

Secure Middleware with JBoss Application Server v5.x

Anil Saldhana
Red Hat Inc
6280

JAZZ00N09

THE INTERNATIONAL CONFERENCE ON JAVA TECHNOLOGY
JUNE 22-25, 2009 ZURICH



netcetera



AGENDA

- > Security as we know it
- > Security Features versus configuration
- > Security in JBossAS5.x
- > Authentication
- > Authorization
- > Audit
- > Password Masking in Microcontainer Beans
- > References

Security as we know it



- > Secure
 - Shoot dog?
 - Drugged food?

Security as we know it



- > Top of the line security
 - Not in action

JAZOON09

THE INTERNATIONAL CONFERENCE ON JAVA TECHNOLOGY
JUNE 22–25, 2009 ZÜRICH



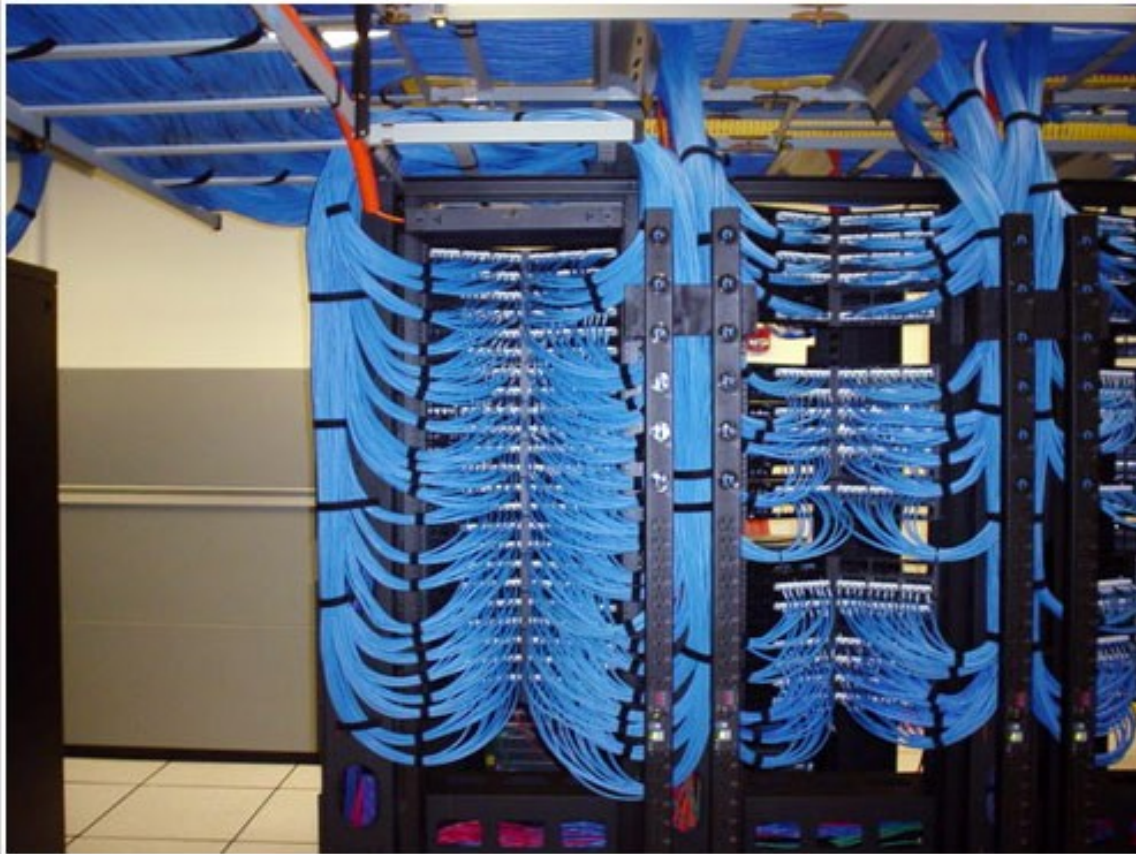
netcetera



Security Features versus configuration

- > Goal is to provide new security features all the time
 - While minimizing additions to configuration

