

---

# Java and Web Services Security Overview

## — Agenda

---

- Java Platform Security
- J2EE Security
- Web Services Security

## – Intro

---

- Java world has an array of security mechanisms :
  - Java Platform Security = JVM level
  - J2EE Roles = Web Application Level
  - HTTP Security = HTTP Server/Application Level
  - SSL Security = Mutual Authentication using X.509
  - Web Services Security – XML using X.509

## – Intro

---

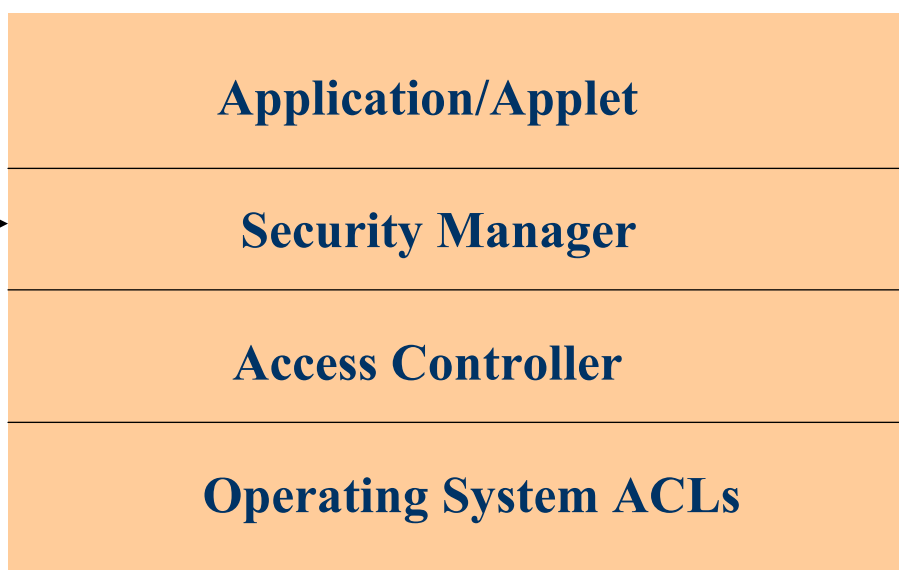
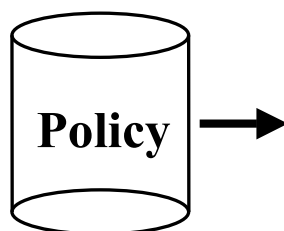
- Why should I care about any of this anyway ?
  - Java deployment increasing
    - Phones & PDAs make Java common term
  - You now have a choice in Desktop and Server O/S
    - Java applications reduce commitment to platform
    - Get away from those legacy Windows systems.....
  - Web Services
    - Heavy bias on Java on non-MS platforms
    - Promise of making platform choice less relevant
    - Reuse existing applications
  - Policy can be applied to downloaded applets and applications ( JNLP )
    - Sandbox is not just for Applets anymore
  - Security has to be designed into an application
    - Developers must provide security as part of their deployment
    - We cannot do this at the end of the development cycle

## — Java Platform Security

- Java Platform Security components
  - JVM Security Manager
  - JAAS (Java Authentication and Authorization Service )
  - JSSE ( Java Secure Socket Extension )
  - JCE ( Java Cryptography Extension )

## — Java Platform Security

### Java Virtual Machine



## — Java Platform Security

---

- By default :
  - Local code = Trusted
  - Applets = Not Trusted – in Sandbox
- The sandbox is available for all application types ( JDK 1.2 and greater ) :
  - -Djava.security.manager on JVM start
- User can assign rights to code based on :
  - Origin
  - Signature
  - Principal

## — Java Platform Security

---

- Sample Permissions
  - java.net.SocketPermission –Socket usage
  - java.awt.AWTPermission – GUI usage
  - java.io.FilePermission – File access
  - java.lang.RuntimePermission – Runtime services
  - You can also add your own

## — Java Platform Security

---

- Policy is kept in file security.policy file
- Two versions - user version overrides system
  - Plug-In :
    - x:\Program Files\Java\j2rex.x.x\lib\security
    - X:\Documents and settings\username
  - JVM
    - X:\jdk\jre\lib\security
    - X:\Documents and settings\username
    - -Djava.security.manager to enable Security Manager
    - -Djava.security.policy=filename/url to override

## — Java Platform Security

---

- Sample Policy

```
grant {  
    permission java.io.FilePermission  
        "${user.home}/text2.txt", "read";  
grant {  
    permission java.net.SocketPermission  
        "129.144.176.176:1521", "connect,resolve";  
}
```

