PRESENTED BY RED HAT

LEARN. NETWORK.
EXPERIENCE OPEN SOURCE.

www.theredhatsummit.com

SECURITY ASSURANCE WITH JBOSS EAP

Anil Saldhana
Lead Middleware Security Architect
Red Hat
May 4, 2011

Agenda

- Middleware Security – JBoss Platforms
- Secure your applications [Web,EJB etc]
  - Security Domain
  - Mask passwords in Clear Text
- Security Features
  - Audit
  - Fine Grained Access Control – XACML
  - Kerberos/SPNego SSO – JBoss Negotiation

Agenda

- Identity Management – PicketLink
- Best Practices and Tips
- Q & A

Middleware Security – JBoss Platforms

Security is critical to middleware

- Balance between features and configuration
- Features such as business access control and audit may be available as part of the stack
  - No need for embedding the logic in applications
- Security Response Team
  - Cross-cutting, multi-geo and round-the-clock.

Secure Your Applications

Security Domain

- Central concept in security configuration for EAP.
- Defines the modules for
  - Authentication
  - Authorization
  - Audit
  - Mapping

Secure Your Applications

Security Domain

- You can define globally in conf/login-config.xml
  - Static configuration. No hot deployment of domains.
- You can define at deployment level using xxx-jboss-beans.xml
  - Provides hot deployment of security domains

Secure Your Applications

Security Domain

```
<deployment xmlns="urn:jboss:bean-deployer:2.0">
  <application-policy  xmlns="urn:jboss:security-beans:1.0"  name="MyDomain">
  <authentication>
      <login-module code="org.jboss.security.auth.spi.UsersRolesLoginModule"  flag="required" />
   </authentication>
    <authorization>
      <policy-module code="org.jboss.security.authorization.modules.DelegatingAuthorizationModule" flag="required"/>
     </authorization>
    <mapping>
      <mapping-module code="org.jboss.security.mapping.providers.DeploymentRolesMappingProvider" type=”role”>
      </mapping-module>
     </mapping>
  </application-policy>
</deployment>
```


Secure Your Applications

How to secure your web applications?

- Define your security constraints in your web.xml
  - Login Config: BASIC, FORM, CLIENT-CERT, DIGEST
- Define your security domain name in jboss-web.xml
  - If you omit, it defaults to “other”
- Provide your security domain configuration in a xxx-jboss-beans.xml file
- Package them into a war file

Secure Your Applications

How to secure your web services applications?

- EJB based web services are authenticated and authorized by the EJB Container.
- POJO based web services
  - Configure a security domain in WEB-INF/jboss-web.xml
  - Configure a WEB-INF/jboss-wsse-server.xml to declare the roles or ***unchecked*** access.

Secure Your Applications

How to secure your EJB3 applications?

- Define your security annotations in your bean classes.
  - @RolesAllowed, @RunAs, @PermitAll etc
- Define your security domain name in jboss.xml
- Provide your security domain configuration in a xxx-jboss-beans.xml file
- Package them into a jar file

Secure Your Applications

Encrypt/Mask Passwords in Clear Text

- Do you like to see passwords in the clear?
- Facilities to encrypt/mask/hash passwords:
  - Data store Passwords
  - Tomcat connector Passwords
  - Messaging Destination Passwords
  - Microcontainer Beans Passwords

Secure Your Applications

Encrypt/Mask Passwords in Clear Text

- Difference between hashing and encryption
  - Hashing involves a one way treatment
  - Encryption is two way but involves a key
    - Masking typically uses Password based encryption

Use Cases

Security Features – Use Case 1

- This is a large company.
  - About 20,000 employees log on to their Windows Machines each day.
  - The Desktop logins are governed by an ***Active Directory*** Domain Controller.
  - After corp desktop login
    - they access about 30 internal web applications.
  - The employee information is stored in AD and divisional databases.