Security Features versus configuration



- > Great System
 - Hard Wiring between objects

JAZZ00N09

THE INTERNATIONAL CONFERENCE ON JAVA TECHNOLOGY
JUNE 22–25, 2009 ZÜRICH



Security Features versus configuration

- > New Features
 - Feeble foundation



JAZZ00N09

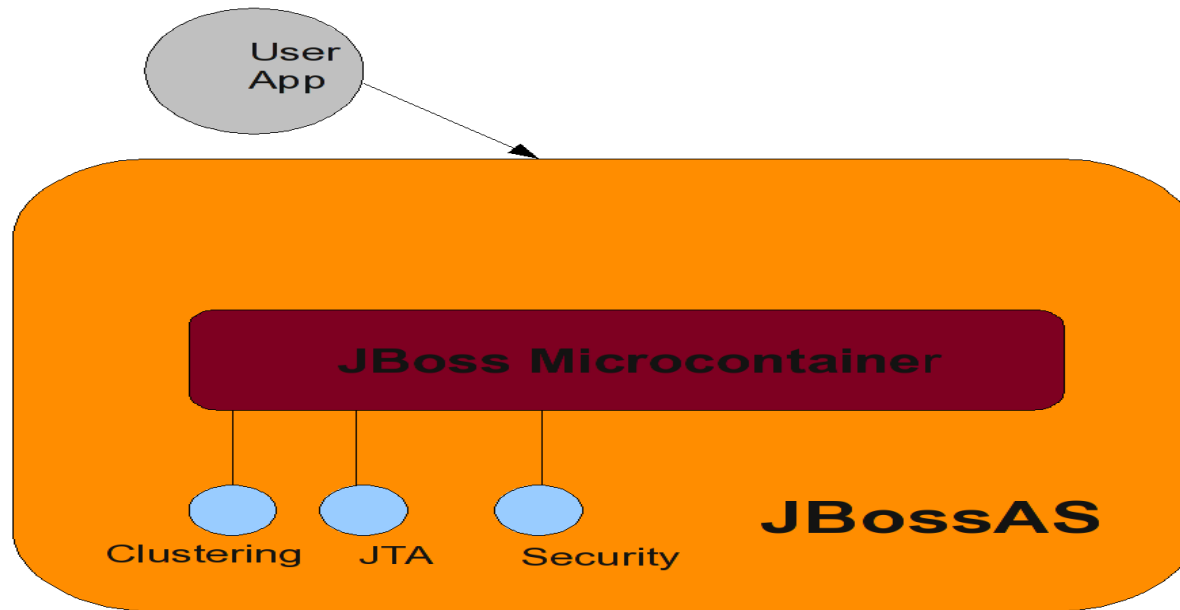
THE INTERNATIONAL CONFERENCE ON JAVA TECHNOLOGY
JUNE 22–25, 2009 ZÜRICH



