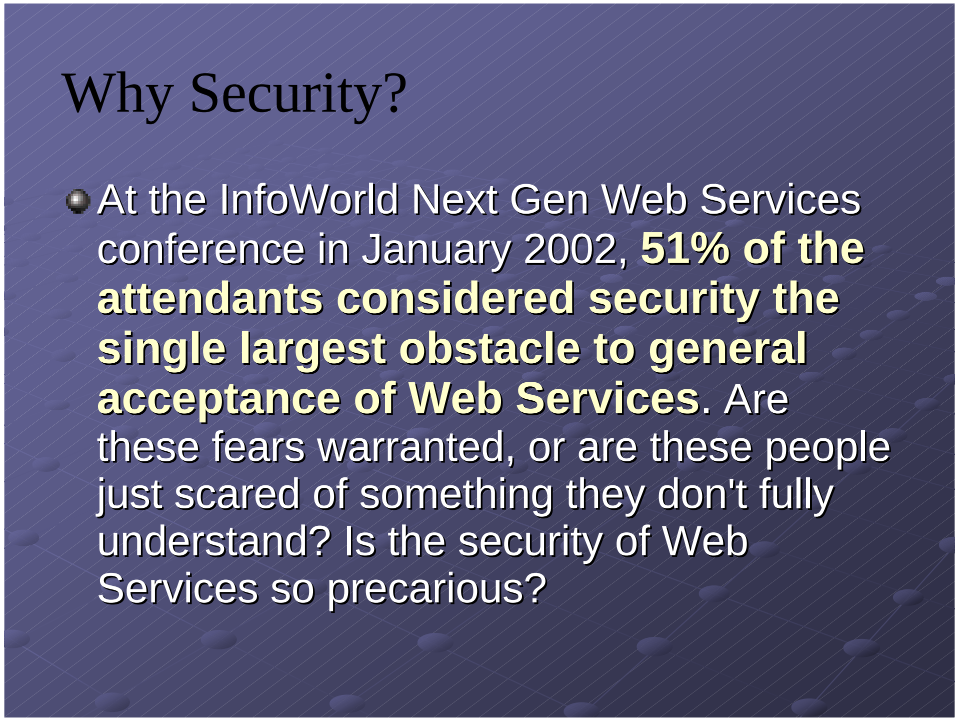
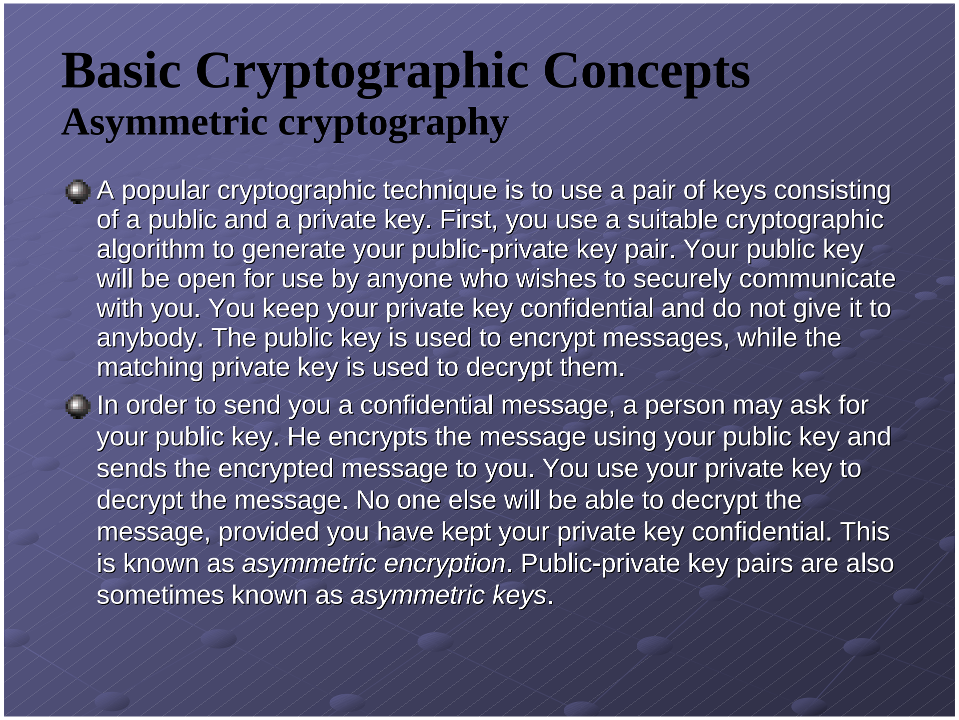
Security in Web Services