Security Features – Use Case 1 - Issues

- Windows Login – 1 password.
- Web Application 1 – 1 password.
- Web Application 2 – 1 password.
- ....
- Web Application 30 – 1 password.
- Web applications run by various divisions at the company, with different needs for roles.

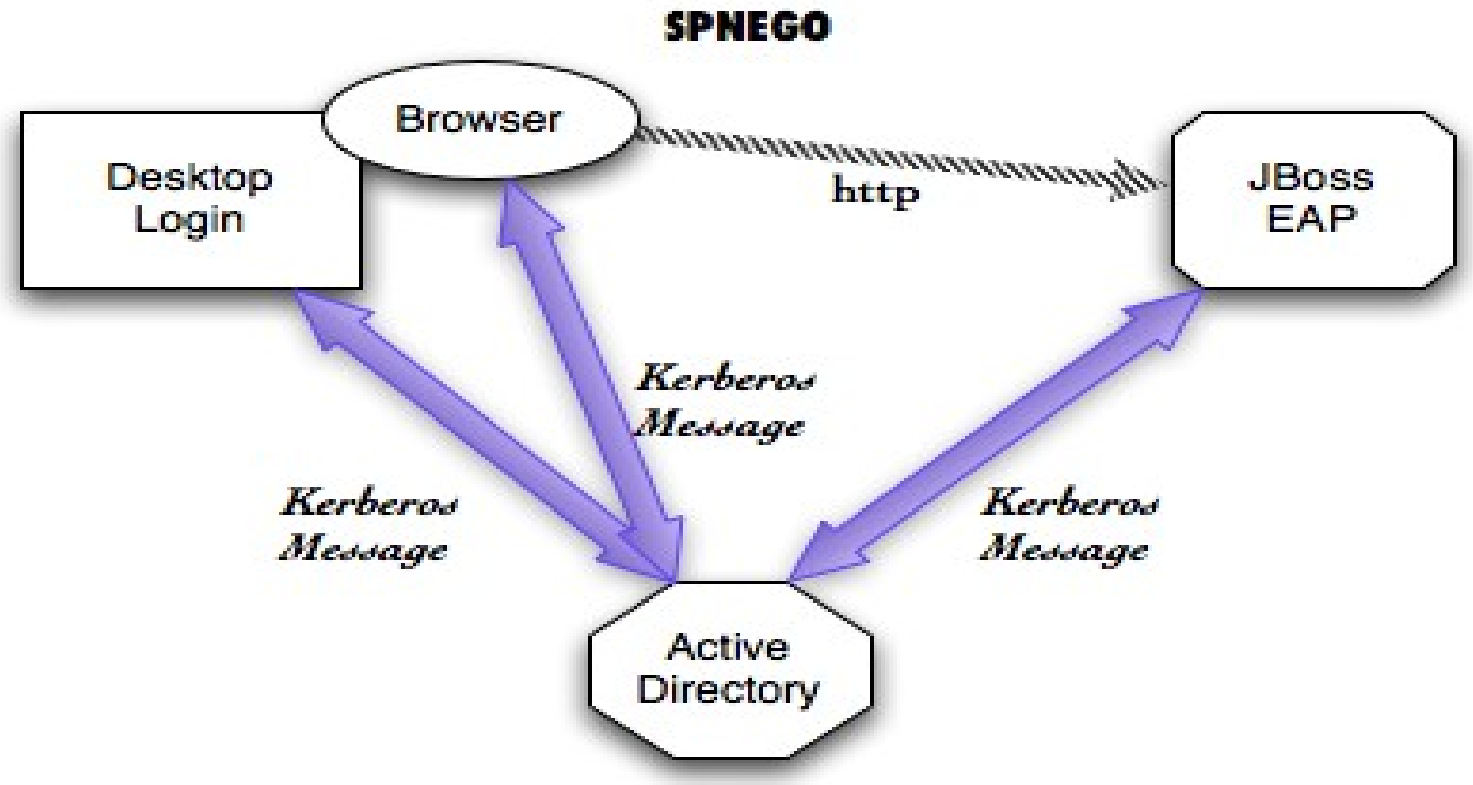
Security Features – Use Case 1

Negotiation

- Fully supported in JBoss EAP 5.0 and beyond.
- Users log into their desktops (windows/linux) governed by a Kerberos based Domain Controller (Active Directory).
    - Web applications on EAP can have seamless SSO.
- Negotiation takes care of the authentication aspect.
    - Roles for the web apps can come from any silo.
        - Due to JAAS login module stacking.
- *EJB3 Applications can have kerberos auth.*

