Is SAML An Effective Framework For Secure SSO?

Category: Security Technology – Secure Access And Defenses

Vinayendra Nataraja

Foundations of Information Assurance - IA 5010

December 2nd 2012

TABLE OF CONTENTS

FIGURES

Introduction

What Is Single Sign-On?

Single sign-on is a process that allows network users to access all authorized network resources without having to log in separately to each resource. When a request is made by the user to view the content of the service provider, it checks for authorization of the request with a trusted entity known as Identity provider. The identity provider identifies the user and validates the request and if successful then allows the service provider to perform the necessary action.

With the emergence of Web 2.0 bubble, the number of sites requiring password as a login authentication has increased drastically. Each website or web application requires the user to remember a password associated to each account. It is very difficult for a human brain to memorize so many different passwords and the user is forced to register with a same password to many websites or write down all passwords on a piece of paper. This causes a huge security hassle as a compromise on a single account or the confidential paper with passwords, would mean that the sensitive data from all the accounts would be at risk. Therefore the current authorization situation is highly unsustainable. Thus there was a need for a service, which stores only one single password for all systems. Single sign-on helps users by seamlessly authenticating all the applications by using protocols such as SAML.

Single Sign On has been widely accepted by various sectors of the industry but especially in the cloud based Software-as-a-service (SaaS) applications. Single Sign on is being used in the healthcare industry as it automates the login process, enabling clinicians to login only once to their desktop in order to gain fast access to all their applications – removing clicks, keystrokes, and complex passwords. Cost savings are based on 9.51 minutes of time clinicians save, on average, every day through simplified access to mission critical applications and patient files. This saves of about $2,675 per year per clinician. (Ponemon Institute© 2011)

What Is SAML?

SAML stands for Security Assertion Markup Language**.** It is a XML based protocol used to exchanging authentication information between identity provider and service provider. SAML protocol specifies extensible and flexible set of data formats, which is used for communication of user authentication between the parties in a distributed network. SAML support also appears in major application server products and is commonly found among Web services management and security vendors. (OASIS 2007)

History And Versions Of SAML

SAML v1.0 was the first version released in November 2002. A revision was of v1.0 was v1.1, which

was released in September 2003 but only small differences. It has seen significant success and gained momentum in all segments of the Internet including financial institutions, higher education, government and other industries.

SAML v2.0 built on the success of v1.1 had very significant changes. Although the two standards address the same use case, SAML 2.0 is incompatible (on the wire) with its predecessor. (OASIS 2007)

SAML Concepts And Architecture

The SAML protocol consists of assertions, protocols, bindings, and profiles

1. *Assertions*

   An assertion is a package of information that supplies one or more statements made by a SAML authority. SAML defines three different kinds of assertion statement that can be created by a SAML authority.

   - Authentication: This indicates that a subject was authenticated at a particular time using a particular method
   - Attribute: Contains some properties about the subject such group and role information
   - Authorization decision: Information about the request to access the specified subject is granted or denied.

2. *Protocols*

   SAML defines set of request and response messages in XML format that by service providers to obtain assertions.

3. *Bindings*

   This defines how the messages are communicated within SOAP the messages.

4. *Profiles*

   Profile of SAML defines constraints and/or extensions in support of the usage of SAML for a particular application

5. *Metadata and Authentication Context*

   These contain details of how the information about the configuration is shared between two communication entities. Also it contains more information about the authenticity of the data. (OASIS 2007)

Working Of SAML

SAML is a protocol that uses XML to transfer information between two parties.

It has more details about the status of authentication of users, what kind of rights and roles the user has and how can the users use the data, depending on their rights and roles. The working of SAML depends on a series of communication between the identity provider and the service provider.