## — Java Platform Security

- Sample Policy

```
grant codebase "http://www.games.com",
    signedBy "Duke",
    principal javax.security.auth.x500.X500Principal "cn=Alice" {
    permission java.io.FilePermission "/tmp/games", "read,
write";};

grant signedBy "mikem" {
    permission java.io.FilePermission "/tmp/*",
"read,write";};
```

## — Java Platform Security

- Use policytool in JDK to edit :



## — Java Platform Security

---

- Java Authentication and Authorization Service ( JAAS )
  - Authentication framework using for example
    - W2K/NT
    - Kerberos
  - Allows JVM to authenticate users for policy checking
    - doAs() method
    - doAsPrivileged() method

## — Java Platform Security

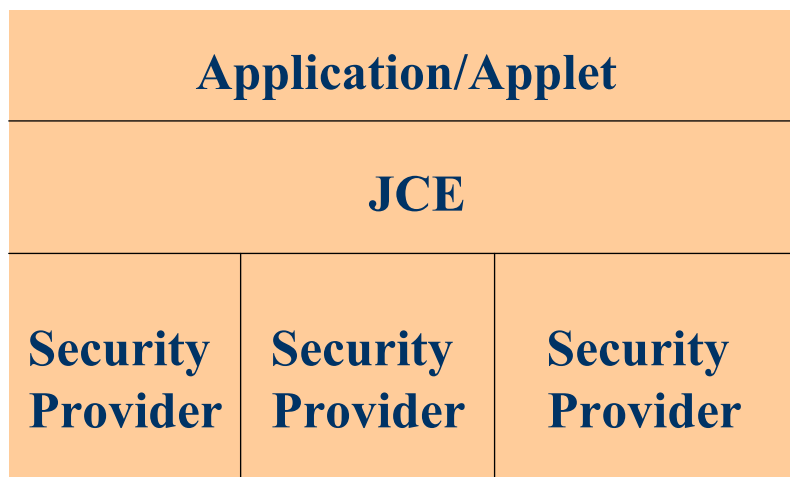
---

- Sample JAAS policy :

```
App1 {  
    com.sun.security.auth.module.NTLoginModule required;  
};  
  
App2 {  
    sample.SampleLoginModule required;  
    com.sun.security.auth.module.NTLoginModule sufficient;  
};
```

## — Java Platform Security

- JCE – Java Cryptography Extension



## — Java Platform Security

- JCE – Java Cryptography Extension
  - Provides abstraction layer for access to crypto algorithms
  - Programmer calls standard Engine classes supplied with JCE
  - JCE and policy ( file java.security ) provides access to Algorithm classes
  - Security providers can be changed without changing application

## — Java Platform Security

- OS/390 and z/OS – SAF access
  - PlatformAccessControl
  - PlatformAccessLevel
  - PlatformReturned
  - PlatformSecurityServer
  - PlatformThread
  - PlatformUser
- <http://www-1.ibm.com/servers/eserver/zseries/software/java/security.html>

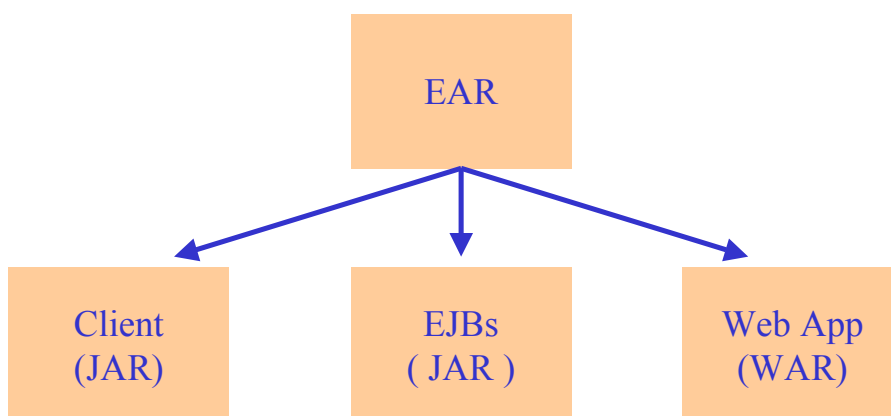
## — Java Platform Security

```
import com.ibm.os390.security.*;
....
System.out.println(PlatformSecurityServer.isActive());
System.out.println(PlatformSecurityServer.resourceTypeIsActive("FACILITY"));
System.out.println(PlatformUser.isUserInGroup(PlatformThread.getUserName(),"SYS1"));
System.out.println(PlatformUser.authenticate("MIKE","MIKEPASS"));
if
(PlatformAccessControl.checkPermission(PlatformThread.getUserName(),"FACILITY","BPX.DAEMON",
PlatformAccessLevel.READ) == null)
{
    System.out.println("User has access to resource");
}
else
{
    System.out.println("User does not have access to resource");
}
System.out.println(PlatformThread.getUserName());
```

