

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT

**LEARN. NETWORK.
EXPERIENCE OPEN SOURCE.**

www.theredhatsummit.com

Securing JBoss Services

Robert C. Broeckelmann Jr.
Ovidiu Feodorov,
Open Source Consultant
24 June 2010

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Agenda

- Introduction
- Our goal
- Ground floor concepts
- Securing JBoss services
- Securing JBoss communications
- Closing

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Introduction

- We will discuss
 - SSL
 - Other network encryption technologies
 - JGroupsENCRYPT
 - JBoss use of these
 - Security-related JBoss configuration
- Technologies used
 - Java 1.6
 - IBM JDK
 - Sun JDK
 - OpenJDK6
 - JBoss 4.3.0_GA.CP04

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Our Goal

- All administrative “chatter” encrypted
 - Including listening endpoints and multicast traffic
 - LDAP communications encrypted via SSL
- Application traffic encrypted via SSL
 - JBossWS
 - EJB2/EJB3
 - JMS/JBossMessaging
 - IIOP/CORBA
 - JBossWeb

SUMMIT

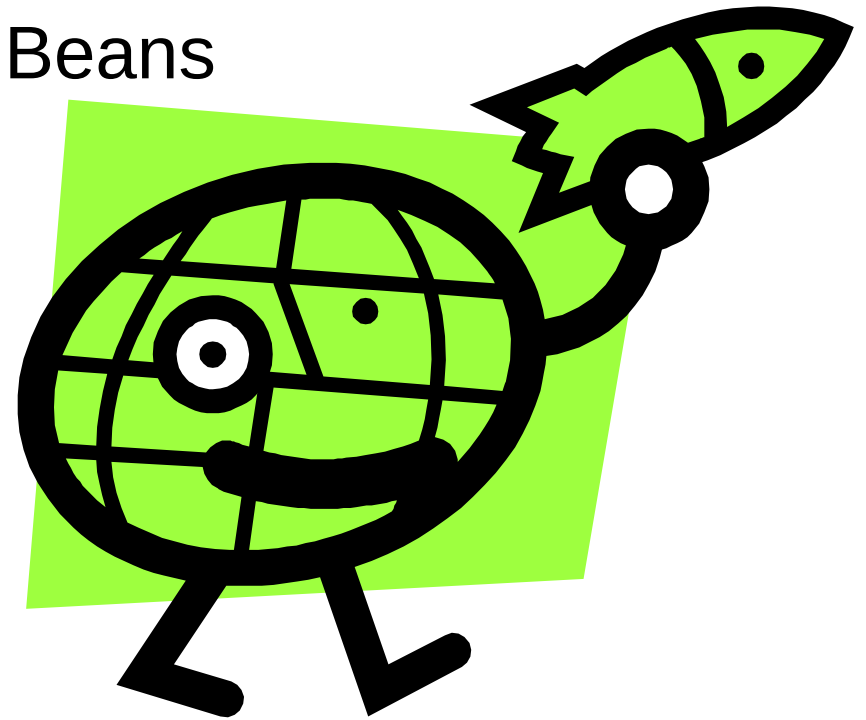
**JBoss
WORLD**

PRESENTED BY RED HAT



Ground Floor Concepts

- SSL – quick introduction
- JSSE – Java Secure Socket Extension
- Java Cryptography Architecture
- JBoss JAASSecurityDomain MBeans



SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



TCP/IP, Sockets, & Encryption

- TCP/IP are fundamental protocols of the Internet
 - Assuming everyone is familiar with these 😊
 - Most communication over the internet is TCP/IP
- Sockets API provides lowest-level programming interaction with TCP sockets
- TCP/IP is clear text communication
 - Fine for accessing your favorite news site (assuming no user id/password, of course)
 - Not so good when looking at your bank account
- An encryption technology must be used to secure network communication flowing over TCP
 - Enter SSL

