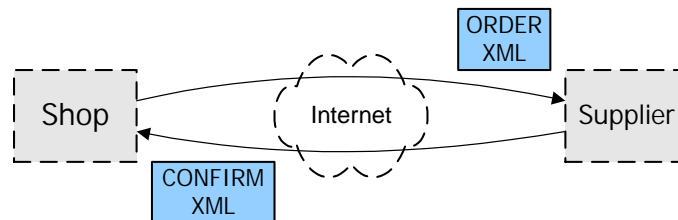


Web Service Security

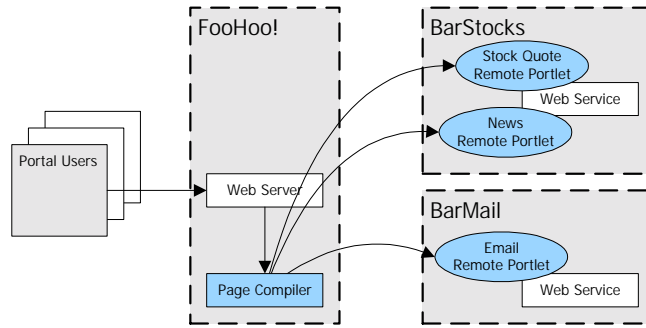
Jan P. Monsch
jan.monsch@csnc.ch

- Use
 - B2B integration with XML documents → SOAP
 - Automation of business processes
 - Fast development cycles



- Example
 - Order processing systems

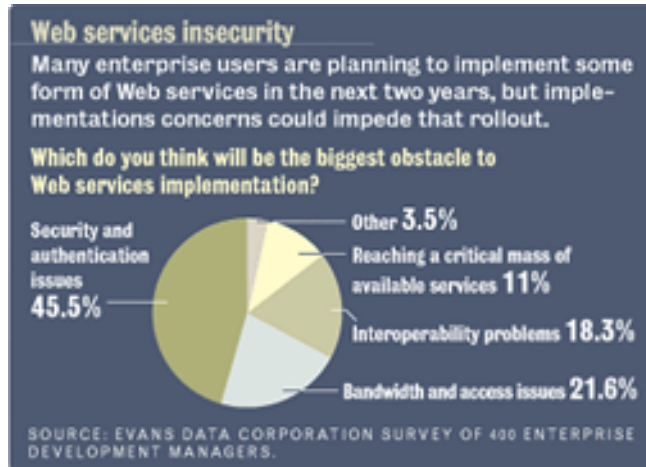
- Example
 - Integration of Web Services into one web portal



- Data- or presentation-oriented Remote Portlets

- Microsoft
 - Pushes Web Service technology to gain greater momentum with their .NET platform in businesses
 - Positioned against J2EE Platform
 - Web Services integral part of .NET
 - Out of the box support in .NET development tools
- IBM
 - Biggest seller of IT services and products
 - Pushes SOAP standards
- Standardization committees
 - W3C: XML, SOAP, XML Signing, etc.
 - OASIS: Web Service Security, Web Portlets

▪ Market notion about Web Services:



▪ Demonstration Web Services

- Amazon
 - <http://www.sephiroth.it/tutorials/flashPHP/webService/files/amazon.html>
- Google
 - <http://www.flash-db.com/Google/>

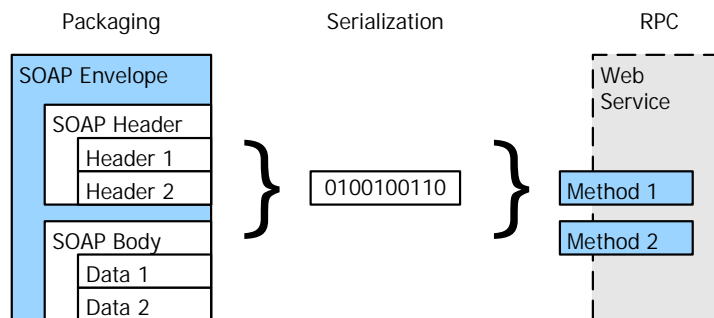
→ For end-users Web Services are invisible

