Bachelor Thesis
Use of DANE to improve the security for identity federations

Sophia Bergendahl, Christoffer Holmstedt

Luleå University of Technology
Dept. of Computer Science, Electrical and Space Engineering

June 12, 2012

ABSTRACT

Identity federations is a way to build a standard for online services similar to the one in real life with identification cards and signatures. However, there are more security aspects to take in to account online. This report analysis the security mechanism used to achieve data integrity in an identity federation and specifically the use of X.509 certificates. As well as an evaluation of the possibility to use DNS-Based Authentication of Named Entities (DANE) to improve the security for an identity federation. The report is a result of literature studies, set up of test environment and discussions with experts. It concludes that improvements can be made on how identity federations handle their own metadata and trust other entities metadata. DANE is today only a draft, but when DANE with TLS/TLSA becomes a RFC standard and/or when a standard for how DANE handles SAML certificates can it be used to improve the initial trust bonding.

PREFACE

It's an interesting world we live in where we are constantly on the move with our cellphones, laptops and other devices online day out and day in, all hours during the day. With the increase of possibilities online we are eager to find a way to build trust online for even more feature rich online services for all of us as citizens.

This thesis work was carried out at .SE (The Internet Infrastructure Foundation) in Sweden, at their Stockholm office in the spring of 2012, to be more precise, from the end of March to the beginning of June. The subject for this thesis was chosen because of interest in internet security. To be able to keep focus and dive deeper into the specific technologies Sophia's focus has been on identity federations and certificates while Christoffer has put more focus on DNS-Based Authentication of Named-Entites (DANE).

We would like to thank Staffan Hagnell (.SE) for giving us the opportunity to come and work for .SE and for the support on the way to our conclusions. We would also like to thank Carl Ljungqvist (Certezza AB) for the first introduction about indentity federations and Rickard Bellgrim (Certezza AB) for giving us much needed help when setting up the testing environment as well as helping us understand how certificates are used in general and within identity federations. For giving us new perspectives about DANE we would like to thank Jakob Schlyter (Kirei) and Leif Johansson (SUNET).

From our university, Luleå University of Technology, we would like to thank our supervisor Ulf Bodin, Dept. of Computer Science, Electrical and Space Engineering, for guiding us in the right direction when struggling with the thesis and putting words to our conclusions. We would also like to thank Johan Carlson, Dept. of Computer Science, Electrical and Space Engineering, for the LaTeX template that we have used for this report.

As a final note we would like to thank all of you that has supported us in our work but didn't get mentioned above. Thank you.

Sophia Bergendahl and Christoffer Holmstedt

CONTENTS

Chapter 1 – Introduction 1

1.1 Background . . . 1
1.2 Problem description . . . 2
1.3 Purpose . . . 2
1.4 Project delimitations . . . 2

Chapter 2 – Method 3

2.1 Approach . . . 3

Chapter 3 – Theory 5

3.1 Identity federations . . . 5
    3.1.1 What is an identity federation? . . . 5
    3.1.2 The technical side of an identity federation . . . 7
3.2 Certificates . . . 7
    3.2.1 X.509 certificates . . . 7
    3.2.2 Certificates in SAML . . . 8
3.3 DNS-Based Authentication of Named Entities . . . 9
    3.3.1 In general . . . 9
    3.3.2 An example . . . 10
3.4 How DANE can be implemented in Shibboleth . . . 12

Chapter 4 – Evaluation 13

4.1 How to improve security in identity federations . . . 13
4.2 How to combine identity federations and DANE . . . 14
    4.2.1 How to publish certificates . . . 14
    4.2.2 How to locate the correct certificate in DNS . . . 17
4.3 How DANE can be implemented in Shibboleth . . . 18

Chapter 5 – Discussion 21

5.1 Our conclusions . . . 21
    5.1.1 Identity federations with SAML2 and DANE . . . 21
    5.1.2 How DANE can be implemented in Shibboleth . . . 22
5.2 How to continue with further research . . . 23
    5.2.1 Secure communication with a federation operator . . . 23

5.2.2 Publishing SAML2 certificate as CERT RR, is it viable? 23
5.2.3 How to locate the correct certificate in DNS 23

APPENDIX A – INSTALLATION AND CONFIGURATION 25
A.1 Bind, OpenDNSSEC and other requirements 25
A.2 Shibboleth Service/Identity Provider 26
A.3 Federation Operator - Metadata Aggregator 26
A.4 DANE Trust Engine Extension 26

CHAPTER 1

Introduction

1.1 Background

In todays society more and more information is shared through internet. A lot of the communication consist of sharing pictures, stories and information with friends and family. The online websites that allow and promote these services are often very easy to use and especially easy to register a new account for. To log in to most of these services the required credentials are often username and password.

Services that require somekind of proof that the user behind the keyboard really is the person he/she says he/she is, are lagging behind in the online era. A solution to this problem is ”BankID” which was introduced in 2003 [1] in Sweden. Other solutions comes from Nordea, SEB, Telia, Posten and Steria [2, p. 256]. All listed solutions have their pros and cons and a new solution is in the horizon. A new solution will be built upon the concept of identity federations [2, p. 23].

One major identity federation that exists in Sweden today is SWAMID, it's for students in higher education and is run by SUNET. Another one is ”Skolfederation.se” which is planned to go into production during 2013. ”Skolfederation.se” is for pupils/parents and teachers in compulsary primary and secondary school run by .SE and SUNET. Other federations that soon will come in to production is ”E-legitimationsnämden” [3] and ”Vård, hälsa, omsorg” according to .SE.

An identity federation does not have the goal to increase security and therefore will probably continue to use the same authentication mechanism that has been used before e.g. username and password. However, the federation provides user friendliness since the user only has to have one username and one password to all services within the federation.

1.2 Problem description

The subject of the thesis is divided into smaller subsection.

- An analysis of the security mechanism used to achieve data integrity in an identity federation and specifically the use of X.509 certificates.

- Two separate evaluations one about the possibility to use DANE to improve the security for an identity federation and another one concern the possibility to implement support for DANE in the open source software package Shibboleth.

- Implementation of DANE in Shibboleth, simpleSAMLphp or equivalent software to improve the security for an identity federation.

1.3 Purpose

The purpose with this thesis is to show that DNS-Based Authentication of Named Entities (DANE) [4, 5, 6] can be used to improve security for identity federations. In the same time proving DNS Security Extension (DNSSEC) [7, 8, 9, 10] useful, since DANE is depending on the use of DNSSEC.

1.4 Project delimitations

As stated above this project will be about DANE and identity federations and as with most, if not all implementation of different protocols and standards, a lot of underlying technology is used. This is also the case with DANE and identity federations.

Within an identity federation there are four main entities which are clients, service providers, identity providers and in most cases a federation operator. All entities might communicate with each other over different protocols and communication schemes. In this project the main focus has been on communication over HTTPS.

A firm delimitation in this report is that of excluding how Transport Layer Security (TLS) and Secure Sockets Layer (SSL) works in detail and why it's deemed as a secure way for transmission between two hosts. Some parts in the report will touch the topic of the TLS/SSL handshaking process but will not go into further detail. The focus in this report is on the actual data that is being transmitted (SAML2 requests and responses) and how it's deemed secure.

Another delimitation is made concerning DANE, DNS and DNSSEC. DANE uses DNSSEC and DNS as an underlying technology and security schemes/protocols. As the focus is on DANE, issues that might exist with DNS and DNSSEC in general is not discussed in this report.

CHAPTER 2

Method

2.1 Approach

To be able to show that DANE can be used to improve the security in identity federation, more knowledge was needed in the specific subjects of concern.

