SUMMIT JBoss WORLD

PRESENTED BY RED HAT

LEARN. NETWORK. EXPERIENCE OPEN SOURCE.

redhat.

TRUSTED SECURITY WITH JBOSS ENTERPRISE APPLICATION PLATFORM

Anil Saldhana, Red Hat Inc
Robert C. Broeckelmann Jr., Nova Ordis, LLC

06.29.2012

About the Speakers

- Anil Saldhana
  - Lead Middleware Security Architect, Red Hat Inc
  - Founder of Project PicketLink
- Robert C. Broeckelmann, Jr.
  - Partner & Principal Consultant, Nova Ordis, LLC.
  - A Services and Development Company, Red Hat Partner

Disclaimers

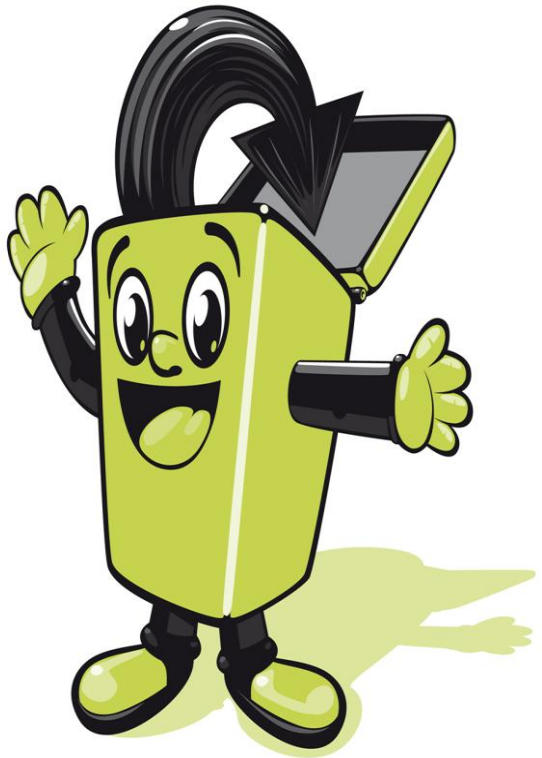
- This presentation contains a few of the possible uses of Red Hat technologies.
- Your situation and requirements probably differ.
- As always, please test in a non-production environment before using in production.
- We are not responsible for the spontaneous combustion of the known universe or any other undesirable outcomes associated with using what is discussed here.
  - Good Luck!

Agenda

- JBoss EAP 6.0 Security
- PicketLink
- Security Concepts
- Related Specs
- Some Real-World Use-Cases
  - Single Sign-On between two JBoss Applications
  - Integration with a Reverse-Proxy
  - Integration with a third-party STS

JBoss EAP 6.0 Security

- Domain model holds the security configuration
  - Security Domain configuration
- Management Interfaces are secured by default
  - Http Interface
  - Native Interface (CLI)
- PicketLink is available
  - SAML based Single Sign On
  - STS based Identity Propagation



JBoss EAP 6.0 Security

- Domain model holds the security configuration
  - Security Domain configuration

