Single Sign On In A CORBA-Based Distributed System

Igor Balabine
IONA Security Architect
Outline

• A standards-based framework approach to the Enterprise application security
• Security framework example: IONA Security Framework (iSF)
• Security framework based SSO solutions
• Summary
Why A Security Framework?

Security Framework:

- insulates middleware applications from the diverse and changing enterprise security infrastructures.
- provides a uniform, vendor-neutral, standards-based approach to communicating security-related requests across the enterprise.
- provides applications a single access point to multiple security services such as authentication, authorization, SSO, PKI, management, and notification services.

Security Framework binds applications with any enterprise security infrastructure!
Security framework approach avoids performance bottlenecks suffered by a centralized approach such as Super Directory!
Security Framework Architecture

Application

App ↔ SF Adapter

Native API calls: no changes in the application code!

Request/Response messages over IIOP/http(s): support for distributed and co-located deployments

Security Service (S2)

S2 ↔ ESS Adapter

Native API example:
- EJBContext.getCallerPrincipal()
- EJBContext.isCallerInRole()

Enterprise Security System (ESS)

Third party security system native protocol

J2EE or WS
Authentication and Authorization Services

- Authentication and authorization services are supported via dedicated adapters.
- Internal protocol: SAML – satisfies purposes and allows extensibility. Could be easily replaced if necessary if internal interface in the application SDK is generic.
- Required authorization models: coarse grain – RBAC (e.g. J2EE, Web Services), fine grain – DAC (e.g. CORBASEC, B2Bi).

**SAML protocol allows communicating arbitrary security assertions between applications and the Security Server!**
PKI Services

- PKI services are supported via dedicated adapters.
- Internal protocol: XKMS – powerful and extensible. Endorsed by industry leaders (Verisign, Entrust, Microsoft).
- Common use: integration with certificate stores.
- Advanced use: certificate validation services.

Many PKI vendors are expected to adopt XKMS: in such installations S2 PKI adapter becomes (almost) a pass through!
Framework Administration

- Solutions integrated with 3rd party systems are managed using native administrative tools, e.g. SiteMinder console for an enterprise which uses Netegrity SiteMinder.
- Framework provides out of the box facilities for Single Sign-On and authorization (RBAC and DAC) services for environments devoid of such functionality, e.g. Windows Domain.
- Framework Auditing Component co-located with the Security Server (S2) provides logs in standard formats (syslog, NT Event Log, Snort) easily consumable by event monitoring systems.

*Framework offloads administrative tasks to 3rd party tools where possible and provides components to manage custom security information!*
Single Sign-on and End-to-End Security

Auth. tokens are passed in http header or in SOAP message

SSO: User seamlessly logs into multiple services after authenticating once to Enterprise Security System via iS2

E2E Security: Services validate authentication tokens with iS2 and receive fine grain user authorization information

“Realm A”

Web Service

“Realm C”

CORBA middleware application

“Realm B”

J2EE application server

Security Server

S2 SSO facility

“Realm D”

Enterprise Backend System, e.g. UNIX Servers

Enterprise Directory, e.g. RACF

Auth. tokens are passed in CSI v2 context
Framework Scalability and Fail-over

Authentication Token structure: \{ issuer id, [backup id,] <unique value>\}

This interceptor is configured to access iS2 instance k

Session info is replicated upon session creation

Security framework clustering schema guarantees that principal’s authorization information is no more than two hops away!
Security Platform Example:
IONA Security Framework (iSF)
Optional iS2 Components

IONA ART or App Server

IONA Security Service

iAzMgr Facility

SSO Facility

iS2 Adapter

Optional: Implements Role Based Access Control model

Optional: Provides session management functions

Interacts with Enterprise Security Service

iSF provides optional built-in components which augment the existing ESS functionality or provide mechanisms absent in the existing ESS!
Single Sign On (SSO) Facility

- Provides session management features to iS2 client applications.
- Issues authentication tokens which clients can use for subsequent access to the services provided by iS2 client applications.
- Authentication token is valid for a certain period configured by SysAdmin.
- Authentication token expires if idle period between two subsequent service requests exceeds maximum configured by SysAdmin.

*iS2 SSO facility provides single sign on functionality across Enterprise security domains!*
iS2 Authorization Manager

- iAzMgr keeps information which supports implementation of the Role Based Access Control (RBAC) model by IONA or third party products.
- iAzMgr stores information about Principals, Roles and privilege scopes called “Realms”.
- iAzMgr answers a simple question: “Which Roles are assigned to this Principal in a given Realm?”
- iAzMgr database of Principals, Roles and Realms is stored in an abstract repository accessed via JDBC.