netcetera



Security in JBAS 5.x



Security in JBAS 5.x

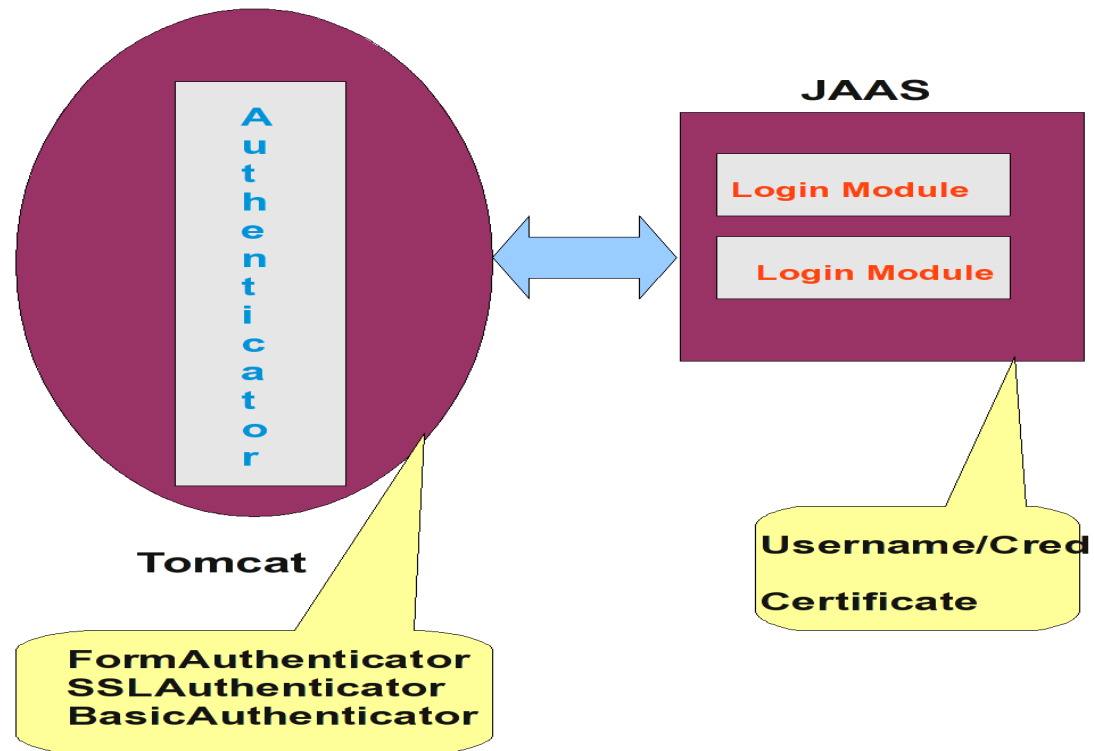
- > Makes use of the JBoss Microcontainer based architecture
- > Introduce new features
 - Authentication: Support for JSR-196
 - Authorization: Pluggable access control stack (Spec, JACC, XACML ...)
 - Auditing
 - Mapping : Role mapping, Principal mapping
 - Password Masking Feature for MC beans
- > Simplify Configuration
 - Security Domain Configuration

Security in JBAS 5.x : Authentication

- > JSR-196: Java Authentication SPI for Containers
- > Allows us to externalize authentication as Server Authentication Modules(SAM)
 - JAAS does not have a notion of a container message
 - ➔ No access to HttpServletRequest, SOAPMessage etc
 - Tomcat authenticator code would be as an example in the SAM