JBoss EAP 6.0 Security – Domain Model

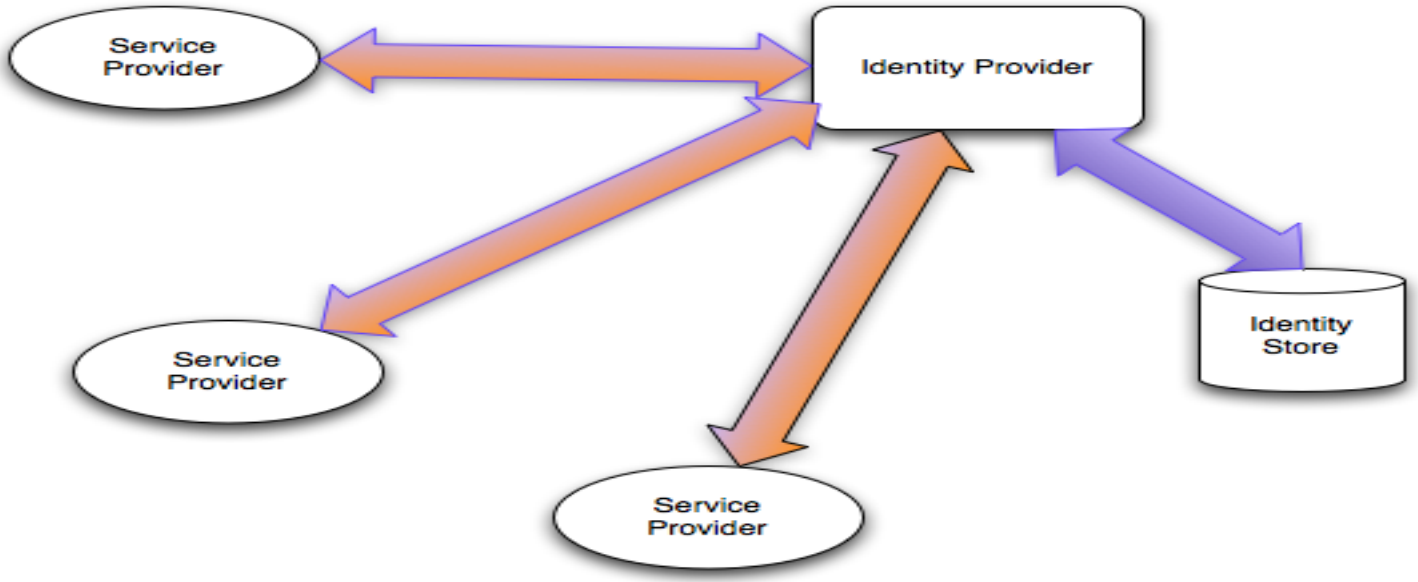
```
<subsystem xmlns="urn:jboss:domain:security:1.0">
   <security-domains>

      ...

      <security-domain name="idp" cache-type="default">
         <authentication>
            <login-module code="UsersRoles" flag="required">
               <module-option name="usersProperties" value="users.properties"/>
               <module-option name="rolesProperties" value="roles.properties"/>
            </login-module>
         </authentication>
      </security-domain>

      <security-domain name="picketlink-sts" cache-type="default">
         <authentication>
            <login-module code="UsersRoles" flag="required">
               <module-option name="usersProperties" value="users.properties" />
               <module-option name="rolesProperties" value="roles.properties" />
            </login-module>
         </authentication>
      </security-domain>

      <security-domain name="sp" cache-type="default">
         <authentication>
            <login-module code="org.picketlink.identity.federation.bindings.jboss.auth.SAML2LoginModule" flag="required"/>
         </authentication>
      </security-domain>

      ...

   </security-domains>
</subsystem>
```


JBoss EAP 6.0 Security – Management Interfaces

- Secured by default
  - Username/Password via http digest mechanism
- Scripts available to add users to management realm
  - add-user.sh /add-user.bat
  - No default user available