Security Features – Use Case 1

Negotiation



Security Features – Use Case 2

My web and ejb applications have business rules that keep changing:

- Employees cannot sell company shares during blackout period.
- Employees cannot view their manager's salary.
- Junior Traders can make trades if they are less than $1m in value.
- Web application is read only during non business hours (9am -5pm).

Security Features – Use Case 2 - Issues

Many of these business rules = access control rules

- Choice is to embed the business rules in the application logic ***or***
- Externalize the access control logic.
- Java EE Container Security for web and ejb applications uses coarse grained access control rules.
- ***FINE GRAINED AUTHORIZATION***
- ***DOMAIN DRIVEN AUTHORIZATION***

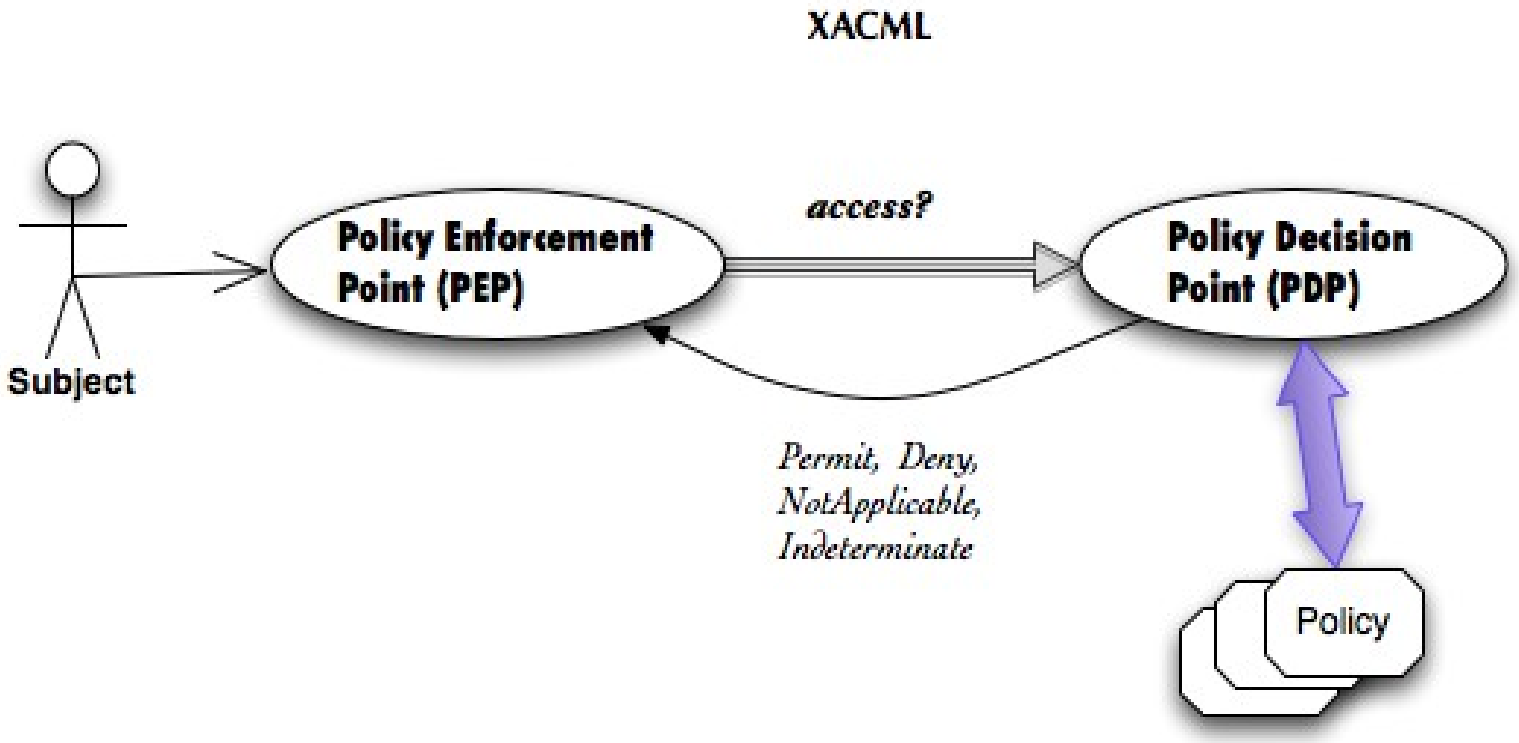
Security Features – Use Case 2 - Detail

XACML Engine

- Oasis XACML v2.0 Compliant.
- Fully supported in JBoss EAP 5.0 and beyond.
- Provide Fine Grained Authorization Capabilities to EJB and Web Applications.
- Business Applications can also make **direct** use of the Engine API.
- Unlike Java EE coarse grained Role Based Access Control, XACML is fine grained attributes driven

Security Features – Use Case 2 - Detail

XACML Engine



Security Features – Use Case 2 Detail

XACML Access Decisions – a combination of rules on

- Subject – user, actor invoking a service
- Resource – something that needs to be protected
- Action – subject is performing on the resource (get,read,write,delete)
- Environment – date,time,ip address

Security Features – Use Case 3

Enterprise has Identity Management Needs

- Manage users, roles, groups, attributes etc.
- Support SAML, WS-Trust, OpenID, OAuth.