- SOAP (Simple Object Access Protocol)
 - Base Protocol for Web Services
 - W3C Standard
 - XML-based RPC protocol (like CORBA)
 - Design Goal: Simple and Extensible

- True cross-everything protocol
 - cross-platform: Java, .NET, etc.
 - cross-operating system: Windows NT, Unix
 - cross-programming language: Java, C#, VB, C++

- Using existing transport protocols
 - HTTP, SMTP (Multi-Recipient), raw TCP/IP, FTP, MQ

- SOAP standard specifies three items
 - Packaging model (SOAP envelope)
 - Serialization mechanism (SOAP encoding rules)
 - RPC mechanism (SOAP RPC representation)



- SOAP standard does **NOT** specify
 - Session Handling, since SOAP itself is stateless
 - Security mechanisms
 - Distributed garbage collection
 - Object-by-reference

- WSDL (Web Service Description Language)
 - Definition of a Web Service (like CORBA IDL):
 - Business functions, e.g. SearchGoogle
 - Data structures, e.g. key words, language, etc.
 - Only thing needed to write an application
- UDDI (Universal Description, Discovery & Integration)
 - Directory with published WSDL definitions

- Demonstration WSDL
 - SOAP clients can be generated on-the-fly

→ XMLSPY

- Register to get a access token
- Download Google API package
- Extract GoogleSearch.wsdl
- Load WSDL of Google search
- Select SOAP message
- Enter parameters
- Send message to Google
- Enjoy results

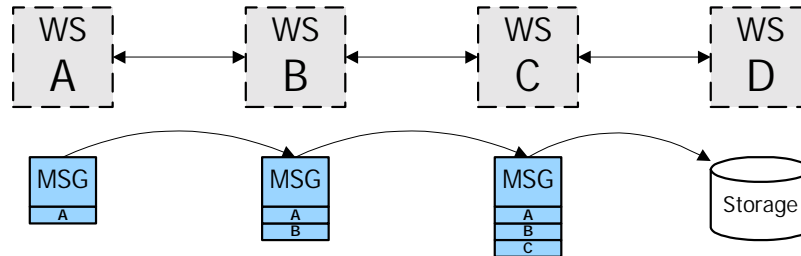
- Web Applications
 - Business logic is hidden behind presentation server



- Web Services
 - Opening up business logic for business partners
 - Direct interface for developers to business logic



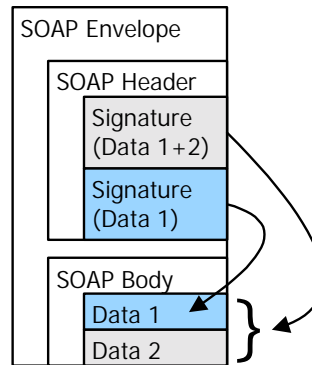
- SOAP are meant to be passed through chained WS



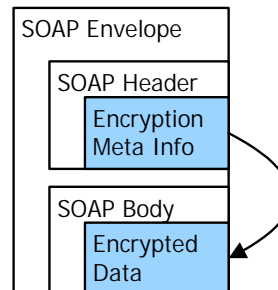
- Mutual authenticated **SSL is not enough**
 - Protection of message during transport only
 - No end-to-end security in chained web services
 - No protection of message in storage, e.g. database

- Countermeasures:
 - SOAP message level security needed
 - XML Signatures & XML Encryption

- Standard
 - W3C Specification, can be applied to SOAP
 - Using existing technology, e.g. X.509, RSA, SHA1
- Features
 - Partial XML document signing possible
 - Overlapping signatures possible.
- Solves
 - End-to-end message authentication & integrity



- Standard
 - W3C Specification, can be applied to SOAP
 - Using existing technology, e.g. 3DES, AES
- Features
 - Partial XML document encryption possible
- **Caveat**
 - In combination with encryption – Signatures need to be encrypted to prevent crypto analysis
- Solves
 - End-to-end message confidentiality