JBoss EAP 6.0 Security – PicketLink

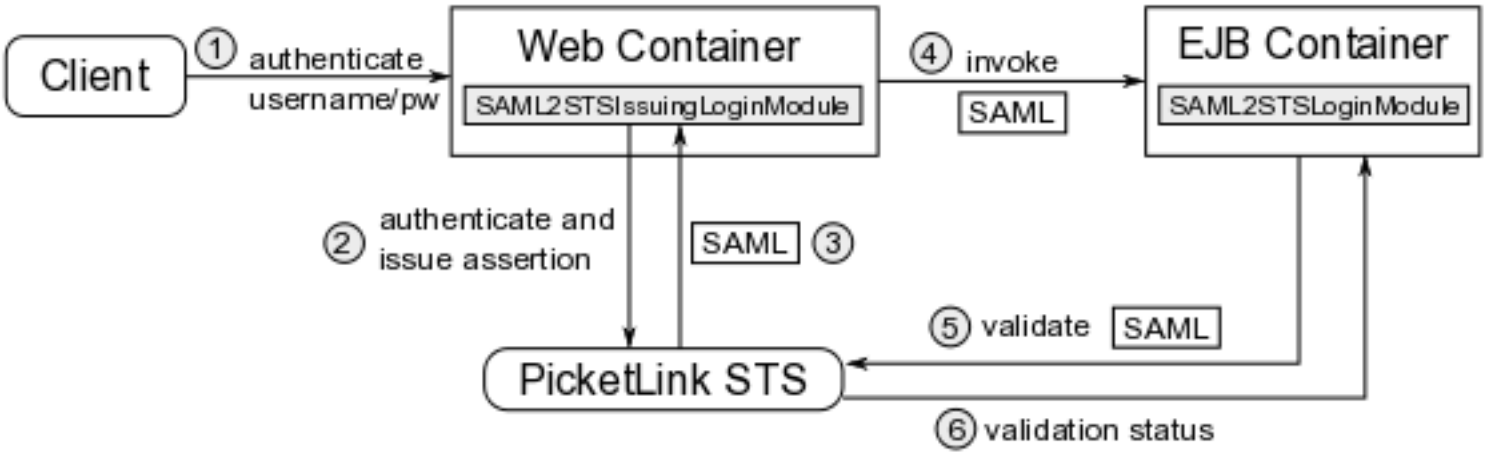
- Provides SAML based Web Browser SSO
  - Identity Provider (IDP)
  - Multiple Service Providers (SP)



SAML WEB BROWSER SSO

JBoss EAP 6.0 Security – PicketLink

- Provides STS based Identity Propagation
  - PicketLink STS
  - STS JAAS Login Modules (act as Clients)



Related Security Concepts

- Security in the Infrastructure
- Standards-Based Security
- Authentication
- Principal
- Subject
- User Repository

More ...

Related Security Concepts

- LDAP
- Security Token
- Authorization
- Identity Propagation
- Security Token Service(STS)
- SSL

Security in the Infrastructure

- Abstract security details away from application code and deployment descriptors to the greatest extent possible.
- Security should be an administrative task, not a development task.
- Generally, see getting 90% towards this goal on a given implementation.

Standards-Based Security Models

- If there is an industry spec. that provides a solution to a security problem, best approach tends to be to use that standard.
- NOT all implementations(vendor products) are
  - Created equal
  - Easy to use

Authentication

- Process of a remote entity (user or system) proving its identity to the system.
- Can be achieved in a variety of ways.
- In our examples, we will use userid and password(for end-user authentication).
- Token Validation–confirm a security token is valid and trusted
  - Validating digital signature
  - Checking expiration timestamp
  - Checking user exists in a User Repository

Principal

- An entity that can be authenticated.
- Could be a system.
  - Batch job.
  - An application.
  - A computer.
- Could be an end user.
  - A Web application user in our case.

Subject

- Refers Java Authentication and Authorization Service(JAAS) Subject
  - JAAS is the Java API/SPI for implementing authentication and authorization mechanisms.
  - Basis of PicketBox/JBossSX.
- The Subject is a Java object that contains Principal objects, public credentials, and private credentials.
- Accessed through a JAAS Context.
- Application code should refer to the JBoss JAAS Subject for all information about an authenticated user.

User Repository

- A collection of user information known to the system.
- May include: usernames, passwords, groups, group membership, and other attributes
- Examples
  - LDAP
  - Flat file
  - Database
- Master copy of all user and group information within the system.
- This is often an LDAP database.

LDAP

- LDAP—Lightweight Directory Access Protocol.
- A specification.
- Very common User Repository in many organizations.
- Contains
  - User objects (plus attributes)
  - Group objects
  - Mappings that describe group membership.

