MSc System and Network Engineering
Research project 2:

Calculating metadata propagation time within eduGAIN

Marcel den Reijer
Marcel.denreijer@os3.nl

October 15, 2019

Abstract

There are several issues about SAML metadata. First is that updating the metadata between parties is done manually or automated with a cronjob[1]. Second, the propagation time may be hard to calculate, because determining the updating of the metadata may be hard. Calculating the propagation time may be too hard since one can not tell the difference between manual updating or event-based updating. Studies on calculating the propagation time and exchanging metadata are limited and therefore this research assumes that there are no defined approaches about how to calculate the propagation time and update the SAML metadata event-based. The purpose of this research is to show a way how to calculate the propagation time and a implementation of how to update SAML metadata event-based. An event-based approach will be better in the case of a security emergency e.g. whereby cryptographic keys need to be replaced. One has to wait for manually updating the metadata or wait for that the cronjob starts, and therefore a better approach is to do it event-based.

Contents

1 Introduction

A collection of inter-operating organizations that agree under a certain rule set is called an "identity" federation. This rule sets typically consist of technical profiles, standards, and policies that provide the trust and security to exchange identity information to get access to third-party services[2]. Exchanging identity information is often done with Security Assertion Markup Language (SAML), Oauth or OpenID[3]. The eduGAIN project is an initiative that interconnects research and educational institutes with a "full mesh" identity federation around the world as shown in figure 1[4].

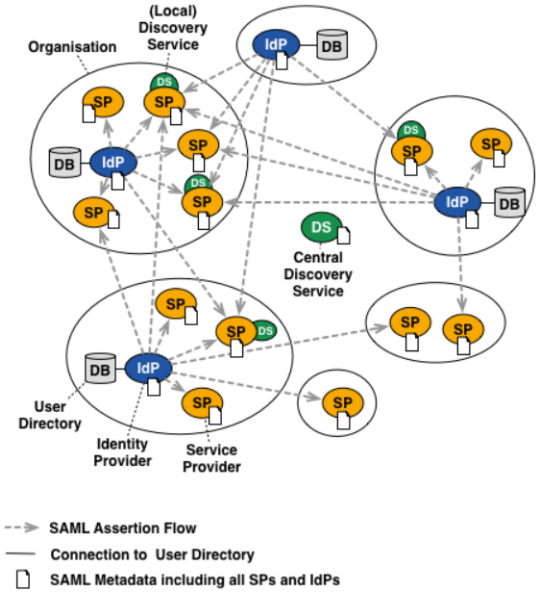

Figure 1: The implemented federation architecture by eduGAIN[4]

Mesh identity federations and the eduGAIN inter-federation Service build their trust by using SAML (explained in section 2) to limit the audience to known actors. Every organization (research or educational institute) is connected to a so-called home federation. To inter-operate securely, every federation depends on exchanging SAML metadata (sometime referred as just metadata). To inter-operate securely, Every metadata file contains the following attributes. Mandatory attributes are entity ID, cryptography keys and protocol endpoints[5].

1.1 Problem statement

The problem statement is that current SAML implementations lacks in updating the metadata event-based[1]. Current SAML implementations e.g. openSAMLphp or PyFF, aggregate, generate or download the metadata of the other party at certain times with a cronjob in the background. At the moment the SAML metadata change at the other party, it takes time till the cronjob is running before the metadata will be updated. In SAML context an Identity Providers (IdPs) is the party that authenticates users and issues identity assertions. An Service Providers (SPs) is a party which evaluates these identity assertions from an IdP to give access to the authenticated user[2]. Responding to security threats, key rollover or even updates to service configuration and attribute release information can be achieved with changes in the metadata configuration of an SP or IdP. The time that it takes for a configuration to flow from the IdP/SP via their home federation, through an inter-federations service such as eduGAIN, and on to other IdPs/SPs, is an important factor in ensuring a consistent configuration throughout the environment. The propagation time of exchanging the metadata is therefore important to know, because different parties may still use old metadata in a case of a compromised IdP or SP.