Brandon Blanck
Xiaoling Hu
Clinton Nielsen
Eric Chowns
Terry McAllister



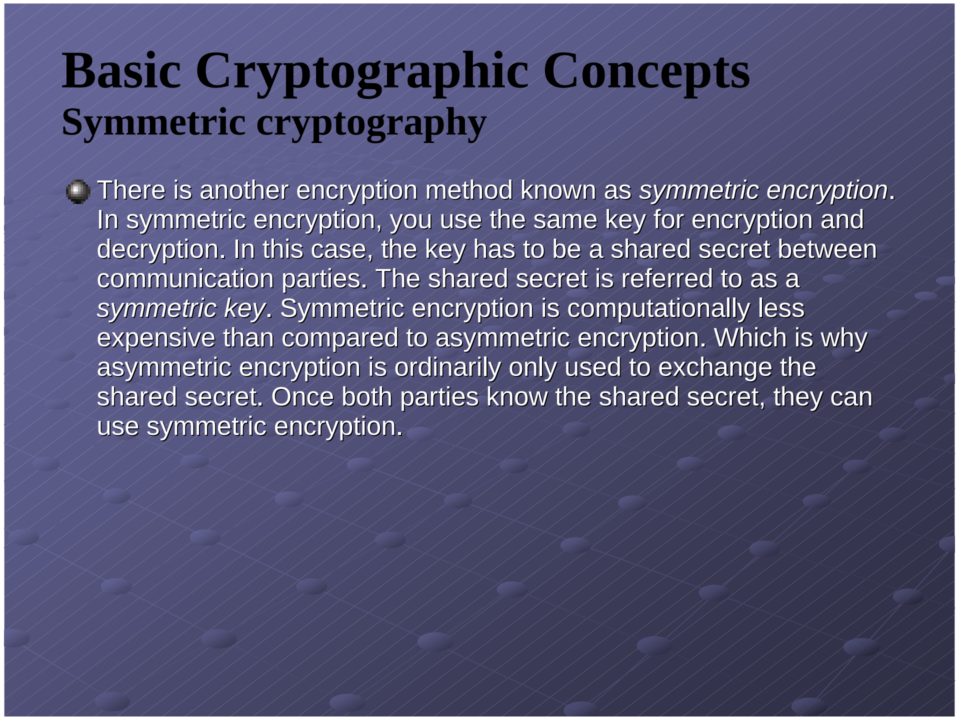
Why Security?

- At the InfoWorld Next Gen Web Services conference in January 2002, **51% of the attendants considered security the single largest obstacle to general acceptance of Web Services**. Are these fears warranted, or are these people just scared of something they don't fully understand? Is the security of Web Services so precarious?



Basic Cryptographic Concepts
Asymmetric cryptography

- A popular cryptographic technique is to use a pair of keys consisting of a public and a private key. First, you use a suitable cryptographic algorithm to generate your public-private key pair. Your public key will be open for use by anyone who wishes to securely communicate with you. You keep your private key confidential and do not give it to anybody. The public key is used to encrypt messages, while the matching private key is used to decrypt them.
- In order to send you a confidential message, a person may ask for your public key. He encrypts the message using your public key and sends the encrypted message to you. You use your private key to decrypt the message. No one else will be able to decrypt the message, provided you have kept your private key confidential. This is known as *asymmetric encryption*. Public-private key pairs are also sometimes known as *asymmetric keys*.



Basic Cryptographic Concepts
Symmetric cryptography

- There is another encryption method known as *symmetric encryption*. In symmetric encryption, you use the same key for encryption and decryption. In this case, the key has to be a shared secret between communication parties. The shared secret is referred to as a *symmetric key*. Symmetric encryption is computationally less expensive than compared to asymmetric encryption. Which is why asymmetric encryption is ordinarily only used to exchange the shared secret. Once both parties know the shared secret, they can use symmetric encryption.

Basic Cryptographic Concepts
Message digests

- Message digests are another concept used in secure communications over the Internet. Digest algorithms are like hashing functions: they consume (digest) data to calculate a hash value, called a message digest. The message digest depends upon the data as well as the digest algorithm. The digest value can be used to verify the integrity of a message; that is, to ensure that the data has not been altered while on its way from the sender to the receiver. The sender sends the message digest value with the message. On receipt of the message, the recipient repeats the digest calculation. If the message has been altered, the digest value will not match and the alteration will be detected.
- But what if both the message and its digest value are altered? That kind of change may not be detectable at the recipient end. So a message digest algorithm alone is not enough to ensure message integrity. That's where we need digital signatures.

Basic Cryptographic Concepts
Digital signatures

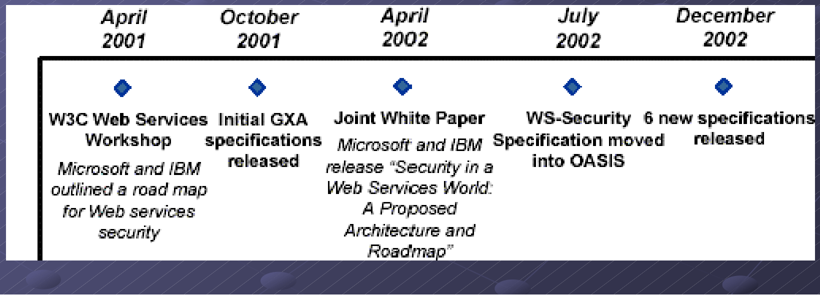
- Keys are also used to produce and verify digital signatures. You can use a digest algorithm to calculate the digest value of your message and then use your private key to produce a digital signature over the digest value. The recipient of the message first checks the integrity of the hash value by repeating the digest calculation. The recepient then uses your public key to verify the signature. If the digest value has been altered, the signature will not verify at the recipient end. If both the digest value and signature verification steps succeed, you can conclude the following two things:
  - The message has not been altered after digest calculation (message integrity); and
  - the message is really coming from the owner of the public key (user authentication).

Basic Cryptographic Concepts
Certificates

- In its most basic form a digital certificate is a data structure that holds two bits of information:
  - The identification (e.g. name, contact address, etc.) of the certificate owner (a person or an organization); and
  - the public key of the certificate owner.
- A certificate issuing authority issues certificates to people or organizations. The certificate includes the two essential bits of information, the owner's identity and public key. The certificate issuing authority will also sign the certificate using its own private key; anyone interested party can verify the integrity of the certificate by verifying the signature.

History

- In April 2001, the W3C conducted a Web Services Workshop whose aim was to explore the direction the W3C should take to standardize the emerging Web services architecture. This was a pivotal event in the Web services world, as one of its outcomes was a set of specifications in October 2001 that presented an architecture for the next generation of XML Web Services. This architecture is known as the Global XML Web Services Architecture, (GXA).