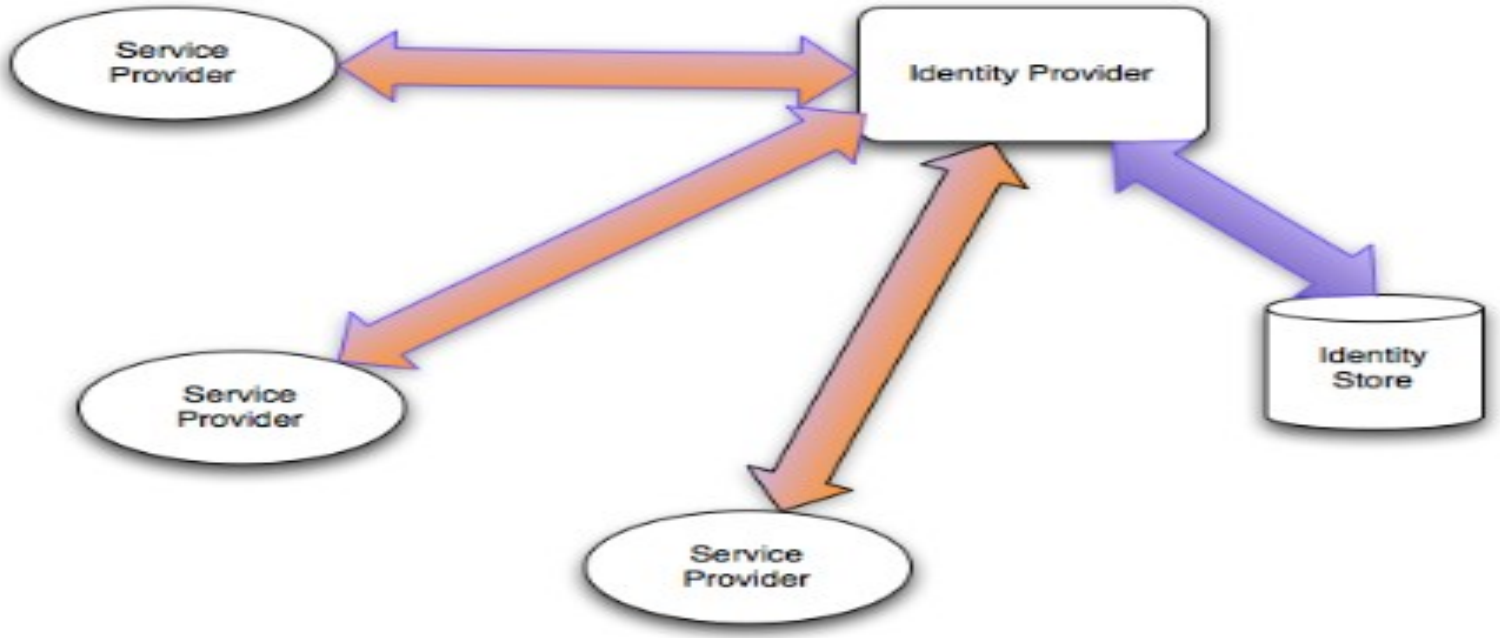
Security Features – Use Case 3 Detail

PicketLink Identity Management

- Identity Model (IDM)
- Federated Identity Support
  - SAML v2.0, WS-Trust, OpenID
