedData)

Il formato Enveloped Data contiene dati crittografati che possono essere decrittografati solo dai destinatari specificati. Include le seguenti informazioni:

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

- Numero versione: con SCEP, viene utilizzata la versione 0.
- Elenco di ogni destinatario e della chiave di crittografia dei dati crittografata correlata. Con SCEP, esiste un solo destinatario (per le richieste: il server CA; per le risposte: client).
- Dati crittografati: vengono crittografati con una chiave generata in modo casuale, ovvero crittografata con la chiave pubblica del destinatario.

PKCS#10

PKCS#10 descrive il formato di un CSR. Un CSR contiene le informazioni che i client richiedono di includere nei propri certificati:

- Nome soggetto

- Una copia della chiave pubblica
- Una password di verifica (opzionale)
- Qualsiasi estensione di certificato richiesta, ad esempio: Utilizzo chiave (KU)Utilizzo chiave esteso (EKU)Nome alternativo soggetto (SAN)UPN (Universal Principal Name)
- Impronta digitale della richiesta

Di seguito è riportato un esempio di RSI:

```
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
```

Informazioni correlate

- Bozza IETF SCEP
- SCEP legacy con guida alla configurazione della CLI
- Configurazione del supporto SCEP per BYOD

Appendice

Richieste SCEP

Formato messaggio di richiesta

Le richieste vengono inviate con un HTTP GET nel formato:

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

Dove:

- **CGI-path** dipende dal server e punta al programma CGI (Common Gateway Interface) che gestisce le richieste SCEP: Cisco IOS® CA utilizza una stringa di percorso vuota.Microsoft CA utilizza **/certsrv/mscep/mscep.dll**, che punta al servizio IIS MSCEP/ Network Device Enrollment Service (NDES).
- **Operazione** identifica l'operazione eseguita.
- **Il messaggio** contiene dati aggiuntivi per l'operazione (può essere vuoto se non sono richiesti dati effettivi).

Con il metodo GET, la parte del **messaggio** è di testo normale o PKCS#7 con codifica DER (Distinguished Encoding Rules) convertita in Base64. Se il metodo POST è supportato, il contenuto inviato con codifica Base64 con GET potrebbe essere inviato in formato binario con POST.

Visualizzazione Struttura

Valori possibili per **le operazioni** e valori di **messaggio** associati:

- **operation** = `OperazionePKIO`: **messaggio**è una struttura SCEP **pkiMessage**, basata su PKCS#7 e codificata con DER e Base64.la struttura **pkiMessage** può essere di questi tipi: **PKCSReq**: PKCS#10 CSR**GetCertInitial**: polling per stato concessione CSR**GetCert** o **GetCRL**: recupero certificato o CRL
- **operazione** = **GetCACert**, **GetNextCACert** o (facoltativo) **GetCACaps**: **Il messaggio** può essere omesso o impostato su un nome che identifica la CA.

Risposte SCEP

Formato messaggio di risposta

Le risposte SCEP vengono restituite come contenuto HTTP standard, con un **Content-Type** che dipende dalla richiesta originale e dal tipo di dati restituito. Il contenuto DER viene restituito come binario (non in Base64 come per la richiesta). Il contenuto di PKCS#7 potrebbe contenere o meno dati crittografati/firmati in busta digitale; in caso contrario, ovvero se contiene solo un insieme di certificati, viene definito PKCS#7 **degenerato**.

Tipi di contenuto

Valori possibili per **Content-Type**:

application/x-pki-message:

- in risposta all'operazione **PKIOperation**, con **pkiMessage** di tipo: **PKCSReq, GetCertInitial, GetCert o GetCRL**
- il corpo della risposta è un **pkiMessage** di tipo: **RapprCert**

application/x-x509-ca-cert:

- in risposta all'operazione **GetCACert**
- il corpo della risposta è il certificato CA X.509 con codifica DER

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

- in risposta all'operazione **GetCACert**
- il corpo della risposta è un PKCS#7 degenerato con codifica DER che contiene i certificati CA e RA

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

- in risposta all'operazione **GetNextCACert**
- il corpo della risposta è una variante di un **pkiMessage** di tipo: **RapprCert**

Struttura di pkiMessage

OID 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 SCEP

- PKCS#7 **SignedData**
- PKCS#7 EnvelopedData (denominato **pkcsPKIEnvelope**; facoltativo, crittografato per destinatario messaggio)
  messageData (CSR, cert, CRL, ...)
- **SignerInfo** con **authenticatedAttributes**:
  transactionID, **messageType**, **senderNoncepkiStatus**, **receiverNonce** (solo risposta)**failInfo** (risposta + solo errore)

MessageType SCEP

- richiesta:
  PKCSReq (19): PKCS#10 CSR**GetCertInitial** (20): polling di registrazione certificati**GetCert** (21): recupero certificato**GetCRL** (22): Recupero CRL
- risposta:
  CertRep (3): risposta a una richiesta di certificato o CRL

PKIstatus di SCEP

- **OPERAZIONE RIUSCITA** (0): richiesta soddisfatta (risposta in pkcsPKIEnvelope)
- **ERRORE** (2): richiesta rifiutata (dettagli nell'attributo failInfo)
- **IN SOSPESO** (3): la richiesta è in attesa dell'approvazione manuale