April 2001
W3C Web Services Workshop
Microsoft and IBM outlined a road map for Web services security

October 2001
Initial GXA specifications released

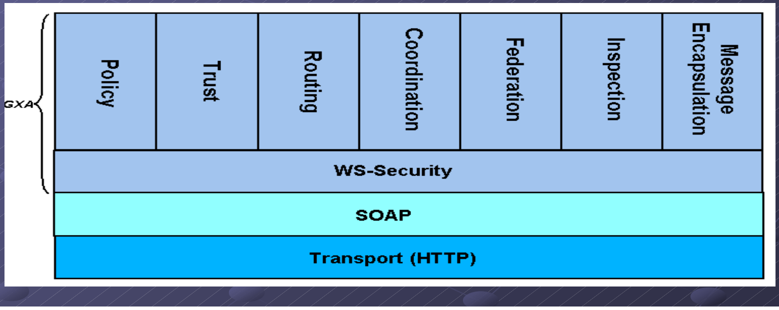
April 2002
Joint White Paper
Microsoft and IBM release "Security in a Web Services World: A Proposed Architecture and Roadmap"

July 2002
WS-Security Specification moved into OASIS

December 2002
6 new specifications released

The Cornerstone of GXA:
WS-Security

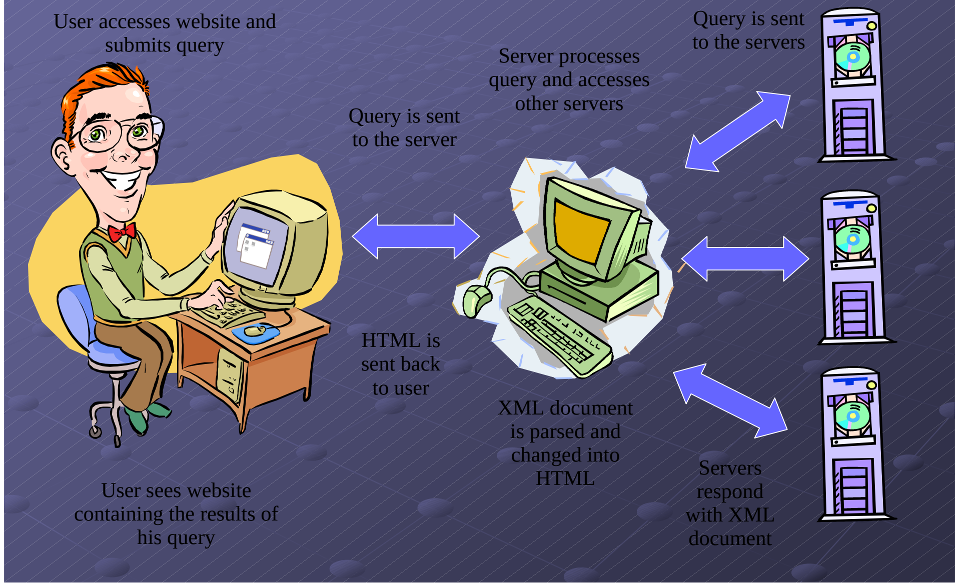
- **Secure Message Exchanges**
- WS-Security defines a standard set of SOAP extensions that implement message-level integrity and confidentiality for secure message exchanges. WS-Security forms the basis for many other GXA specifications, and thus is considered "The Cornerstone of GXA." This is indicated in the figure below, which shows GXA's placement in the "Web Services Stack:"



Introduction to Web Services

- Relatively new
- Used to exchange information between applications
- Security is a big concern
- Generally XML based
- We will focus primarily on SOAP

How Web Services Work


User accesses website and submits query
Query is sent to the server
Server processes query and accesses other servers
Query is sent to the servers
HTML is sent back to user
XML document is parsed and changed into HTML
Servers respond with XML document
User sees website containing the results of his query

A Useful Example

- We have considered a hotel business in class
- Expand this to be a complete vacation booking company
- Consider booking hotels, car rentals, entertainment, restaurant reservations, etc.
- One service cannot provide all this information

A Useful Example cont.

- Vacation booking website needs to communicate with other websites
- Needs access to available hotels, car rentals and so on
- Web Services are useful! The vacation service simply sends a request to other services for the information, then displays it for the user
- Everyone wins: helper sites get more business, vacation site makes a profit and the user doesn't have to visit many sites to do one thing

Basic Security

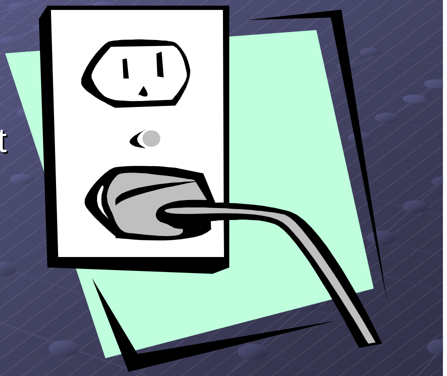
Network Level Firewall

- Monitors all incoming traffic
- Checks identity of information requesters
- Authenticates users based on their identities
- Allows for encrypted messages

Basic Security cont.

Secure Socket Layer (SSL)

- Protocol for transmitting private documents
- Uses a private key to encrypt data
- URLs start with https: rather than http:
- Protects information during transmission

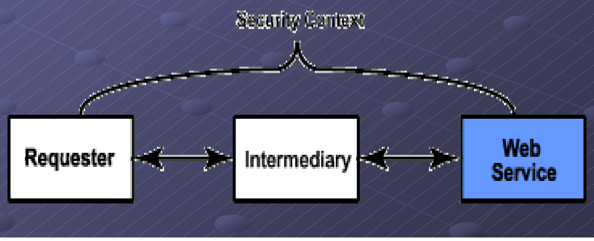


Basic Security cont.

- Firewalls and SSL are not enough
- Firewalls help to protect the server against unwanted access
- SSL helps to ensure that the data transmitted is safe and secure
- What about a user accessing application to application services (e.g. vacation site to hotel booking service)?
- User could gain access to potentially dangerous information
- Need another form of security to prevent this

Basic Security (cont.)