1.2 Scope

The primary goal of this research is to design a method/approach or a protocol to exchange SAML metadata event-based and calculate the propagation time in a full meshed identity federation. But this research will not cover a method for continuously detecting updates to metadata. It shows only a way to a solution for calculation the propagation time of metadata and finds a way to update SAML metadata event-based.

2 Background information about SAML

2.1 SAML Architecture

Security Assertion Markup Language (SAML) is defined (by OASIS) as a framework for exchanging security-related information between trusted parties based on XML. Within the SAML specification, there are three entities defined as explained in the introduction. These entities are IdP, SP and the end-user. SAML consists of different components as shown in figure 2[6].

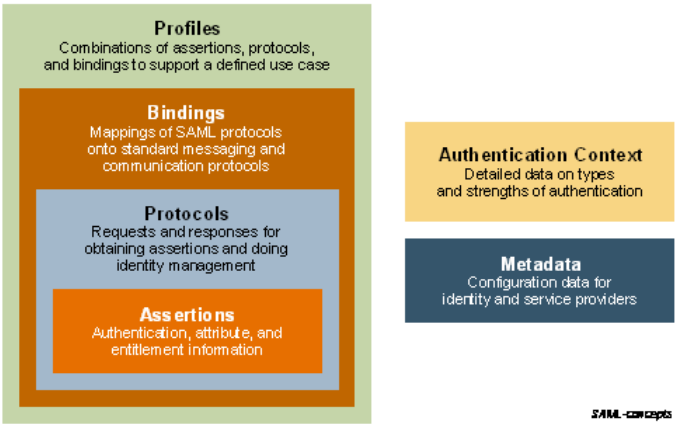

Figure 2: SAML Architecture - Basic Concept[6]

The first component is called "Profiles". Profile defines a certain rule set for the usage of the SAML syntax for sending security information. The second component is called "Bindings". Bindings define how SAML assertions can be exchanged, using underlying communication protocols e.g HTTP. The third and fourth component is called "protocols" and "Assertions". These components define the semantics and syntax for XML-encoded assertions. These assertions describe authentication, authorization and attribute information. The fifth component is called "Authentication Context". Authentication Context describes different kinds of authentication mechanisms by defining a syntax for describing authentication context declarations. The last component is called "Metadata". As said in the introduction, the metadata defines how a SAML entity described its configuration data e.g. cryptographic Keys for signing/verifying, service endpoint URLs, service configuration or attribute release information.

2.2 Protocol Flow

As said in the introduction if this paper SAML is used in a federated identity, but SAML has the ability for Single Sign-On. The SAML protocol uses first an Authentication request together with a supported front-channel binding e.g. HTTP Redirect or HTTP POST. Figure 3. shows how SAML works[7].

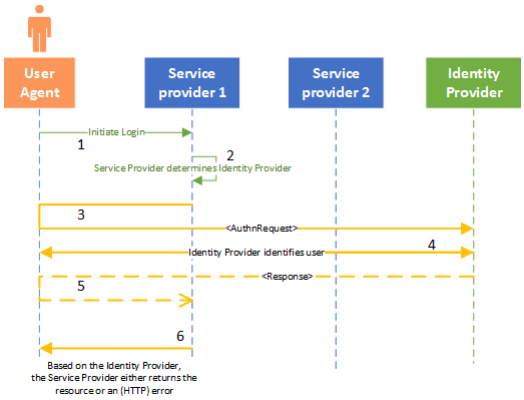

Figure 3: SAML 2.0 - Single Sign On [7][8]

According to the SAML specifications[7], the first message which is sent from the user agent to an SP is a login request. The SP determines the IdP and the binding that can be used by performing a lookup into the metadata of that IdP. The SP sends after the lookup an authentication request and redirects the user agent to the inlog portal of the IdP. The user has to authenticate himself to the IdP and the IdP redirects the user-agent back to the SP. The last step is that the SP may send a status code to the user agent based on the IdP.