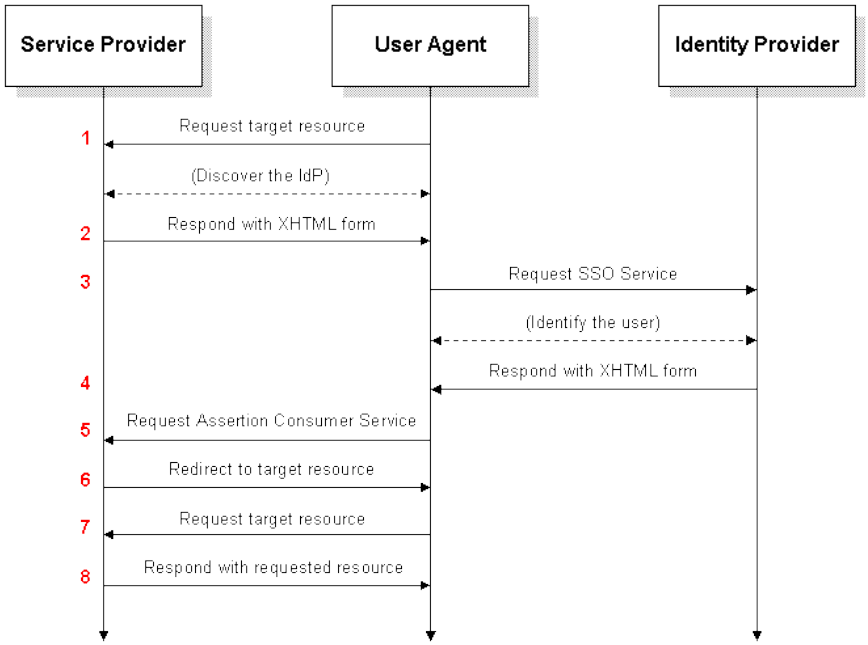

Figure 1: Working of SAML

Source: Tom Scavo, "http://upload.wikimedia.org/wikipedia/en/0/04/Saml2-browser-sso-redirect-post.png"

Here is a general working of SAML Single Sign-On.

- Suppose user wants to access the site of service provider. First the service provider checks if the user is already authenticated, if not it redirects the user to the identity provider to prove the identity.
- The identity provider first verifies the identity of the user and then using the password that the user provides, authenticates the user.
- It then passes the authentication information back to the identifier, which verifies if indeed the information sent is from the identity provider.
- Once the service provider gets the information about the user, it creates a session for the user and displays the requested information.

Acceptance Of SAML As A Framework

SAML framework is widely used and according to a study almost 5000 organizations have invested in SAML solutions. Automated sign-on is business critical to SaaS vendors serving large and mid-sized customer organizations. It brings significant and measurable improvements to the most important objectives of a SaaS business: driving engagement and usage, fighting churn and minimizing unit costs. There are many SAML Open Source Implementations namely Enterprise Sign On Engine, simpleSAMLphp, Lasso, OpenSSO, OpenSAML, Shibboleth, SourceID etc. (emillion 2012)

How Are Web Services Secured In SAML?

The communications between the parties are subjected to variety of threats. The nature of the threat depends on many factors such as the type of communication, the type of systems that are used for communication, the protocol used to communicate the information etc. There is a need to safeguard all these threats and implement a safe communication between the parties. SAML is intended to communicate the authentication information in a safe manner and protect against unauthorized usage. System entities use the SAML protocols for constructing and exchanging security assertions.

Use of Token for exchange of information

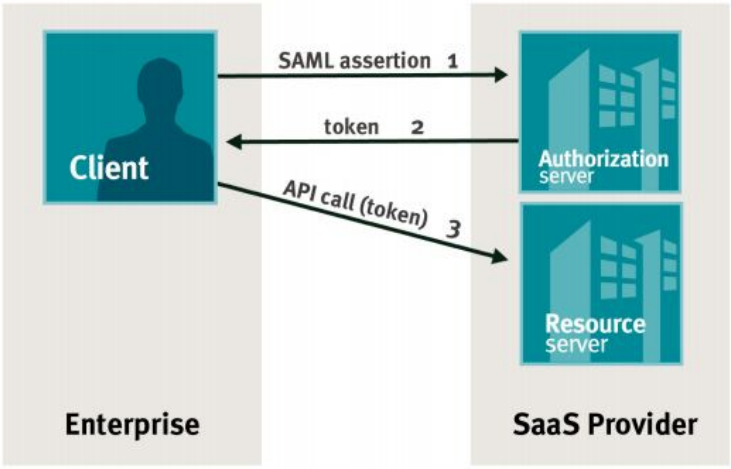

Figure 2: Exchange of Tokens, Source: Swapnil, "http://blog.nexright.com/"

The above diagram shows how exchange of information between the SaaS provider or Service provider happens by the use of tokens. There are three types of tokens that are defined in SAML

1. *UsernameToken Profile*

The UsernameToken profile defines how the username and password can be encapsulated into a security token. The profile defines two different ways to transmit the password:

- Password Text: the password is sent in clear-text
- Password Digest: a digest derived from the password is sent

In order to communicate with an existing Web Service the service provider can use Password Digest that requires the additional parameters such as Nonce, Created date, etc in the Security Header. The Service providers can just use the password to be sent as plain text but it might cause a security concern. (OASIS 2003)

