

Web Services Security Threats and mitigation

Jonathan Gershater
Layer 7 Technologies



October 29th, 2008



Agenda

- Web Services review
- Web Services Threats
- Web Services Mitigation
- Conclusion and Questions

Web Services Stack

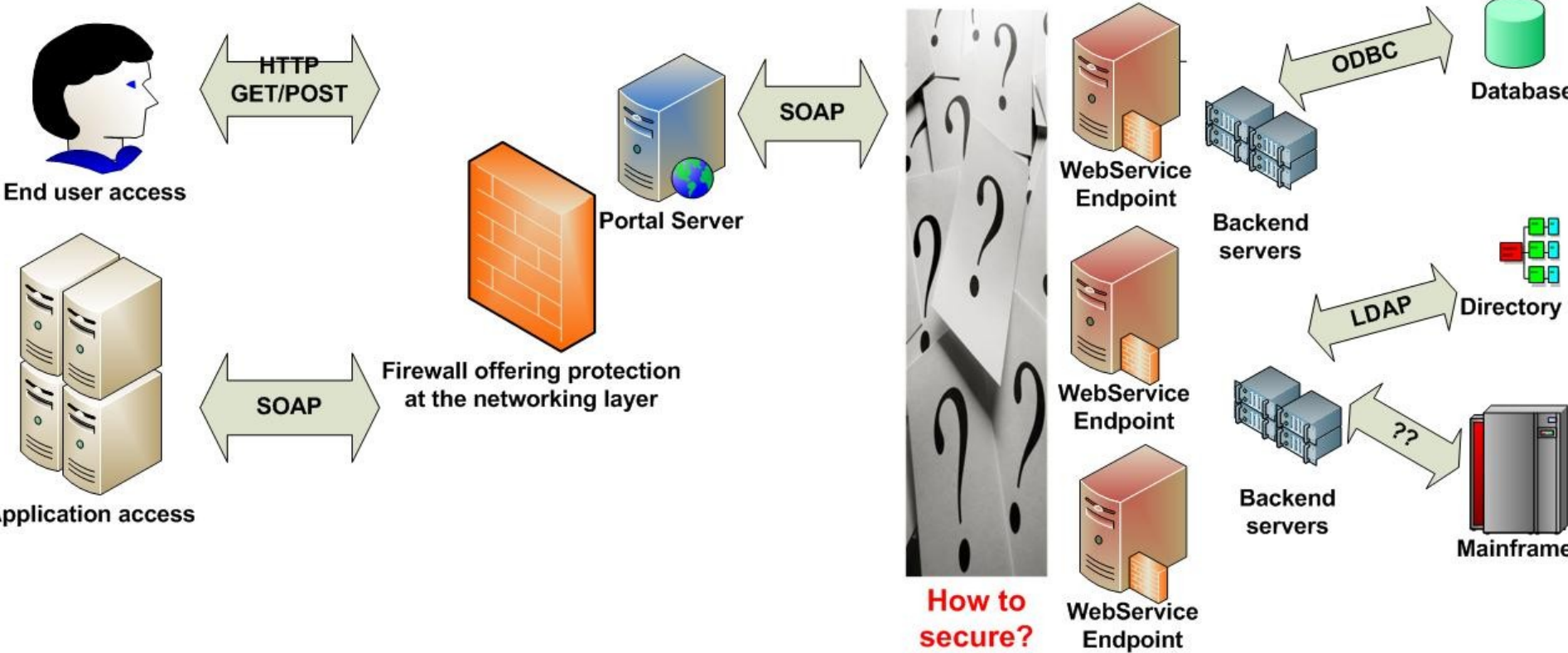
Presentation Layer XML, AJAX, Portal, Other
Security Layer WS-Security
Discovery Layer UDDI, WSDL
Access Layer SOAP, REST
Transport HTTP(S), JMS, Other



Common web services scenario

Integrated applications offering loosely coupled services consumed by

- * a user at a browser (via a portal)
- * other applications



Common Threats

- 1 Message Alteration
- 2 Confidentiality
- 3 Falsified Messages
- 4 Man in the Middle
- 5 Principal Spoofing
- 6 Forged claims
- 7 Replay of Message Parts
- 8 Replay Entire message
- 9 Denial of Service

<http://www.ws-i.org/Profiles/BasicSecurity/SecurityChallenges-1.0.pdf>



Data integrity

Data has not been changed, destroyed, or lost in an unauthorized or accidental manner.

Data in a web services context means a SOAP message (or portions: one or more SOAP headers, a body, or attachments).

Transport Data Integrity – securing the the method of transport (bus/car/train...)

Resolution:SSL/TLS

SOAP Message Integrity – securing the message (the passenger, after he gets off the bus)

Resolution: XML Signatures

Data confidentiality

Data is not made available or disclosed to unauthorized individuals or processes.

Eavesdroppers or other unauthorized parties cannot view confidential message content.

Transport Data confidentiality – (*again securing the bus*)

resolution: SSL/TLS

“Transport confidentiality is generally inappropriate for these requirements since it terminates with the transport session “

SOAP Message confidentiality

resolution: XML encryption



Message uniqueness

A specific XML message is not resubmitted for processing.

Attacker could resend all or selective parts of a message causing undesirable side effects. Or an XML DOS.

Threat #7 & 8 & 9

Transport layer

SSL/TLS: between the node generating the request, insuring for downstream nodes that this is a unique request

Message layer

The sending SOAP nodes: Sign the SOAP message header, creation time [expiration time] and optional user data.

Receiving node: Verify the signature and check that the creation time is not stale.

XML parsing Threats

A malformed IP message doesn't break a router.