Figure 3. shows that the first message is an "AuthnRequest" message. This message contains the, for example, an "Issuer" element, which contains a unique ID for the requested SP. The "AuthnRequest" message has to be accepted by the IdP OR SP or otherwise, an HTTP error status code will be sent back[7].

3 Previous research

Studies on propagation time within identity federations are limited in the literature, but Alex Stuart researched this subject in 2018[9], within the UK federation (the home federation of the United Kingdom and a member of eduGAIN). Stuart proposed a method for measuring the propagation time from the metadata of SPs to IdPs using SAML2.0 "AuthnRequest" messages. These kinds of messages probes whether updated metadata has been read by an IdP or not. The way SAML2.0 works is by sending an "Authnrequest" message to the IdP. The IdP will accept and respond successfully if the metadata of the probing SP contains the Issuer "entityID" and "AssertionConsumerServiceURL" attributes. The method proposed by Stuart does not rely on specific details of the UK Federation, but it is generalized in a way that his method could be applied in eduGAIN as well. The step taken in his method is as follows: Firstly, for each run of the probe, a new SP has to be registered. Secondly, a new endpoint has

to be added to the SP by deploying a new SP instance. Thirdly, a handler URL is added.

4 Research question

According to the introduction, the main research question will be:

How to calculate the propagation time of metadata throughout SAML identity federations within eduGAIN?

This main research question is divided into the following sub-questions:

- *Can manual vs automatic metadata updates be detected by looking at metadata propagation times?*
- *What levels of cohesion can be found within federations?*
- *What should be the design requirements for updating SAML event-based*

5 Metadata

5.1 Manual vs Automatic detection

Detection if the metadata is updated whether manually or automatically is pretty hard to decide. One approach is to detect it can be done as follows: For every member, every x time frame minutes the aggregated metadata file of eduGAIN or a home federation have to be downloaded. For every file, a hash is created and compared to previous hashes based on their timestamps. When the hash is changed one knows when and how often the metadata of all members has changed. Looking to "change frequencies", one can determine the metadata file is changed automatically. This can be done when the frequencies have a certain pattern. Determining the detection of updating the SAML metadata file manually is very hard since it could also be event driven[10]. The frequencies pattern should be in this case irregular en event driven could also be manually or automatically. This approach assumes that the metadata only changes when the real content change. Even a change in the expiration date could change the hash without that the real data has changed. Therefore a better approach will be detecting differences in metadata by downloading it every x time frame. One can compare the actual data with previous data and may conclude the real data has changed. But still determining if it's done manually is too hard.

5.2 Cohesions of IdPs and SPs

Home federations contain usually two metadata files. The first metadata file is in this research so-called a "local" metadata file and this consists of all the IdPs and SPs of research facilities or institutes, which are a member of that

home federation. The second metadata file is a subset of the local metadata file and published on eduGAIN. The so-called "published" metadata file. There is always a third metadata file maintained by eduGAIN. This is an aggregated version of all published XML files of all members of eduGAIN. Formally, eduGAIN knows only about the IdPs and SPs published by the home federations. Determining the levels of cohesions, the local SAML XML metadata files are compared with the published XML metadata files on eduGAIN and the eduGAIN aggregated XML metadata file. The published XML metadata file of each member is a subset of the local SAML XML metadata file. Every XML file contains the following attributes: "<md:SPSSODescriptor>" which determines an SP and "<md:IDPSSODescriptor>" which determines an IdP[11]. Using regular expressions on every XML metadata-file allows one to count how many IdPs and SPs are declared. The result of this can be used to calculate the coverage of all IdPs and SPs within the members of eduGAIN. This section shows the number of IdPs and SPs in total per type of metadata file. Counting the IdPs and SPs is important to correlate the propagation time with the number of IdPs and SPs. When one looks to the numbers of SPs and IdPs as listened in table 1., eduGAIN knows about 15.041% of all SPs and 27.932% of al IdPs comparing the differences between the local en published XML metadata files. For all the published SPs eduGAIN has aggregated 99.558% of them all and it is also noticeable that eduGAIN has 2.406% more IdPs aggregated then published by comparing the eduGAIN XML metadata file and the published XML metadata file. The differences between the amount of IdPs and SPs in different files is because home federations and eduGAIN add some IdPs or SPs manually. Appendix A. shows a full list of the level of cohesion per member. In this appendix, one can see also a coverage level in percentages. It seems that some members use the same published XML metadata file as for the local XML metadata file. This can be concluded by counting the same IdPs and SPs in both files and use XMLdiff to see the differences between these files.