Security Token

- A self-contained collection of information that systems can pass around that describes a Principal.
- May contain (we'll assume ours does):
  - User ID.
  - List of Groups.
  - Other attributes(maybe from LDAP).
- May utilize:
  - Encryption
  - Digital signature
  - Timestamp
- SAML2 spec addresses.

Authorization

- Process by which the system makes a decision of whether an authenticated principal has permission to access a resource.
- A resource could be:
  - Web Application path (Servlet, JSP, etc)
  - EJB (or EJB method)
  - Web Service
- Will often be based upon:
  - Static information –e.g., LDAP Group membership or a user attribute
  - Dynamic information –e.g., authentication method.

Identity Propagation

- Process by which one system transmits identity of a requestor to another system.
- Identity Propagation usually achieved through some form of token.
- We are using SAML2 tokens in this discussion unless otherwise indicated.

Security Token Service(STS)

- Defined by WS-Trust spec.
- Composed of Web Service(s) that perform operations on Security Tokens(create, delete, renew, transform).
- Client trusts STS.
  - SSL(server certificate)
  - Shared key
  - WS-Security
  - Other mechanisms
- Likewise, client must provide credentials to the STS to establish trust & a principal(for our purposes, the authenticated end user) known to the User Repository.
  - We'll call these the input credentials.

Security Token Service(STS)

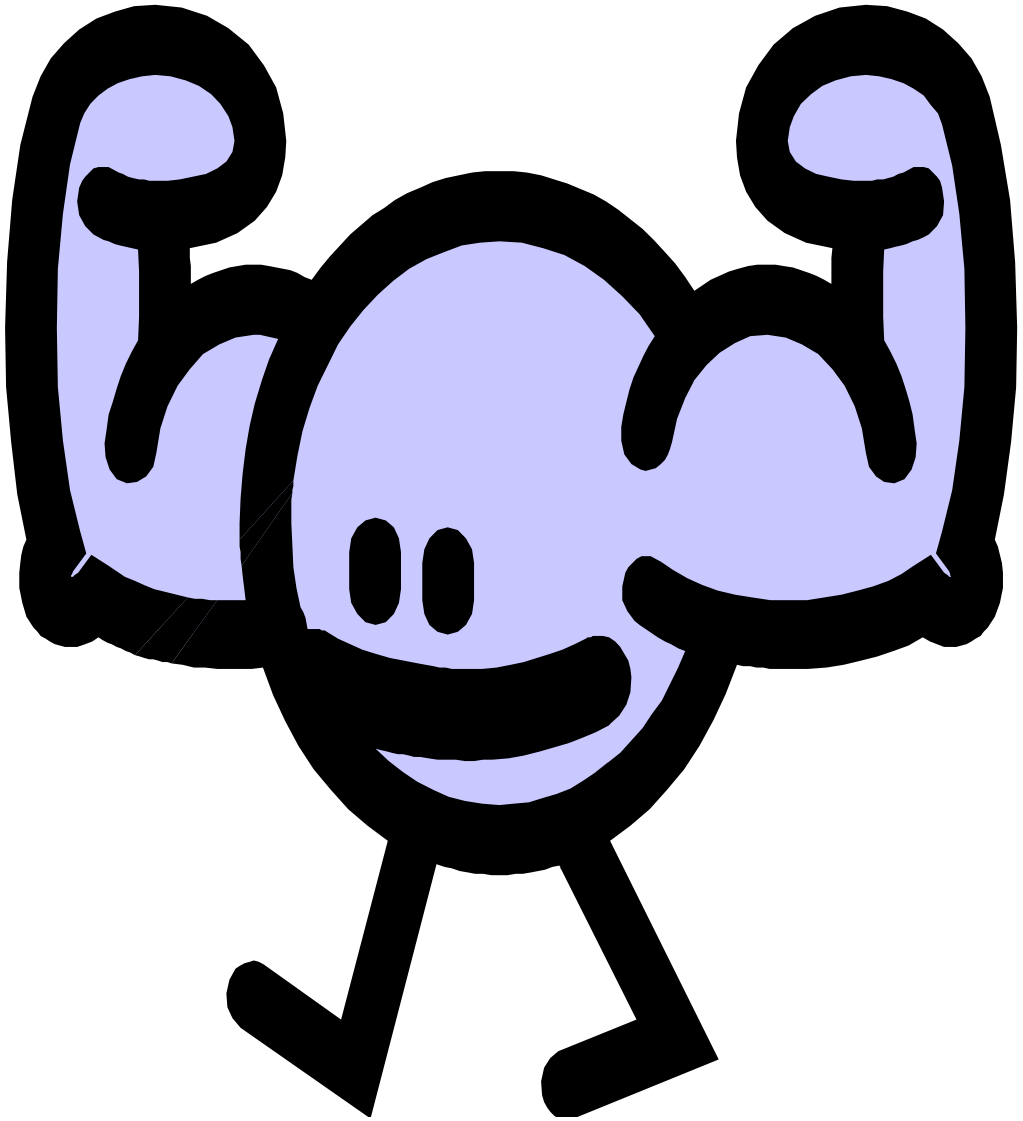
- STS provides assertions about the principal described by input credentials in the form of a Security Token.
  - We'll call this the output credential.
- This output credential should be in a format that all nodes (or at least most of them) in the distributed system understand.
- The STS can be used for all token transformations.
  - Central management of digital signature keys/certificates for security tokens.
  - Central management of
    - token generation.
    - token transformations.