But....a malformed XML message can disable the XML parser inside an application.

Service Code Threats

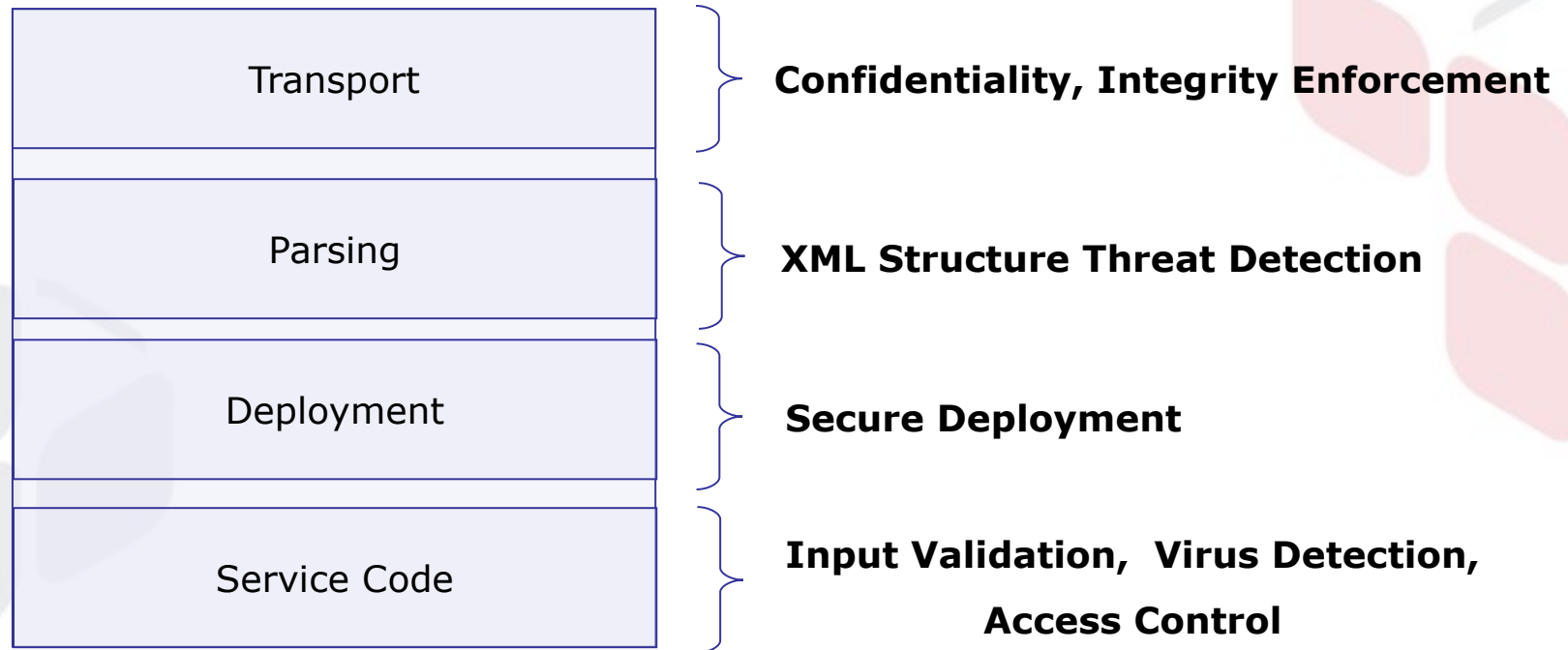
SQL Code Injection

- Code is injected within an XML element
 - ♦ `<SQL>SELECT name FROM DB1 WHERE name = 'Joe'</SQL>` changed to
 - ♦ `<SQL>SELECT * From DB1 WHERE name = *</SQL>`

Virus/Spyware/Malware Injections

- XML Attachments (MTOM, DIME, MIME) are used as a delivery mechanism for virus

Web Services Hardening



Secure Deployment

UDDI and WSDL are the yellow pages of your available service. But, this registry should be secured.

UDDI, WSDL

Virtualize Internal Services to consumers through creation of virtual endpoints described by generalized WSDL and UDDI descriptions.



Input Validation (parameter tampering)

The service code layer

- Is where business logic is coded
- Easiest to break into
- Thus most critical to protect.

Basic Parameter Validation

- Avoid string data types, allows anything to enter.
- Validate Integer values for length.

Specifically Parameter Validation

- Example: Social Security Numbers or zip codes

XML Schema provides a tool to validate message parameters according to predetermined business usage.



Virus Detection (virus, spyware, malware)

XML cannot execute a virus but can be its courier, via:

- SOAP with Attachments, MTOM, WS-Attachments
 - ◆ A Web Service executes an application stored within the SOAP attachment or issues the SOAP attachment to another system for later execution.
 - ◆ **Mediation:** Attachments should be scanned with a virus scanner, (unavailable in traditional virus scanning engines.)
- Base64 encoded malicious program
 - ◆ Web Service or other application is programmed to decode BASE64 values and execute resulting binary.
 - ◆ **Mediation:** If this is the intended purpose for a large XML element and validation cannot be accomplished, the element should be decoded and then scanned by a Virus Scanning Engine, (unavailable in traditional virus scanning engines.)

Authentication:

Who are you?

- *HTTP BASIC, LDAP, KERBEROS etc.*

Authorization:

What are you allowed to do?

- *URL access*
- *Access web service operation*

Auditing

Who did what?

- *Compliance*
- *Audit trails etc.*

XML Secure Span appliances and software solutions

- XML Appliance
- Software installed on your servers
- VMWare image
- Optional XML secure tunnel "XML VPN"