## J2EE Security

- Application builder defines Roles
  - Defined in web.xml and ejb-jar.xml – part of application
  - Users mapped to roles
    - In web.xml ( Tomcat )
    - Custom Registry ( WebSphere )
    - SAF EJBROLES class ( z/OS )
- Application Packaged into a number of files
  - WAR – Web Application Archive
  - EAR – Enterprise Application Archive
- Application servers deploy files from this format

## J2EE Security



## — J2EE Security

- Roles used in two ways:
  - Declarative
    - Access to URLs, beans restricted by role
  - Programmatic
    - Application calls `isUserInRole` method – part of `ServletRequest` class

```
if ( req.isUserInRole("UpdateAccount"))
{
    /* Update Account logic .... */
}
```

## — J2EE Security

- Declarative Security – from `web.xml`

```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>The Entire Web Application</web-resource-name>
    <url-pattern>/*</url-pattern>
  </web-resource-collection>
  <auth-constraint>
    <role-name>tomcat</role-name>
  </auth-constraint>
</security-constraint>
```

## — J2EE Security

---

- Declarative Security – from ejb-jar.xml

.....

```
<method-permission>
  <role-name>everyone</role-name>
  <method>
    <ejb-name>CustomerBean</ejb-name>
    <method-name>*</method-name>
  </method>
</method-permission>
```

.....

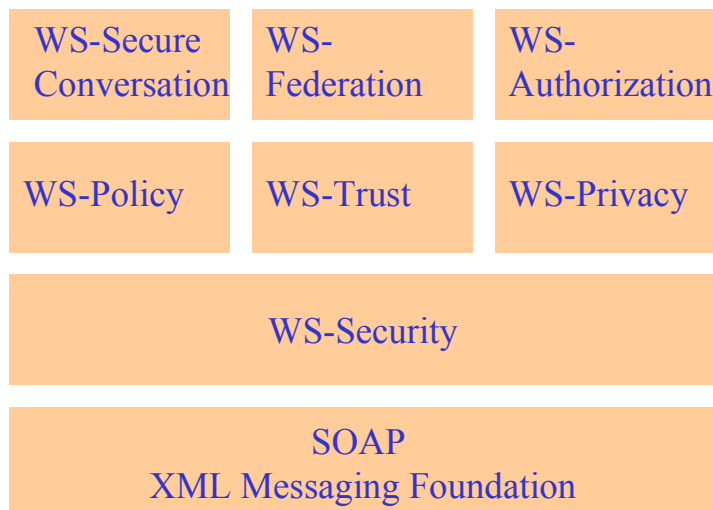
## — Web Services Security

---

- SOAP (Simple Object Access Protocol ) introduced....
  - No security support in protocol!
    - Can use Basic HTTP authentication over SSL
    - Can use SSL client certificate auth enabled by Web Server
  - IBM and Microsoft join forces to address this....
    - Extending XML schema

## — Web Services Security

- Web Services Roadmap (IBM and MicroSoft )



## — Web Services Security

- WS-Security
  - Foundation for services
  - SOAP extended to use
    - XML Signature
    - XML Encryption
    - Security Tokens
      - Userid/password
      - Kerberos Ticket

## — Web Services Security

---

- WS-Policy
  - Exchange requirements and capabilities between senders and receivers
- WS-Trust
  - Establish and manage trust relationships
- WS-Privacy
  - Declare privacy policies and compliance

## — Web Services Security

---

- WS-SecureConversation
  - Manage and authenticate messages between parties
- WS-Federation
  - Manage identity across organization boundaries
- WS-Authorization
  - Manage authorization data and policies

## Links

---

- Java Tutorial
  - <http://developer.java.sun.com/developer/onlineTraining/Programming/JDBCBook/index.html#contents>
- WS-Security Roadmap
  - <http://schemas.xmlsoap.org/specs/ws-security/WS-Security-Roadmap.htm-roadmap>
- Oasis WS-Security Site
  - [http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=wss](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss)
- IBM WebSphere SDK for Web Services (WSDK) Version 5.1
  - <http://www-106.ibm.com/developerworks/webservices/wsdk>
- Z/OS JVM SAF info and docs
  - <http://www-1.ibm.com/servers/eserver/zseries/software/java/security.html>
- Z/OS JAAS
  - <http://www-1.ibm.com/servers/eserver/zseries/software/java/jaas.html>