SOA Security – Is it a Treacherous Journey?

Doc Shankar
IBM Distinguished Engineer
Federal CTO Office
dshankar@us.ibm.com

Agenda

- What is service-oriented computing?
- What is the business value of SOA?
- Why is security harder in SOA?
- How do we get attacked?
- Have the security fundamentals changed?
- Do standards solve everything?
- What are the security functions & technologies required for SOA?
- What are the security implications of Web Portals?
- What does it take to securely web service-enable legacy applications?

Key terms and acronyms*

- Trustworthy Computing
- Trusted Computing
- Trusted OS
- Trusted Guards
- Trust
- Information Security
- Information Assurance
- Privilege
- Portal
- Sandbox
- Governance
- End-to-end security
- PKI
- XML
- SOAP
- WSDL
- UDDI
- SAML
- XACML
- TTP
- PDP
- PEP
- WS – *

* Not a complete list

What is Service-Oriented Computing?

- Make a collection of software services accessible via standardized protocols whose functionality can be automatically discovered and integrated into applications
- Distributed Systems Evolution
  - Computer
  - Computer Utility
  - Computer Networking
  - LAN
  - Client/Server
  - Internet
  - Web Applications
  - Web Services
  - Cross Organizational Web Services
  - SaaS/PaaS/IaaS
  - Cloud Computing

SOA is a computing paradigm emphasizing dynamic discovery, composition & interoperability

Business value of SOA

- Tremendous opportunity for companies to integrate
  - Across departments
  - Across systems
  - Across enterprises
- Integration helps
  - Simplify business processes
  - Discover business services
  - Automate business processes
  - Improve speed to market
  - Allow companies to react quicker to changes
  - Share data & services
  - Rejuvenate legacy systems

To the end users, this is nirvana. But to security folks this is their worst nightmare!

Why Security is harder in SOA?

- Greater accessibility to data
- Dynamic application-to-application interaction
- Dynamic discovery
- Relative autonomy (lack of human intervention)
- Trust establishment
- Cross organizational interactions
- Cross domain/MLS interactions
- High assurance

Attack Categories

- Unsafe Programs
- Misconfigured Programs
- Buggy Programs
  - Buffer Overflows
  - Parsing Errors
  - Formatting Errors
  - Bad input to cgi bin
- Malicious Programs
  - Trojans
  - Virus
  - Worms
  - Rootkits
  - Botnets
- Applications
  - Cross site scripting
  - Injection flaws
  - Malicious file execution
  - (See top 10 on OWASP)
- Eavesdropping
- Spamming
- IP Spoofing
- Phishing
- Pharming
- DoS/DDoS
- People
  - Social Engineering
  - Weak passwords
  - Sloppy Admins.

Attack Categories – Web Services

- Message Alteration
- Loss of Confidentiality
- Falsified Messages
- Man in the middle
- Principal Spoofing
- Forged Claims
- Message Replay
- Denial of Service
- Privilege Escalation
- Command Injection
- Malicious Code Attacks

Customer Pain Points

- P - Privacy (Confidentiality)
- A - Authorization (Authentication)
- I - Integrity
- N - Non-Repudiation

T

I

The fundamentals of security haven't changed for a long time. However, in the last few years due to viruses, worms, intrusions & DDoS attacks, another one has been added called "Assured Information Access".

The Problem with Standards

- Too Many
  - Network layer (IPSec)
  - Transport Layer (SSL/TLS)
  - XML Security (XML Encryption, XML Signature)
  - Message Security (WS-Security, WS-SecureConversation)
  - Reliable Messaging (WS-Reliability, WS-RelaibleMessaging)
  - Access Control (SAML, XACML)
  - Policy (WS-Policy)
  - Security Management (WS-Trust, XKMS)
  - Identity Management (WS-Federation, Liberty Alliance)
- Overlapping
  - OASIS, W3C, IETF, Liberty Alliance,…
- Interoperability
  - Constantly being updated

There is a need for more formal specification of standards

Security Functions/Technologies

- Authentication
- Identity Management
- Trust Management
- Web Services Policies
- Authorization
- Message Confidentiality & Integrity
- Accountability/Auditing
- Non-Repudiation
- Availability
- Discovery Service Security
- Contract Negotiation
- Secure Software