2. *X.509 Certificate Token Profile*

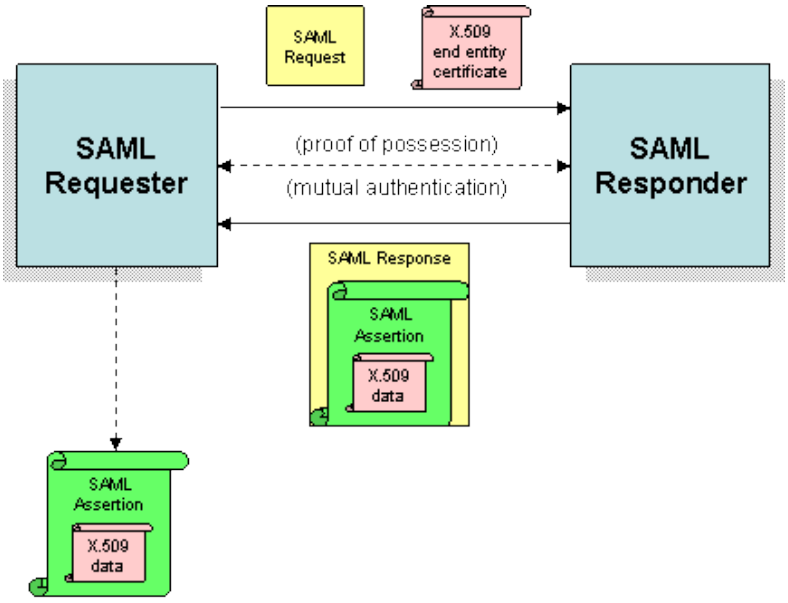

Figure 3: Exchange of X.509 certificate profile token

Source: globus.org, "http://dev.globus.org/wiki/SAMLHoKAssertionRequest"

An X.509 certificate is binding of an identifier and a public key, verified by the registration authority, and signed by a certificate authority. A user (end entity) creates a public/private key pair and sends the unsigned certificate (often called a certificate signing request) to the certification authority. The user is optionally vetted to ensure he/she is who he/she claims to be by the registration authority. Once assured, the certificate authority signs the certificate and returns it to the user. At some point down the road, the user's certificate might be compromised. He/she reports the issue and the certificate is added to a certificate revocation list (CRL). This blacklist is published in a repository that may be queried by other end entities using the serial number on the certificate. (Stallings and Brown 2011)

3. *SAML Token Profile*

SAML Token Profile uses three subject confirmation methods:

- Bearer
- Holder-of-key
- Sender-Voucher

SAML with bearer confirmation offers a few additional advantages over the plain old username token. Foremost, the assertion can contain not just a username (subject) but also a bundle of attributes that can further serve to identify the user, define users' roles, or be otherwise consumed by the service. In addition, the

assertion does capture the notion of what entity is issuing the assertion (asserting the user identity). Lastly, some SOA stacks may not be able to handle a username token with no password.

The identity provider being the Holder-of-key encrypts the outgoing SOAP envelope using the public key of SOAP client. Then it places it in SAML token and signs it with its own private key. Next, the SOAP client adds that SAML token to the SOAP header and constructs the SOAP body, signs it with its own key, and makes a SOAP client call. The service provider validates the SAML token by using the public key of the SOAP client.

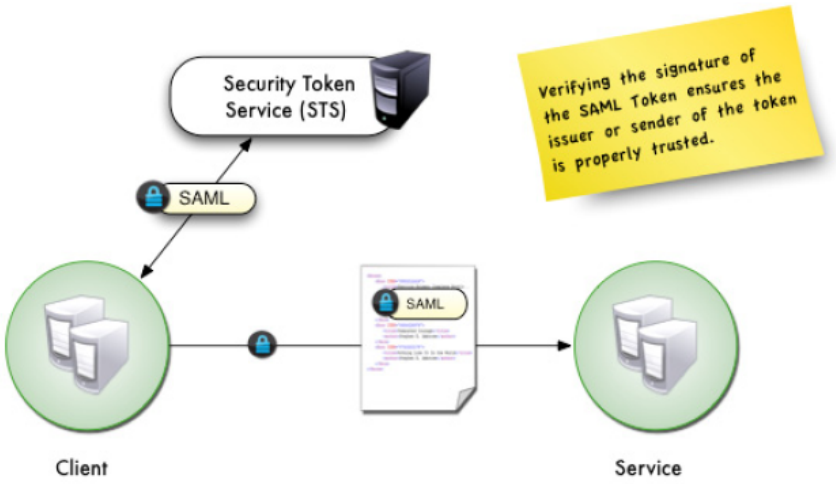

Figure 4: Exchange of SAML token, Source: Pratap Reddy Pilaka, "http://pratapreddypilaka.blogspot.com/2012/01/basics-of-saml.html"

There are two types of certificates involved in SAML Sender-Voucher. The identity of the sender and the identity of the issuer. The server also needs to sign the response. How the certificate is included in the response is determined by the policy of the web-service producer, and impacts set-up on the client. (OASIS 2003)