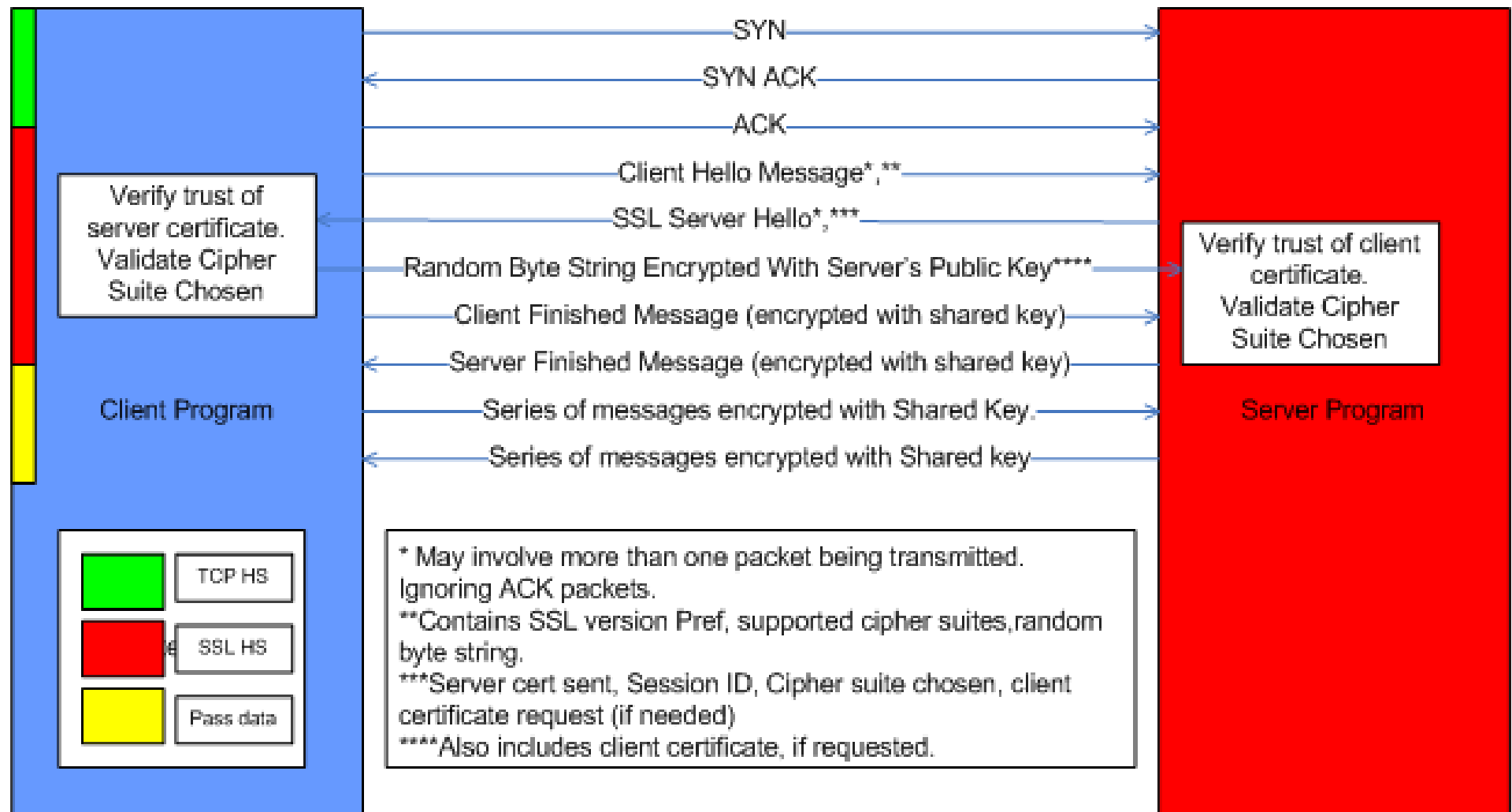


SSL/TLS – Secure Sockets Layer

- Secure Sockets Layer (SSLv3) and its descendent Transport Layer Security (TLSv1) are the corner stone of encrypted communication over the internet today
- Original SSL protocol developed by Netscape
 - V2 first released in February, 1995
- Sits above TCP in the network protocol stack
- Implementation libraries provide an API similar to the Sockets API



SSL/TLS—In depth



SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



JSSE – Java Secure Sockets Extension

- JSSE is an implementation of SSL and an API for Java
- Part of JDK since v1.4^[3]
- Concepts:^[6]
 - X509 Certificates
 - Keystores/Truststores
 - KeyManager/TrustManager
 - Providers
 - Socket Factories
- Java provides significant support for SSL/TLS
 - javax.net.ssl package
- Using SSL in Java can be effectively reduced to an SSLSocketFactory / SSLServerSocketFactory initialization strategy
- JBoss provides a configuration wrapper around JSSE Socket Factories—JBoss JaasSecurityDomains