The beginning of the project consisted mainly of a literature study about DANE, identity federations and the SAML 2.0 standard, which the federation that .SE is involved in is based on. Several discussions took place with experts in the different subjects to get other perspectives than those recieved from the literature study.

A test environment was set up to develop a deeper understanding about identity federations and SAML, as well as the software Shibboleth that was used. It was with recommendations by Staffan Hagnell the software choice to use Shibboleth was made, instead of simpleSAMLphp or equivalent software. The test environment also included OpenLDAP as user credential storage, Bind as DNS server and OpenDNSSEC to be able to sign all DNS resource records.

With an improved understanding and knowledge, an analysis was possible and conclusions from the analysis could be used to start the implementaion phase. During the entire project a few hours per week were dedicated for writing this report.

CHAPTER 3

Theory

3.1 Identity federations

3.1.1 What is an identity federation?

A simple identity federation example is when signing in to ”antagning.se”. The user visits ”antagning.se”, needs to visit ”my pages”, to do so he/she presses ”Log in”. Assuming the user is already studying in Sweden perhaps at Luleå University of Technology he/she can select this university among the listed universities. The user has made his/her choice and is redirected to the selected university's student portal and is identified through the university. When the user is verified he/she is redirected back to ”antagning.se” with an approval message.

An identity federation can be based on different standards. The identity federation used in the thesis work testing environment is based on OASIS Security Assertion Markup Language (SAML 2.0) [11, 12, 13, 14, 15, 16, 17]. The SAML technical overview [17, p. 8] describes SAML as an XML-based framework both for describing and exchanging security information between business partners online.

In SAML standard ”antagning.se” is a Service Provider (SP) [16, p. 11] and the identification at Luleå University of Technology is the Identity Provider (IdP) [16, p. 7], see figure 3.1. The Discovery Service (Disco) is where the user could chose from several universities at ”antagning.se”. Furthermore, SAML also specifies the Attribute Authority (AA) that provides a ”ticket” containing information about the user from a third party perspective. A theoretical AA example is ”Bolagsverket” (Swedish Companies Registration Office). ”Skatteverket” (Swedish Tax Agency) wants to allow individuals to see their business' tax account and sends an attribute request to ”Bolagsverket” that will answer this request with a ”ticket” containing information that allows or disallows the individual to see his/her tax account [2, p. 284].

Outside of the SAML standard, the identity federation can also have a Federation Operator (FO). It is possible to have an identity federation both with and without a

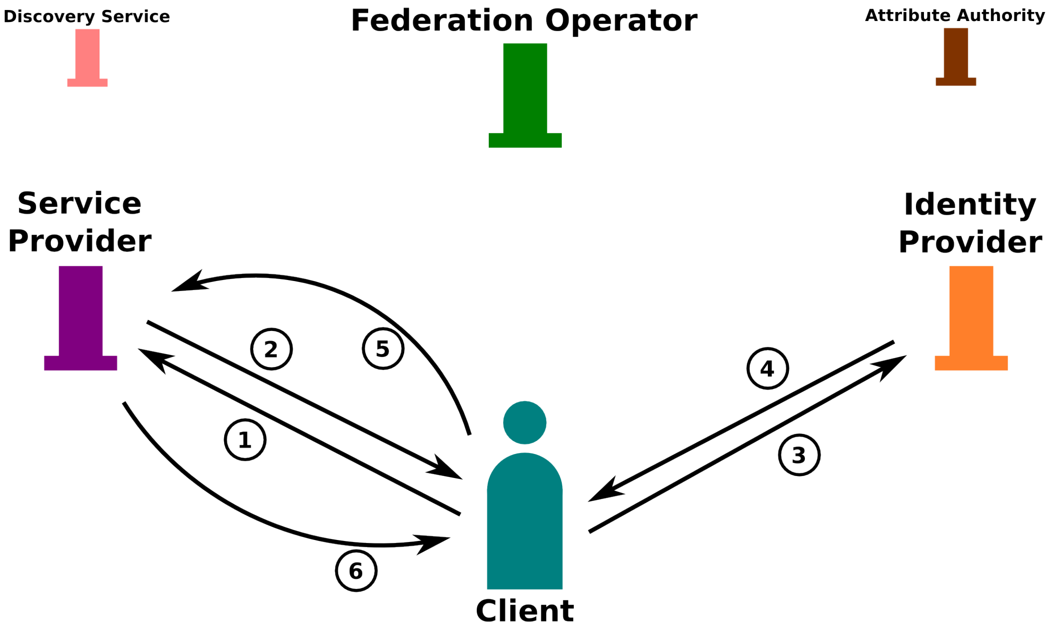


Figure 3.1: Shows how the client visits the service provider (1). The client want to see restricted resources at the service provider, whom demands that the client is identified first. Therefore, the service provider redirects the client to the identity provider (2) and (3). At the identity provider, the client is authenticated and sent back to the service provider (4) and (5). The client is now allowed to see restricted resources at the service provider (6).

federation operator. The federation operator provides digitally signed aggregated metadata [18, p. 3]. The SAML Technical Overview [17, p. 16] says ”Metadata defines a way to express and share configuration information between SAML parties. For instance, an entity’s supported SAML bindings, operational roles (IDP, SP, etc), identifier information, supporting identity attributes, and key information for encryption and signing can be expressed using SAML metadata XML documents. SAML Metadata is defined by its own XML schema”. The key information refers to the shared certificates (public keys) that are used when connecting and sending requests and responses messages between the entities. With that said every service provider and identity provider holds the shared certificates between each other within its metadata.

The need of a federation operator grows as the amount of service- and identity providers increase, since it is easier to turn to the federation operator to make sure the metadata is up to date instead of having to turn to each other. The only thing is that the service- and identity providers has to keep track of a shared certificate with the federation operator, meanwhile, the federation operator provides them with a metadata file containing all certificates within the federation. A federation operator also provides a unity between the entities in the federation. This unity can be used so that every entity share administration, has the same technical set up and so on.

3.1.2 The technical side of an identity federation

The example with ”antagning.se” is in SAML referred to as a SP-initiated web SSO [17, p. 12], where SSO stands for Single Sign-On and SP for Service Provider. SSO is when it is possible to sign in at one place and use that identification at other places as well.

SP-initiated web SSO uses The Web Browser SSO Profile [15, p. 14] that says that ”to implement this scenario, a profile of the SAML Authentication Request protocol is used, in conjunction with the HTTP Redirect, HTTP POST and HTTP Artifact bindings”.

In more technical terms the SP-initiated web SSO scenario is in the SAML Profiles document [15, p. 15] described as the client (principal) sends an HTTP request via an HTTP user agent asking to access a protected resource at the service provider. The client is not authenticated and therefore the service provider obtains with the authentication request protocol the location to the identity provider. The service provider then issues an ”AuthnRequest” message that the user agent deliver to the identity provider through an HTTP redirect, post or artifact binding. At the identity provider the client is identified through some back-end engine with user credentials. The identity provider then issues a ”Response” message that the user agent deliver to the service provider through HTTP post or artifact binding. This message holds at least one authentication assertion as well as attributes that describes the user or it may hold an error. Depending on the response message the client will receive access to the service provider or not.

3.2 Certificates

3.2.1 X.509 certificates

This section describes how certificates are used to send a private message from Alice to Bob, the technical steps from encryption of the message to the validation of trust. The example is based on the introduction on SSL/TLS from the Apache Foundation [19].