Authentication

Authentication of a party is to determine the identity of another party by a processing of identification and verification. Two types of authentications are needed in SAML:

1. *Active session authentication*

Authentication is provided in a non-persistent manner in the communication channel used to transport SAML. This authentication maybe either unilateral, that is from the session initiator to the receiver or bilateral. (OASIS 2010)

2. *Message-Level*

Message level authentication is an application layer service and facilitates the protection of message data between applications. SOAP based communications introduces the notion of Message Level

authentication. In message level authentication, security information is contained within the SOAP message, which allows security information to travel along with the message. (OASIS 2010)

Confidentiality Of The Message

To achieve message confidentiality, the parties in a SAML conversation may use the confidentiality protection mechanism in the underlying SOAP transport. For SOAP messages used over HTTP, this would be HTTP/S. The confidentiality of the message can also be achieved by encrypting the message using a strong encryption mechanism. Two types of confidentiality must be taken in account:

- In Transit
- Message Level confidentiality (OASIS 2003)

Data-Integrity Of The Message

Data integrity is the ability to confirm that a given message as received is unaltered from the version of the message that was sent. Integrity can be achieved by using the following.

- XML Digital Signature

To ensure message integrity, the parties in a SAML conversation may add a XML Digital Signature to the SAML query.

- HTTP/S with Certificates

The parties MAY use the underlying transport of the SOAP conversation to ensure message integrity. For SOAP messages sent over HTTP, this would be HTTP/S with client certificates. (OASIS 2003)

Key Management And Binding Of Identity To The Key

Keys are used in SAML protocol to protect the confidentiality and integrity of the message. Key management is also used to manage certificates and other security options regarding the encryption, decryption, and digital signing of email messages. Therefore there is a need to safeguard these keys. Adequate protection needs to taken to protect the keys by using strong encryption and password policies for accessing the data. Verifying the binding of key to an entity requires additional validation and the use of public key infrastructure helps in establishing the key-to-individual binding. Some federations still rely on traditional PKIX-style hierarchical PKI for key management. Others, including many large-scale SAML federations, rely on an alternative key-management method. In this method, collections of keys are collectively signed, resulting in an object that behaves like a combination of a PKI certificate and CRL (certificate revocation list). Incidentally, this bag-of-keys model has been considered by the KARP (Keying and Authentication for Routing Protocols) working group as the basis for routing protocol key management (Naithan 2008)

Common to all approaches to key management is a federation that consists of those identity providers and service providers that share trust in a set of keys. Such a trust framework is called a ring of trust. The

deployment of OpenID typically relies on a single global ring of trust that encompasses all OpenID IdPs, and SPs. In fact, it is entirely possible that the success of OpenID is due in large part to the absence of a requirement on key management. Conversely, SAML federations often do require key-and-trust management, which constitutes a major part of the work involved in running a SAML federation.

Use Of TLS/SSL Over HTTP

SSL can be implemented in SAML by attaching an authorization token to the message. The SAML token is expected to carry some authorization information about an end user. Because HTTPS piggybacks HTTP entirely on top of TLS, the entirety of the underlying HTTP protocol can be encrypted. This includes the request URL, query parameters, headers, and cookies. However, because host addresses and port numbers are necessarily part of the underlying TCP/IP protocols, HTTPS cannot protect their disclosure. In practice this means that even on a correctly configured web server eavesdroppers can still infer the IP address and port number of the web server that one is communicating with as well as the amount and duration of the communication, though not the content of the communication. (Naithan 2008)

What Are Security Risks In SAML Based System?

There are various security risks associated with SAML protocol. These are discussed below.

Risks Associated With SAML Assertions

Once an assertion is issued it is not in the issuer's control anymore. Hence the issuer has no control over how long the assertion will exist before it is made use of. Issuer might also not have to chance to check to whom the assertions are shared with. Therefore the SAML authorities need to be careful about what information needs to be added to the assertion. (Naithan 2008)

Risks Associated With SAML Protocol

The protocol is susceptible to denial of service (DOS) attack as the effort required for processing of each assertion is proportionally much greater than the effort required by an attacker to generate the request. (Naithan 2008)

Risks Associated With SAML Protocol Bindings

There are various risks associated with the SAML bindings, which can be seen below.

1. *Eavesdropping*

Eavesdropping is a threat to confidentiality of data. SAML profile may under such a threat if the web services are not secured. In order to protect against this threat methods such as assertions, assertion references, attribute information, should be in an encrypted format. Care should be taken such that only valid

parties can view this information.

2. *Replay*

Replay is a type of an attack in which the attacker fraudulently intercepts the data and retransmits it to the receiver. Signed SAML does not detect or prevent replay attacks. Also, assertions that contain sender-vouches confirmation impose no restriction on the entities that use or re-use these assertions.