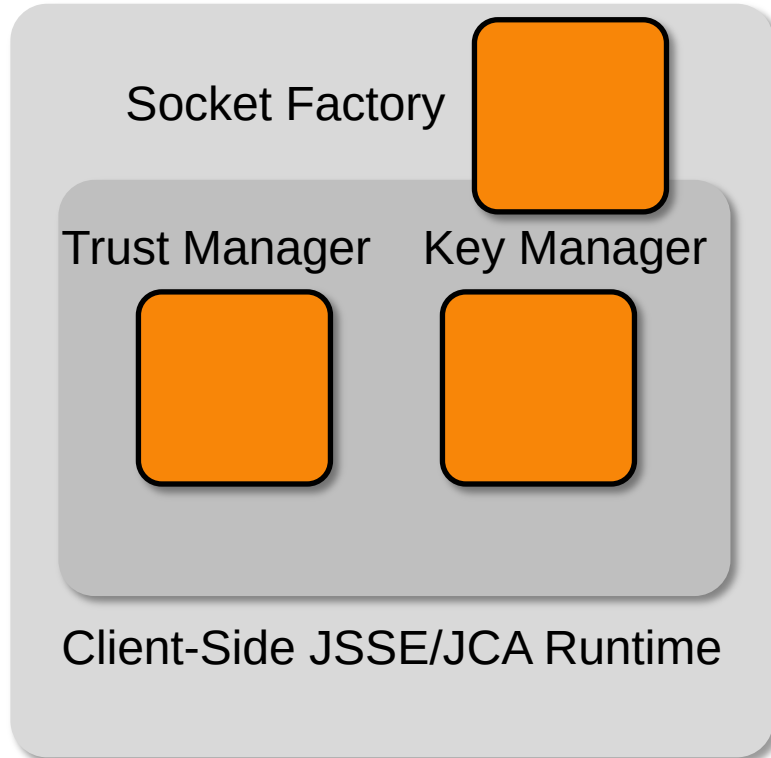
SUMMIT

**JBoss
WORLD**

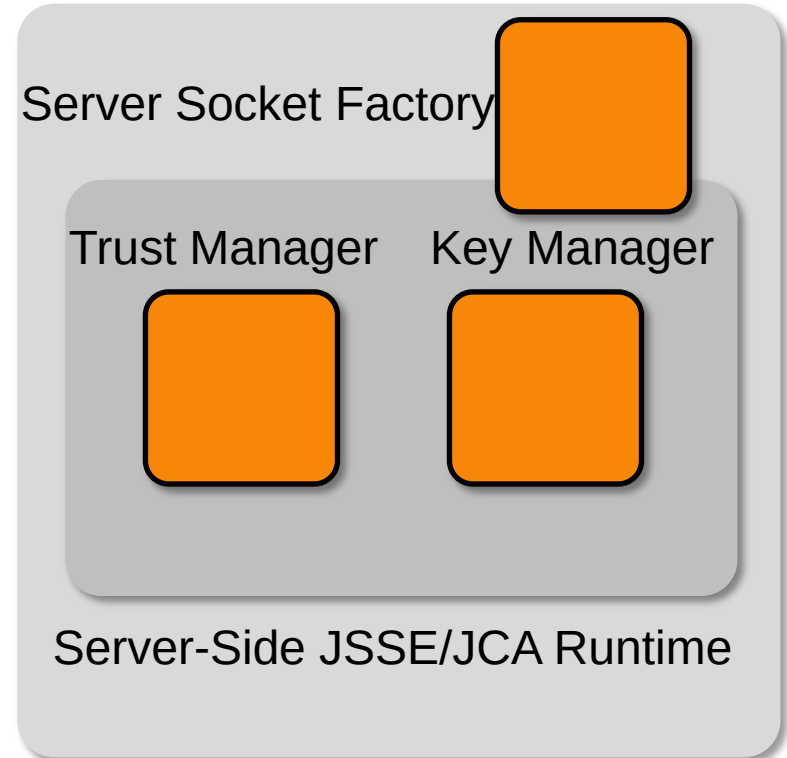
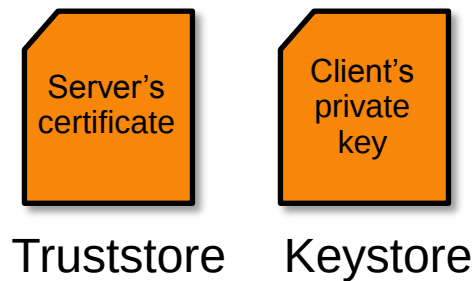
PRESENTED BY RED HAT



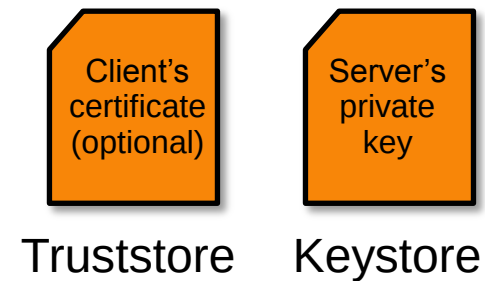
JSSE Key Elements



Client JVM



ServerJVM



SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Java Cryptography Architecture

- Java Cryptography Architecture(JCA) defines a provider architecture and APIs for: [2]
 - Digital signatures
 - Encryption
 - Certificate validation
 - Key management
 - Random number generation
 - Others
- JCE—Java Cryptography Extensions— part of JCA in Java 6
 - Adds more cryptography algorithm/functionality support
- JSSE uses JCA provider architecture.
 - Implemented as a set of providers & APIs
- JSSE implementations use JCA/JCE functionality for cryptographic operations

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Security in JBoss

- Security in JBoss is reduced essentially to
 - Securing communication between client and server
 - Providing authentication services

Implemented via various security interceptors that delegate authentication to a JBoss security manager, which accesses a credential storage

- Providing authorization services

Similarly implemented, with the difference that the JBoss security manager accesses a role storage

SUMMIT

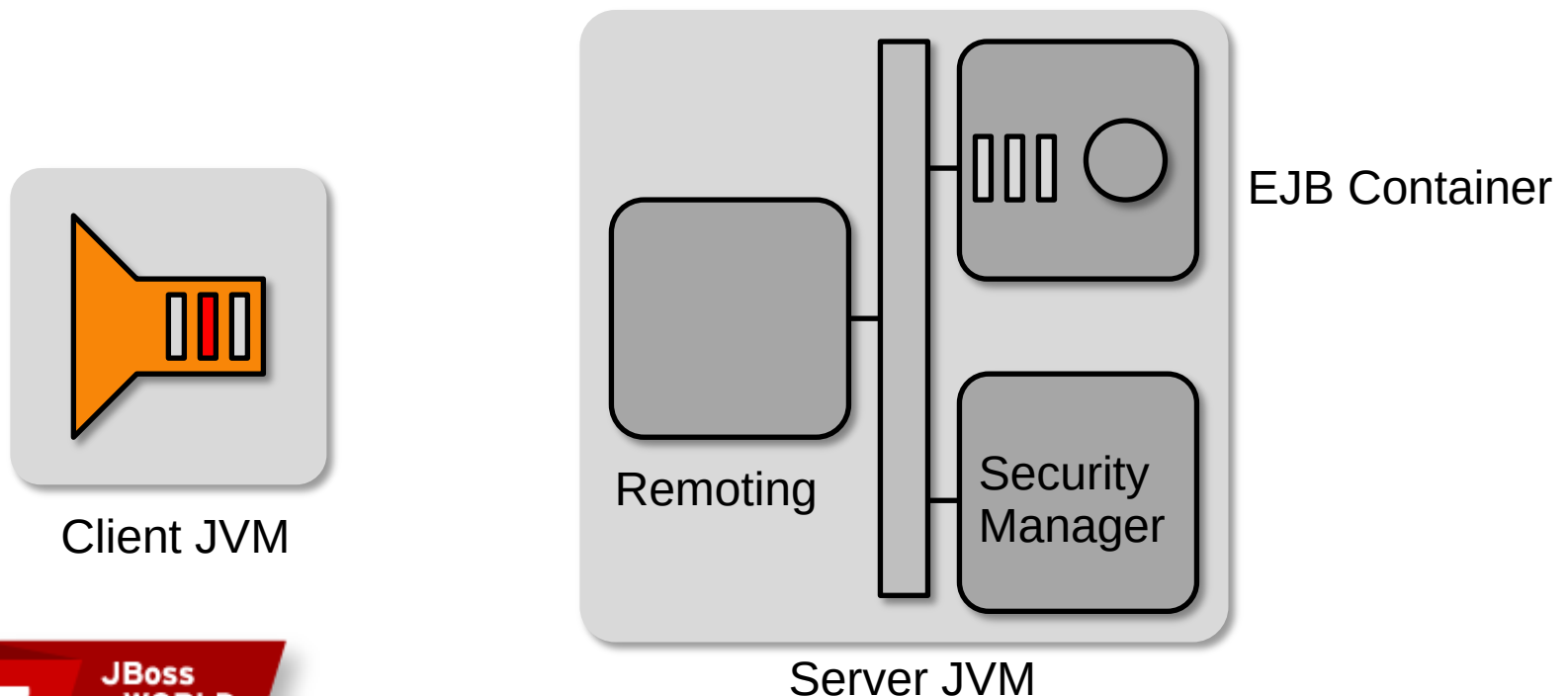
**JBoss
WORLD**

PRESENTED BY RED HAT



Step-by-Step Security in JBoss - Programmatic Client Login

- The client runtime communicates to the server “who” it is
- The caller’s identity (user name, id, SSN, etc.) and credentials are gathered via a client-side programmatic JAAS login
- The caller’s identity is submitted to the server along with credentials
- Authentication/authorization will be performed on the server
- All subsequent invocations require an authenticated identity as well