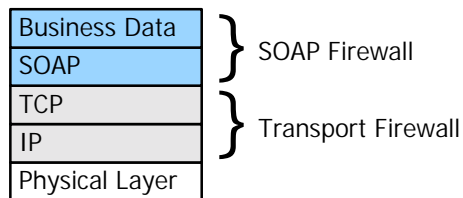


- SOAP message resend or replay due to
 - System problems
 - Attacker trying to reuse a signed message

→ Countermeasures

- Introduce serial number to SOAP message
- Add expiry timestamp
- Sign serial and timestamp

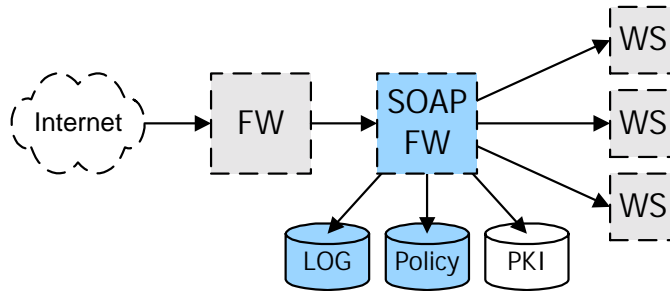
- SOAP utilizes HTTP or SMTP for transport
 - Messages slip through firewalls
 - Standard firewall does not inspect the application layer



→ Countermeasures

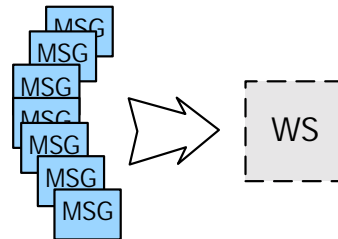
- SOAP firewalls needed

- SOAP Firewall
 - is plugged between standard FW and Web Services
 - checks SOAP messages for integrity
 - authenticates, authorizes messages



- cannot inspect encrypted content!

- XML parsing
 - SOAP messages larger than a binary format
 - Slow, since XML must be checked against schemas

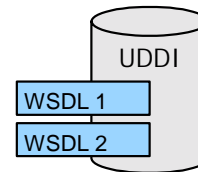


- Possibility of Denial of Service attacks

→ Countermeasures

- SOAP firewall needed
 - Denial of Service prevention
 - Fast XML parser

- UDDI registry
 - stores Web Service descriptions



- Unprotected registries allow an attacker to
 - collect information about the available Web Services
 - immediately generate a SOAP client to start penetration

→ Countermeasures

- Deny anonymous requests to the UDDI registry
- Restrict user access to UDDI objects to a minimum

- Same issues apply to Web Services as to web applications, e.g.
 - SQL injection
 - Verbose error messages (Developer level messages)
 - Data isolation
 - Fine grained function access control
 - Session handling, when needed
 - Incorrect configuration of servers
 - Exception handling
 - Buffer Overflows

→ Countermeasures

- Security-aware application designers and developers

- Web Services
 - Emerging technology → Hype
 - Powerful for both developers and hackers
 - Not proven to fulfill business needs

- Security
 - Complex since security does not end at company perimeter
 - Ink on the security standards still very wet

- Standards
 - SOAP, XML Signature, XML Encryption → www.w3c.org
 - Web Service Security, SAML → www.oasis-open.org

- SOAP Client:
 - XMLSPY 5 Enterprise Edition → www.xmlspy.com

- Public Web Services:
 - Google Search → api.google.com
 - Amazon → associates.amazon.com/exec/panama/associates/join/developer/kit.html

- SOAP Firewalls
 - VordelSecure → www.vordel.com
 - SOAP Content Inspector → www.xtradyne.com
 - Reactivity Service Firewall → www.reactivity.com

- Other recommended reading:
 - "A Guide to Building Secure Web Apps" → www.owasp.org
 - "Some thoughts about SOAP versus REST on Security" www.prescod.net/rest/security.html