3. *Message insertion*

A fabricated request or response is inserted into the message stream. An incorrect message such a "True" might be inserted into the message, which might cause the system to behave differently.

4. *Message deletion*

A valid request or response might be deleted from the message stream. The message deletion attack would also prevent a request from reaching a responder.

5. *Message modification*

Modification of the request to alter the details of the request can result in significantly different results. This is a threat to integrity of the message. SAML assertions and SOAP message content should be signed by the sender so that receiver can ensure that the information that provides binding between the subject to the message content has been protected by the attesting entity against modification.

6. *Man-in-the-middle*

Assertions with a holder-of-key subject confirmation and SOAP bindings are susceptible to Man-in-the-middle attack. An authentication mechanism should be put in place to prevent such kind of attacks. (IBM 2004) The diagram below explains how a MITM attack can occur on a SAML SSO solution.

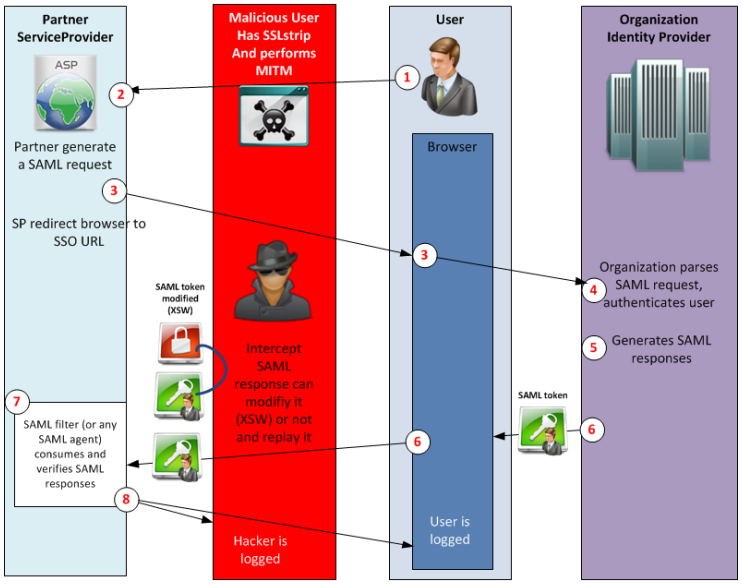

Figure 5: Man in the middle attack, Source: Nagib, "http://www.secutalk.ch/?p=650"

The above diagram shows a use case of Man in the middle attack. The user first tries to login

to Service Provider website. The service provider checks if a valid session token exists. If not it generates a SAML request and sends it to the identity provider. Identity provider authenticates the user and sends a SAML response back to the service provider. The attacker intercepts this message and gets the SAML response. The attacker then modifies the response by altering the message contents but keeping intact the signature of the identity provider. Attacker then sends the altered message to the service provider. The Service provider checks for signature and then authenticates the attacker as a valid user. (SecuTalk 2012)

Risks Associated With SOAP Over HTTP

In SOAP over HTTP the client can only make SOAP calls, not receive them, SOAP is no more insecure than any other application which POSTs XML files to a web server. The clients are safe unless the server have been subverted; the server is vulnerable, and does need to be secured.

Since SOAP data is received by the server, but not sent to the client, one can understand that the threat is primarily aimed at the server itself. The following are methods of attack, and how Web Services can be exploited to fulfill these attacks:

- *Buffer Overflows*

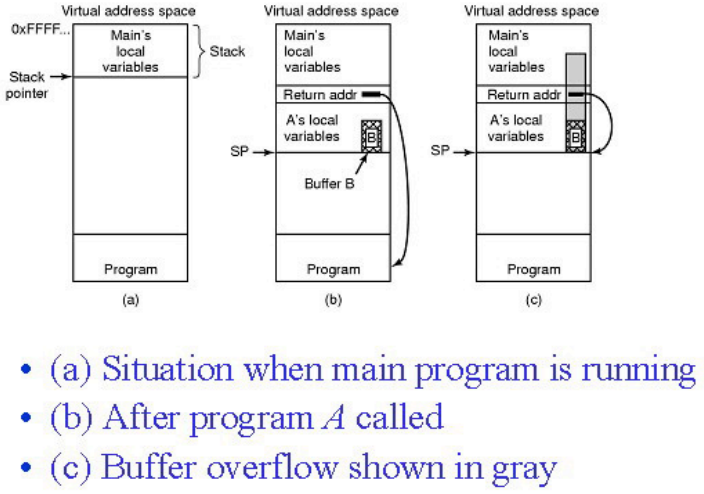

Figure 6: Illustration of Buffer Overflow, Source: bufferoverflowattack.com, "http://bufferoverflowattack.com/"

