Security in a SOA-Based Enterprise

May 23, 2005

Paul Patrick
Chief Security Architect,
BEA Systems

Paul.Patrick@bea.com

Challenges Facing an Enterprise

- Increase operational efficiency
  - Improve productivity from existing systems
  - Streamline and optimize mission process
  - Built-in re-use and adaptation, no 'starting over'
  - Turn IT into a mission asset
- Provide a unified view of the mission
  - Eliminate silos and create end-to-end visibility into and across the business
  - Provide specific views of processes and information
- While …
  Achieving faster time to value

Service Oriented Architecture Vision

Server-centric	Network-centric
Monolithic Architecture	Service architecture
Manual Configuration	Adaptive Configuration
Limited Discovery	Dynamic Discovery
Hub-based	Loosely coupled
Hard-coded physical addressing	Service-based Naming & Addressing
Early Binding	Late Binding
Point-to-Point Routing	Adaptive Routing
Silo Security	Distributed Security

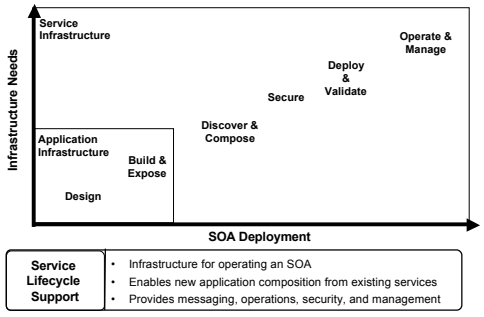
Review of Today's Environments

- Today's environments are "liquid"
  - Changes must be made in moments, not weeks in order to satisfy changes in operational situations
    Adapting to changes in mission situation
    Re-tasking a mission for a new target of opportunity
- Today's environments are distributed
  - Impractical to require manual changes at each application
  - Impractical to redeployment for config/security changes
- Today's environments are non-stop
  - Changes can't require restarts
  - Requirements for 24x7 becoming more common

Operational Aspects Now Critical

- Operational aspects now take "center stage"
  - Focus shifts from ease of development to deployment
  - Security, monitoring, diagnostics, configuration
- These aspects tend not to be thought of until deployment
  - Leads to re-design or re-architecture which causes delays
  - Tends to result in "patch work" approach
    Home-grown solutions to "work-around" issues
    Perimeter security with no defense for inside attacks
    Operation monitor/control integration done ad-hoc
  - Results in "Accidental Architecture"

Service-Based Infrastructure vs. Application Infrastructure



SOA Concepts with Security Implications

- Heterogeneous Environments
- Communication Styles
- Distributed Security Enforcement
- Service Proliferation

Heterogeneous Environments

- Today's environment are heterogeneous
  - Each with its own identity and authentication models
  - Each with its own security policy enforcement schemes
- Identity "bridging" key to unified security model
  - WS Security specs do NOT fully address the issue
  - Many apps do not support concept of "assertion"
  - Wrapping legacy apps and proxies still leave issue of identity mapping
- Unified policy language key to cost effective management

Identity Propagation Approaches

- "Message-Based Authentication"
  - WS-Security (Standard)

    "Signer" of message used as identity
    Not always proof of sender's identity
- "Assertion"
  - SAML (Standard)

    "Conical" representation of identity and attributes
    Mechanism to obtain attributes associates with an identity
- "Re-Authentication"
  - WS-Trust (Specification)

    Mechanism allowing one form of "trusted" credential to be "exchanged" for another form of credential

Communication Style

- Approach may require "plumbing" understanding
  - Synchronous communications

    Protection typically by secure connection (e.g. SSL/TLS)

    Identity dropped at connection termination point

    Requires all identity to have digital certificates
  - Asynchronous communications

    Introduce new threats to be considered

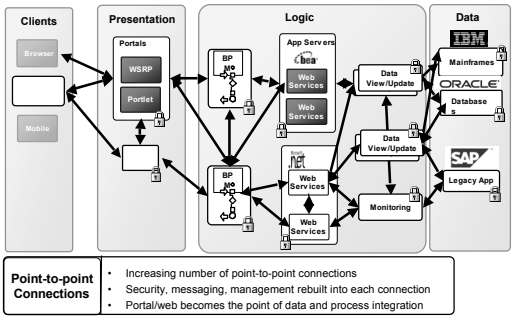
    Messages "at rest" open to tamper/unauthorized viewing due to potential covert channels

    Developers must be aware of counter measures

    Digital Signature for tamper detection

    XML Encryption for confidentiality
- Not all end points support same mechanisms

Point-to-Point Service Connections



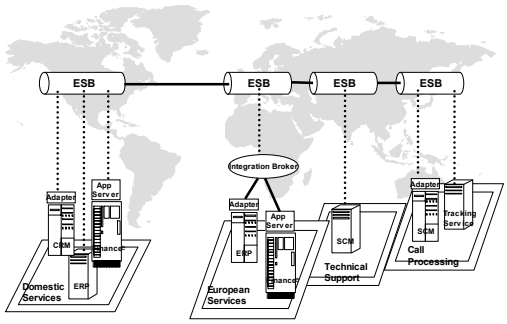
Distributed Security Enforcement

- Distributed environments tend to have multiple enforcement points
  - Each point typically has a unique policy language
  - Increases operations cost as number of silos increase
- Typical solution is enforcement at the perimeter
  - Leaves apps/components open to insider attack
- "Decisions in Context"
  - Decisions must have access to context to allow security enforcement to be removed from application logic