SUMMIT

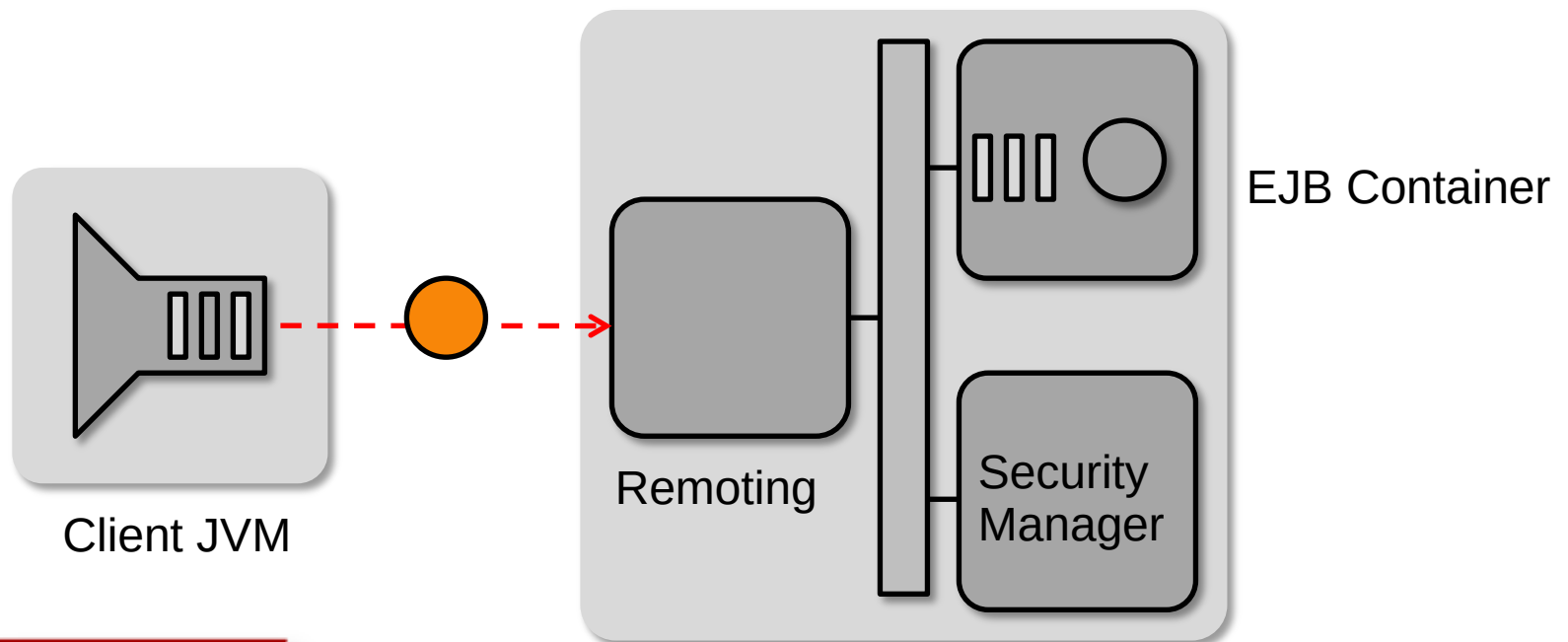
JBoss
WORLD

PRESENTED BY RED HAT



Step-by-Step Security in JBoss – Security Context Propagation

- Principal and credentials are sent to the server with each invocation
- Web applications use servlet container's login mechanism (BASIC, FORM, etc.) to provide identity and credentials
- EJB client's security interceptor query the JBoss SecurityAssociation for principals and credentials
- Both identity and credentials are serialized as part of the invocation