- Tech Preview in JBoss EAP 5.1, SOA-P 5.0
- IDM Module supported in JBoss EPP5

Security Features – Use Case 3 Detail

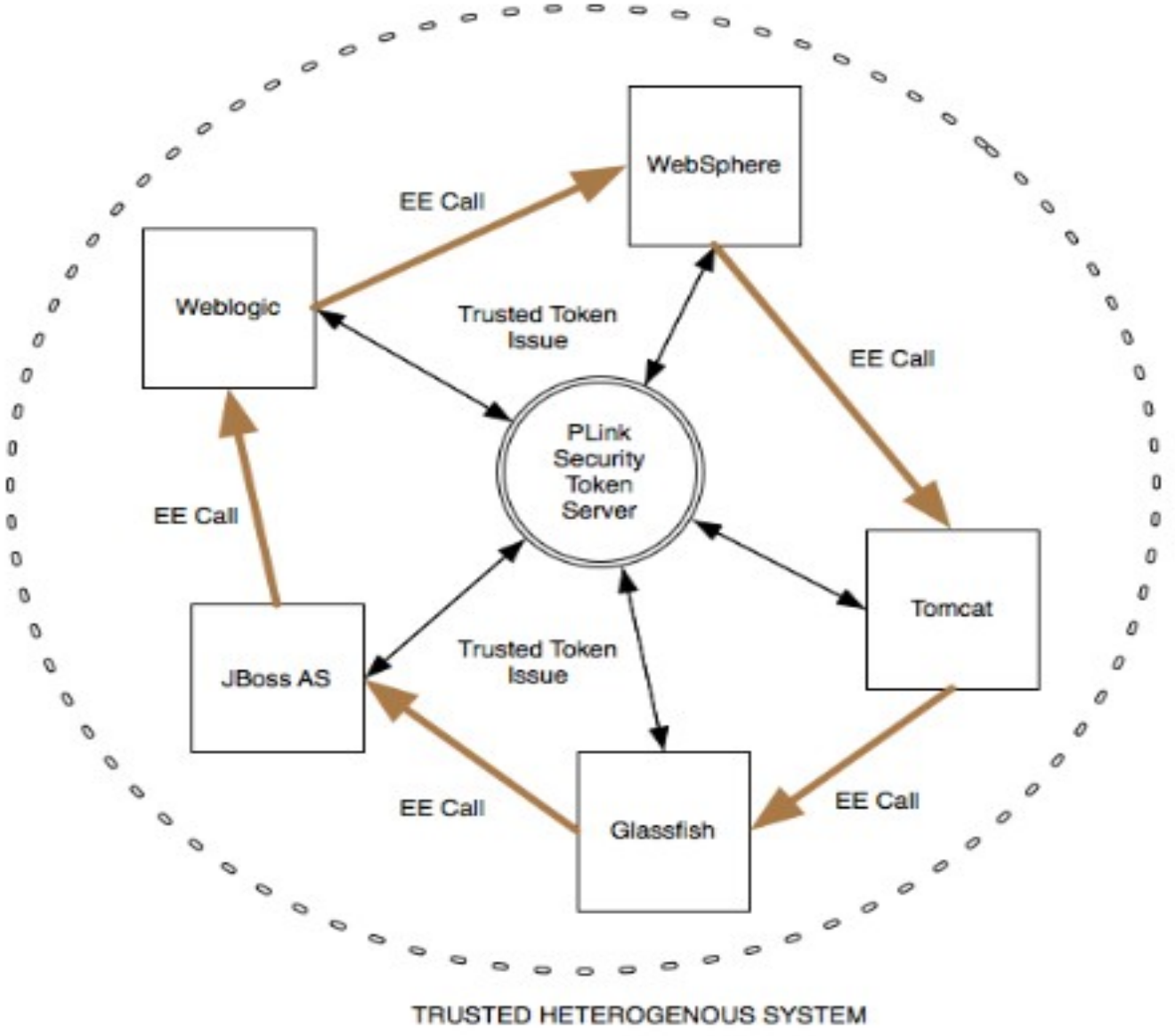
PicketLink Architectures – Web Based Identity Provider



