Simple Certificate Enrollment Protocol

Ives Steglich

Munich, 12th October 2004
Overview

• SCEP Simple Certificate Enrollment Protocol
  • Goals
  • Basics
  • Message Format
  • Messages
  • Transaction Model
  • Requests

• Integration into OpenCA
  • Interface
  • Supported Operations
  • Open Issues
Overview

• SCEP Simple Certificate Enrollment Protocol
  • Goals
  • Basics
  • Message Format
  • Messages
  • Transaction Model
  • Requests

• Integration into OpenCA
  • Supported Operations
  • Open Issues
SCEP :: Goals

Primary

- CA and RA public key distribution
- Certificate enrollment
- Certificate revocation (manual)
- Certificate query
- CRL query

Secondary

- Certificate renewal
- CA rollover
- Confidentiality of internal networkdata?
SCEP :: Basics

Transport protocols

- HTTP (Get & POST)
- LDAP

Cryptographic Protocols & Containers

- PKCS#7 – Envelop and Confidentiality
- PKCS#10 – Certificate Requests

Cryptographic Algorithms

- RSA - no others supported till now for keys
- DES - used in PKCS#7 encryption portion
- MD5 - as digest for encrypted message part

Others

- Message based protocol: Request -> Response
- Actions always triggered by Client
SCEP :: Messages

PKCSReq
CertRep
GetCertInitial
GetCert
GetCRL
GetCACert
GetCACertChain (since rev. 3)
GetCACaps  (since rev. 10)
GetNextCACert  (since rev. 10)
SCEP :: Transaction Model

certificate non existent

PKCSReq

GetCertInitial

certificate pending

CertRep

(Status = SUCCESS)

or timeout occurred

certificate issued

CertRep (Status = REJECTED || FAILURE)
SCEP :: Basic Messageformat PKCS#7

**pkiMessage:**
- outer PKCS#7 container

**pkcsPKISigned**
- contains signed attributes
- transactin attributes
- etc.

**pkcsPKIEnveloped**
- contains encrypted information
- PKCS#10 Request
- issued certificate
- crl
- empty
- depending on request and reply from the scep-interface
Authenticated Transaction Attributes

- **transactionID**: unique transaction identifier - required
- **messageType**: how to handle the content / what to expect - required
- **pkiStatus**: only in response
- **failinfo**: only in error condition
- **senderNonce**: prevent reply attacks – required in request and response
- **recipientNonce**: prevent reply attacks – required in response
Messageformat :: Authenticated Transaction Attributes II

messageTypes

PKCSReq (19) Permits use of PKCS#10 certificate request
CertRep (3) Response to certificate or CRL request
GetCertInitial (20) Certificate polling in manual enrollment
GetCert (21) Retrieve a certificate
GetCRL (22) Retrieve a CRL or CRL-Distributionpoint

pkiStatus

SUCCESS (0) request granted
FAILURE (2) request rejected
PENDING (3) request pending for manual approval.
### Messageformat :: Failcodes

<table>
<thead>
<tr>
<th>failinfo</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>badAlg (0)</td>
<td>Unrecognized or unsupported algorithm identifier</td>
</tr>
<tr>
<td>badMessageCheck (1)</td>
<td>Integrity check failed</td>
</tr>
<tr>
<td>badRequest (2)</td>
<td>Transaction not permitted or supported</td>
</tr>
<tr>
<td>badTime (3)</td>
<td>Message time field was not sufficiently close to</td>
</tr>
<tr>
<td></td>
<td>system time</td>
</tr>
<tr>
<td>badCertId (4)</td>
<td>No certificate could be identified matching</td>
</tr>
<tr>
<td></td>
<td>the provided criteria</td>
</tr>
</tbody>
</table>

OpenCA mainly uses `badRequest` for ANY error inside OpenCA so kind of problems may be difficult to trace on client-side.
**SCEP :: Communication Examples**

**CA/RA Certificate Distribution**  
(performed only once usually)

- Get CA/RA Certificate  
  (HTTP GET or POST Request)
- CA/RA Certificate Download  
  (HTTP Response Message)
- Compute Fingerprint  
- Call CA Operator  
- Receive Call  
- Check Fingerprint

**Enrollment of an Certificate**

- PKCSReq:  
  PKI cert. enrollment msg
- CertRep: pkiStatus = PENDING[1]  
  no data attached (in PKCS#7)
- GetCertInitial: polling msg [n-1]
- CertRep: pkiStatus = PENDING[n]  
  no data attached (in PKCS#7)
- GetCertInitial: polling msg [n]
- CertRep: pkiStatus = SUCCESS  
  certificate attached (in PKCS#7)

**Manual Approval**  
or Cert. Request
Based on PKCS#10 contains

- subject "the requestor's subject name"
- challengePassword
- extensions (x.509 v3)

ChallengePassword

- automatic enrollment;
  (requires preauthentication of clients)
- as revocation pin for verification against the ca
SCEP :: Requests II

Renewal

- if used issued Cert instead of selfsigned request should be handled as renewal

- if request is send after half of validity time may also be handled as renewal request

- behavior dependent on CA Policy

CA Rollover

- use newly introduced msg: GetNextCACert if supported by CA

Request New Cert for new CA/RA Cert

- use the new CA/RA cert in the requesting envelop instead of the actual CA/RA cert
Overview

• SCEP Simple Certificate Enrollment Protocol
  • Goals
  • Basics
  • Message Format
  • Messages
  • Transaction Model
  • Requests

• Integration into OpenCA
  • Supported Operations
  • Open Issues
Integration into OpenCA

cmdline based toolkit openca-scep (c-code)
  - can parse and create scep-conform pkcs#7 msg
  - interface similar to openssl cmd-interfaces
  - doesn’t do transaction or error handling

OpenCA created a new interface called SCEP
  - consists of two functions
    - one for ca/ra certificate distribution
    - one for the operations itself
  - transaction state and error checking managed inside those functions
  - openca database keeps track of transactions
OpenCA :: Supported Operations

PKCSReq
CertRep
GetCertInitial
GetCert
GetCRL
GetCACert
GetCACertChain
GetCACaps (planed for 0.9.3)
GetNextCACert (planed for 0.9.3)
OpenCA :: Open Issues

Handling of renewals

- not implemented yet
- planed for 0.9.3

Preauthentication and automatic processing

- OpenCA backend doesn’t support automatic enrollment and preauthentication yet
- planed for 0.9.3

Crldistributionpoints

- sends back always CRL in response
- CDP instead of CRL not implemented yet
- planed as configuration option for 0.9.3

CA Rollover

- planed for 0.9.3