SUMMIT

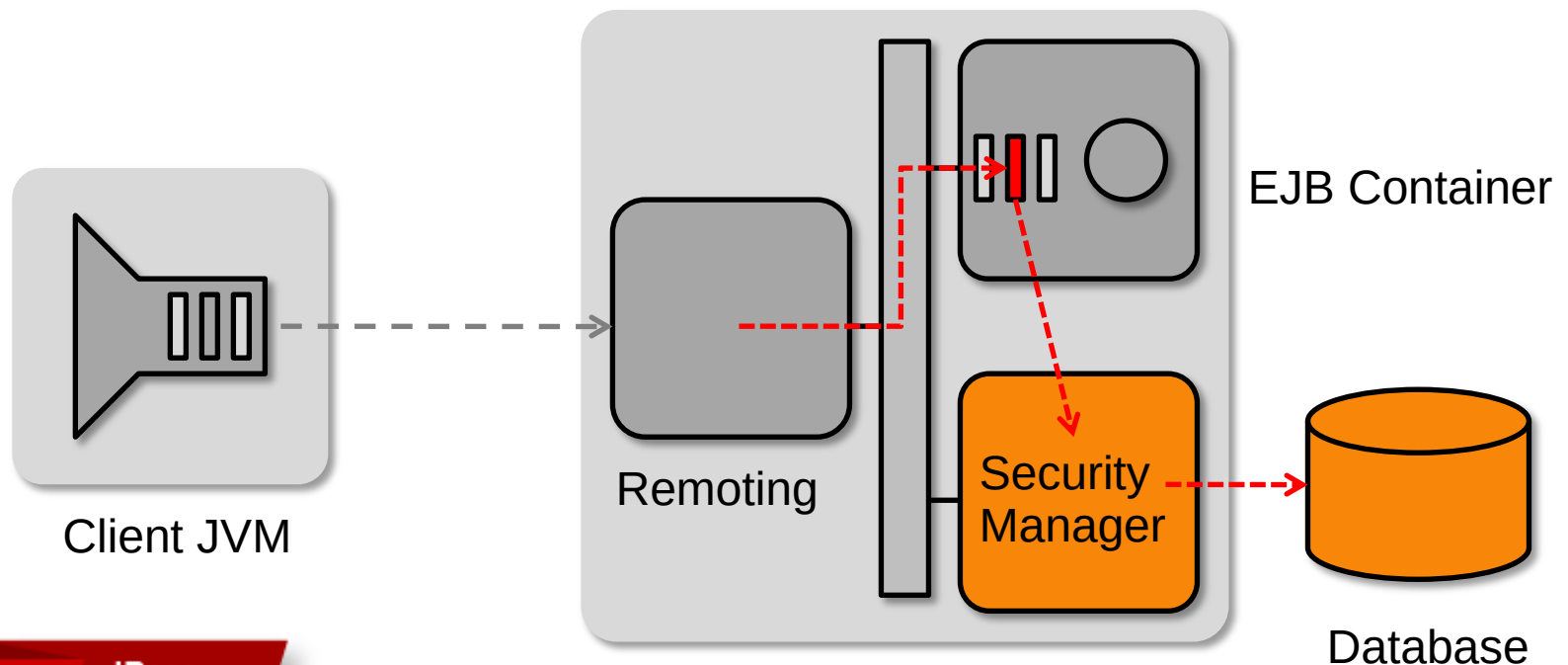
JBoss
WORLD

PRESENTED BY RED HAT



Step-by-Step Security in JBoss - Authentication

- The server verifies the caller's identity
- JBoss' security manager uses JAAS pluggable authentication modules to integrate with existing security infrastructure (operating system authentication, flat file authentication, database authentication, LDAP authentication)
- Once the caller has been authenticated, the principal is associated with the calling thread via a Server-Side SecurityAssociation



SUMMIT

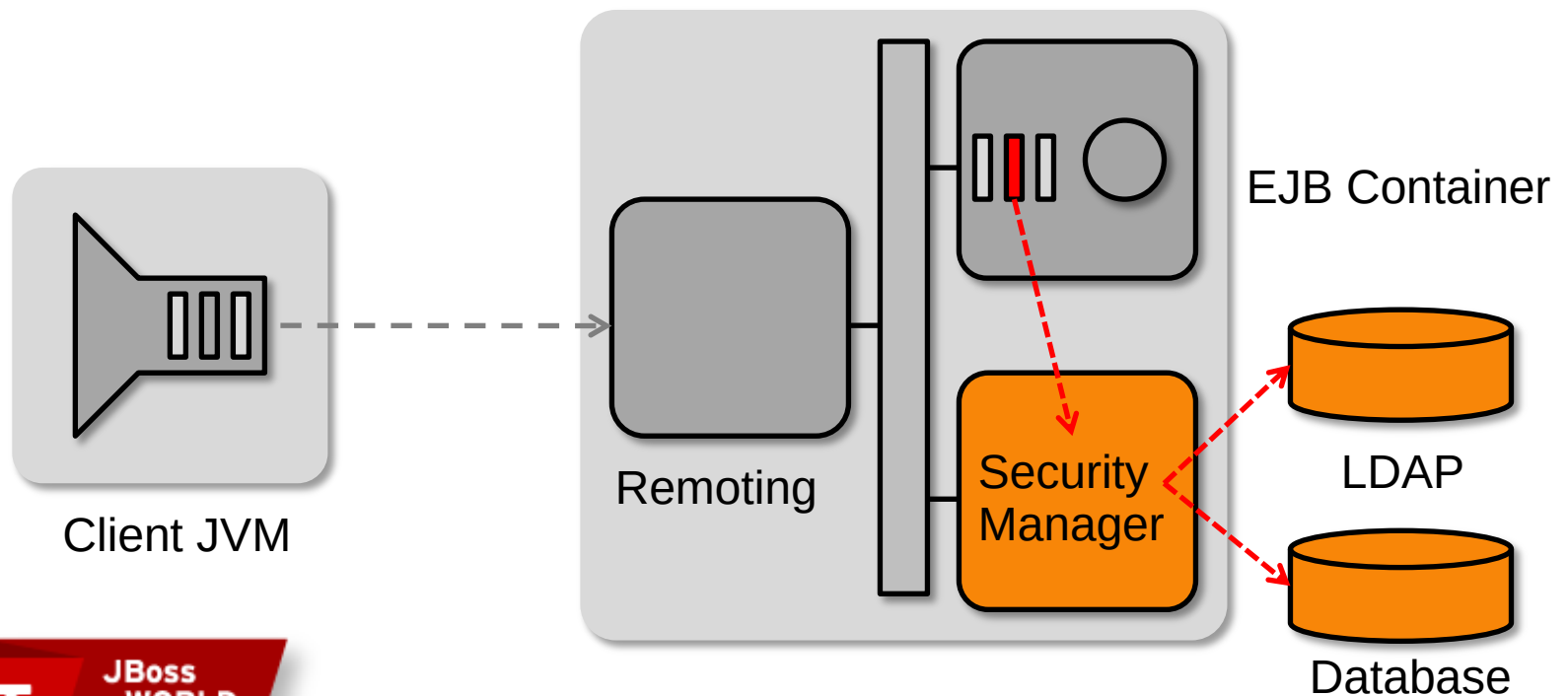
JBoss
WORLD

PRESENTED BY RED HAT



Step-by-Step Security in JBoss - Authorization

- Once the caller's identity is known and verified, the server checks whether the caller has the required privileges to invoke the operation
- Roles are also retrieved via JAAS PAM
- PAM modules can be combined, authentication and authorization can rely on information coming from different sources