	SPs	**IdPs**
Local XML file	16561	10862
Published XML file	2491	3034
eduGAIN XML file	2480	3107

Table 1: An overview of all SPs and IdPs within the local, published and eduGAIN XML metadata file.

5.3 Calculating propagation time

In this research, the propagation time is used to denote the time for an IdP or SP to distribute the metadata to other IdPs or SPs in the case also through home federations and eduGAIN. The propagation time is calculated as follows. First, the transmission time of the complete metadata file should be calculated. The formula for calculating the transmission time in seconds is as follow[12]:

$transmissiontime = filesize/Bitrate$

Where the file size is the size of the metadata file in bits divided by the bit rate. Let us consider a file with a size of 41 MB and one download it with a bit rate of 50Mb/s. Then the result with 10% overhead will be $42991616 \times 8/(50 \times 10^6) * 1.1 = 7.6sec$. When one knows the file transmission time, one has also added the processing time of aggregating all metadata data together. The processing time is the time taken by the aggregating process from the first CPU cycle until the last CPU cycle. The total formula for calculation the propagation time within eduGAIN will be as follow:

$Propagationtime = t_1 + p_1 + t_2 + p_2 + t_3$

where t_1 is the transmission time between the IdP or SP which initiated to the home federation. p_1 is the procession time of aggregating all SAML metadata by the home federation. t_2 is the transmission time between the home federation and eduGAIN. p_2 is the time of aggregating the SAML metadata by eduGAIN. The last variable is t_3 and this is the transmission time of IdPs or SPs.

5.3.1 Processing time

Figure 4. shows the processing time (pull-based) in seconds of the eduGAIN SAML metadata over 20 times aggregating all metadata files of all members, which was in total 5587 IdPs and SPs (2480 SPs+3107IdPs). PyFF (a python library used by eduGAIN for pulling metadata from home federations and aggregate them) takes an average time of 35.32 seconds to propagate. Figure 4. shows also the processing time when the number of IdPs and SP are 2343. The processing time of this amount has an average of 19,27 seconds and for 1735 IdPs and SPs 15,05 seconds. Even if the amount of processing is too low, one can see a correlation between the time of processing the metadata and the number of IdPs and SPs.

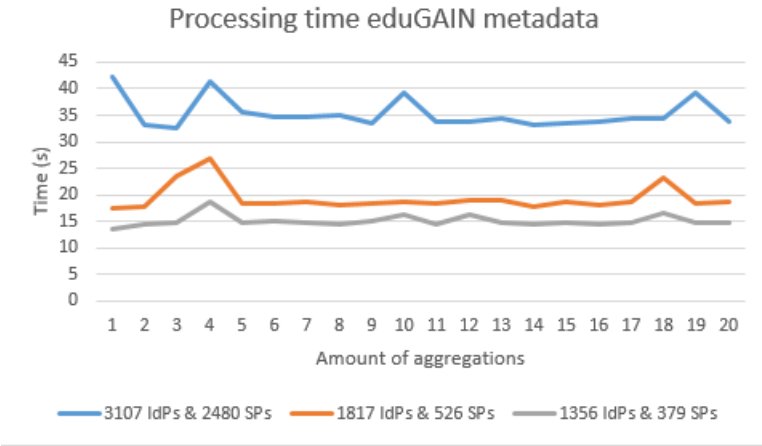

Figure 4: Propagation time eduGAIN XML metadata calculated on a Dell Poweredge R210 system with Intel Xeon l3426 1.86 GHz, 8GB RAM 1066 MHz and 500GB hdd WD-Re3-WD5002ABYS 7.200rpm.