- Data may be received and forwarded on by intermediary
- Data integrity and confidentiality maybe lost in the forwarding process
- To fix this problem, we need a mechanism to provide end-to-end security

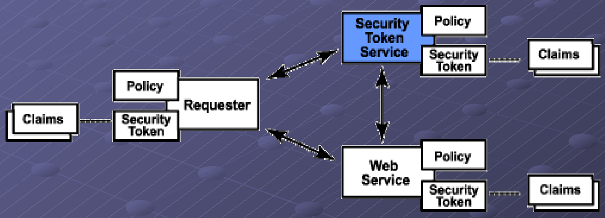


Basic Security (cont')

- The following WS security model can fix the problem
  - A Web service can require that an incoming message prove a set of *claims* such as name, key, etc
  - " A requester can send messages with proof of the required claims by associating *security tokens* with the messages. Thus, messages both demand a specific action and prove that their sender has the claim to demand the action"
  - "When a requester does not have the required claims, the requester or someone on its behalf can try to obtain the necessary claims by contacting other Web services."

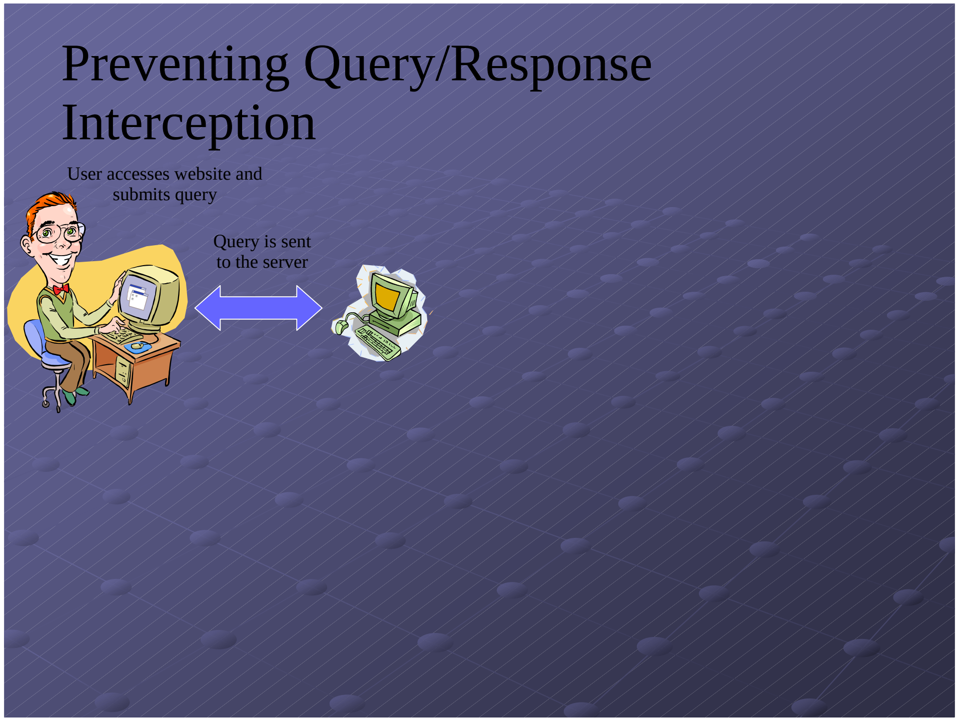
Basic Security (Cont')