SSL—Secure Sockets Layer

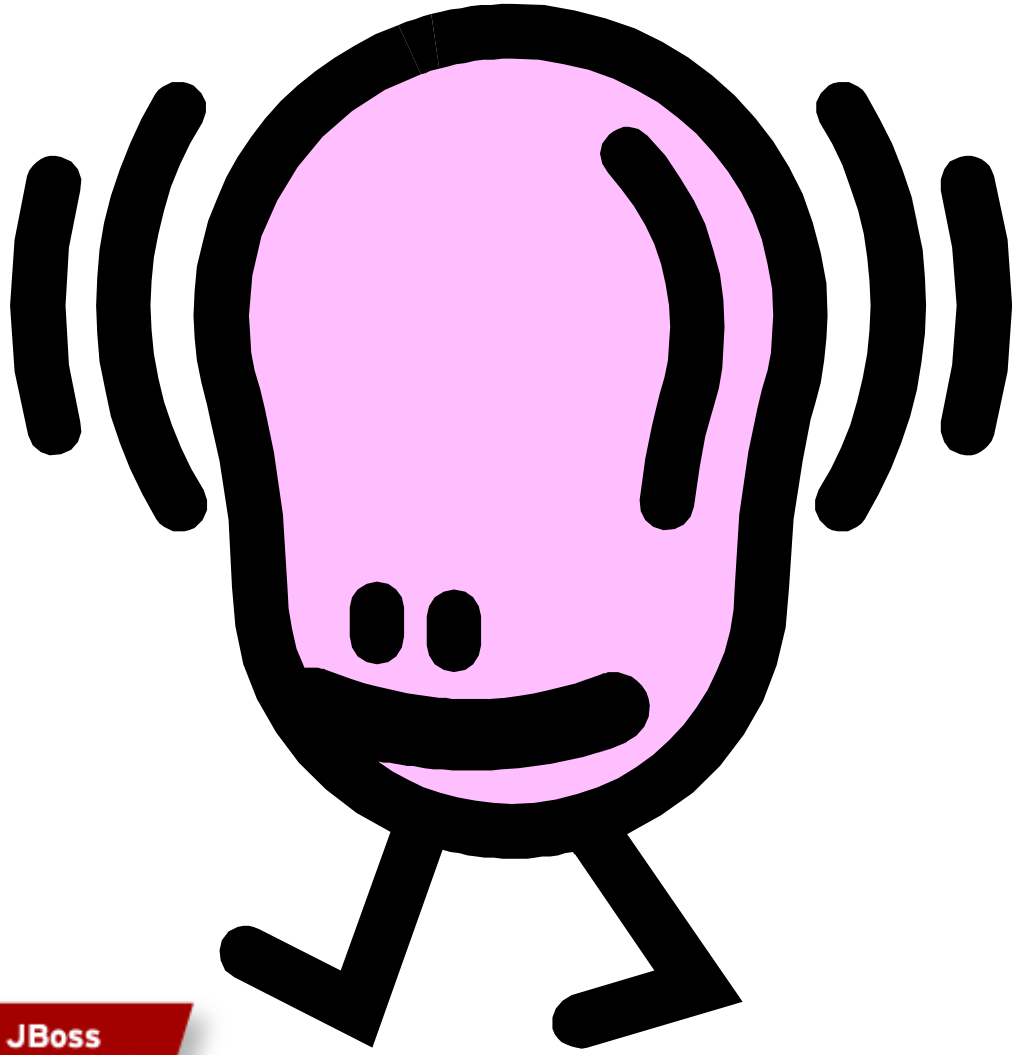
- Secure Sockets Layer (SSL) provides transport-layer security between each tier of a distributed system.
- Provides for integrity and confidentiality
- Mutually Authenticated SSL refers to the requirement of the client presenting a valid x509v3 certificate.
- Could also use alternatives (such as WS-Security Integrity & Confidentiality for SOAP Web Services)

