Journal of Mobile, Embedded and Distributed Systems, vol. III, no. 1, 2011
ISSN 2067 – 4074

PKI Interoperability Based on Online Certificate Validation

Dinu Smădu
Military Technical Academy
George Coşbuc Blvd, no 81-83, Sector 5,
Bucharest, zip code 050141
ROMANIA
smadudinu@gmail.com

Abstract: One of the most important problems related to Public Key Infrastructures is the validation of the digital certificates. Certificate validation services can be based on offline and/or online schemes. Offline schemes have the major disadvantage that they cannot always give an up-to-date response. On the other side, the most used protocol for online validation, the Online Certificate Status Protocol [1], also has its drawbacks. It can only state if a certificate has been revoked or not. RFC 5055 [2] defines a more complex protocol, the Server-based Certificate Validation Protocol (SCVP), capable of building and validating the certification path. To implement a basic functionality of this new protocol, we will start from an existing project, the CADDISK and we will try to implement an OpenSSL module.

Key-Words: SCVP, PKI, certificate validation, OpenSSL, path discovery, path validation, CADDISK

1. Introduction

The initial purpose of public key cryptography was that of securing the communication between two entities, which do not know each other. For example, when Bob wants to send a message to Alice, he needs to know her public key, to be able to send her encrypted messages. The problem arises when the number of users is very large. To solve the problem of linking users to their public key, a relatively small number of authorities should be used. These authorities are called Certification Authorities (CAs). A CA digitally signs a data structure that contains the pair: identity-public key. This data structure is called a certificate. CAs are critical for large scale PKIs.
One of the most important problems related to PKI is the validation of these digital certificates. Each certificate is meant to be valid for a certain period of time, but there are situations when a certification needs to be revoked, before the end of the validity period. For example, if a user's private key is compromised, he must call for the revocation of his certificate. This revocation is done by the emitting Certification Authority, at the request of the owner or for other reasons.
With the increase in security demands at the application level, the need to more precisely validate the certificate has grown. The certificate validation solutions are divided in two categories: off-line schemes and on-line schemes.
The off-line schemes are based on certificate revocation lists, that are updated at certain periods of time, and have the main disadvantage that they do not always use up-to-date information when validating a certificate.
On the other side, on-line schemes are much more accurate and should be compulsory for applications that require a high security level (e.g. Banking applications).

This is a post conference paper. Parts of this research have been published in the Proceedings of the 3rd International Conference on Security for Information Technology and Communications, SECITC 2010 Conference (printed version).

1.1 OCSP

OCSP (Online Certificate Status Protocol) is a client-server protocol, used to verify

the revocation status of a certificate. An OCSP client sends a request containing the ID of the certificate to be validated and the OCSP server responds with a message containing the respective revocation status.
The main purpose of OCSP was to be scalable and to offer more precise revocation information.
For each query, the server may give one of the three answers:

- good – the certificate has been revoked by the time the request has been received
- revoked – the certificate has been temporarily or definitively revoked
- unknown – the server does not have enough information about the certificate

The standard allows for multiple queries to be sent in the same request.
The main disadvantage of this protocol is the centralized architecture, and the fact that each response must be digitally signed by the server. The response time of the server may thus suffer.

1.2 Lightweight OCSP

Derived from OCSP, the Lightweight OCSP protocol was proposed by IETF PKIX [5] and aims to solve the problems OCSP had in dealing with very large scale PKIs. It thus uses pre-computed responses, it reduces the size of the messages and it caches the responses.
Clients are not permitted to sign the requests; all the extensions are eliminated, excepting the Nonce extension (which is not recommended to be used). Also, only one certificate id may be contained in a request.

2. SCVP

Sometimes it is not enough to know if a certificate is revoked or not and the client may need to have access to additional data.
The OCSP and its derivatives cannot build and validate the certification path and do not support other types of certificates (e.g. Attribute certificates [6]).

In 1999 the Server-based Certificate Validation Protocol was proposed [7] and after a rather large number of modifications, in 2007 IETF PKIX Working Group proposed the RFC 5055 [2].
SCVP allows the PKI clients to delegate the whole validation process of a digital certificate to a SCVP server.
It is much more general and complex than OCSP. The most complex component in PKI applications is the process of validating the certification path. The main purpose of SCVP is to facilitate the implementation of PKI clients by delegation of Path discovery and/or path validation to a dedicated server. It also permits the centralized management of the digital certificates validation policy.

2.1 Path discovery

To validate a certificate, a path between it and a trusted point must be created. This is what path discovery consists of. One or more candidate paths are created and then passed over to the path validation algorithm. Even if each certificate in the path is valid, this doesn't necessarily mean that the path is valid. There are several other constraints, like the size of the certification path.

2.2 SCVP architecture

There are two possible approaches to the SCVP architecture.