The buffer overflow situation exists if software makes an attempt to place much more data inside a buffer than it could keep or even when software attempts to place data in a memory space area past a buffer. Buffer overflows could be activated by inputs that are designed to perform program code, or maybe modify how software works. This might cause erratic software behavior, such as memory space access errors, wrong final results, a crash, or perhaps a break of system secureness. Buffer overflow has become the most common kind of application security vulnerability. Almost all application developers understand what a buffer overflow vulnerability is actually, however buffer overflow attacks towards the two legacy and also newly-developed programs remain very typical. (Stallings and Brown 2011)

- *XML Injections*

XML Injection is an attack technique used to manipulate or compromise the logic of an XML application or service. The injection of unintended XML content and/or structures into an XML message can alter the intend logic of the application. Further, XML injection can cause the insertion of malicious content into the resulting message/document. XML is so ubiquitous that it can also be found in use on the web application server, in browsers as the format of choice for XMLHttpRequest requests and responses, and in browser extensions. Given its widespread use, XML can present an attractive target for XML Injection attacks due to its popularity and the default handling of XML allowed by common XML parsers. Where the browser is an active participant in an XML exchange, consideration should be given to XML as a request format where authenticated users, via a Cross-Site Scripting attack, may be submitting XML that is actually written by an attacker.

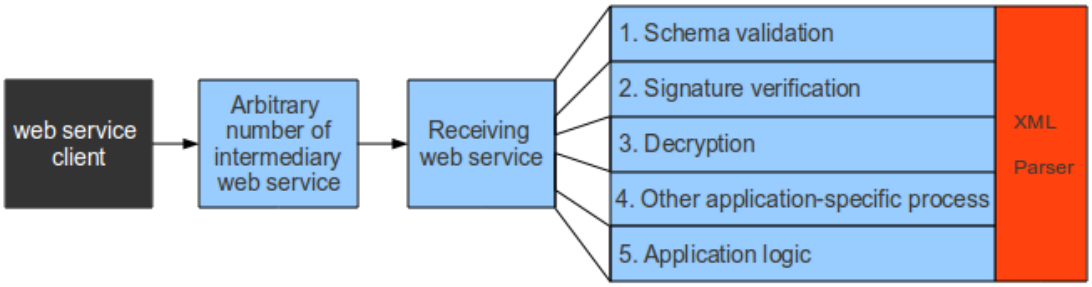

Figure 7: XML Injection Attack, Source: rub.de, "http://clawslab.nds.rub.de/wiki/index.php/XML_Injection"

Red = attacked web service component

Black = location of attacker

Blue = web service component not directly involved in attack

In the above diagram we can see that the attacker using just a web service client will be able to perform malicious attack on the XML parser by using XML injection methods.

- *Session Hijacking*

Session hijacking is an attack in which the attacker intercepts and controls the valid session created by source and destination entities. The figure shows a high-level example of session hijacking. The primary type of session hijacking attack, the most graceful and elegant of them, is the concept of session theft, which is actually becoming the man in the middle with the classic security paradigm of a man in the middle attack. The attacker gets in the middle of a session and actually captures the information going back and forth, including the TCP establishment, the sequence numbers, the network identifications, the MAC addresses, and the port numbers, and then intercepts. At some point the attack becomes either the client or the server and gets in the way and becomes a true man in the middle and proxies the traffic back and forth.

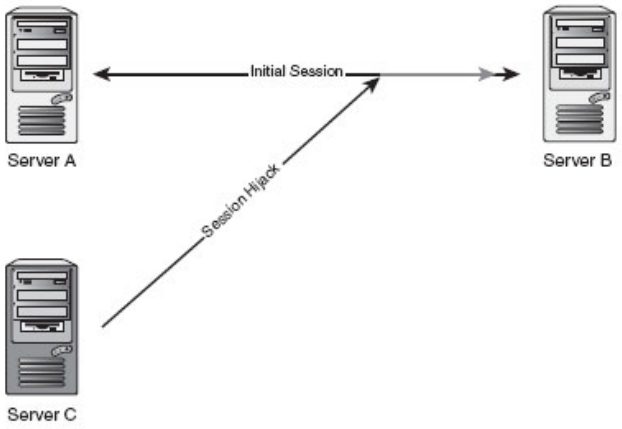

Figure 8: Session Hijacking Illustration Source: Himanshu Dwivedi "http://searchstoragechannel.techtarget.com/feature/Fibre-Channel-frame-weaknesses"

If this is done correctly, the attacker convinces the server that the attacker is actually the client. The client sometimes can't quite tell that the server is the server or not. Often times this breaks the connection between the client and the server. That doesn't really matter. If the attacker steps in the middle and becomes the client to the server the attack is probably going to be successful.