- The above model is illustrated below



Preventing Query/Response Interception

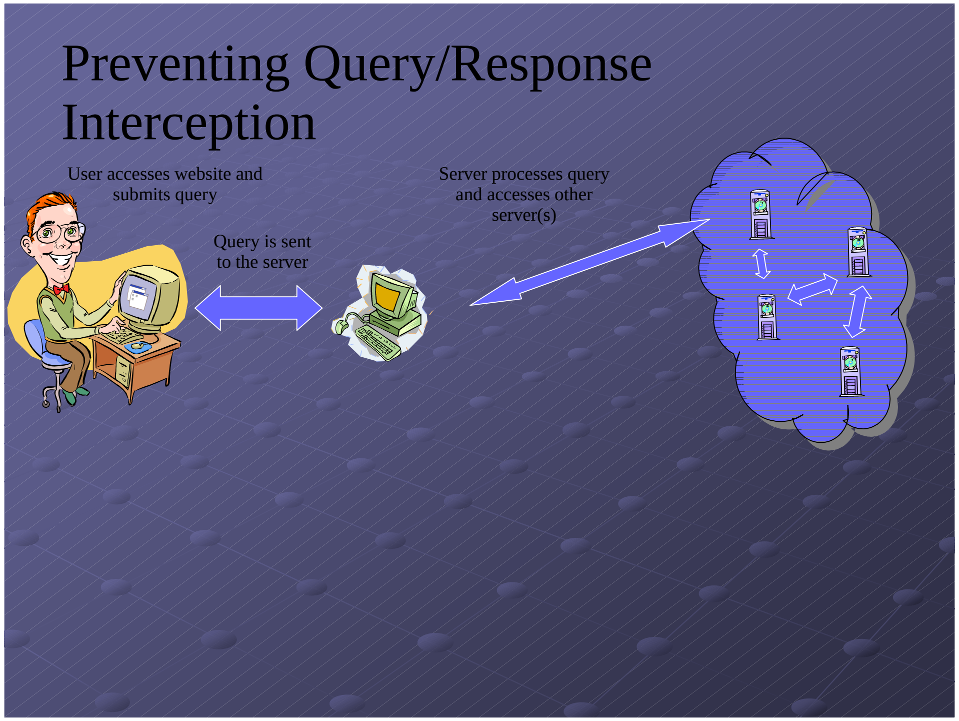
User accesses website and submits query



Preventing Query/Response Interception

User accesses website and submits query

Query is sent to the server

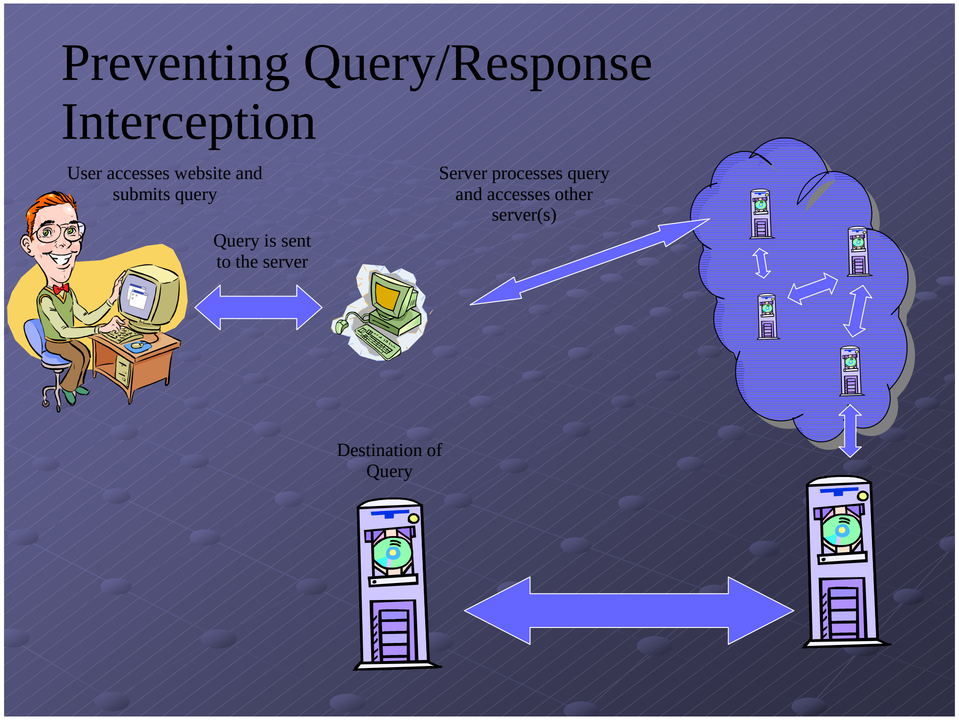


Preventing Query/Response Interception

User accesses website and submits query

Query is sent to the server

Server processes query and accesses other server(s)



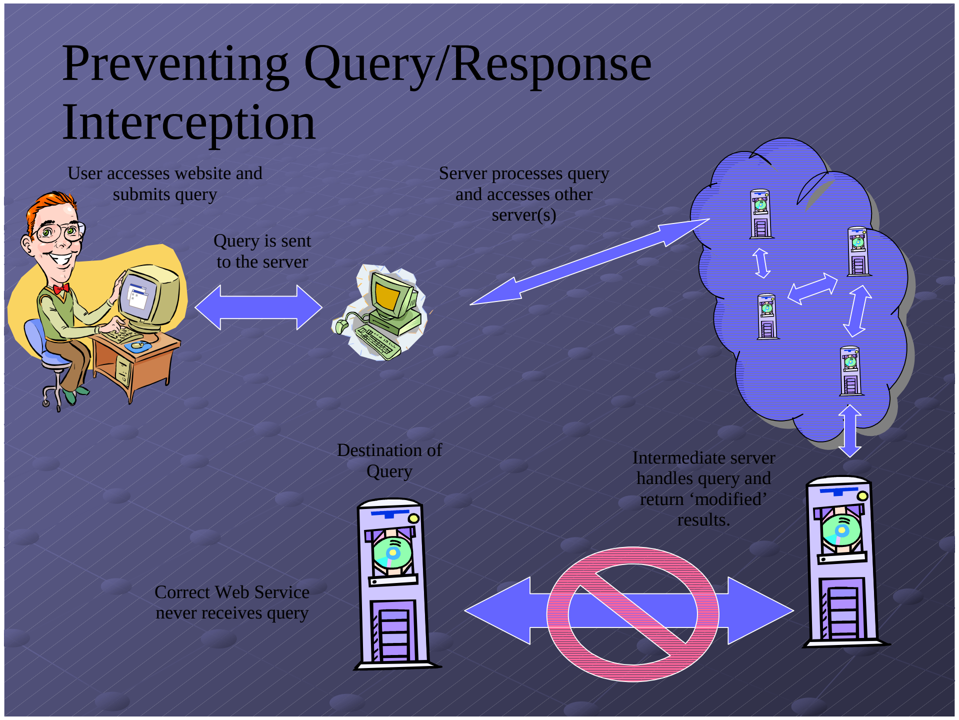
Preventing Query/Response Interception

User accesses website and submits query

Query is sent to the server

Server processes query and accesses other server(s)