SAML WEB BROWSER SSO

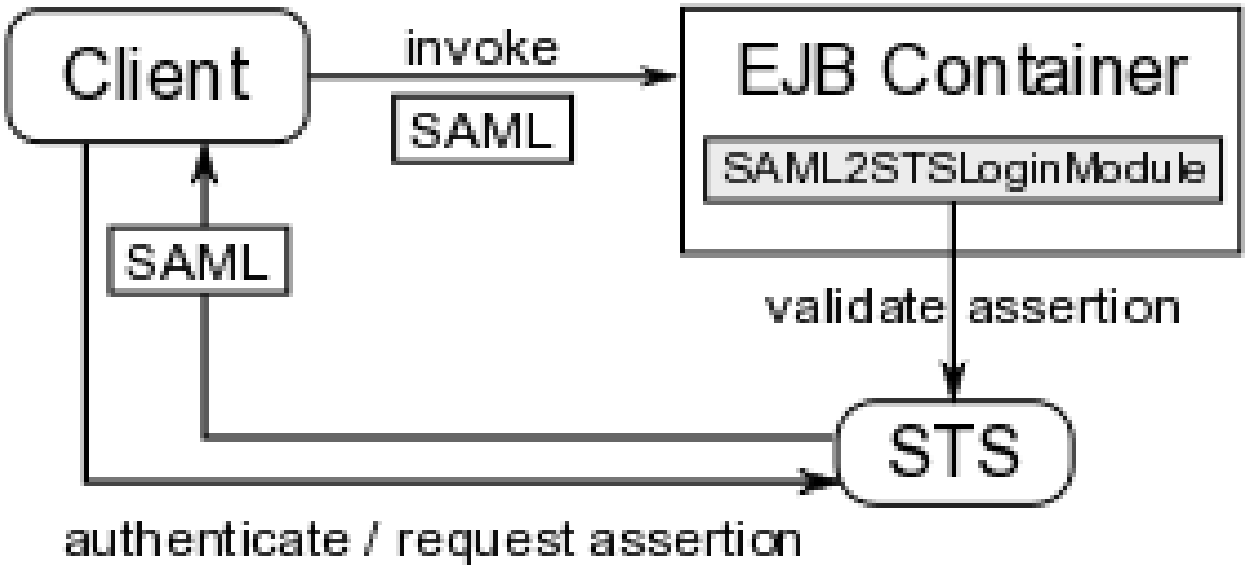
Security Features – Use Case 3 Detail

PicketLink Architectures – WS-Trust Based STS



Security Features – Use Case 3 Detail

PicketLink Architectures – STS for EJB Applications



Security Features – Use Case 3 Detail

Levels of Assurance (NIST 800-63 Special Pub e-Auth)