JaasSecurityDomain

- Primary SSL configuration mechanism in JBoss
- Most JBoss functionality that uses SSL provides a JaasSecurityDomain reference mechanism
- Defined in any *-service.xml file under \$SERVER_HOME/deploy
- The JaasSecurityDomain is an extension of the JaasSecurityManager[1]
 - adds the notion of a
 - KeyStore,
 - KeyManagerFactory
 - TrustManagerFactory
 - supports SSL and other cryptographic use cases
- General-purpose configuration mechanism for building
 - SSLServerSocketFactory objects
 - SSLSocketFactory objects

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



JaasSecurityDomain Configuration Example

```
<mbean name="jboss.security=JaasSecurityDomain,domain=A_Security_Domain"
```

JaasSecurityDomain Name

```
    code="org.jboss.security.plugins.JaasSecurityDomain">
```

JaasSecurityDomain Class

```
    <attribute name="KeyStoreType">JKS</attribute>
```

Keystore Type

```
    <!-- ServerAlias will key initilaization of an SSLServerSocketFactory -->
```

```
    <attribute name="ServerAlias">my-server-alias</attribute>
```

Server Certificate Alias

```
    <!--ClientAlias will key initialization of an SSLSocketFactory -->
```

Client Certificate Alias

```
    <attribute name="ClientAlias">my-client-alias</attribute>
```

Keystore Password

```
    <attribute name="KeyStorePass">changeit</attribute>
```

Keystore URL

```
    <attribute name="KeyStoreURL">file://path/to/some/keystore.jks</attribute>
```

```
    <attribute name="TrustStoreType">JKS</attribute>
```

Truststore Type

```
    <attribute name="TrustStoreURL">file:///path/to/some/truststore.jks</attribute>
```

Truststore URL

```
    <attribute name="TrustStorePass">changeit</attribute>
```

Truststore Password

```
    <attribute name="ClientAuth">>false</attribute>
```

Client Authentication Required?

```
    <constructor>
```

```
        <arg value="A_Security_Domain"
```

```
        type="java.lang.String" />
```

```
    </constructor>
```

```
</mbean>
```

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



JAASSecurityDomain Benefits

- It's possible to use SSL by configuring JSSE using default or command-line System Properties.
 - May be appropriate for simple cases.
 - Limits you to one Keystore.
 - Puts password in Java arguments command line.
 - No control over which Certificate Alias is used.
- Use of JBoss JAASSecurityDomain MBeans overcomes these limitations.
 - Otherwise, must manually create a SocketFactory object for each use-case in Java code.
- Provides a central mechanism to configure SSL endpoints used by JBoss.

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



JAASSecurityDomain Enhancements

- Accommodate multiple keys in the same keystore.
 - Specify a Server Certificate Alias
 - ServerAlias attribute
 - Specify a Client Certificate Alias
 - ClientAlias attribute
- Should a Server Endpoint require Client Authentication?
 - ClientAuth attribute
- Ability to specify JSSE Provider arguments via JAASSecurityDomain MBean.
 - Note, Red Hat development did this.



Securing JBoss Communication

- JBossWeb – SSL Tomcat Connectors
- JBossRemoting Unified Connector Port
- JRMPInvoker Service Port
- IIOP Listening Port
- EJB3 Remoting Connector
- JMS Remoting Bisocket Port
- JNDI/HAJNDI
- Encrypted cluster multicast communication
- JBossWS client-side
- LdapExtLoginModule

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



JBossWeb

- Web/Servlet/JSP container
- Configuration File:
 - \$SERVER_HOME/deploy/jboss-web.deployer/server.xml
- Tomcat Connectors
 - Can reference a JAASSecurityDomain: security-domain attribute.
- Notes
 - Certificate Alias name and Client Authentication must be specified in server.xml.



JBossWeb SSL Configuration

```
<Connector SSLImplementation="org.jboss.net.ssl.JBossImplementation"
```

```
    protocol="HTTP/1.1"
```

```
    ciphers="..."
```

```
    SSLEnabled="true"
```

```
    maxThreads="150"
```

```
    clientAuth="false"
```

```
    secure="true"
```

```
    keyAlias="my_alias"
```

```
    connectionTimeout="20000"
```

```
    scheme="https"
```

```
    port="12345"
```

```
    sslProtocol="SSLv3"
```

```
    address="${jboss.bind.address}"
```

```
    securityDomain="java:/jaas/TOMCAT_SSL"
```

Does Connector Use SSL?

Is Client Authentication required?

Is the Connector Secure?

Server certificate alias

SSL Protocol/Version.

JassSecurityDomain JNDI Name

```
/>
```

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



JBossRemoting Unified Connector(EJB2.x), EJB3 Remoting Connector(EJB3), JMS Remoting Bisocket Connector (JMS)

- Used by remote clients to make EJB2.x, EJB3, and JMS calls.
- Config Files
 - conf/jboss-service.xml -- Unified Connector
 - deploy/ejb3.deployer/META-INF/jboss-service.xml – EJB3 Remoting Connector
 - deploy/jboss-messaging/remoting-sslbisocket-service.xml – JMS Remoting Bisocket Connector
- Depend on:
 - Mbean references a ServerSocketFactory MBean.
 - ServerSocketFactory MBean references JAASSecurityDomain.