6 Protocol design

6.1 Protocol requirements

This section is going about the minimal requirements that do resolve at least the problem statements. The minimal requirements are a way to update the metadata automatically based on events and calculating the propagation time. There are two ways to exchange the metadata between two parties. These pull and push based[13]. The difference between them is that one pushes data and other pull data. Based on the intention of an IdP or SP, the home federation and eduGAIN only accept members to its federations. When a push-based approach will be used, there will be a need for authentication[14]. When the home federation and eduGAIN have white-listed their members a pull-based approach will e better so we do not need to think about authentication and stuff like that, because home federation and eduGAIN will pull the metadata. They may know an update of an IdP or SP sends them a notification. Besides a pull-based approach, there are certain data which need also to be sent to calculate the propagation time. This data are the transmission time and aggregating time. Eventually, the transport may also be secured with the latest TLS version.

6.2 Proposed protocol design

The protocol as shown in figure 5. will be proposed as follow and is kept as simple as possible. The SAML party who wants to update its metadata sends a notification to the other SAML party. The other party will pull the SAML

metadata from the party who initiated an update notification and sends back an acknowledgment with a timestamp about who long it took before pulling the metadata was completed. The loopback pointer determines the continuous running process of looking whether the metadata has been updated or not.

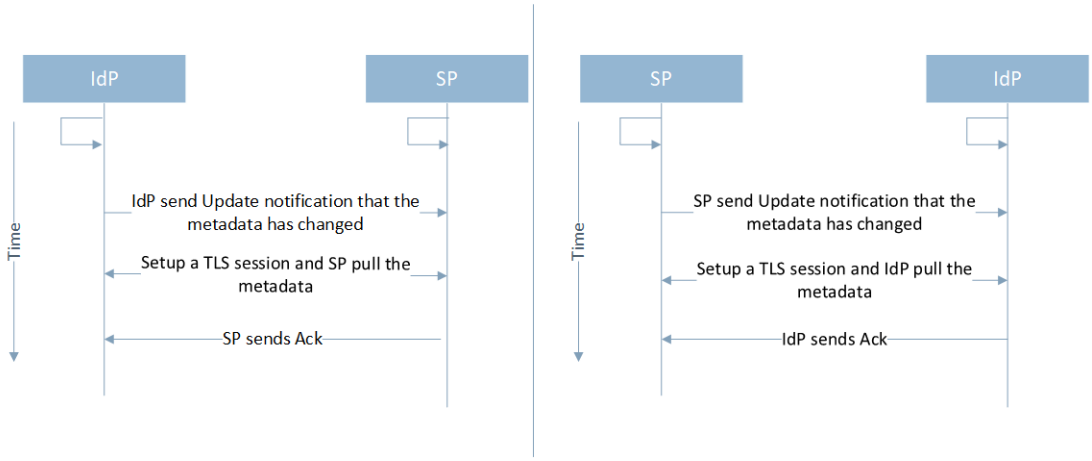

Figure 5: Protocol flow updating metadata between IdP/SP and SP/IdP

Figure 6. shows the acknowledge packet and the pull packet. The Acknowledgements contains the transmission time of pulling the metadata file and a hash. The pull packets consist of multiple fields and are again not mandatory. It is only mandatory for those who want to know the propagation time. Every party will add transmission time and processing time. In the end, eduGAIN will be able to calculate the propagation time.

