Security in Web Services

Brandon Blanck
Xiaoling Hu
Clinton Nielsen
Eric Chowns
Terry McAllister

Introduction to web services

Web services are a new and exciting technology that is rapidly gaining popularity. They are used primarily as a way to communicate between applications. For example, there could be a vacation booking site that communicates with the airlines, car rentals, restaurants, and entertainment sites. This application to application communication is accomplished through web services.

In general, web services are XML based. They define an XML schema and a Web Services Definition Language, or WSDL. These two files describe how to communicate with the web service. The available methods/functions, their parameters, and return values are defined by these two files. Using the WSDL, a service can easily communicate with other services.

Much of this communication is transparent to the user. The user simply accesses a website and submits a query. This query is sent to the server, where it is parsed and processed, and sent to other relevant servers. These servers then respond with the result (typically an XML document), which the original server then parses and generates an HTML page. This HTML page is sent back to the user, and the user views the website containing the result of their query.

Because web services are often sending sensitive information – such as user names and/or passwords, or information that can be used maliciously (as will be presented), security is an essential component to any proper web-service.

Basic security concepts

There are some basic security steps that should be undertaken by a server. A network level firewall is very important. The firewall monitors all incoming and outgoing traffic and will block any traffic that does not meet its requirements (e.g. the traffic requests a closed port). Identity of the information requesters can be determined at the firewall, and they can be authenticated or rejected based on this. The firewall also allows for the passage of encrypted information.

Secure Socket Layer (SSL) is a protocol used for transmitting sensitive information. It uses a private key for encrypting data, and ensures that the data is protected during transport. Sites that use SSL start with https: rather than http:.

Message digests are another method used to secure communications. Message digests are like hashing functions – they ‘digest’ data to calculate a hash value called a message digest. The digest value can then be used to verify the integrity of the data. The data, as well as the digest, can be altered however, and as such, message digests are not enough.

Digital Signatures are used with message digests to ensure message integrity. Digital Signatures have the added benefit of verifying a message source. Keys are used to produce and verify digital signatures. Typically, a message will be ‘digested’, and the digested version of the message will be signed. The recipient of the message can then repeat the digest calculation, and use the public or private key to verify the signature. If the digest value has been altered, then the digital signature will not verify.

Finally Digital Certificates are used to transmit information about the sender of a message. In a basic form, they contain two pieces of information: the identification of the owner of the certificate, and the public key of the owner. The certificate is then signed using the owner's private key, and, as such, can be verified as stated earlier using the owner's public key.

Web service security

The main concern with security in web services is when a user is able to access potentially dangerous information, which firewalls and SSL do not protect against. It is possible for a user to access data that is only intended for applications, thus compromising it. Consider our vacation service example. Perhaps the hotel service has some discounted rates to give the vacation service under certain circumstances. If a malicious user taps into the hotel service, bypassing the vacation service, he would then have access to these rates.

Another possible problem occurs when data is sent through an intermediary. At the intermediary, data integrity and confidentiality could be lost. To prevent this, and unwanted access, we need a mechanism to provide end-to-end security.

WS Security Model

Fortunately, a model exists to provide end-to-end security. It is called the WS Security Model. With this, a web service can require that any incoming message prove a set of claims, such as a name, key, etc. To prove these claims, the message can be sent with a set of security tokens. Thus, messages both request that an action be carried out, and prove that they have a claim (or permission) to request that the action be carried out. If the requester doesn't have the required claims, rather than being rejected outright, the requester or someone on its behalf can try to obtain the necessary claims by contacting other Web services.

WS Security defines a <Security> tag within a SOAP message that contains all security-related information. Some important fields under the <Security> tag are the username and password, which are self-explanatory, and the nonce. A nonce is a parameter that varies with time, intended to limit or prevent the unauthorized replay or reproduction of a file. A WS security header will also include the message digest, the message encryption tag, and a description (or URL) of all of the algorithms used to digest and encrypt the message.

To ensure that received messages are always in an expected form, all messages are also canonicalized. Canonicalization is a process whereby all logically equivalent XML files will be broken down into identical octet streams when sent over a network. Canonicalization can be compared to sorting, though it is usually not quite so simple. The canonicalization algorithm is also included in the WS security header.

The entire WS Security block is sent in clear text. Because of this, it is essential that it is sent over a secured channel; otherwise the password is available to anyone looking. If the block absolutely must be sent over an unencrypted channel, the password should be obscured by creating a password digest as described in the WS Security specification — by creating an SHA1 digest using the nonce, timestamp and password.

XML Firewall

An XML firewall sits between the Internet and the web service. It receives all requests destined to the SOAP server and ensures that the message received is identical to the one that the requester sent and that the sender is indeed a trusted business partner. These two aspects are defined as ‘Message Integrity’ and ‘User Authentication’. If the message integrity is found to be in order, the XML firewall reads the requester's identification information from the SOAP request and verifies it usually through means of a digital signature. If the requester is found to be a trusted business partner, the XML firewall allows the request pass onto the SOAP server. Essentially, the XML firewall enforces the rules set out by the WS Security model.