Public key cryptography algorithm transforms a message from Alice in to a private (encrypted) message to Bob, the message is unreadable until it is decrypted. Encrypted messages can only be decrypted with a secret key, in public key cryptography two keys are used that both can encrypt and decrypt messages. However, if one key is used to encrypt the message only the other key can decrypt it, which make publication of the public key possible while keeping the private key secret. Additionally, if Alice and Bob has shared keys, Alice can encrypt the private message with the public key and only Bob with the private key can decrypt the message.

The possibility that someone has made changes to the message still exist since the public key is public. ”Message digest” even called one-way function or hash function can be used to guarantee that the message has not changed. Message digest creates a short, fixed-length representation of the message about to be sent and sends the message and the summary to Bob. Then, when Bob receives the message he makes a summary as well and compares it with Alice's.

In addition, the message digest has to be securely sent as well, which is made possible with digital signatures. A digital signature is made by the private key by encrypting a digest of the message and other information as well, sequence number for example. It is Alice that creates the digital signature and includes it in the message digest to Bob and since no one can change the digest and still sign it, the integrity is kept. Moreover, to make sure reuse of the signature can not be made at a later date, the signature contains a sequence number that is unique.

Furthermore, Alice needs to know that the shared public key is with Bob as well as Bob needs to verify that the message signature really was signed by Alice's private key. With a certificate that validates the other's identity, confirms the public key and is signed by a Certificate Authority (CA), Alice can be sure she is talking to Bob and vice versa.

Certificates and security

Certificates are not secure in themselves the security is built on cryptographic algorithms and most often on public-key cryptography as in the example above. Cryptographic algorithms, which is a part of cryptanalysis, uses keys to protect data, in public-key cryptography a public encryption key and a private (secret) decryption key is used. [20, p. 252]

The RSA algorithm has become almost synonymous with public key cryptography according to Computer Networking, A top down approach [21, p. 726]. This book also informs that RSA uses arithmetic modulo-n operations as well as it describes how the recipient generates his/her private and public RSA key. RSA is secure since there are no known algorithms that quickly can factor large prime numbers and the larger the generated prime numbers are, the more secure the keys become.

3.2.2 Certificates in SAML

Saml2int [22] describes an Interoperable SAML V2.0 Web Browser SSO Deployment Profile that is based on SAML Web Browser SSO Profile in the SAML Profiles document [15]. The deployment profile describes the behaviour and options that all parties are supposed to follow. Furthermore, it addresses the content, exchange and processing of SAML messages. For example in the Saml2int profile [22], the service provider should at endpoint protect response messages using Transport Secure Layer (TLS) or Secure Sockets Layer (SSL) and identity providers should do the same when receiving authentication request. However, if an identity provider does not use TLS/SSL, XML Encryption should be used and in its response message return an encrypted assertion. In addition, if service providers does not use TLS/SSL as recommended, then the service providers metadata should include a suitable key descriptor for XML Encryption.

3.3 DNS-Based Authentication of Named Entities

3.3.1 In general

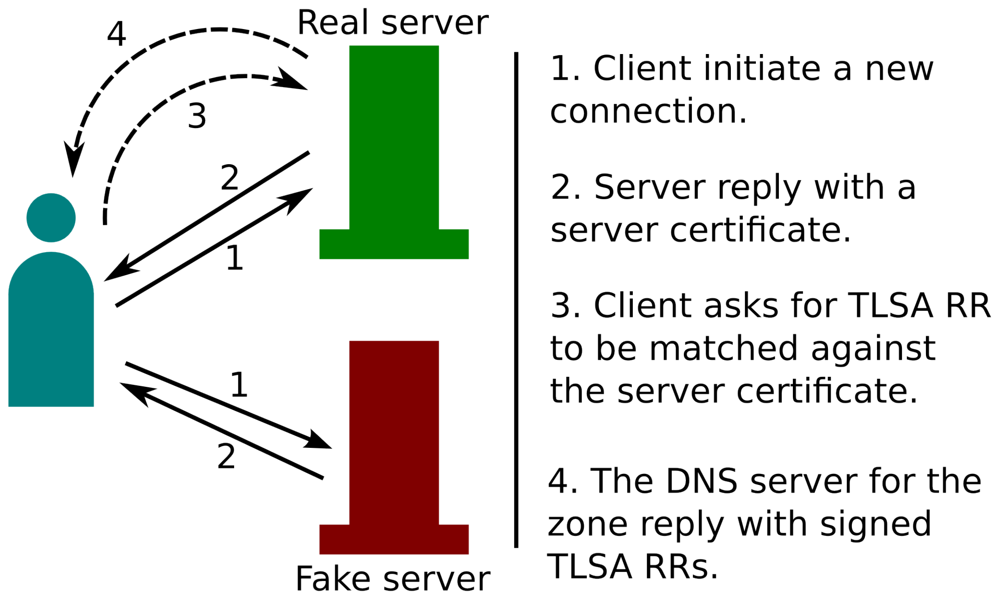
DANE, DNS-based Authentication of Named Entities, is as of writing still a new concept and technology. It's introduced with the informational RFC6394 document [4] describing some use cases and some parts specified in more detail in the Internet-Draft *"The DNS-Based Authentication of Named Entities (DANE) Protocol for Transport Layer Security (TLS)"* [5]. After the specification for TLS communication, S/MIME is up next, which will describe how to map an emailaddress to a resource record. Exactly how this will work and which resource record type that must be used is not yet clear according the to the draft [6]. Make note that these Internet-Drafts from the Internet Engineering Task Force (IETF) is still "work in progress" as they are still drafts and may change.

The problem that DANE tries to solve is the current issue with Certificate Authorities (CAs) where anyone of them may give out a certificate for any domain name. It might not be likely that a CA loses its private key but might sign a certificate with information that is not correct or the CA might sign a key that is intended for signing/encrypting emails but is then used in other settings.

A hacker that gets hold of a perfectly valid certificate from a well-known CA for a domain he/she doesn't own will be able to lure unexpected visitor to a fake site with TLS/SSL active. As the model with CAs works today where they can sign any domain name and the more common webbrowsers (Mozilla Firefox, Microsoft Internet Explorer, Google Chrome) trust most of these CAs, the certificate from the hacker will give clients connecting to the server with the false certificate a "false-positive". The domain name matches the common name in the certificate and it's signed by one of all trusted/well-known CAs. This is where DANE comes in as a solution.

The following explanation is also presented in figure 3.2. If the client connecting to the false server could get some information about which certificate is the true one to use, the client would know that the false certificate is a false one and not to trust it. In short, this works by trusting the DNSSEC infrastructure and the owner of a domain name can publish a TLSA DNS resource record (for TLS communication) which tells the client which certificate is the correct one to use. If the hacker tried the same attack with DANE fully operational he/she would not succeed cause the client would know that the false certificate from the hacker (fake server) is actually a false one. The client would in this case immidiately drop all established connections, if any, and never initiate a new TLS connection.

This effectively moves some responsiblity from the CAs to the domain owner. The domain owner now has full control of which TLS certificates are the valid ones.



1. Client initiate a new connection.

2. Server reply with a server certificate.

3. Client asks for TLSA RR to be matched against the server certificate.

4. The DNS server for the zone reply with signed TLSA RRs.

Figure 3.2: A client connect to either a real or fake server. The third and fourth step is done within the Domain Name System and requires DNSSEC active. (It is assumed that both the real and the fake server has a server certificate signed by some well-known CA for the same domain.)

3.3.2 An example

How DANE works in detail for TLS communication is specified in the Internet-Draft earlier mentioned [5] but to be able to follow later reasoning a short presentation will be given here. Let's take an example, Alice wants to get hold of some resource from Bob that needs to be protected so only Alice and Bob knows about it and nobody tampers with it while in transit. This is where TLS comes into play. Alice initiate a TLS handshake as a TLS client with Bob which in this example is acting as TLS server. Everything works fine and with TLS they setup a secure channel between themselves.

Hold on, how does Alice know that she is communicating with "the" Bob she believes it is or is it someone else that is trying to fake the identity of "the" Bob? To perhaps give a better explanation figure 3.3 describe this dilemma.

In the initial TLS handshake Alice retrieves a server certificate that she needs to validate with a third party, a CA. This is a certificate from Bob that is signed by his CA of choice, which basically says that the CA in question has confirmed, in some out-of-band way, that it's the true Bob, Alice is communicating with.

In this scenario, similiar to the scenario with the "hacker" above, there is no way for Bob as server/domain owner to limit which certificates that is allowed to be used for his

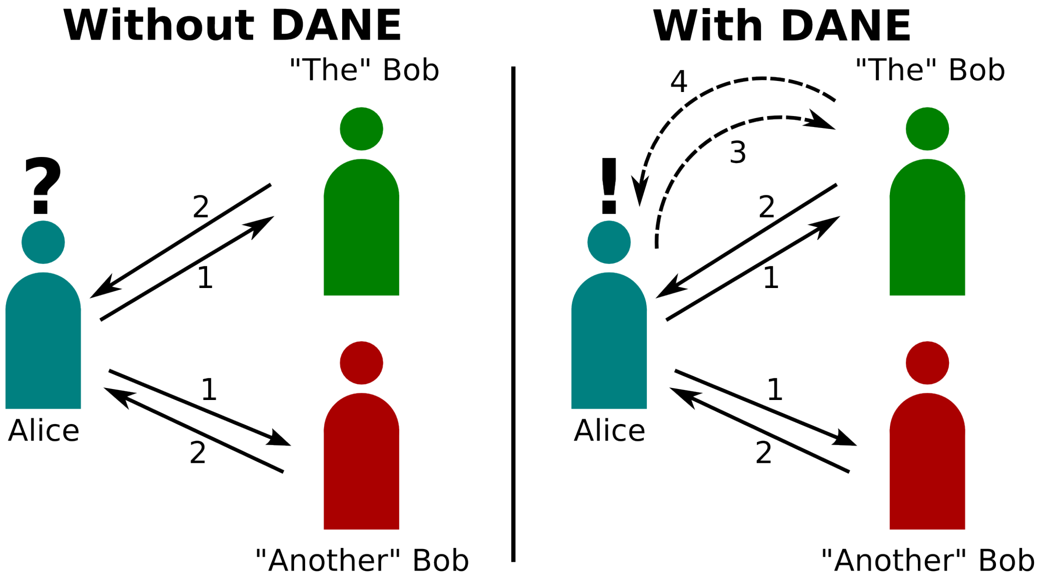


Figure 3.3: Without DANE the connecting client can't be sure without some out-of-band procedure or prior communication who is the real Bob. With DANE "the" Bob can publish a TLSA RR which will determine which TLS server certificate is the real one. (This assumes that both the client and "the" Bob trust the DNSSEC infrastructure).

domain. A hacker with a certificate for Bob's domain signed by any well-known CA will be able to set up his/her valid TLS server in the name of Bob. Alice will then never know if it really is "the" Bob she is communicating with next time.

To prevent this Bob can publish TLSA DNS resource records (TLSA RRs)[5, ch. 2] with either the full certificate or a hash value of it. Alice can when initiating the TLS handshake also ask for the TLSA RRs from Bob. With DNSSEC in place she can be sure that she recieves the TLSA RR from the true Bob and if the resource records matches the given certificate that was given in the TLS handshake Alice is sure it's to "the" Bob she is opening a secure channel.

Depending on the TLSA RR information and local policies Alice might have to do the the normal certificate chaining to some trusted anchor as well or she will accept a self-signed certificate from Bob. TLSA resource records is a new DNS record type that is still not approved as a standard so it might change in the future before being approved.

3.4 How DANE can be implemented in Shibboleth

Shibboleth is not a single computer program, it's more of a software package consisting of an identity provider written in java and a service provider written in C++ as the main parts. As dependency for the identity provider and service provider software, Opensaml exist both in Java and C++. Other projects are a centralized discovery service, embedded discovery service and metadata aggregator among others. Even if Shibboleth can be viewed as a software package, all parts can and usually are installed seperately on different machines.

DNS-Based Authentication of Named Entities (DANE) is still a draft for TLS and S/MIME communication and the RFC available is only an informational RFC focusing mainly on TLS use cases. The main concept is that with DNSSEC in place it's possible to use the DNS system as a trusted source for retrieval of some extra information that confirms what earlier was made in some other out-of-band way. As an example it's possible to trust a TLS certificate retrieved when establishing that connection because it's possible to retrieve extra information through the DNS system that proves it's the right certificate in use.

Focus has been put on the identity provider part in Shibboleth. It has good documentation on how to extend it. Make note that the identity provider software is currently at version 2.3.6 and a new major version is on the way as 3.0.0. The identity provider software has both an API and different extension points.

CHAPTER 4

Evaluation

4.1 How to improve security in identity federations

Messages sent between entities within the identity federations depends on the data integrity being held. However, the most difficult part is setting up SAML's initial trust relationship between entities, the first exchanging of keys. The SAML metadata document says *"...to aid in establishing trust in the accuracy and legitimacy of metadata, including use of XML signatures, SSL/TLS server authentication, and DNS signatures. Regardless of the mechanism(s) used, relying parties SHOULD have some means by which to establish trust in metadata information before relying on it."* [12, p. 29].

With that said, after the first certificate exchange has been made a trust relationship is built and the data integrity is up and running. The question is, how can the first certificate exchange be made securely? One solution might be a third party, a CA or the use of DNSSEC or maybe both to validates that the certificates is the right ones. Another one could be to use Pretty Good Privacy (PGP) [23] to sign or encrypt the certificate before transmission. With this solution, the soon to be certificate receiver first sends a PGP encrypted message containing its contact information to the certificate sender, that decrypts the message. Finds out the receivers phonenumber and calls him/her to confirm the message sender and key used in the PGP message. When it is confirmed that it came from the receiver, the certificate sender sends the certificate to the receiver.

Furthermore, the communication standard described in SAML could be more strict. That involves the first certificate exchange just mentioned. Another example is the certificate itself, the SAML bindings document says that the bindings which use TLS/SSL has to authenticate clients using X.509 certificate [14, p. 8] and the metadata profile document says *"In the case of an X.509 certificate, there are no requirements as to the content of the certificate apart from the requirement that it contain the appropriate public key. Specifically, the certificate may be expired, not yet valid, carry critical or non-critical extensions or usage flags, and contain any subject or issuer. The use of the certificate structure is merely a matter of notational convenience to communicate a key and has* *no semantics in this profile apart from that. However, it is RECOMMENDED that certificates be unexpired."* [13, p. 11].

By changing the SAML documents guidelines in general to be less open for interpretation it could improve the security. For example to not recommend that the certificates has not expired, but express that they should or must not be expired. Moreover, it is a vulnerability that the metadata file can contain expired certificates.

Without stricter guidelines an online check of the certificate could also be a solution to the expired certificate problem. The Online Certificate Status Protocol (OCSP) [24] could be used to perform this online check.

4.2 How to combine identity federations and DANE

As stated earlier it's possible to divide federations into two categories, those with a federation operator and those without. In the analysis that has been done the solution with a federation operator is the most probable and viable setup in the long run, it's basically much easier to scale, both technically and non-technically. It's with this configuration in mind the following evaluation has been made.

With a federation operator publishing all the metadata about all service providers, identity providers and other entities how do the trusting entites rely on the federation operator? Is it the correct federation operator the trusting entities connect to or is it someone trying to "impersonate" the role of the federation operator as a man-in-the-middle attack.

As of now the trust must be made in some out-of-band way e.g. a simple download of the federation operator's certificate and then confirm it's the correct one via telephone, email communcation between administrators or meet in person. This is not very practical when the federation grows bigger with tenths or hundreds of service- and identity providers or is it the only way?

4.2.1 How to publish certificates

SAML2 certificate as CERT resource record

The first solution that comes into mind is that it might be possible to publish the federation operator's certificate in the Domain Name System and more specifically as a CERT RR[25]. This would mean that when a requesting entity (service provider, identity provider or any other entity) needs to update the metadata about all other entities it connects to the federation operator and downloads the metadata. At the same time the requesting entity sends out a DNS lookup (with DNSSEC active) for a corresponding CERT RR. When both the metadata and the CERT RR is retrieved, a comparison can be made and confirm or dismiss the validity of the metadata. This is done after step 3 and 4 in figure 4.1.

A problem that still exists with this approach is how will each entity transmit metadata about itself to the federation operator and how will the metadata be validated by the federation operator (Step 1 and 2 in figure 4.1). To keep building on the same solution each entity would have to publish respective certificate as a CERT RR themselves in their own domain zone. They would also need to sign their own metadata which is not the case in current standards. Now the initial upload of metadata from an entity to the federation operator could be intitated, perhaps through a web interface, with or without some "Pretty Good Privacy" (PGP) solution e.g. OpenPGP[23]. The federation operator then fetches the metadata as well as the corresponding CERT RR from the entity and if it's the correct signature on it the federation operator can publish it with the metadata from all other entities. Within one cache interval all entities will now fetch the complete updated metadata from the federation operator.

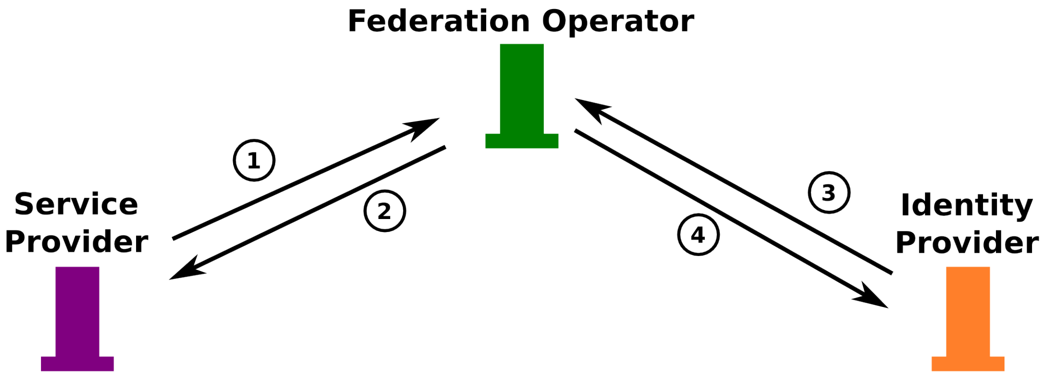


Figure 4.1: A complete cycle of transmitting metadata securely from the Service provider to the Identity provider via a federation operator. Step 1, the service provider initiate an update of metadata. Step 2, the federation operator requests the metadata from some well known location. Step 3, the identity provider updates the local metadata cache when the cache timer has expired by requesting new metadata from the federation operator. Step 4, the federation operator respond with the new metadata. After each response that includes a certificate in some form another request is made to the domain name system for CERT RRs to be able to validate the original certificate.

New questions arises here, is it practical to store SAML2 certificates in the Domain Name System as a CERT RR or are there better alternatives? The second question that comes to mind is more related to DANE. If CERT RRs is used as stated above it would require DNSSEC otherwise its value would be of no use. Is it not possible to just use the TLSA RR with TLS certificates as it is and not introduce another layer that CERT RRs would become within the same system (DNS and DNSSEC)?

Only use TLS (with TLSA RR)

Let's start with the question would it be enough within identity federations to only use TLS with TLSA and not to publish any SAML certificates?

The solution is already at the point where all entities within a federation needs to sign their own zone with DNSSEC as an underlying infrastructure. Instead of publishing SAML2 certificates in the Domain Name System it might be possible to just use TLS with TLSA RRs and deem that all the information communicated between any two entities is valid as long as it's over TLS that has been established with TLSA RRs procedures.

This would mean that when a new entity, let's say a service provider, joins a federation the service provider must send its metadata to the federation operator. As an example the administrator for the service provider might visit a webpage over HTTPS (with validity check for TLSA RR) to initiate the transfer of metadata, step 1 in figure 4.2. The federation operator then opens a new connection over HTTPS (with validity check for TLSA RR) for the metadata from the service provider, step 2 in figure 4.2. This request, from the federation operator back to the service provider, that initiated the communication the first time is to make sure that the metadata that is fetched is from the right service provider. When the metadata is recieved the federation operator can publish it with the metadata for all other entities, step 3 in 4.2

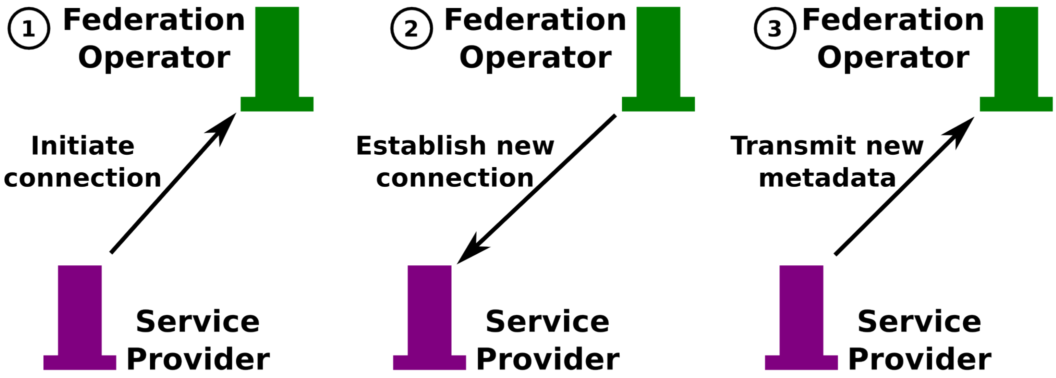


Figure 4.2: A simple procedure to establish first time communication with a new member in an identity federation.

Publishing SAML2 certificate as CERT RR, is it viable?

Earlier in this chapter the question arose if it is practical to store SAML2 certificates in the Domain Name System. Due to time constraints this is out of scope of this report and further investigations has to be done in this area. For further discussions in this report it's assumed that it's a viable solution to store SAML2 certificates in the Domain Name System as CERT RRs.

Using the best of two worlds

It's often the simple solution that is the best one but would it be worth it to use TLSA validated connection for the HTTPS communication together with the SAML2 certificates in CERT RRs as explained in earlier sections?

To simplify the view an argument could be made that the validity check on the SAML2 certficate downloaded over the HTTPS secure channel and then verified against the CERT RR is just another safety check within the same "layer" as the secure channel was established in. The reason for this is that the TLS connection is validated first with information from the DNSSEC infrastructure, when requesting the TLSA RR. The second validation is made depending on the TLSA RR information and local policies, it confirms that the TLS server certificate chain is to be trusted or not. The third validation would be when the SAML2 certificate is compared against the CERT RR available within the DNSSEC infrastructure.

As noted the first and the third validation are both trusting the DNSSEC infrastructure, and therefore the combination of both using TLS with TLSA and SAML2 certificate with CERT RRs would at first glance not add any extra benefits. Though under some circumstances someone might be able to hack into a webserver for some entity and change the certificate used for SAML2 signing. Common production setup is to put the DNS servers on seperate instances than the webserver and if only the webserver is compromised the third validation check against the CERT RR would in this case fail. So in the end the question boils down to if the communication is secure to the correct host, is this enough to trust everything that the host sends?

4.2.2 How to locate the correct certificate in DNS

The last section was about how to distribute the certificate within the Domain Name System and if it's necessary or not. The CERT RR is mentioned as a possible solution and there might be several other solutions as well. If it's decided upon that a TLSA validated TLS connection is not enough to trust the data that is being sent the SAML2 certificates must be published as well in the Domain Name System. With a single or just a few certificates that signs several metadata files for the federation operator it's relatively easy to publish the certificates in the Domain Name System and fetch all certificates based on the domain/subdomain. Though it would put an unwanted limit that the federation operator is only allowed to use a few certificates per DNS zone. Of course it would work with several hundred certificates but it wouldn't be a very good solution. This is because each time some entity would check if one specific certificate is the correct one the requesting entity would have to download all certificates through the Domain Name System and then one by one try to match them against the certificate

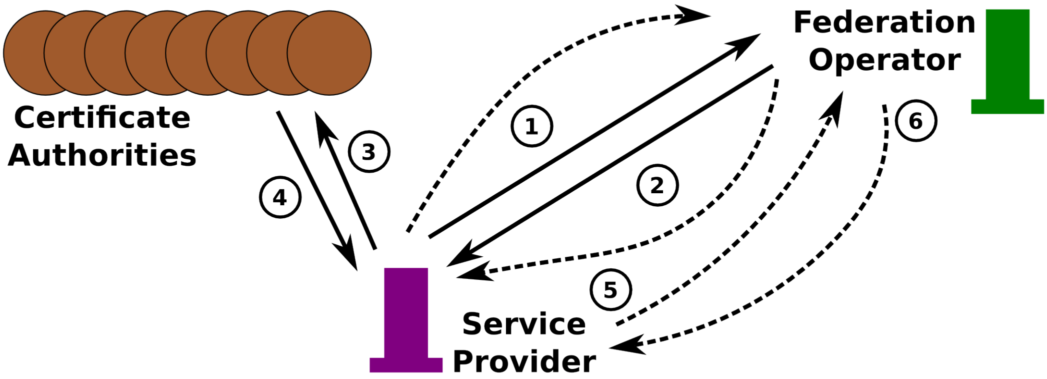


Figure 4.3: Three validation checks divided into 6 different requests/responses. Step 1 and 2 is the initial communication when the TLS communication is established with TLSA. Step 3 and 4 is when local policy demands that the certificate recieved within the metadata has to be validated that it chains to a CA. Step 5 and 6 is the last validation step when the Service Provider requests corresponding CERT reosource records from the federation operator and checks that the certificate used to sign the metadata is available as a CERT RR.

used in the fetched metadata file.

This means that somekind of general algorithm/solution is required to locate the correct certificate published in the Domain Name System when an entity recieves a signed metadata file. One way of doing this might be with the "Dynamic Delegation Discovery System (DDDS)"[26, 27, 28, 29] and the NAPTR resource record[28]. Exactly how this could be done is out of the scope of this report.

The main issue and possible solution in this section was originally introduced by Leif Johansson (SUNET).

4.3 How DANE can be implemented in Shibboleth

DANE as a technology is on the way to becoming a standard within TLS communication and sooner or later also for S/MIME. There is not even a draft available for the use together with certificates concerning other usages than those, therefore there is no way to "implement" DANE for Shibboleth (concerning the SAML2 certificates that is).

As stated in previous chapter the software for the identity provider part is written in Java. It has several extension points where the functionality of different parts can be extended. Due to the number of extension points some reading had to be done to figure out exactly which extension type was best suited for the purpose of a DANE extension.

First possibility was the "Metadata Provider" extension point. The "Metadata Provider" extension point is the part in the identity provider that control the retrieval of metadata. An example to this is the "File Backed HTTP Metadata Provider", it basically fetches the metadata from an URL and stores a cache locally on the server.

The second extension possiblity was a ”Metadata Filter” extension. An example to this one is the ”Required Valid Until” which checks some arguments in the retrieved metadata if the metadata has a ”valid until” argument set not to far in to the future.

Third and fourth extension type that might have worked for the purpose of a DANE implementation are the ”Security Policy” and ”Security Policy Rule” extension types. Both these relates to security concerning individual messages between service and identity providers and not the initial loading of metadata, which in the end ruled these out as viable solutions.

The solution finally chosen to be implemented was a trust engine extension. A trust engine is a part of Shibboleth that processes either metadata or other messages to validate it or dismiss the message. An example trust engine is the ”Static Explicit Key Signature Trust Engine” that takes one or more certificates compares it with signed metadata and confirms that the signature is the right one.

CHAPTER 5

Discussion

5.1 Our conclusions

5.1.1 Identity federations with SAML2 and DANE

We started out with the task to identify where we could improve security within idenitity federations. We specifically looked into the data integrity part of SAML2, how X.509 certificates are used and how it would be possible to improve security with the help of DNS-Based Authentication of Named Entities.

Our conclussion is that as soon as the initial configuration for each entity in an identity federation is done and all entities are up and running, data integrity is as secure as the public/private key cryptography scheme. However, the difficulty is still in the initial setup, how to decide who to trust when you don't trust anyone online. As well as building trust when performing an emergency key rollover, when the key one day unexpectedly becomes invalid. Procedures for this is as of today not specified in any detail, it's up to each federation to decide on their own. Common practice is as said in earlier chapters to share key or certificate information in some out-of-band way e.g. meeting in person or call each other. We belive that it is this initial sharing of information that would benefit the most if it could be done solely over the internet and it's possible if we put trust in the DNSSEC infrastructure.

DANE is one way to go and is a valid solution for identity federations. However, exactly how to store SAML2 certificates and how to retrieve the correct certificate for each validation of metadata is still an open question since we haven't been able to reach that far. We have listed some solutions such as using CERT resource records and perhaps locating the correct certificate with "Dynamic Delegation Discovery System (DDDS)" procedures.

Furthermore, our belief is that within future SAML specifications and/or when establishing new federations, more strict rules concerning metadata must be agreed upon to secure the data integrity of the metadata. Future agreements should include requirements that TLS with TLSA validation is used when sharing metadata. Another requirement should be to publish the certificates within the DNS/DNSSEC infrastructure. This is to make sure that both the channel that the communication is taking part over is secure as well as the data stored on the webserver accessed is the proper one e.g. if downloading a certificate over a HTTPS connection.

5.1.2 How DANE can be implemented in Shibboleth

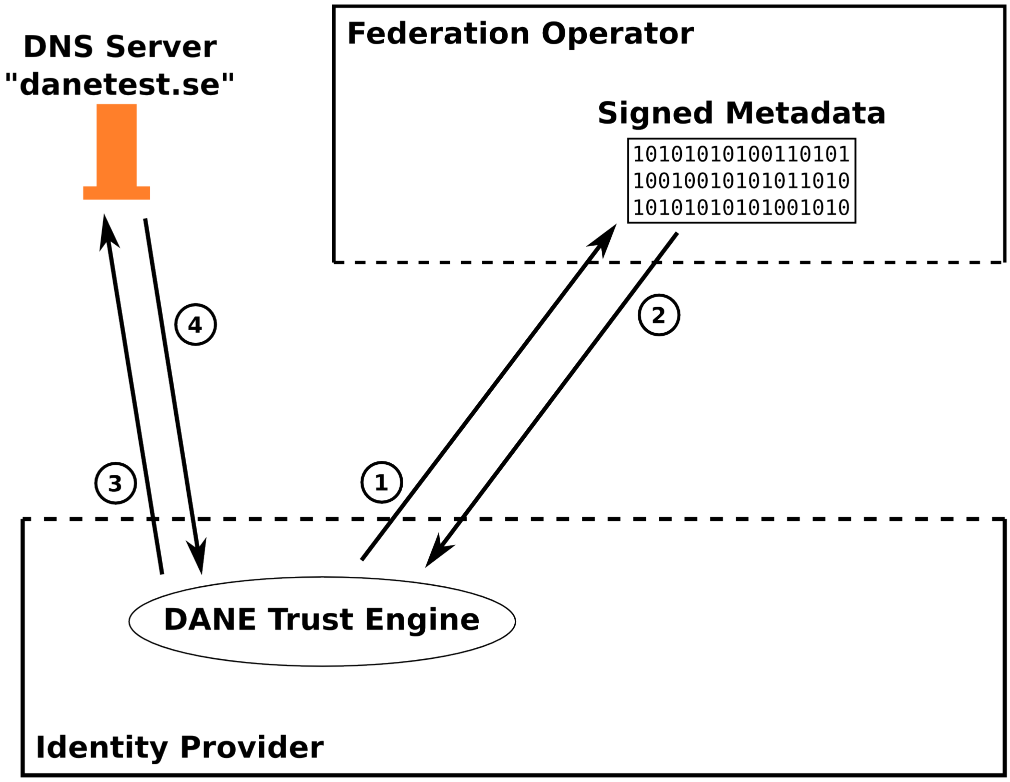

DNS Server
"danetest.se"
Federation Operator
Signed Metadata
10101010100110101
10010010101011010
10101010101001010
④
②
③
①
DANE Trust Engine
Identity Provider

Figure 5.1: Four steps corresponding to the requests/responses vital to the DANE Trust Engine.

To start of at the right level we need to restate that DANE as of today, has no standard nor draft available outside the topic of TLS or S/MIME communication. What we set out to implement is more of a proof-of-concept that when/if there is a standard available for SAML certificates, or perhaps certificates in general, our implementation might be a solution to build upon. It's also important to remember that we have only looked at the identity provider software out of all computer programs available under the Shibboleth name. When/If DANE would be implemented into Shibboleth, development has to be made to all parts that publish and consume metadata.

We started out focusing on developing a metadatafilter extension for Shibboleth. After some digging it became clear that it would be a trust engine extension that was better suited for our purpose.

Our DANE trust engine is intended to be configured from the standard xml-configuration files. The trust engine recieves the metadata as argument (Step 1 and 2 in figure 5.1) and then requests a CERT RR from the domain name system with the DNSSEC flags set (Step 3 in the same figure). If the reponse comes back without DNSSEC flags set the information is not to be trusted otherwise it will compare the certificate retrieved with the one supplied as argument. Depending on the outcome of that comparison the trust engine will either confirm or dismiss trust for the metadata.

Due to time constraints and limited previous knowledge in Java/Spring Framework we were unable to finish the implementation in time. Installation and configuration guides are available in appendix A as well as references to our code available online.

5.2 How to continue with further research

5.2.1 Secure communication with a federation operator

The communication between the federation operator and the other entities such as identity providers and service providers has no standardized way in how it should work. How does the federation operator know that it really is the right identity provider it's talking with, without any prior communication? Should the federation operator be the one that initiate connection to the other entities or vice versa? Last, but not least, how should this be solved technically to minimize the manual labour and automate as much as possible?

5.2.2 Publishing SAML2 certificate as CERT RR, is it viable?

In section 4.2.1 it's mentioned that SAML2 certificates could be stored in CERT RRs. Further research need to be done to establish exactly how this could be accomplished. With CERT RR it's possible to store the certificate in it's full format, only a hash of it or just a url to the location where it's stored[25, ch. 2.1]. What would be suitable for SAML2 certificates and is this really a viable solution for the identity federation scheme?

5.2.3 How to locate the correct certificate in DNS

In section 4.2.2 "Dynamic Delegation Discovery System (DDDS)"[26, 27, 28, 29] and the NAPTR resource record[28] is mentioned as a possible solution to the "matching problem". How does an entity that has recieved a signed metadata file locate the correct certificate within the Domain Name System?

APPENDIX A

Installation and configuration

For this project git together with github has been used for revision control. To be able to keep the installation and configuration instructions up to date even after this report has gone into print we have decided to write a summary of all used software in this appendix and then put all concrete instructions on github.

As git is a distributed version control system there is no specific repository that is our "master" that we can link to. Due to this both of our repositories could be used but to not flood this appendix with redundent/"duplicate" links all links in this appendix will be to the repository available under Christoffer's github account.

This report in LaTeX as well as all of our instructions are available at
`https://github.com/christofferholmstedt/BachelorThesis`

The DANE Trust Engine Extension is available at
`https://github.com/christofferholmstedt/shibIdPextension`

A.1 Bind, OpenDNSSEC and other requirements

To be able to use DANE, DNSSEC needs to be up and running as a prerequisites. In this project Bind as DNS server together with OpenDNSSEC to sign the zone were used.

To sign the zone you need some kind of key store from where OpenDNSSEC can fetch the keys. If a Hardware Security Module is not available SoftHSM can be used for this, also available from the OpenDNSSEC website. (`http://www.opendnssec.org/`)

For user authentication on the identity provider server we have used OpenLDAP [30]. Complete installation and configuration instructions are available at our github repository.

A.2 Shibboleth Service/Identity Provider

We installed the service- and identity provider software on different virtual machines to better be able to mimic a production setup. It should be possible to install them on the same machine but as the service provider use Apache HTTP Server and the identity provider use Apache Tomcat some complications may arise concerning the configuration on which webserver should listen on which ports.

Complete installation and configuration instructions are available at our github repository.

A.3 Federation Operator - Metadata Aggregator

One of the Federation Operator technical purposes is to collect, combine and share the metadata from and to respective service/identity provider. For our testing environment we downloaded the metadata from respective provider to the federation operator. We then got help from Leif Johansson (SUNET) who provided us with a link to the scripts that is used to sign the metadata for identity federation "Skolfederationen".

More information is available at the github repository mentioned above.

A.4 DANE Trust Engine Extension

The code for the DANE Trust Engine Extension is available at github `https://github.com/christofferholmstedt/shibIdPextension`. As of today the code in the master branch compile just fine and Shibboleth loads the extension but it's not possible to activate the extension by configuring it in the 'relying-party.xml' file with Shibboleth. The current code is a copy of the "Static Explicit Key Signature Trust Engine" which is built in with the standard distribution of Shibboleth. Some outcommented code is for doing a DNS lookup tested in a seperate code project and should work when our extension is able to start properly.

Instructions on how to install, configure and continue the development on the extension is available at the git repository.

Bibliography

[1] "This is bankid, financial id-technology." `http://www.bankid.com/en/What-is-BankID/`. Read: April 2012. Unknown author and date of publishing.

[2] S. Jönsson, "E-legitimationsnämnden och svensk e-legitimation." `www.regeringen.se/content/1/c6/15/82/56/74c79ddf.pdf`, December 2010. Read: April 2012.

[3] E-legitimationsnämnden, "E-legitimationsnämnden." `http://www.elegnamnden.se/`, 2012. Visited: May 2012.

[4] R. Barnes (BBN Technologies), "Request for Comments. Use Cases and Requirements for DNS-Based Authentication of Named Entities (DANE). RFC6394." `https://datatracker.ietf.org/doc/rfc6394/`, October 2011. Read: April 2012.

[5] P. Hoffman (VPN Consortium) and J. Schlyter (Kirei AB), "Request for Comments, draft. The DNS-Based Authentication of Named Entities (DANE) Protocol for Transport Layer Security (TLS)." `https://datatracker.ietf.org/doc/draft-ietf-dane-protocol/`, March 2012. Read: April 2012. (This Internet-Draft is still 'work in progress' when writing this report).

[6] P. Hoffman (VPN Consortium) and J. Schlyter (Kirei AB), "Request for Comments, draft. Using Secure DNS to Associate Certificates with Domain Names For S/MIME." `https://datatracker.ietf.org/doc/draft-hoffman-dane-smime/`, March 2012. Read: April 2012. (This Internet-Draft is still 'work in progress' when writing this report).

[7] R. Arends (Telematica Instituut), R. Austein (Internet Systems Consortium), M. Larson (VeriSign, Inc.), D. Massey (Colorado State University), and S. Rose (National Institute for Standards and Technology), "Request for Comments. DNS Security Introduction and Requirements. RFC4033." `https://datatracker.ietf.org/doc/rfc4033/`, March 2005. Read: March 2012.

[8] R. Arends (Telematica Instituut), R. Austein (Internet Systems Consortium), M. Larson (VeriSign, Inc.), D. Massey (Colorado State University), and S. Rose (National Institute for Standards and Technology), "Request for Comments. Resource

Records for the DNS Security Extensions. RFC4034.” `https://datatracker.ietf.org/doc/rfc4034/`, March 2005. Read: March 2012.

[9] R. Arends (Telematica Instituut), R. Austein (Internet Systems Consortium), M. Larson (VeriSign, Inc.), D. Massey (Colorado State University), and S. Rose (National Institute for Standards and Technology), “Request for Comments. Protocol Modifications for the DNS Security Extensions. RFC4035.” `https://datatracker.ietf.org/doc/rfc4035/`, March 2005. Read: March 2012.

[10] M. StJohns, “Request for Comments. Automated Updates of DNS Security (DNSSEC) Trust Anchors. RFC5011.” `https://datatracker.ietf.org/doc/rfc5011/`, September 2007. Read: April 2012.

[11] “Assertions and protocols for the oasis security assertion markup language (saml) v2.0.” `http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf`, March 2005. Read: April 2012.

[12] “Metadata for the oasis security assertion markup language (saml) v2.0.” `http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf`, March 2005. Read: March 2012.

[13] S. Cantor (Ohio State University), “Saml v2.0 metadata interoperability profile.” `http://docs.oasis-open.org/security/saml/Post2.0/sstc-metadata-iop-cs-01.pdf`, August 2009. Read: April 2012.

[14] “Bindings for the oasis security assertion markup language (saml) v2.0.” `http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf`, March 2005. Read: April 2012.

[15] “Profiles for the oasis security assertion markup language (saml) v2.0.” `http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf`, March 2005. Read: March 2012.

[16] “Glossary for the oasis security assertion markup language (saml) v2.0.” `http://docs.oasis-open.org/security/saml/v2.0/saml-glossary-2.0-os.pdf`, March 2005. Read: March 2012.

[17] N. Ragouzis (Enosis Group LLC), J. Hughes (PA Consulting), R. Philpott (EMC Corporation), E. Maler (Sun Microsystems), P. Madsen (NTT), and T. Scavo (NCSA/University of Illinois), “Security assertion markup language (saml) v2.0 technical overview.” `http://www.oasis-open.org/committees/download.php/27819/sstc-saml-tech-overview-2.0-cd-02.pdf`, March 2008. Read: Mars 2012.

[18] T. Nilsson (Certezza AB) and R. Bellgrim (Certezza AB), "Skolfederation.se, teknisk infrastruktur (saml), revision 0.71." `https://www.skolfederation.se/wordpress/wp-content/uploads/2012/03/SE_Skolfederation-Teknisk-infrastruktur-SAML-rev-071.pdf`, April 2012. Read: April 2012.

[19] T. A. S. Foundation, "Apache: SSL/TLS strong encryption: An introduction." `http://httpd.apache.org/docs/2.2/ssl/ssl_intro.html`, 2012. Read: April 2012.

[20] D. Gollmann (Hamburg University of Technology), *Computer Security (Third Edition)*. ISBN: 978-0-470-74115-3, John Wiley and Sons, Ltd, 2011. Read: May 2012.

[21] J. F. Kurose (University of Massachusetts, Amherst) and K. W. Ross (Polytechnic Institute of NYU), *Computer Networking, A Top-Down Approach (International Edition)*. ISBN: 978-0-13-136548-3, Pearson Addison-Wesley, 2010. Read: May 2012.

[22] A. Åkre Solberg (UNINETT), S. Cantor (Ohio State University), E. Maler (Sun Microsystems), L. Johansson (Stockholm University), J. Hodges (Neustar), I. Young, N. Klingenstein, and B. Morgan, "Interoperable saml 2.0 web browser sso deployment profile." `http://saml2int.org/profile`. Read: Mars 2012. Unknown date of publishing.

[23] J. Callas (Network Associates, Inc.), L. Donnerhacke (IKS GmbH), H. Finney (INetwork Associates, Inc.), and R. Thayer (EIS Corporation), "Request for Comments. OpenPGP Message Format. RFC2440." `https://datatracker.ietf.org/doc/rfc2440/`, November 1998. Read: April 2012.

[24] S. Santesson (3xA Security), "Online Certificate Status Protocol Algorithm Agility RFC6277." `http://datatracker.ietf.org/doc/rfc6277/`, June 2011. Read: April 2012.

[25] S. Josefsson, "Request for Comments. Storing Certificates in the Domain Name System (DNS). RFC4398." `https://datatracker.ietf.org/doc/rfc4398/`, March 2006. Read: April 2012.

[26] M. Mealling (VeriSign), "Request for Comments. Dynamic Delegation Discovery System (DDDS), Part One: The Comprehensive DDDS. RFC3401." `https://datatracker.ietf.org/doc/rfc3401/`, October 2002. Read: April 2012.

[27] M. Mealling (VeriSign), "Request for Comments. Dynamic Delegation Discovery System (DDDS), Part Two: The Algorithm. RFC3402." `https://datatracker.ietf.org/doc/rfc3402/`, October 2002. Read: April 2012.

[28] M. Mealling (VeriSign), “Request for Comments. Dynamic Delegation Discovery System (DDDS), Part Three: The Domain Name System (DNS) Database. RFC3403.” `https://datatracker.ietf.org/doc/rfc3403/`, October 2002. Read: April 2012.

[29] M. Mealling (VeriSign), “Request for Comments. Dynamic Delegation Discovery System (DDDS), Part Four: The Uniform Resource Identifiers (URI) Resolution Application. RFC3404.” `https://datatracker.ietf.org/doc/rfc3404/`, October 2002. Read: April 2012.

[30] O. Foundation, “Openldap.” `http://www.openldap.org/`, 2012. Visited: April 2012.