Authentication

- Problem
  - How do you allow a user authenticated on one system to use services within a SOA?
- Functions
  - Identify participants in transactions
  - Limit access to resources
  - Create personalizations
  - Authenticate on one system & use services within SOA (SSO)
  - Service Chaining (Impersonation)
- Technologies
  - HTTP-based token authentication
  - SSL/TLS-certificate based authentication
  - SOAP-based token
  - WS-Security (supports UN/PW, X.509, Kerberos, SAML)
  - WS-Trust/WS-Federation (supports federated authentication)

WS-Security provides authentication (as well as Confidentiality & Integrity at the SOAP message level)

Identity Management

- Problem
  - How do you securely identify within and across an organization?
  - Org A may use X.509, Org B may use Kerberos tickets & Org C may use UN/PW
- Functions
  - Entity's registration & verification
  - Issuance of digital identity & credentials
  - Used for authentication at SOA entry point
- Technologies/Architecture
  - Isolated Identity management
    - Service providers act both as credential provider & identity provider
  - Federated identity management
    - A group of providers agree to recognize identifiers from one another
    - Cross organizational SOA more involved
  - Centralized identity management
    - Providers rely on a single TTP to provide credentials & identifiers to requestors
    - In a cross-organizational SOA, orgs need to trust each others TTP

In an SOA, an entity's identity forms the basis for both authorization & trust

Trust Management

- Problem
  - How do we establish trust relationship between services
- Functions (Trust federation frameworks)
  - Liberty Alliance(an early alliance for addressing FIM;converged to SAML 2.0))
  - Shibboleth(initiative for addressing FIM;open source; converged to SAML 2.0)
  - WS-trust (extends WS-security with security token management)
  - WS-Federation (expands WS-trust with federation)
  - Higgins (Open Source Initiative, User control over their private information)
- Technologies/Architecture
  - Pair wise trust
    - Each entity shares key information
    - SAML assertion verification enough
  - Brokered trust
    - TTP used to exchange key information
    - Hard to prevent communication between entities
  - Community trust
    - Use of an external PKI
    - Remote entity's key retrieved through the PKI interface

Web Services Policies

- Problem
  - WSDL describes protocol bindings & message formats; what about meta data?
- Functions (Web services policy framework)
  - Policy assertions defined for a number of WS-* specifications
  - Use XML to write these expressions
- Technologies/Architecture (evolving)
  - WS-Security Policy
    - Assertions specifying confidentiality, integrity & information about security tokens
  - WS-RM Policy
    - Parameters necessary for reliable message delivery
  - WS-Addressing
    - Mechanisms that allow web services to communicate addressing information (routing data)

Authorization

- Problem
  - How do manage authorization & access control credentials in a SOA?
- Functions (Authorization models)
  - (RBAC) Role-based access control
  - (ABAC) Attribute-based access control
  - (PBAC) Policy-based access control
  - (RAdAC) Risk adaptive access control
- Technologies/Architecture
  - SAML
    - Security assertions that specify how an entity was authenticated, by whom, entity's attributes, what can it do & access, etc.
    - Written in XML
    - Example – "Paul was authenticated on 06/22/07, 12:13:08 via X.509 & the assertion is valid for one hour"
  - XACML
    - Security assertions that specify what an authenticated entity can do or access
    - Policies written in XML
    - Can be used with SAML (SAML token can say check with a specific XACML policy for access)
    - Example – "members with group attribute of "designers" can perform read & write on http://server.ibm.com/design/docs.html

Message Confidentiality & Integrity

- Problem
  - TLS assures the security of messages only during transmission!
- Functions (Message layer security)
  - End-to-end security (intermediaries can’t be trusted)
  - Security needed at message layer independent of transport layer
  - Security of stored messages
- Technologies/Architecture
  - HTTPS
    - HTTP over SSL/TLS
  - XML Encryption
  - XML Signature
  - XKMS
  - WS-Security
  - XML Gateways