Destination of Query



Preventing Query/Response Interception

User accesses website and submits query

Query is sent to the server

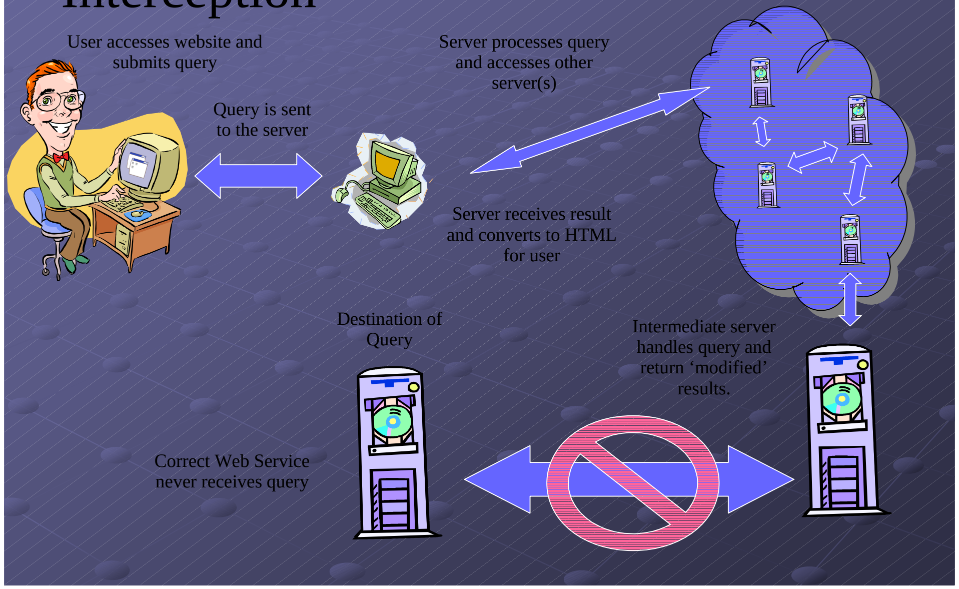
Server processes query and accesses other server(s)

Destination of Query

Intermediate server handles query and return ‘modified’ results.

Correct Web Service never receives query

Preventing Query/Response Interception



User accesses website and submits query

Query is sent to the server

Server processes query and accesses other server(s)

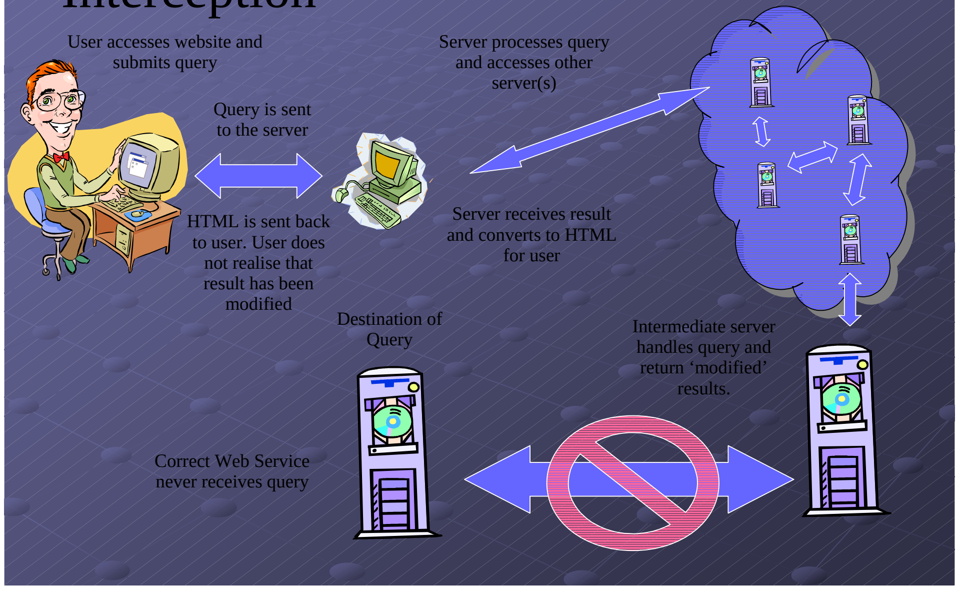
Server receives result and converts to HTML for user

Destination of Query

Intermediate server handles query and return ‘modified’ results.

Correct Web Service never receives query

Preventing Query/Response Interception



User accesses website and submits query

Query is sent to the server

Server processes query and accesses other server(s)

HTML is sent back to user. User does not realise that result has been modified

Server receives result and converts to HTML for user

Destination of Query

Intermediate server handles query and return ‘modified’ results.

Correct Web Service never receives query

Preventing Query/Response Interception

- WS-Security defines a <Security> header block within a SOAP message that contains all security-related information. For example, the following <Security> header is used to define direct trust using a username and password (all namespace declarations in this article are omitted for brevity):

```
<?xml version="1.0" encoding="utf-8"?>
 <S:Envelope>
  <S:Header>
   <wsse:Security>
    <wsse:UsernameToken wsu:Id="MyID">
     <wsse:Username>Zoe</wsse:Username>
     <wsse:Password>MyPassword</wsse:Password>
     <wsse:Nonce>FKJh...</wsse:Nonce>
     <wsu:Created>2001-10-13T09:00:00Z</wsu:Created>
    </wsse:UsernameToken> .....
   </wsse:Security>
  </S:Header>
 <S:Body wsu:Id="MsgBody"> ..... </S:Body>
</S:Envelope>
```

- In the above example, the username and password are supplied along with a nonce and timestamp. A *nonce* is a parameter that varies with time, intended to limit or prevent the unauthorized replay or reproduction of a file (this includes what is common referred to as a "replay attack"). **Because the password is clear text, it should be sent on a secured channel**; otherwise, it should be obscured by creating a password digest as described in the WS-Security specification—by creating a SHA1 digest using the nonce, timestamp and password).