- The SCVP servers go through all the necessary steps for validating a certificate (path discovery and path validation according to the central policy). This implies that the client applications completely trust de validation sever
- Only the certificate path and/or the revocation information is sent to the client, which does its own validation

SCVP is also a request-response protocol. The client sends a request to the SCVP server, which in turn replies with a response with the results of the validation requests.
A client request contains:

ISSN 2067 – 4074

- The certificate(s) to be validated or for which information is needed
- The actions that have to be done by the server
- The validation policy to be used
- Optional elements (e.g. reference time),

The certificates in the request can be public key certificates or attribute certificates. The certificate can be included in the request (not possible with OCSP).
For public key certificates, the standard defines three types of verifications:

- Building a certification path up to a trusted root
- Building a validated certification path up to a trusted root
- Building a validated certification path up to a trusted root and verifying the revocation state for the certification path.

For attribute certificates, the verifications consist of:

- Building a certification path up to a trusted root, for the issuer of the attribute certificate.
- Building a validated certification path up to a trusted root, for the issuer of the attribute certificate.
- Building a validated certification path up to a trusted root, for the issuer of the attribute certificate and verifying its revocation state.
- Building a validated certification path up to a trusted root, for the issuer of the attribute certificate and verifying the path's revocation state and the revocation state of the attribute certificate.

As one can see, there are several scenarios for the SCVP protocol. A client may want to be responsible for the validation of the certification path, and it would only demand the path from the server. If the client doesn't specify any option, the server will respond with de validity state of the built certification path.
The building and the validation of the certification path is done based on one or more trust anchors. A certain validation policy specifies the rules and the parameters which should be used by the server in order to validate a certificate. This policy is included in the client's request to the server. The client may also send in this request other variables, like the validation algorithm to be used, the accepted trusted anchors, etc.
The SCVP server publishes the references to the available validation policies, and if these policies have certain parameters that can be modified, the default values are also published.

2.3 The SCVP model

Actors:

- The SCVP Client
- The SCVP Server
- Other SCVP Servers
- SCVP server database
- OCSP Servers

There are two possible types of request that the client can send to the server. The client can either request the available validation policies or send certificate based requests (path discovery, path validation). Of course, there are two corresponding responses, the validation policies or the validation responses for the second type of request. To obtain the revocation state of a certain certificate, the SCVP server can make requests to the other SCVP servers, to OCSP servers or it can check its own database. It is also possible to use certificate revocation lists.
The server's response can either contain the requested information, or an error message.

2.4 Security

SCVP uses CMS encapsulation for the request/response messages. The protocol offers the possibility to compute a MAC for the message or to digitally sign it.
The requests sent to the server can be digitally signed and encapsulated in SignedData CMS message.
In the case of MAC based authentication, a key exchange protocol can be used, and the request is encapsulated in an AuthenticatedData CMS message.
The SCVP responses may be protected based on the validation result (success or error) and on the communication channel in use (e.g. TLS). In most of the

cases the protocol states that the answers should be protected, except for the error messages if the client explicitly demands unprotected error messages.

3. SCVP implementations

Even if the first SCVP draft is 11 years old, only a few software programs support this protocol, and almost none offer its full functionality.
The main software companies that offer commercial SCVP solutions are Axway[8], Ascertia [9], Tumbleweed and CoreStreet[10].
The CADDISK [11] project is an open source project that aims the provide PKI security by using DNSsec[11] and LDAP[12]. The team working on this project started the implementation of a SCVP solution, but unfortunately the project was abandoned.
Based on the existent sources, our goal is to create an OpenSSL compliant SCVP version, that should offer the basic validation functionality,
The CADDISK sources have been recompiled and now we have a working client-server architecture on a Ubuntu based machine.
For now the server is not capable of processing any requests, it only answers with a standard error message, but more functionality is to be added soon.

4. Conclusions

There is an obvious need for a new software solution, capable of processing complex certificate validation queries and the Server-based Certificate Validation Protocol is probably the best protocol to be chosen to fulfill these requests.
Because in the RFC, the described functionality of the protocol is quite vast, a software solution that would cover all of the aspects mentioned there is rather difficult to implement. But most of the vendors do not have this in mind. Building a powerful base that would be capable of validating the certification paths would be an improvement over the existing protocols.
Because of the interest shown in SCVP, the base of an open-source implementation would be a big step, because it would allow for rapid developing and testing.

References

[1]OCSP - http://www.ietf.org/rfc/rfc2560.txt

[2]SCVP RFC 5055- http://www.rfc-editor.org/rfc/rfc5055.txt

[3] CADDISK- http://www-lor.int-evry.fr/~maknavic/CADDISC/

[4] OPENSSL- http://www.openssl.org/

[5] RFC 5019- http://www.rfc-editor.org/rfc/rfc5019.txt

[6] Attribute certificates- http://www.ietf.org/rfc/rfc3281.txt

[7] SCVP draft

[8]http://www.axway.com/products-solutions/email-identity-security/identity-security/va-server

[9]http://www.ascertia.com/Products/TFSCVP/Default.aspx?m=eidvalid&s=tfscvp

[10]http://www.globalforte.com/download00034678/Tumbleweed/Products/Validation_Authority_Validator_Toolkit.pdf

[10] http://www.corestreet.com/toolkits/

[11] http://www.dnssec.net/

[12] http://en.wikipedia.org/wiki/LDAP