Accountability/Auditing

- Problem
  - How do you implement accountability in a (dynamic) service chain?
  - Lack of auditing standards for Web Services
- Functions
  - Auditing requires a secure, distributed logging facility
  - Needs to use digital signatures for integrity
  - All intermediaries need to log information about captured SOAP messages
  - Many COTS & GOTS products use their own non-standard logging mechanism
- Technologies/Architecture
  - NIST SP 800-92 provides guidance, but not specific to SOA
  - Need for a specific SOA audit framework using other WS-*

Non-Repudiation

- Problem
  - How do you ensure that a party to a contract or a communication cannot deny the authenticity of their signature on a document or the sending of a message that they originated ?
  - Would a digital signature always guarantee non-repudiation?
- Functions
  - Non-repudiation is a way to guarantee that
    - the sender of a message cannot later deny having sent the message
    - the recipient cannot deny having received the message
  - Digital Signature
- Technologies/Architecture
  - XML Signature
  - Identrus for secure e-payments

Availability

- Problem
  - How do you provide assured information & service access?
  - What is the relationship between availability, QoS & Reliability?
- Functions
  - Availability is intended to ensure that QoS & Reliability are maintained even the Web Service is being attacked?
  - Design to include redundancy of critical functions
  - Design exception and error handing capabilities
  - Gracefully degrade performance
  - Prevent deadlocks & recursion
- Technologies/Architecture
  - WS-Reliability
  - WS-ReliableMessaging

Discovery Service Security

- Problem
  - How does one locate the different web services on the internet?
- Functions
  - UDDI is an open industry initiative, sponsored by OASIS, enabling businesses to publish service listings and discover each other and define how the services or software applications interact over the Internet.
    - White Pages - address, contact, and known identifiers
    - Yellow pages - industrial categorizations based on standard taxonomies
    - Green pages - technical information about services exposed by the business
  - UDDI v3 supports authentication, authorization & digital signatures
- Technologies/Architecture
  - UDDI v2
  - UDDI v3
  - WSDL
  - Service publishing API
  - Service Inquiry API

Web Portal (WP) Security & Trust

- WPs provide the connectivity to SOA – 2 roles
  - Web services for users
  - Web requestor for web services in the SOA
- WPs interact with provider services & identity providers
  - Perform actions on behalf of users
  - Middleman between the user and services
- WP security requirements
  - Provide authentication information about users
  - Coordinate with provider's authorization mechanisms
  - Provide auditing services
  - Provide confidentiality, integrity & non-repudiation of messages to both the user and provider
- WP security solutions
  - SAML (assertions for authentication & authorization)
  - WS-security (confidentiality, integrity & non-repudiation)
  - WS-trust (federated authentication & authorization)

Web service-enabling legacy applications

- Allows SOA applications to leverage the legacy functionality
  - Exposes legacy applications & back end data bases to new threats
  - Provides new avenues of attacks
- Making it discoverable adds more exposure/risk
- Enabling web applications
  - Use of SSL/TLS with PKI certificate or uid/pw
  - Web application has to trust the authentication system's assurance
  - Web application performs it's own authorization or uses a central authorization service
  - Each application process must adhere to principal of least privilege
- Enabling non-web applications
  - Requires mapping
    - Web service credentials to legacy credentials
    - Incoming/outgoing web service messages to legacy application protocols
  - Application has to trust the authentication system's assurance & mapping
  - Application performs it's own authorization or uses a central authorization service
  - Enabling easier if business objects & APIs are built into application
  - Enabling harder if business logic & GUI are intertwined

If possible, upgrade legacy applications to support SOA technologies; otherwise mapping of identities & privileges needed

Conclusions

- Standards do not provide all of the required properties to develop robust, secure, and reliable web services
  - QoS
  - QoP
  - Availability
  - Audit
  - Accountability
  - Service description
  - Automatic service discovery
  - Electronic negotiation
- Overlapping standards
- Standards maintained by different organizations
  - W3C
  - OASIS
  - Liberty Alliance
  - Industry forum headed by MS & IBM
- Conflicting standards
  - WS-Reliability/WS-RelaibleMessaging (W3C, OASIS, Industry forum)
  - Liberty Alliance, WS-Federation, Shibboleth
  - Transitive trust models evolving
- Security & integrity of UDDI is hard
- Non-repudiation hard