Preventing Query/Response Interception

- Messages that are received need to satisfy the following two criterion:

Preventing Query/Response Interception

- Messages that are received need to satisfy the following two criterion:
- The message has not been altered while on its way to the hotel's web service (**message integrity**)

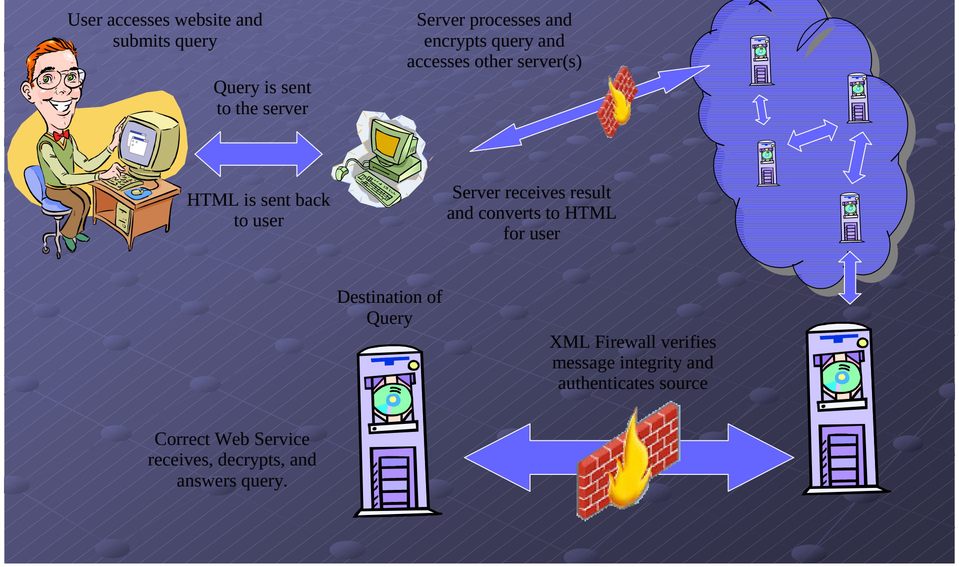
Preventing Query/Response Interception

- Messages that are received need to satisfy the following two criterion:
- The message has not been altered while on its way to the hotel's web service (**message integrity**)
- the requester is really a trusted business partner (**user authentication**).

Preventing Query/Response Interception

- New sequence of events:
  - The tour operator sends a method invocation SOAP request to the hotel's web service. The request includes all the relevant security information (message integrity and user authentication).
  - The hotel's web service is protected by an XML firewall, which receives all requests destined to the SOAP server. The XML firewall checks that the message as received is identical to the one that the requester intended to send.
  - If the message integrity is found to be in order, the XML firewall reads the requester's identification information from the SOAP request and makes sure that the user is really a trusted business partner.
  - If the requester is found to be a trusted business partner, the XML firewall allows the request pass onto the SOAP server.

Preventing Query/Response Interception


User accesses website and submits query
Server processes and encrypts query and accesses other server(s)
Query is sent to the server
HTML is sent back to user
Server receives result and converts to HTML for user
Destination of Query
XML Firewall verifies message integrity and authenticates source
Correct Web Service receives, decrypts, and answers query.

Preventing Query/Response Interception

- Four elements to create XML signature:
  - Signature Element with Three children: SignedInfo, SignatureValue, KeyInfo
  - SignedInfo Elements:
    - CanonicalizationMethod
    - SignatureMethod
    - Reference
  - Canonicalize the SignedInfo element using the CanonicalizationMethod
  - KeyInfo Elements:
    - KeyName

Preventing Query/Response Interception

- Three steps to verify an XML signature (Done by the XML Firewall):
  - Canonicalize the SignedInfo Element
  - Verify the integrity using value contained in Reference element
  - Verify the Signature using the Signers public/private key

- Now the message is verified to have not been altered, and it's source is authenticated – the XML firewall will allow the query to be sent to the Web Service Server.

Identity Theft

- In the past 5 years, over 5 million people have been victimized by identity theft
- One common for of identity theft is sending credit cards, SIN numbers, and other personal information over the Internet
- Unprotected information can be viewed by hackers during online transactions

I have never been happier, or prouder, to be Seymour Skinner!
You're not Seymour Skinner!

For Your Eyes Only

- One of the main goals of WS-Security is maintaining confidentiality
- No one except the sender and receiver should see the information being sent through the web service
- Encryption and digital signatures are common forms of protecting information
- The two major types of cryptography are Asymmetric and Symmetric (discussed earlier)

Levels of Encryption

- Transport Level:

  - Encrypt the communication protocol used by web services (e.g. TCP/IP, HTTP)
  - SSL is standard for encrypting TCP/IP and was developed by Netscape (In your face Microsoft!)
  - Easy to implement in HTTP servers
  - Ideal for intranet where content is tightly controlled and monitored
  - WS-Security aims to maintain interoperability among different protocols. Transport level encryption restricts web-services to SOAP over HTTP only

XML Level

- XML Encryption:

  - Defines an XML schema that represents an encrypted image, document, or XML file
  - Allows you to encrypt the entire file or just parts of the message
  - Can also be used in other areas, such as encrypting database entries

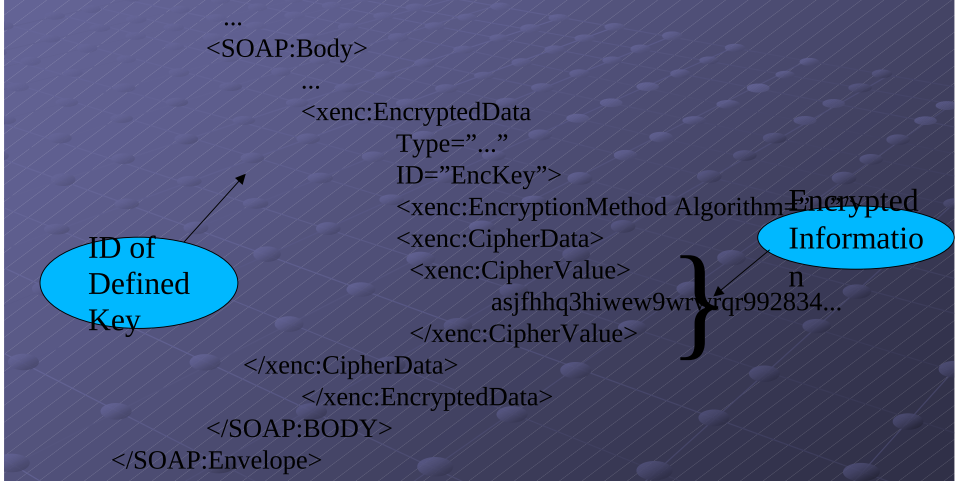
How Does SOAP use XML Encryption

Simple Encryption Example

```
<?xml version=”1.0” encoding=”utf-8”?>
<SOAP:Envelope xmlns: ...>
        ...
          <wsse:Security>
                    ...
[redacted]
                    </xenc:EncryptedKey>
                    <ds:Signature>
                              ...
                    </ds:Signature>
          </wsse:Security>
          ...
```

Symmetric key definition

Encryption Example Cont.