JBossRemoting Unified Connector(EJB2.x) SSL Configuration

```
<mbean name="jboss.remoting:service=Connector,transport=socket"
```

```
  code="org.jboss.remoting.transport.Connector"
```

```
  display-name="Socket transport Connector">
```

```
<attribute name="Configuration">
```

```
<config> Remoting Unified Connector MBean
```

```
<handlers>
```

```
  ServerSocketFactory MBean
```

```
    <handler  
      subsystem="invoker">jboss:service=invoker,type=unified</handler  
>
```

```
</handlers>
```

```
<invoker transport="socket">
```

```
...
```

```
<attribute  
  name="serverSocketFactory">jboss.remoting:service=Server  
  SocketFactory,type=RemotingSecurityDomain</attribute>
```

```
...
```

```
</invoker>
```

```
</config>
```

```
</attribute>
```

```
<depends>jboss.remoting:service=NetworkRegistry</depends>
```

```
<depends>jboss.remoting:service=ServerSocketFactory,type=Remotin  
gSecurityDomain</depends>
```

```
</mbean>
```

```
<mbean  
  code="org.jboss.remoting.security.domain.DomainServerSocketFa  
  ctoryService"
```

```
  name="jboss.remoting:service=ServerSocketFactory,type=Remotin  
  gSecurityDomain"
```

```
  display-name="SecurityDomain Server Socket Factory">
```

```
  <attribute  
    name="SecurityDomain">java:/jaas/REMOTING_MASSL</attri  
    bute>
```

```
  JaasSecurityDomain reference
```

```
  <depends>Jboss.security:service=JaasSecurityDomain,domai  
  n=REMOTING_MASSL</depends>
```

```
</mbean>
```

```
  ServerSocketFactory MBean Reference
```

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



EJB3 Remoting Connector(EJB3) SSL Configuration

EJB3 Remoting Connector MBean

```
<mbean code="org.jboss.remoting.transport.Connector"
```

```
name="jboss.remoting:type=Connector,name=DefaultEjb3Connector,transport=socket,handler=ejb3">
```

```
<attribute name="Configuration">
```

```
<config>
```

ServerSocketFactory MBean

```
<handlers>
```

```
<handler
  subsystem="AOP">org.jboss.aspects.remoting.AOPRemoting
  InvocationHandler</handler>
```

```
</handlers>
```

```
<invoker transport="socket">
```

```
...
```

JaasSecurityDomain Reference

```
<attribute
  name="serverSocketFactory">jboss.remoting:service=Se
  rverSocketFactory,type=SecurityDomain</attribute>
```

```
...
```

```
</invoker>
```

```
</config>
```

ServerSocketFactory Mbean Reference

```
</attribute>
```

```
<attribute
  name="InvokerLocator">sslsocket://${jboss.bind.address}:12
  345</attribute>
```

```
<depends>jboss.aop:service=AspectDeployer</depends>
```

```
<depends>jboss.remoting:service=ServerSocketFactory,
  type=SecurityDomain</depends>
```

```
</mbean>
```

```
<mbean
```

```
code="org.jboss.remoting.security.domain.DomainServerSoc
  ketFactoryService"
```

```
name="jboss.remoting:service=ServerSocketFactory,type=Se
  curityDomain"
```

```
display-name="SecurityDomain Server Socket Factory">
```

```
<attribute
  name="SecurityDomain">java:/jaas/EJB3_MASSL</attrib
  ute>
```

```
<depends>JBossSX:service=JaasSecurityDomain,domain=SYS
  TEM_DEFAULT_SSL</depends>
```

```
bean>
```

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



JMS Remoting Bisocket Connector (JMS) SSL Configuration

```
<mbean code="org.jboss.remoting.transport.Connector"
name="jboss.messaging:service=Connector,transport=sslbisocket"
display-name="SSL Bisocket Transport Co
```

```
nconnector">
```

Remoting Bisocket Connector MBean

```
<attribute name="Configuration">
```

```
<config>
```

```
<invoker transport="sslbisocket">
```

```
....
```

```
<attribute name="clientSocketClass"
isParam="true">org.jboss.jms.client.remoting.ClientSocketWrapper
</attribute>
```

```
<attribute
name="serverSocketClass">org.jboss.jms.server.remoting.ServerS
ocketWrapper</attribute>
```

```
<attribute
name="serverSocketFactory">jboss.messaging:service=Serve
rSocketFactory,type=SSL</attribute>
```

```
...
```

```
</invoker>
```

```
<handlers>
```

```
<handler
```

ServerSocketFactory Reference

```
subsystem="JMS">org.jboss.jms.server.remoting.JMSServerInvoc
ationHandler</handler>
```

```
</handlers>
```

```
</config>
```

```
</attribute>
```

```
<depends>jboss.messaging:service=ServerSocketFactory,type
e=SSL</depends>
```

```
</mbean>
```

ServerSocketFactory MBean

```
<mbean
code="org.jboss.remoting.security.domain.DomainServerSocketFa
ctoryService"
name="jboss.messaging:service=ServerSocketFactory,type=SSL"
display-name=
```

```
"JMS SSL Server Socket Factory">
```

```
<attribute
name="SecurityDomain">java:/jaas/JBM_MASSL</attribute>
```

```
<depends>jboss.messaging:service=JaasSecurityDomain,domain=JB
M_MASSL</depends>
```

```
</mbean>
```

JaasSecurityDomain Reference

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



JRMPInvoker & JRMPInvokerHA Services

- JRMPInvoker is used by twiddle and other tools to make JMX over RMI calls into a JBoss container.
- JRMPInvokerHA is a highly available version of the service.
- Config Files
 - \$SERVER_HOME/conf/jboss-service.xml
 - \$SERVER_HOME/deploy/cluster-service.xml



JRMPInvoker Service SSL Configuration

JRMP Invoker Service MBean

```
<mbean name="jboss:service=invoker,type=jrmp"  
  code="org.jboss.invocation.jrmp.server.JRMPInvoker">
```

...

RMI SSL ClientSocketFactory Name

```
<attribute  
  name="RMIClientSocketFactory">org.jboss.security.ssl.RMISSLClientSocketFactory<  
/attribute>
```

RMI SSL ServerSocketFactory Name

```
<attribute  
  name="RMIServerSocketFactory">org.jboss.security.ssl.RMISSLServerSocketFactor  
y</attribute>
```

JaasSecurityDomain Reference

```
<attribute  
  name="SecurityDomain">java:/jaas/JRMP_INVOKER_MASSL</attribute>
```

...