Relevant Specs

- HTTP
- SAML2
- WS-Trust
- WS-Security
- SSL/TLS
- X509
- JAAS
- Many minor ones…



Real-World Use Cases



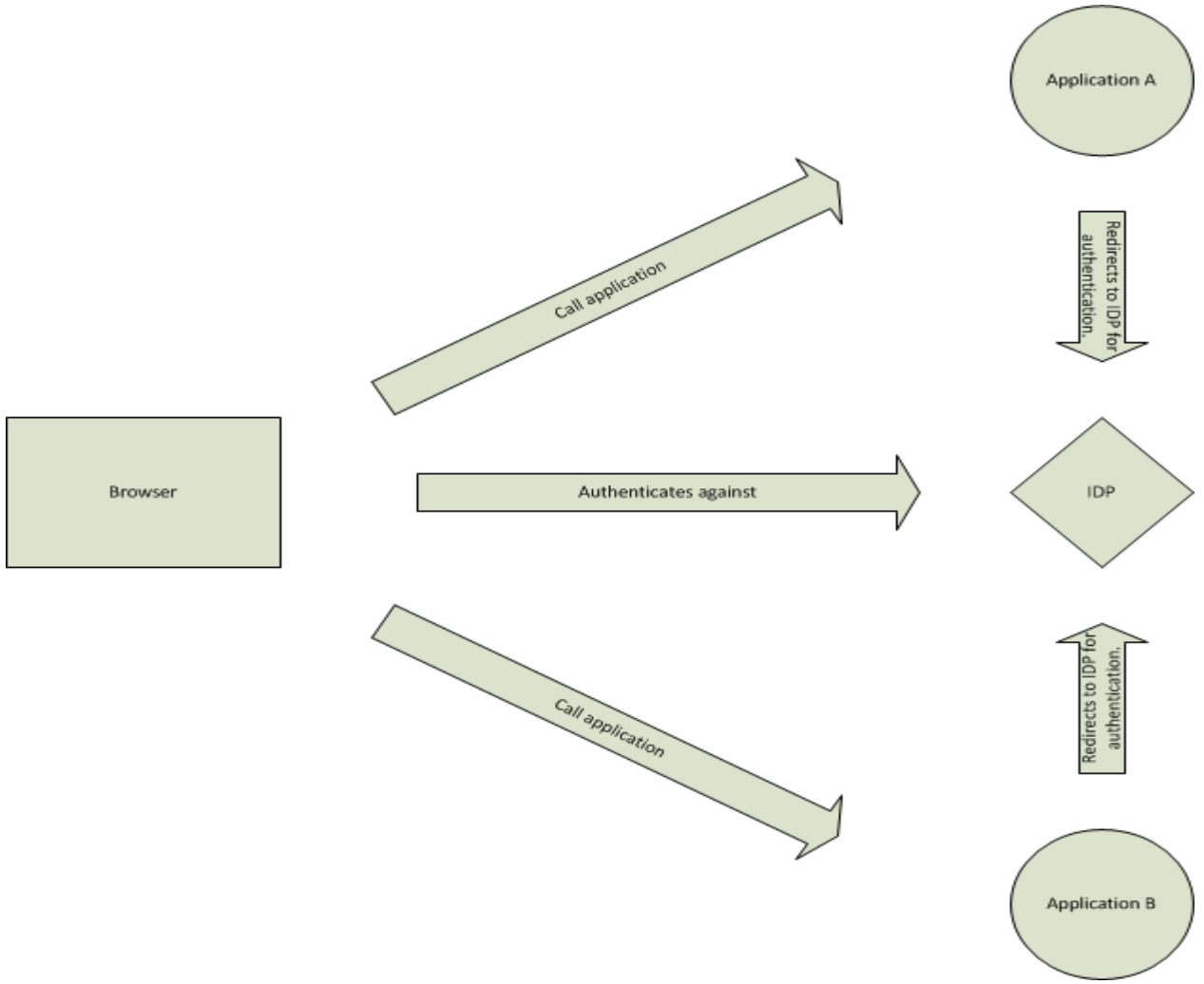
Real-World Use-Cases

- An assortment of use-cases from clients Nova Ordis, LLC has worked with.
- One of my former coworkers covered several interesting use cases at JBossWorld last year.
  - I'm intentionally covering different use cases.
- Use Cases
  - SSO between Web Applications
  - Integration with a Reverse Proxy
  - Integration with a third-party STS