*iAzMgr keeps authorization information in environments devoid of robust authorization facilities such as Windows Domain!*
### iAzMgr Feature: Scoped Roles

<table>
<thead>
<tr>
<th>Realm</th>
<th>Role</th>
<th>Principal</th>
</tr>
</thead>
<tbody>
<tr>
<td>HR</td>
<td>Employee</td>
<td>Alice</td>
</tr>
<tr>
<td>Engineering</td>
<td>Manager</td>
<td>Bob</td>
</tr>
<tr>
<td>Accounting</td>
<td>Administrator</td>
<td>Carol</td>
</tr>
<tr>
<td></td>
<td>Operator</td>
<td></td>
</tr>
</tbody>
</table>

**iSF implements a superset of the proposed NIST RBAC standard compatible with the J2EE requirements!**
A “smart” iSF adapter allows to use RACF as a RBAC repository!
Putting It All Together

First invocation: user credentials are passed, e.g. Alice:secret

Server sends back session authentication token received from ISF

SSO: Client passes session authentication token for all subsequent invocations

Server implements a credential collector which passes client credentials to ISF via the exposed IS2 Client SDK interface

Authorization:
1. Server maintains a table which maps requested actions to authorizations (“roles”)
2. Server queries the AuthenticatedPrincipal interface provided by ISF to determine if the Client possesses a necessary role (“isCallerInRole”).

End-to-end security: Server propagates Client’s session authentication token when delegating the task to the next tier service.

To Authn + Authz System (3rd party)
Security Framework Based SSO Solutions
“A Spanish subsidiary of a Swiss Insurance”

Authenticate: Alice

Session token = T(W)

EJB-Authorization “per request”
“isCallerInRole”, “getCallerPrincipal”

Servlet

EJB application

Security Platform Server

AA Adapter S

Enterprise Security Store

SunOne LDAP
Deployment with iSF

Client Application

IIOP Firewall

{iS2 \text{ Competence System Adapter}}

Authentication Service

{S1: \text{CORBA Services}}

ESS 1

{S2: \text{CORBA Services}}

ESS 2

Accessibility list:
S1: role1, role2, ...
... S2n: role1, role5, ...

"Bank Specific" Svc Locator

Transport: IIOP only!

Maintains a list of available services and makes access control decisions

RACF

Policy Enforcement Point: ASP CORBA Authorization Interceptor enforces access control to CORBA services

Optional: Routes authenticated Client’s requests and redirects requests without authentication tokens to the Authentication Service.

Authenticates clients and caches authorization data

Requests service

A collection of services which requires authorization from CD1

A collection of services which requires authorization from CD2

End to end is nothing. \textit{END 2 ANYWHERE is everything.}™
SSO For CORBA Clients


1. Alice

2. CORBA Client C

3. CORBA Login Server

4. “Alice”, “secret”

5. O2K CSI v2 plug-in

6. To iS2 and Authz System

LoginUP.login(“Alice”, “secret”)
Login.login(<token>)
LoginSSL.login()

"Alice", "SSO Token"

"iSF remedies CSI v2 deficiencies and provides a robust SSO solution for CORBA applications!"
“A Government Agency”

• Requirements:
  – Use Kerberos credentials to authenticate and authorize requests to non-Kerberized services
  – Support secure invocation of Kerberized services by non-Kerberized CORBA applications using delegated Kerberos credentials

• Solution:
  – Use CSI v2 context for transmitting Kerberos service request tokens
  – Use iSF to authorize Kerberos users for invoking non-Kerberos services
“Government Agency” Deployment

Servers S1 and S2 are registered with Kerberos

Delegated credential (Proxy STkt or TGT)

Alice

Kerberos Client (GSS API)

CORBA Client C

O2K CSI v2 plug-in

Alice’s delegated Kerberos credential

Authorization info

LDAP

iS2

CORBA Server S1

O2K CSI v2 plug-in

Alice’s delegated Kerberos credential

Another Kerberos Service

Kerberos KDC

Kerberos TGS

O2K CSI v2 plug-in

CORBA Server S2
Summary - Security Platform benefits

• Security Framework approach provides applications a robust integration broker layer with Enterprise wide security services.
• Framework based architecture is flexible and allows integration with diverse security solutions from Windows domain to OS/390 RACF.
• Security framework covers important aspects of security such as authentication, authorization, SSO and PKI services.
Additional Information

IONA Security Framework (iSF) and its application to providing security services to J2EE, CORBA applications and Web Services is described in the “Orbix E2A Security” white paper.

You may download it at
http://www.iona.com/whitepaper.htm