Identity Theft and Encryption

In the past 5 years, over 5 million people have been victimised by identity theft. One common cause is the sending of sensitive information over the Internet: credit cards, social insurance numbers, etc. Unprotected information can be viewed by hackers and other malicious users during online transactions.

One of the main goals of WS Security is maintaining confidentiality. Only the sender and receiver should see the information that is sent through the web service. Encryption, digital signatures, and certificates are common forms of protecting this information. There are numerous algorithms for encrypting/decrypting plain text messages.

Asymmetric cryptography is one such form. It uses a public and private key pair, generated by a common algorithm. The public key is used to encrypt messages, and is available to anyone that wants to communicate with you. The private key, on the other hand, is used to decrypt messages, and is only known to you.

Symmetric cryptography is another form. There is one key that is used for encryption and decryption. Both sides must know what that key is. Unfortunately, because of this, the key must be exchanged among everyone, and the more people that know the key, the more likely it is to be compromised.

As previously mentioned, data can be encrypted at the transport level using SSL. This is very easy to implement on HTTP servers. However, because WS Security aims to maintain interoperability among different protocols, this is not ideal. SSL restricts web services to SOAP over HTTP.

A better way is to use XML encryption. This defines an XML schema that represents a document, image, or XML file. Using XML encryption, the entire file

or just parts of the message can be encrypted. For added security and portability, different parts of the message can be encrypted using different encryption techniques. This makes it so the entire message is much harder to decode, and can allow one message to be sent to several different parties – each with the ability to decrypt their respective portion.

XML encryption is not just restricted to web services; it can also be used in other areas such as encrypting database entries.

Denial of Service Attacks

A denial of service, or DoS, is an incident in which a user or organization is deprived of the services/resources that they normally expect to have. For example, the e-mail server could become overwhelmed due to a DoS attack, and temporarily be disabled. In general, a DoS attack does not result in the theft of information, or other security loss.

One typical type of attack is a Buffer Overflow. This occurs when more traffic is sent than is expected by the server, causing, as one would expect, the buffers to become full and overflow.

A SYN Attack is another DoS. This attack exploits the handshaking that occurs between client and server. In this attack, numerous connection attempts are sent very rapidly, and then do not respond. This causes the first packet to be left in the buffer, causing other, legitimate requests to go unprocessed.

A teardrop attack is an exploitation of the Internet Protocol (IP). When the packet is too large and split into fragments, the attacker inserts a confusing value into the offset field of one of the packets. If not handled properly by the operating system, the entire system can crash.

Smurf attacks occur by sending ping requests to a broadcast address. The ping requests are spoofed to look like they are from the target site. When the receiving site responds, the target site receives all the ping responses and can be overwhelmed.

Unfortunately, there is not really any way to fully protect against DoS attacks. Much of it lies with the programmers to make sure that their code is secure and not vulnerable to overflows and other attacks. Likewise, the administrator must keep everything up to date, since new bugs are discovered all the time, and fixes are released for them. Sadly though, many times you must simply learn from your mistakes, and fix security holes as soon as possible.

WestJet vs. Air Canada: An Example of Poor Web Service Security

Air Canada alleges WestJet used automated software, and the password and pin of a former employee, to access an online database containing all of Air Canada's information on ticket sales. By gaining access, Air Canada claims that WestJet was able to identify Air Canada's most profitable routes, plan expansion into new routes, and adopt pricing strategies to force Air Canada out of new markets.

However, the former employee had legitimate access to the site. So how could Air Canada have detected and prevented this breach?

This brings up some interesting questions. Why are former employees able to access the site? Should their access not be suspended upon termination?

According to Air Canada, 240,000 queries were made to their service between May 15, 2003 and March 19, 2004. Doing the math, this yields approximately two queries a minute, every minute of every day during this period. Obviously, this is not normal.

Unfortunately, this is not a case that WS Security could prevent – the former employee had legitimate access. Sadly, the blame lies mostly on Air Canada – one, for allowing former employees access to their site, and two, for not being aware of the sheer number of queries being generated. Of course, WestJet is not without blame, but Air Canada, had it had better security, should have caught this much sooner than it did.

This specific example illustrates the idea that no matter how many external security devices are employed – such as encryption and firewalls etc. – if a business leaves a loophole in the security, then there is still a hole. Businesses need to be aware that although security devices are essential, good security practice is just as essential.

Conclusion

In this day and age, with so many people and businesses using the Internet, security is of utmost concern. Security in web services is no different. Business sites, especially, must be secure in order to instill trust in their customers. WS Security, when used in addition to other basic security such as firewalls and SSL, allows for a very secure web service. This is not to say it is invulnerable, however. Servers must still be kept up to date with the latest patches, and there is no replacement for a good server administrator who notices when things are amiss. But for a service to be secure there is no reason to not implement WS Security.