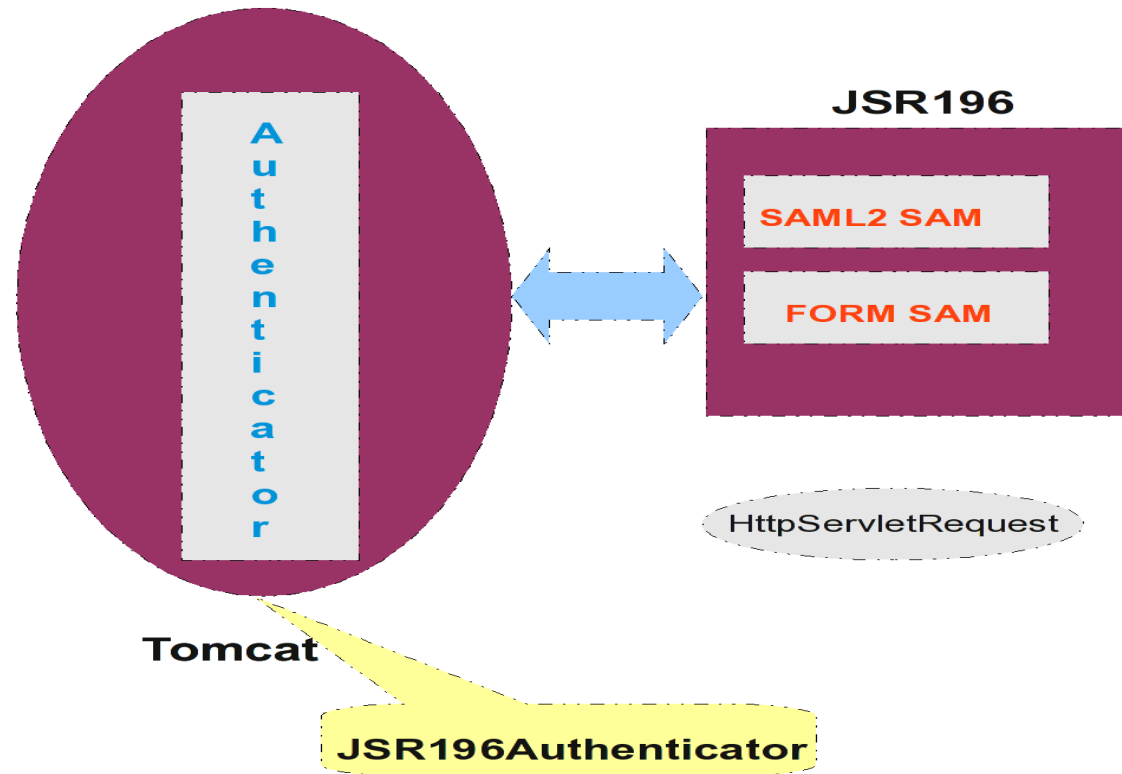
Security in JBAS 5.x : Authentication

> JSR-196



Security in JBAS 5.x : Authentication

> JSR-196



<http://anil-identity.blogspot.com/search/label/jsr-196>

JAZOON09

THE INTERNATIONAL CONFERENCE ON JAVA TECHNOLOGY
JUNE 22-25, 2009 ZURICH



Security in JBAS 5.x : Authorization

- > Pluggable Access Control Stack for Web and EJB Applications.
- > Apply spec access control, JACC, XACML (or custom) to web and ejb applications in a pluggable fashion.
 - Mix and match (JACC for Web, XACML for EJB)
- > Value added feature that still maintains Java EE RBAC compliance
- > Fine grained access control
 - **JBossXACML**: Oasis XACML v2.0 support
 - **JBossACL** :Instance based access control

<http://server.dzone.com/articles/security-features-jboss-510-1>

<http://server.dzone.com/articles/security-features-jboss-510-2>

<http://server.dzone.com/articles/security-features-jboss-510-3>

Security in JBAS 5.x : Audit

- > Enable auditing of security events in web and ejb applications
- > Plug in various auditing providers
 - Default provider is a Log4J provider

[*http://server.dzone.com/articles/security-auditing-jboss*](http://server.dzone.com/articles/security-auditing-jboss)

Security in JBAS 5.x : 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:)