JaasSecurityDomain Dependency

```
<depends>Jboss.security:service=JaasSecurityDomain,domain=JRMP_INVOKE  
R_MASSL</depends>
```

```
</mbean>
```

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



JRMPInvokerHA Service SSL Configuration

```
<mbean code="org.jboss.invocation.jrmp.server.JRMPInvokerHA"
  name="jboss:service=invoker,type=jrmpha">
  <attribute name="ServerAddress">${jboss.bind.address}</attribute>
  <attribute name="RMIObjectPort">12345</attribute>
  <attribute
    name="RMIClientSocketFactory">org.jboss.security.ssl.RMISSLClientSocketFactory<
  /attribute>
  <attribute
    name="RMISSLSocketFactory">org.jboss.security.ssl.RMISSLServerSocketFactory<
  /attribute>
  <attribute
    name="SecurityDomain">java:/jaas/JRMP_INVOKER_MASSL</attribute>
  <depends>jboss:service=Naming</depends>
  <depends>JBossSX:service=JaasSecurityDomain,domain=JRMP_INVOKER_M
  ASSL</depends>
</mbean>
```

JRMPInvokerHA MBean

JaasSecurityDomain Reference

JaasSecurityDomain Reference

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



IIOP Listening Port

- Used by remote CORBA clients.
- IIOP Protocol
- Config File:
 - `$SERVER_HOME/deploy/iiop-service.xml`
 - SSL Port is defined in `$SERVER_HOME/conf/jacorb.properties`.



IIOP Listening Endpoint SSL Configuration

```
<mbean code="org.jboss.iiop.CorbaORBService"  
  name="jboss:service=CorbaORB">
```

IIOP Service MBean

...

```
<attribute name="SecurityDomain">java:/jaas/IIOP+SSL</attribute>
```

...

JaasSecurityDomain Reference

```
<depends>Jboss.security:service=JaasSecurityDomain,domain=IIOP+SSL</depends>
```

...

JaasSecurityDomain Dependency

```
<depends>jboss:service=Naming</depends>
```

```
</mbean>
```

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



JNDI/HAJNDI

- Remote JNDI queries are usually done over unencrypted channels.
 - Generally, non-sensitive information.
- There are exceptions.
- JBoss supports making JNDI requests over HTTPS using the httpha-invoker.sar service.
 - See \$SERVER_HOME/deploy/httpa-invoker.sar/META-INF/jboss-service.xml.
- Must configure an SSL Connector for JBossWeb (see previous slide).



Encrypted Cluster Communication

- JGroupsENCRYPT can be used to encrypt JGroups communication.
- Can be used to secure TCP or UDP communications.
- Default JChannels in “all” container.
 - boss.service=\${jboss.partition.name:DefaultPartition}
 - boss.cache.service=EJB3SFSBClusteredCache
 - jboss.cache.service=EJB3EntityTreeCache
 - jboss.cache.service=TomcatClusteringCache
 - jboss.messaging.service=PostOffice -- DataChannelConfig (uses TCP by default,
 - jboss.messaging.service=PostOffice – ControlChannelConfig
- How it works:
 - Private/Public Key are generated by container.
 - Public Key shared among cluster members.
 - Encrypts a shared key.
 - Shared key used encrypt/decrypt each transmitted multicast packet.

Example

```
...
<VERIFY_SUSPECT ... />
  <ENCRYPT
    encrypt_entire_message="true"
    sym_init="128"
    sym_algorithm="AES/ECB/PKCS5Padding"
    asym_init="512" asym_algorithm="RSA"/>
  <pbcast.NAKACK ... />
...
```

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



JBossWS – Native SOAP Web Services

- Want to abstract details of SSLSocketFactory initialization away from application code.
- Red Hat added
StubExt.PROPERTY_SOCKET_FACTORY
 - Must be set programatically.
- Passes SocketFactory into JBossRemoting
- Provide custom SSLSocketFactory that does a JNDI lookup on a JaasSecurityDomain.

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



ExtLdapLoginModule SSL Communication

- LdapExtLoginModule can be used to define a LDAP User Repository in JBoss.
- Uses JDK JNDI LDAP Connection Pooling functionality to maintain connections to LDAP.
- JDK library allows an SSLSocketFactory property to be set called[2]:
 - `java.naming.ldap.factory.socket`
- Can be set as a JAAS LoginModule attribute in `$SERVER_HOME/conf/login-config.xml`
- Takes SSLSocketFactory class name.
- Socket Factory can do a JNDI lookup on a JaasSecurityDomain.



Closing Thoughts

- We have briefly reviewed
 - the JBoss Services that may need to be secured.
 - How to secure these Services.
 - Use of JaasSecurityDomain Mbeans to secure these Services with SSL.
 - Enhancements to JaasSecurityDomain MBeans.
- We have assembled information regarding JBoss's SSL use into one place.



References

- [1] <http://community.jboss.org/wiki/JaasSecurityDomain>
- [2] <http://java.sun.com/products/jndi/tutorial/ldap/security/ssl.html>
- [3] <http://java.sun.com/javase/6/docs/technotes/guides/security/jsse/JSSERefGuide.html#Introduction>
- [4] http://www.redhat.com/docs/en-US/JBoss_Enterprise_Application_Platform/4.3.0.cp07/html/Server_Configuration_Guide/The_JBoss_Security_Extension_Architecture-The_JaasSecurityDomain_MBean.html
- [5] <http://community.jboss.org/wiki/JaasSecurityDomain>
- [6] <http://java.sun.com/javase/6/docs/technotes/guides/security/jsse/JSSERefGuide.html>

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Thank You

- Questions?????

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



FOLLOW US ON TWITTER

www.twitter.com/redhatsummit

TWEET ABOUT IT

[#summitjbw](https://twitter.com/summitjbw)

READ THE BLOG

<http://summitblog.redhat.com/>

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT