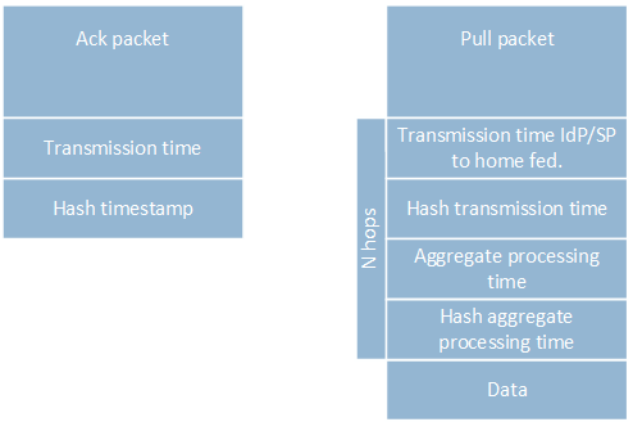

Figure 6: Pull Ack packet

6.2.1 Implementation within eduGAIN

The implementation of the SAML metadata exchange protocol will slightly different since there are some side backs. First, the metadata goes from IdPs/ SPs upstream. This means that the metadata goes from IdP/SP to the home federation, from home federation to eduGAIN and back to IdPs/SPs. There are also IdPs and SPs which are not advertised to eduGAIN. The implementation will be proposed as shown in figure 7. The flow is as follow:

- First, an IdP or SP sends an update notification to its home federation after it has detected that its metadata has changed.
- Second, a TLS session has been established and the certain home federation will pull the metadata.
- Third, the home federation will send back an acknowledge time with the transmission time. After this, the home federation will process the new aggregated metadata.
- fourth, the home federation will broadcast a pull notification to all Idp M/SP M (which are the IdPs and SPs who are a member of the home federation. IdP M/ SP M are the IdPs and SPs just like IdP N or SP N but are not advertised to eduGAIN. Since the home federation has aggregated the new metadata file, it sends at the same time an update notification to eduGAIN.
- fifth, the home federation and IdP M/SPs M set up a TLS session so that the IdP M/ SP M pull the metadata from their home federation. At the same time, there is also a TLS session between the home federation and eduGAIN and eduGAIN may pull the metadata from the home federation. After this step eduGAIN aggregate the new metadata.

- sixth, eduGAIN sends an update notification back to IdPs N and SP N. IdPs N and SP N setup a TLS session with eduGAIN and pull the new metadata and sends an acknowledge back with the time of transmission between these parties.