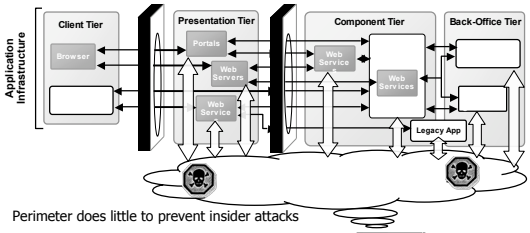
    Remote decisions may require copy of request and other environment information to be forwarded
  - Accountability and Traceability key to auditing requirements

    Often required to not only report attempted operation but also what was supplied and returned

Autonomous and Federated: "Virtual Enterprise"



Classic Perimeter Security Scheme



Perimeter does little to prevent insider attacks

- Who's watching what's occurring
- Where's the sensitive data protection
- *Insider represents greatest threat due to understanding of organization's mission and systems*

Security Enforcement: What Do People Do Today

- Enterprises forced to build aspects of security enforcement directly into business logic
  - Utilize 3rd-party security products directly
    - **Single vendor Lock-in**
    - **Complex integration into each application**
  - Roll their own policy throughout applications
    - **Security policy is "hard coded" into business log**
    - **Distraction from primary focus of building applications**
- Lacks the dynamic characteristics required by today's demanding "liquid" environments

Information Sensitivity

- Virtually all segments have data sensitivity issues
  - Insufficient to only checking if requestor is allowed to performed action
- Query-Based Security Dilemma
  - Control of information being returned is difficult
  - Classic authorization approaches provide insufficient granularity
- Key Concepts:
  - "Dynamic Separation of Duty"
  - "Need to Know"
- Audit trail of Access is MANDITORY

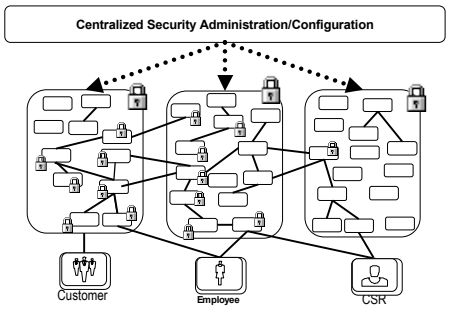
Security Policy Languages

- WS-SecurityPolicy (Specification)
  - XML-based language to describe security requirements of an endpoint
    - **Authentication mechanisms supported**
    - **Sign/Seal requirements**
- XACML 2.0 (Standard)
  - XML-based language to describe authorization/role entitlement policies
    - **Xpath resource references allows element level policy**
    - **Obligations could be used to control redaction**

Service Proliferation

- Proliferation as more services are available
  - Each new service has the opportunity to introduce "Yet Another Security silo"
- Each security silo
  - Increases the number of instances to be managed
  - Increases the risk of "voids" in security coverage
  - Increases the operational cost
- Results in "patch work" approach
  - Home-grown solutions to "work-around" issues
  - Least common denominator solution

Centralized Administration



"Liquid Security"

- The security aspects of an application can be changed from a central location in "real time"
  - Changes in security policy must be able to be done without rebooting, redeploying, or re-coding
  - Distribution of changes needs to be transactional to ensure integrity and consistency of enforcement

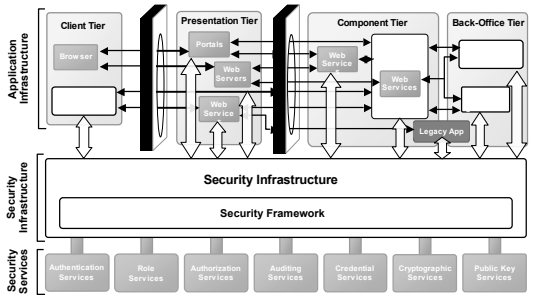
Application Security Best Practices

- Externalize management of identity and policy from application
- Externalize policy enforcement from business logic in application code
  - Support for separation of duties and responsibilities
- Protection enforced as close to target as possible
  - Provide "context" necessary to handle business-like decisions
- Service-based Security Architecture
  - Open, flexible, and extensible

Future Proofing Security Infrastructure

- Network-based security is an essential element of an overall enterprise security infrastructure, but its not enough by itself
- Applications must be an "active participant" in its protection
  - Focused on "controlling what the good guys can do"
- Agility, Flexibility, and Adaptability are key:
  - Standards provide extreme confidence, trust, and interoperability
  - Abstractions protect applications from flux in standards and technology
  - Evolution **must** be planned for as things will change

Infrastructure Approach to Application Security



Key Benefits of Infrastructure Approach

1	Strengthens Application Security •Common, consistent security infrastructure and services •Centralized control and visibility of security policy
2	Improves IT Efficiency •Re-use of security infrastructure and security services •Investment protection in current and future 3rd party security products through interoperability
3	Increases Business Agility •Burden of security policy administration shifted from developers to administrators •Developers can focus on more value-added activities
4	Facilitates Compliance (SOX, HIPPA, GLB, etc.) •Externalizing authorization and auditing from the application facilitates compliance with evolving regulatory and privacy initiatives

Summary

- Awareness of security implications as a result of:
  - Heterogeneity of the environment
  - Service Proliferation
  - Communication Styles
  - Distribution of Security Enforcement
- Remote decisions has implications
  - Reliability, Availability, Scalability, Performance
  - Perimeter approach leaves apps open to insider attack
- Service-Based Security Infrastructure
  - Provides isolation from changing standards
  - Allows flexibility

Thank You