- Level 1
  - Little/no confidence in asserted identity's validity
  - OpenID and OAuth
- Level 2
  - Some Confidence.
  - Password based systems or SAML assertions using password based mechanism

Security Features – Use Case 3 Detail

Levels of Assurance (NIST 800-63 Special Pub e-Auth)

- Level 3
  - High Confidence.
  - Soft/hard crypto tokens, OTP....
- Level 4
  - Very High confidence. PKI, Smart Cards.

Security Features – Use Case 3 Detail

Which standard is relevant to you?

- Community Type Environment
  - Need Level 1 assurance?
    - Internet scale? Decentralized? Then OpenID and OAuth.
- Enterprise Type Environment
  - Level 2 assurance of identity
    - SAML assertions based on password authentication.
  - Level 3 or 4 assurance of identity
    - SAML assertions based on PKI or Smart Cards.

Security Features – Use Case - 4

Thousands of users access my web and ejb applications. It is very difficult to know:

- Who logged in?
- At what time?
- From which IP Address?
- What EJB operation? Which WEB URI?
- Who was denied access and why?

Security Features – Use Case 4 Details

Audit Capabilities

- You can audit your Web and EJB applications.
- Depends on ***enabling*** Java EE Container Security.
- Configuration enables granularity of auditing for web requests.

Security Features – Audit

2008-12-05 16:08:38,997 TRACE [org.jboss.security.audit.providers.LogAuditProvider] (http-127.0.0.1-8080-17:)
[Success]policyRegistration=org.jboss.security.plugins.JBossPolicyRegistration@76ed4518;Resource:=[org.jboss.security.authorization.resources.WebResource:contextMap={policyRegistration=org.jboss.security.plugins.JBossPolicyRegistration@76ed4518,securityConstraints=[Lorg.apache.catalina.deploy.SecurityConstraint;@6feeae6,
resourcePermissionCheck=true},canonicalRequestURI=/restricted/get-only/x,request=[/web-constraints:cookies=null:headers=user-agent=Jakarta Commons-HttpClient/3.0,authorization=host=localhost:8080,]
[parameters=],CodeSource=null];securityConstraints=SecurityConstraint[RestrictedAccess - Get
Only];Source=org.jboss.security.plugins.javaee.WebAuthorizationHelper;resourcePermissionCheck=true;Exception:=;

2008-12-05 16:08:41,561 TRACE [org.jboss.security.audit.providers.LogAuditProvider] (http-127.0.0.1-8080-4:)
[Failure]principal=anil;Source=org.jboss.web.tomcat.security.JBossWebRealm;request=[/jaspi-web-basic:cookies=null:headers=user-agent=Jakarta Commons-HttpClient/3.0,authorization=host=localhost:8080,][parameters=][attributes=];2008-12-05 16:07:30,129 TRACE [org.jboss.security.audit.providers.LogAuditProvider]
(WorkerThread#1[127.0.0.1:55055]:)

Best Practices, Tips – Field Experiences

- Make use of Java EE Container Security.
- Security Domain Configuration as part of your deployment.
- Register to obtain security patches and updates.
- Keep your system up to date.
- Adopt standards based architecture as far as possible.
  - Biggest challenge seen in customers migrating to Jboss
  - Proprietary tokens – LTPA,IV-Creds -> PicketLink

Q & A

Thank You!!!

LIKE US ON FACEBOOK

www.facebook.com/redhatinc

FOLLOW US ON TWITTER

www.twitter.com/redhatsummit

TWEET ABOUT IT

#redhat

READ THE BLOG

summitblog.redhat.com

GIVE US FEEDBACK

www.redhat.com/summit/survey