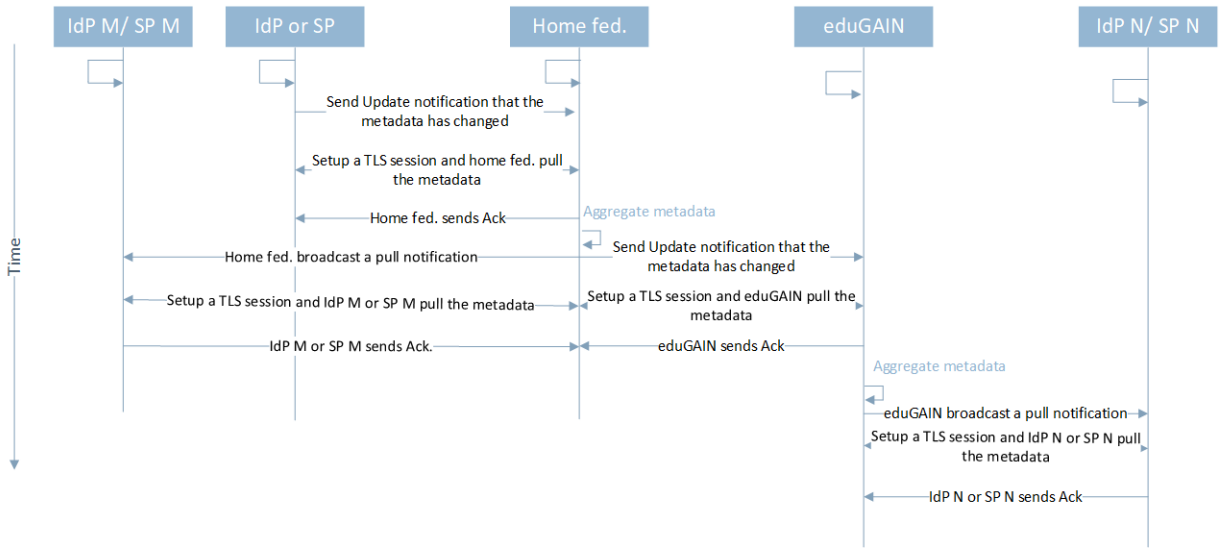

Figure 7: Protocol flow implementation in eduGAIN

7 Discussion

The implementation or design of exchanging metadata has several issues. The requirements as described in this research were not taken into consideration about application security. Every time the next hop, for example, the home federation may change the transmission time between the IdPs/ SPs and the home federation itself. A federation builds upon trust, but it is possible in this design to change the times so that eduGAIN gets the wrong information to calculate the propagation time. Another issue of this design is at the moment when two IdPs or SPs updating the metadata from the same home federation at the same time. The same counts for two home federations to eduGAIN. The way how the propagation time is also not precise. The propagation time is more roughly calculated because other delays were not taken into account. For example, queuing delays or the delays between when the metadata file is received and the start of the aggregating process. The last consideration of applying a TLS session needs to be done because the SAML metadata has been signed. So when it changed the signature does not match anymore and since the data is considered is public data it is not needed.

8 Conclusion

This research shows just a design of how SAML metadata can be updated automatically based on events and how to calculate roughly the propagation time. To answer the main research question will be as follow: "*How to calculate the propagation time of metadata throughout SAMLidentity federations within eduGAIN?*". The protocol design exists in four stages. The first stage and not discovered in this research is that a process continues checks if the SAML metadata has been changed, no matter if it is the metadata of an IdP, SP, home federation or eduGAIN. The second stage is that a notification has been sent to the other party in the federation. The third stage is setting up a TLS session for the transport security and pulling the metadata from the SAML party who initiated update notification. The last stage is sending an acknowledgment back with the transmission time.

9 Future perspectives

This research area can be extended for example by investigating the following three subjects. The first subject is researching if one can do via external assessment of metadata exchange, clashing different versions of metadata. The idea of this research is to investigate whether it is possible to use older metadata files to downgrade security policies. The second subject which can be done is calculating the propagation time in an environment where every party has implemented the Metadata Query Protocol (MDQ). When home federations will become a member of eduGAIN, the eduGAIN metadata file will grow. MDQ is a dynamical way to get certain data out of SAML XML metadata files and so it is not needed the download one big file. This may save bandwidth and it is imaginable that certain countries with low bandwidth want to use a protocol like this, therefore a research when using a protocol like this is also needed. The last subject may be researching if and what bilateral agreement may be exposed by looking at metadata exchange. Since every research institute may set up federations, they can set up direct federations with other facilities and not through eduGAIN. Via this way, security policies of eduGAIN may be bypassed and there is less trust. In a case where an IdP or SP may be compromised, the IdP or SP maybe bypass the security policies and updating the metadata or block the IdP or SP may likely be forgotten. A nice idea or result of this research will be a visually mapped graph of federations.

References

[1] simplesamlphp. Automated metadata management. https://simplesamlphp.org/docs/stable/simplesamlphp-automated_metadata, 2019.

[2] A. Todosijevic. Identity federations and edugain. https://wiki.geant.org/display/eduGAIN/Identity+Federations+and+eduGAIN, September 2017.

[3] N. Heijmink. Secure single sign-on. https://www.ru.nl/publish/pages/769526/z_researchpaper_sso_final_nick_heijmink_s4250559.pdf, July 2015.

[4] S. Andrėj. Federation architectures. https://wiki.geant.org/display/eduGAIN/Federation+Architectures, September 2017.