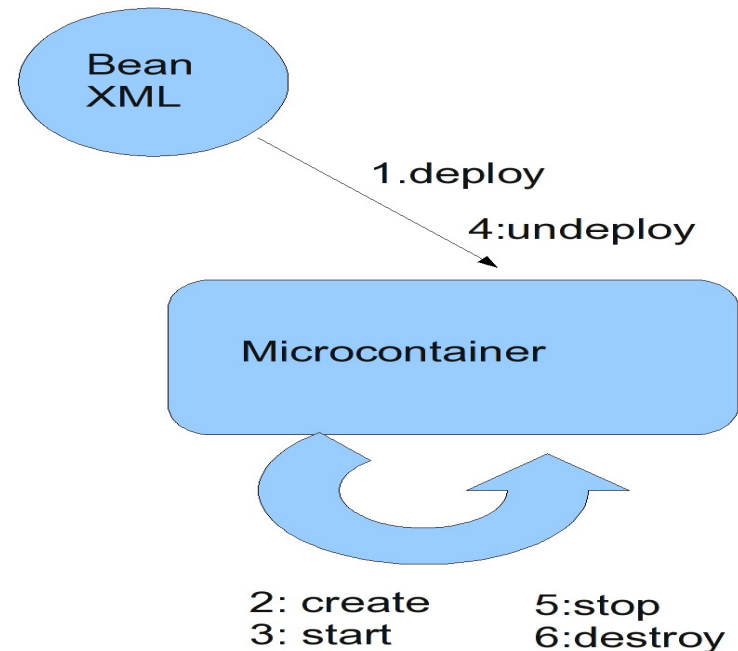
[**Failure**]principal=anil;Source=org.jboss.web.tomcat.security.JBossWebRealm;request=[/jaspi-web-basic:cookies=null:headers=user-agent=Jakarta Commons-HttpClient/3.0,authorization=host=localhost:8080,][parameters=][attributes=];2008-12-05 16:07:30,129 TRACE [org.jboss.security.audit.providers.LogAuditProvider]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

Security in JBAS 5.x : Mapping

- > Map Roles
 - Application role to deployment level role
- > Map Principal
 - X509 principal to a simple name

Password Masking for Microcontainer Beans

- > Make use of the JBoss MC life cycle callbacks
 - Inject the password at “create”



<http://server.dzone.com/articles/security-features-jboss-510-0>

Password Masking for Microcontainer Beans

```
<bean name="SecurityStore"  
class="org.jboss.jms.server.jbossxx.JBossASSecurityMetadataStore">  
  <property name="suckerPassword">CHANGE ME!!</property>  
  <property name="securityDomain">messaging</property>  
  <property name="securityManagement">  
    .<inject bean="JNDIBasedSecurityManagement"/></property>  
</bean>
```



Before

Password Masking for Microcontainer Beans

```
<bean name="SecurityStore"  
class="org.jboss.jms.server.jbossx.JBossASSecurityMetadataStore">  
  <property name="suckerPassword">CHANGE ME!!</property>  
  <property name="securityDomain">messaging</property>  
  <property name="securityManagement">  
    .<inject bean="JNDIBasedSecurityManagement"/></property>  
  <!-- Password Annotation to inject the password from the common  
password utility -->  
  <annotation>@org.jboss.security.integration.password.Password(security  
Domain=messaging,methodName=setSuckerPassword)</annotation>  
</bean>
```



After

Simplified Configuration

- > Single security domain configuration for Authentication, Authorization, Auditing, Mapping, Acl etc
- > Again, a feature of JBoss Microcontainer is leveraged.

<http://server.dzone.com/articles/security-features-jboss-510>

Simplified Configuration

```
<application-policy name="MyDomain">
  <authentication>
    <login-module code="org.jboss.security.auth.spi.UsersRolesLoginModule"
      flag="required">
      <module-option name="usersProperties">users.properties</module-option>
      <module-option name="rolesProperties">roles.properties</module-option>
      <module-option name="password-stacking">useFirstPass</module-option>
    </login-module>
  </authentication>
  <authorization>
    <policy-module code="org.jboss.security.authorization.modules.DelegatingAuthorizationModule"
      flag="required"/>
  </authorization>
  <rolemapping>
    <mapping-module code="org.jboss.security.mapping.providers.DeploymentRolesMappingProvider">
    </mapping-module>
  </rolemapping>
</application-policy>
```

Anil Saldhana

<http://anil-identity.blogspot.com>

Red Hat Inc

anil.saldhana@redhat.com

JAZOON09

THE INTERNATIONAL CONFERENCE ON JAVA TECHNOLOGY
JUNE 22-25, 2009 ZURICH



netcetera

