Use of WebDAV for Certificate Publishing and Revocation

<draft-chadwick-webdav-00.txt>

d.w.chadwick@kent.ac.uk

Rationale

- LDAP is not a good mechanism for the distribution of certificates and CRLs for a number of reasons
  - Cannot usually access LDAP servers through firewalls
  - Cannot search for certificates or CRLs based on their contents
  - Cannot retrieve individual certificates when a user has several
  - Can be difficult to install and use – problems with ;binary and not ;binary
- Whole area of revocation still poses problems

Revocation Problems

- CRLs can get extremely large (several megabytes) for downloading every day or every few hours
- Where do we get them from – problems with LDAP through firewalls
- Some systems therefore use short lived certificates without revocation e.g. Grids, SAML
- What is the ideal frequency of issuing CRLs or length of short lived certificates? Issuer makes the choice but RP takes the risk (and pays the cost)
- Conclusion. Should revisit the best way of distributing certificates and revocation information

Ideal Solution

- Certificates should only need to be issued once and last for as long as is needed, and be reusable as many times as needed by as many relying parties as the user wishes. This offloads work from the issuer.
- Distribution protocols should not be inhibited by firewalls, therefore should be based on http
- Revocation information should instantly be made available to all relying parties
- Should place the responsibility for determining when to retrieve revocation information onto the relying parties, since these bare the risks
- Checking certificate validity should be as efficient as possible

Conceptually based on the REST Principles

- **Representational State Transfer (REST)** principles came from PhD dissertation about the web by Roy Fielding, author of RFC 2616 (HTTP1.1), see http://roy.gbiv.com/pubs/dissertation/rest_arch_style.htm
- Based on client server interactions in which servers are stateless (i.e. no cookies since these are sitewide not server specific)
- Any state retained by the server must be modeled as a resource and all resources are uniquely addressable by a URL i.e. the web is the state machine
- Each server response should indicate if it is cacheable or not cacheable

Applying REST to X.509

- The state of a certificate is either that it

URL ? • EXISTS — Certificate at this URL

- Or

URL ! • DOES NOT EXIST →
  - WAS NEVER CREATED — Nothing at Either URL
  - Or
  - WAS CREATED BUT IS NOW REVOKED — CRL at this URL

Applying URLs to these states we have

We define 2 new Authority Information Access Methods

- Certificate URL (URL ?)
- Revocation URL (URL !)

- We embed these URLs in every certificate that we issue

- Any Relying Party can now instantaneously check on the state of a certificate by contacting these URLs

We use the WebDAV protocol, an enhancement of HTTP

- WebDAV protocol [RFC 2518] provides extensions to HTTPv1.1 protocol [RFC 2616]
- Each X.509 certificate or CRL becomes a WebDAV resource
- WebDAV uses GET to read a resource, PUT to create a resource or overwrite an existing one and DELETE to delete a resource
- WebDAV supports creation of sets of resources called collections
- So all certificates of a user become a WebDAV collection
- WebDAV specifies the MKCOL method to replace HTTP PUT or POST for collections, since latter are allowed to overwrite existing content at the specified URL, whereas MKCOL will fail if there is any existing content at the specified URL. We need this to stop a certificate issuer from overwriting a user's existing collection when creating a new collection

The AIA Access Methods

```
AuthorityInfoAccessSyntax  ::=
        SEQUENCE SIZE (1..MAX) OF AccessDescription
  AccessDescription  ::=  SEQUENCE {
        accessMethod          OBJECT IDENTIFIER,
        accessLocation        GeneralName  }
```

Define two new accessMethods, webdavCert and webdavRev, as follows:

webdavCert OBJECT IDENTIFIER ::= { id-ad n}

webdavRev OBJECT IDENTIFIER ::= { id-ad m}

accessLocation must be a URL pointing to the WebDAV server where the certificate or CRL can be found

But what if a RP does not have a Cert?

- E.g. during certificate path processing a RP may need the certificate of an issuer
- We could say, tough, an RP must always be pushed ALL the certs that it needs from the client
  - But users don't always know what all of these are, and it can make it more complex for users
- OR
- We could define a way for an RP to pull any certs that it needs, given the DN of an entity
  - Means we should standardise a way of naming WebDAV repositories so that standard URLs can be constructed

WebDAV Naming Model

- WebDAV resources are named by URLs, where the hierarchical names are delimited with the “/” character
- The name of a collection ends with /
- Use the rules of [RFC 4514] to convert X.500 DNs into strings, with the exceptions that we replace the comma "," separator between RDNs with the "/" character which is the WebDAV separator, and replace spaces with %20
  - E.g. X.500 DN “c=gb, o=University of Kent, cn=David Chadwick", will be named in a WebDAV repository with the URL:
  - https://server.dns.name/c=gb/o=University%20of%20Kent/cn=David%20Chadwick/
- We make each subject DN the name of a collection of certificates
- We make each issuer DN concatenated with cn=CRLs the name of a collection of CRLs
  - E.g. CRLs of “c=gb, o=University of Kent” CA will be located at https://server.dns.name/c=gb/o=University%20of%20Kent/cn=CRLs/

[RFC 4514] LDAP String Representation of Distinguished Names

Security Considerations

- WebDAV revocation is vulnerable to DOS attacks. But so are OCSP and normal CRLs.
  - To mitigate can posses an old CRL, but can also download a complete WebDAV collection of CRLs from the issuer
- HTTP provides public access to the certificate, which may violate the privacy of the certificate subject
  - Use HTTPS and client authentication with access controls and/or return encrypted responses
- Intermediate Web servers may cache copies of frequently accessed web pages to improve performance
  - Web server MUST use the no-cache cache-response-directive when returning certificates and Not Found CRLs

Ideal Solution Revisited

- Certificates should only need to be issued once and last for as long as is needed ✓
- Certs may be reusable as many times as needed by as many relying parties as the user wishes. ✓
- Distribution protocols should not be inhibited by firewalls, therefore should be based on http ✓
- Revocation information should be instantly made available to all relying parties ✓
- Should place the responsibility for determining when to retrieve revocation information onto the relying parties, since these bare the risks ✓
- Checking certificate validity should be as efficient as possible ✓

Any Questions??

Supplementary Slides

- Only if time, and to answer specific questions

Unresolved Issues

- Should we allow multiple URLs for a certificate or CRL to be included in each issued cert?
  - This will limit DOS attacks but increase latency of certificate validation if one or more URLs are not available
- The discovery problem - How can RPs know which WebDAV repository to contact to pull a user's certificate?
  - Currently DNS name of WebDAV server needs to be configured in by out of band means

Naming Individual Certs and CRLs

- Each PKC has the file suffix .p7c and filename of issuer DN plus certificate serial number, using the rules in RFC 4514 to create strings from DNs
  - E.g. " cn=CSCA, o=university of kent, c=gb+SN=123445.p7c"
- Each AC has the file suffix .ace
- Filename of role ACs is created from the contents of its attribute values plus the serial number of the certificate
  - E.g. a role AC with the embedded attribute type PermisRole with attribute value Project Manager, and certificate serial number of 12345 would create the filename "PermisRole=Project Manager+SN=123456.ace".
- Filename of a policy AC is created from the unique name of the policy embedded in the policy attribute value,
  - E.g. a policy with the name "AstroGridUsers" would produce the filename "Policy=AstroGridUsers.ace".
- Each CRL has the file suffix .crl and filename of the serial number of the certificate it revokes.
  - E.g. a CRL that revokes a certificate with serial number 1234 would produce the filename "serialNumber=1234.crl".

Searching for Certs

- WebDAV supports document properties using XML for name/value pairs. Stored as meta data with the resource
- So could store role=manager, serial number=1234 etc. as certificate properties
- Unfortunately the standard WebDAV protocol does not support retrieval of documents with specific property values, only with property names
  - E.g. Find all certificates with a serial number property. Useless !!
- WebDAV searching and locating capability (DASL) started in 1998, but the work was never completed and IETF working group shut down in 2000, but work has continued in the background and implementation are now reputed to exist
- We may be able to add this capability in the future