Risks Associated With The Profiles Of SAML

While SAML is designed to support a variety of application scenarios, the profiles for authentication defined in the original standard are designed around applications. When channel binding is not used, protection against Man in the Middle attacks is left to lower layer protocols such as TLS, and the development of user interfaces able to implement that has not been effectively demonstrated. Failure to detect a man in the middle attack can result in phishing of the user's credentials if the attacker is between the client and IdP, or the theft and misuse of a short-lived credential if the attacker is able to impersonate a RP. (Cantor and Josefsson 2012)

XML Signature Wrapping Attacks

The two steps of processing XML documents with XML signatures are signature validation and function invocation. If the attacker manages make both the XML views different then XML signature wrapping attacks exist. The attacker modifies the XML structure by inserting forged elements but fools the system easily. This attack is successful because both the signature validation and function invocation looks at different parts of XML. Thus the attacker can modify the XML by keeping the required fields constant.

The below diagram illustrates how the XML signature wrapping attack happens. The attacker first moves the original content to a newly created wrapper content and then inserts arbitrary content with a different id as shown below. (Somorovsky, et al.)

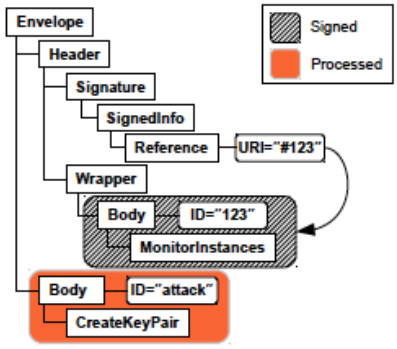

Figure 9: XML Signature Wrapping Attack Illustration, Source: Nagib, "http://www.secutalk.ch/?p=650"

HTTP Referrer Attack

The HTTP Referrer attack takes places when the attacker intercepts the message between the service provider and the Identity provider, modifies the message and changes the referrer to the attackers website. This way when the identity provider sends the message back to the service provider, it sends it to the attackers website instead. Later the attacker can use the response to authenticate with the service provider. (Somorovsky, et al.)

Signature Exclusion Attacks

This attack is possible if the system is designed poorly such that the message is not validated if signature is excluded. If the security logic does not find the Signature element, it simply skips the validation step. (Somorovsky, et al.)

How Can SAML Be Made More Secure?

Use Of Access Control System

Access controls systems are systems that provide security features which control how users and systems communicate and interact with other systems and resources. Since DDOS attack floods the server with multiple requests, the use of access control by coupling it with authentication mitigates the attacks by allowing only some requests to go through and blocking all other requests. Any client level authentication used must have the ability to provide authentication traceability and not itself be vulnerable to forgery. (Naithan 2008)

Use Of Two Factor Or Multi-Factor Authentication

Since using a Single Sign-On, only one password can be used for multiple systems. Using two factor or multi-factor authentication can enhance the security of the system. Three major kinds of authentication include verification by something a user knows (such as a password), something the user has (such as a smart card or a security token), and something the user is (such as the use of biometrics). Therefore this increases the

number of probability of an attack.

Use Of Strong Encryption And Password Policies

The hardest passwords to crack, for a given length and character set, are random character strings. If long enough passwords are used, they resist brute force attacks and guessing attacks. However, such passwords are typically the hardest to remember. The imposition of a requirement for such passwords in a password policy may encourage users to write them down. Therefore a training program to educate the users of the risks involved and strong password polices needs to be put into place.

Aware Of The Information Included In The Assertions

The security assertions that are passed from identity provider to service provider uses SOAP, therefore they are susceptible to man-in-the-middle attack. Care should be taken such that the information added in the assertions does not reveal any confidential information such as the user password. Also the password needs to be encrypted by a strong encryption algorithm. It is also better for such communication to happen over a secure channel such as SSL.

Only Process What Is Hashed

The hashed parts of an XML document are those parts that are serialized as an input to a hash function, and where the hash value is stored in a Reference element. This excludes all parts of the document that are removed before hash calculation by applying a transformation, especially the enveloped signature transform.

Xml Schema Validation

The XML message that is sent from identity provider to service provider needs to be validated, to check if there is any change in the XML schema. This validation will make sure man in the middle attacks are avoided. The validation also helps to check if integrity and confidentiality of the message is intact.

Verify And Validate The Signature

Verification and validation of the signature needs to be done by the service provider to check if the message is altered in anyway. The XML Signature verification in the assertion is an enveloped signature if and only if both objects are identical. The XML Signature validation is the verification of every enveloped signature is exclusively done on the DOM tree of each corresponding assertion.