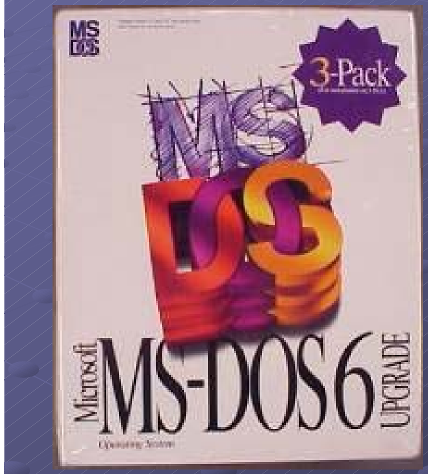


```
    ...
  <SOAP:Body>
          ...
        <xenc:EncryptedData
                Type=”...”
                ID=”EncKey”>
                <xenc:EncryptionMethod Algorithm=”
                <xenc:CipherData>
                  <xenc:CipherValue>
                        asjfhhq3hiwew9wrwrqr992834...
                  </xenc:CipherValue>
        </xenc:CipherData>
        </xenc:EncryptedData>
  </SOAP:BODY>
</SOAP:Envelope>
```

ID of Defined Key

Encrypted Information

Denial of Service (DoS)



- Incident in which a user or organization is deprived of the services/resource they would normally expect to have
- Does not usually result in the theft of information or other security loss

Typical types of attacks

- Buffer Overflow: send more traffic to the network address than the size of its data buffers
- SYN Attack: send numerous connection attempts very rapidly, then fail to respond. The first packet is then left in the buffer and legitimate requests can no longer be fulfilled.

Typical types of attacks cont.

- Teardrop Attack: exploits IP’s fragmentation of packets. Attacker inserts a confusing value into the offset of the second or later packets, which, if not properly dealt with, causes the system to crash.
- Viruses: generally not specifically targeted, but simply unlucky enough to be affected.

Typical types of attacks cont.

- Smurf attack: attacker sends an IP ping request to a site, specifying it to be broadcast to numerous hosts within the network. The ping request is spoofed to look like it is from the target site. When the receiving site responds, the target site receives all the ping responses and can be overwhelmed.

Denial of Service Prevention

- Sadly, no real “cure”
- Much of it lies with the programmers to make a secure site, not vulnerable to overflows, etc.
- Admin must make sure everything is kept up to date
- Many times, simply must learn from your mistakes!

Web Services Security

WestJet Vs Air Canada

A Case Study

Overview

- 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.
- http://www.ctv.ca/servlet/ArticleNews/story/CTVNews/1089309424645_84718624?s_name=&no_ads=#

Importance

- By gaining access to this confidential data, Air Canada claims WestJet has been able to:
  - Identify and target Air Canada’s most profitable routes.
  - Plan expansion into new routes.
  - Adopt pricing strategies aimed at forcing Air Canada out of markets
- “By Knowingly misappropriating Air Canada's confidential information, Westjet has gained a valuable springboards in starting new routes and termination other routes, both within Canada and the United States, and avoiding costly mistakes,” Air Canada’s court filing said.

Prevention and Detection

- The former employee had legitimate access.
- How could Air Canada have detected/prevented this breach?
  - Wisdom of having an online accessible database of all ticket sales?
  - Wisdom of having said database accessible to former employees.
  - Former Employees have access to this application, for the purpose of booking their free flights every year.

Expected User/Application Behavior

- Air Canada alleges that WestJet accessed it’s site, using this employees ID, 240,000 times between May 15, 2003 and March 19, 2004.
  - ~240,000 accesses in a ~300 day period
  - ~800 accesses per day
  - ~33 access per hour
  - ~2 accesses per minute
- Common behavior?

Question to think about.

- Is the environment appropriate?
  - Should it be available to the world?
- Is it necessary?
  - Is it more powerful than it needs to be?
- Is it abusable?
  - Can abuse be detected?

Sources

- www.hanford.gov/oci/maindocs/ci_r_docs/Air_Canada_accuses_WestJet_of_**espionage**.pdf
- www.**jetsgo**.net/en/086rel_20040908_01.shtm
- http://www.ctv.ca/servlet/ArticleNews/story/CTVNews/1089309424645_84718624?s_name=&no_ads=

Sources

- http://www.developer.com/design/article.php/10925_2171031_1
- http://webservices.xml.com/pub/a/ws/2003/04/01/security.html
- http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnglobspec/html/ws-security.asp
- http://www.webservicesarchitect.com/content/articles/fernandez01.asp
- http://www-106.ibm.com/developerworks/webservices/library/ws-secmap/
- http://whatis.techtarget.com/definition/0,289893,sid9_gci213591,00.html
- http://webservices.xml.com/pub/a/ws/2003/03/04/security.html
- www.hanford.gov/oci/maindocs/ci_r_docs/Air_Canada_accuses_WestJet_of_**espionage**.pdf
- www.**jetsgo**.net/en/086rel_20040908_01.shtm
- http://www.ctv.ca/servlet/ArticleNews/story/CTVNews/1089309424645_84718624?s_name=&no_ads=