[5] Wikipedia. Saml metadata. https://en.wikipedia.org/wiki/SAML_Metadata, 2019.

[6] OASIS. Saml v2.0 technical overview. https://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-tech-overview-2.0.html, 2008.

[7] OASIS. Saml v2.0 profiles. https://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf, 2005.

[8] M. den Reijer F. Makioui. Thinking in possibilities for federated log out. https://www.os3.nl/_media/2016-2017/courses/rp1/p07_report.pdf, 2017.

[9] A. Stuart. Measuring metadata propagation in the uk federation. https://github.com/alexstuart/MetadataPropagation, June 2018.

[10] A. Bakker. Feedback research project. https://webmail.os3.nl/?_task=mail&_mbox=INBOX, 2019.

[11] R. Horbe. Saml v2.0 metadata guide. https://www.oasis-open.org/committees/download.php/51890/SAML%20MD%20simplified%20overview.pdf, January 2014.

[12] Wikipedia. Transmission time. https://en.wikipedia.org/wiki/Transmission_time, 2019.

[13] J. Spacey. Pull vs. push technology. https://simplicable.com/new/pull-vs-push-technology, 2017.

[14] J. Poole. Thoughts on push vs pull architectures. https://medium.com/@_JeffPoole/thoughts-on-push-vs-pull-architectures-666f1eab20c2, 2018.

Appendix A: Statistics about the level of cohesion

Members of eduGAIN	Local XML (SPs)	Published XML (SPs)	Local XML (IDPs)	Publishe
Algeria/ARNaai	2	2	2	
Argentina/MATE	3	0	6	
Armenia/AFIRE	3	3	1	
Australia/AAF	224	3	49	
Austria/ACOnet	n/a	n/a	n/a	
Belarus/FEBAS	0	0	1	
Belgium/Belnet	121	1	43	
Brazil/CAFe	116	0	253	
Canada/CAF	55	5	72	
Chile/COFRe	33	8	6	
Croatia/AAI@eduHr	n/a	n/a	n/a	
China/CARSI	36	2	85	
Cyprus/CyNet	0	0	2	
Czech/eduID.cz	208	18	136	
Denmark/WAYF	16	16	61	
Ecuador/MINGA	0	0	2	
Estonia/TAAT	n/a	n/a	n/a	
Finland/HAKA	350	13	53	
France/FER	1184	69	296	
Georgia/GIF	1	1	2	
Germany/DFN AAI	259	114	286	
Greece/Grena	78	13	102	
Hong Kong/HKAF	8	1	9	
Hungary/eduid.hu	155	49	37	
India/INFED	25	6	42	
Iran/IRFED	1	1	2	
Ireland/Edugate	214	13	46	
Israel/IUCC	n/a	n/a	n/a	
Italy/IDEM	119	37	100	
Japan/GakuNin	226	28	161	
Korea/KAFE	27	0	16	
Latvia/LAIFE	41	1	18	
Lithuania/LITNET	34	1	18	
Luxembourg/eduID	n/a	n/a	n/a	
Malaysia/SIFULAN	10	2	7	
Moldova/LEAF	3	3	2	
Morocco/eduIDM.ma	n/a	n/a	n/a	
Norway/FEIDE	7	7	1	
Oman/Oman KID	0	0	2	
Oman/OMREN	3	3	17	
Pakistan/PKIFED	1	0	4	
Poland/PIONIER	10	1	13	
Portugal/RCTSaai	1	1	26	
Russia/FEDUrus	14	0	10	
Russia/RUNNET	15 2	2	4	
Sigapore/SGAF	7	0	2	
Slovenia/ArnesAAI	252	4	37	
South Africa/SAFIRE	18	3	38	
SPAIN/SIR	110	3	1	
Sri Lanka/LIAF	4	1	9	
Sweden/SWAMID	2188	40	2437	
Switzerland/SWITCHaai	1490	20	66	