Avoid Using Referrer Tag

Since the referrer tag can be spoofed easily, the SAML exchange should avoid using it and instead use a dereferrer redirect before the message is sent from identity provider to service provider. A second approach is to enforce the one-time usage property of the SAML artifacts also at the destination site. (Somorovsky, et al.)

Conclusion

SAML is a good, flexible, and extendible framework but comprehensive security architectures should be considered while implementation.

SAML does a couple of things that clear a lot of clutter in the security portability space. It defines a standard mechanism for representing information that needs to be exchanged, and it defines a standard for exchanging that information. But there are various risks that are involved in transferring data from the identity provider to service provider. To mitigate risk, SAML systems need to use timed sessions, HTTPS, and SSL/TLS.

- *All possible attacks needs to taken into considerations*

  There are innumerable attacks that might occur while using SAML such as Man-in-the-middle, replay, eavesdropping, etc. There should be a clear plan before implementation to mitigate such risks and leave no vulnerability for attackers to exploit and make the system secure.

- *The data transferred should be encrypted using strong encryption algorithm*

  Since one of the main concerns with any Internet application is the security of the data that is being communicated over unreliable networks. There are various ways that the attacker can sniff the data from the communication channel. Therefore care should be taken for proper encryption of the data while transferring also making the channel secure.

A secure SAML framework has lot of potential benefits; few of them are listed below:

- User passwords never cross the firewall, since user authentication occurs inside of the firewall and multiple Web application passwords are no longer required
- Web applications with no passwords are virtually impossible to hack, as the user must authenticate against an enterprise-class IdM first, which can include strong authentication mechanisms
- Service Provider initiated SAML SSO provides access to Web apps for users outside of the firewall. If an outside user requests access to a Web application, the Service Provider can automatically redirect the user to an authentication portal located at the Identity Provider. After authenticating, the user is granted access to the application, while their login and

password remains locked safely inside the firewall

- Centralized federation provides a single point of Web application access, control and auditing, which has security, risk and compliance benefits
- The message transferred would not be prone to man-in-the-middle attacks and therefore the message can be transferred from the source to recipient without much risk.

SAML provides standardization for the communication of identity information between identity provider and service provider. It is also relatively easy to implement as compared to other Single Sign-On solutions. Further it provides seamless integration with cross-domain websites taking many of the security aspects into account. But challenges in the field of security do remain and there are various means by which an attacker can put the system at risk as discussed in the paper. The organization implementing SAML should take into consideration the current security challenges and try to mitigate the potential risks.

REFERENCES LIST

1. Cantor, Scott, and Simon Josefsson. *SAML Enhanced Client SASL and GSS-API Mechanisms.* October 17, 2012. http://tools.ietf.org/html/draft-ietf-kitten-sasl-saml-ec-04 (accessed November 18, 2012).
2. emillion. *Is SAML all you need to offer business customers SaaS single sign-on?* 1 1, 2012. http://www.emillion.biz/articles/is-saml-all-you-need-to-offer-business-customers-saas-single-sign-on (accessed 11 18, 2012).
3. Grob, Thomas. *Security Analysis of the SAML Single Sign-on Browser/Artifact Profile.* Prod. IBM Zurich Research Laboratory.
4. IBM. *XML Security: Ensure portable trust with SAML.* Mar 23, 2004. http://www.ibm.com/developerworks/xml/library/x-seclay4/ (accessed Nov 18, 2012).
5. Naithan, Jayesh. *SAML proposal for securing XML web services.* Project Paper, MN: University of St. Thomas, Saint Paul, 2008.
6. OASIS. *saml.xml.org.* 10 18, 2007. saml.xml.org (accessed Oct 21, 2012).
7. OASIS. *Security and Privacy Considerations for 3 the OASIS Security Assertion Markup 4 Language (SAML) V1.1.* , September 2, 2003.
8. —. *Technology Reports: Security Assertion Markup Language (SAML).* Feb 23, 2010. http://xml.coverpages.org/saml.html (accessed Nov 18, 2012).
9. Ponemon Institute©. *How Single Sign-On Is Changing Healthcare.* Independently conducted,: Ponemon Institute, 2011.
10. SecuTalk. *Single Sign On with SAML threats and risks.* Aug 17, 2012. http://www.secutalk.ch/?p=650 (accessed Nov 18, 2012).
11. Somorovsky, Juraj, Andreas Mayer, Jorg Schwenk, Marco Kampmann, and Meiko Jensen. *On Breaking SAML: Be Whoever You Want to Be.*.
12. Stallings, William, and Lawrie Brown. *Computer Security: Principles and Practice.*: Prentice Hall, 2011.