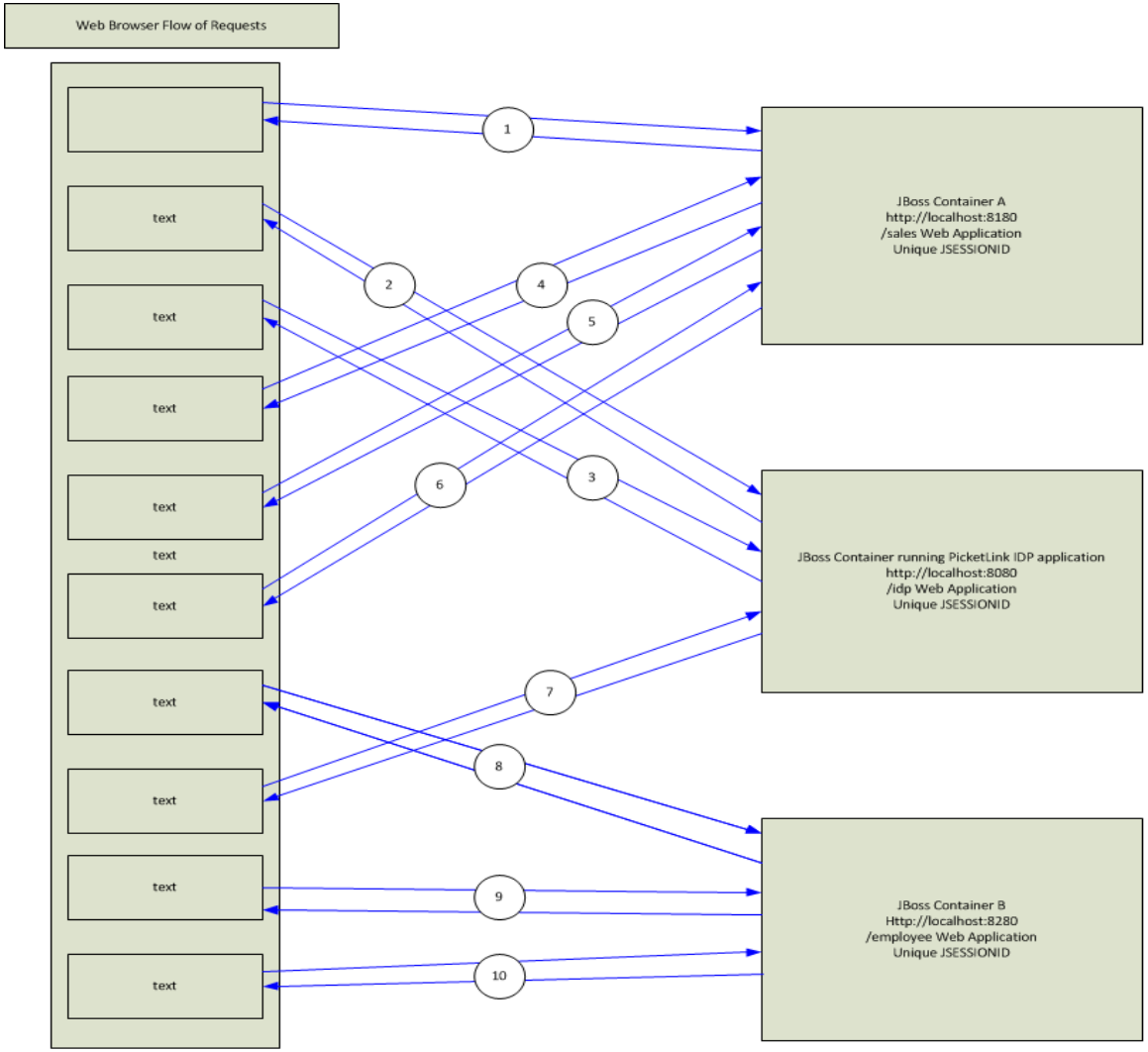
Use Case 1: Single Sign-On between two JBoss Applications

- SSO = Log in once; have access to many applications.
- Doesn't have to be JBoss container specific.
  - Very powerful.
- Could be within a trusted security realm or between two different security realms (federation).
- Described in https://community.jboss.org/wiki/SAMLWebBrowserSSOOnJBossAS70
- Going to describe the interaction between browser app A, IDP, and App B.
- Note, these web applications could be hosted on any web platform
  - WebSphere, Weblogic, .NET, etc.

Single Sign On Details



Browser->Container Interaction



1	Request: GET /sales/ Response: 302 Redirect /idp/?SAMLRequest=blah
2	Request:GET /idp/?SAMLRequest=blah Response: 200 (Login Form)
3	Request: POST /idp/j_security_check Response: 302 redirect /idp/?SAMLRequest=blah
4	Request: GET /idp/?SAMLRequest=blah Response: 302 redirect /sales/?SAMLResponse=blah
5	Request: GET /sales/?SAMLResponse=blah Response: 200 (welcome page for sales app)
6	Request: GET /sales/piechart.gif Response: 200 image (or 304 not modified on subsequent calls)
7	Request: GET /employee/ Response: 302 Redirect /idp/?SAMLRequest=blah
8	Request: GET /idp/?SAMLRequest=blah Response: 302 Redirect /employee/?SAMLResponse=blah
9	Request: GET /employee/?SAMLResponse=blah Response: 200 (welcoem page for employee app)
10	Request: GET /employee/careermap.png Response: 200 image (or 304 not modified on subsequent calls)

- All applications direct login requests to the IDP application.
- User tries to log into application A and redirects them to the IDP.
- The IDP provides a login page and relies upon JEE Security + Form Based authentication.
- User establishes a session with the IDP application that is tracked through a JSESSIONID cookie.
- IDP application returns a SAML token to the calling application(through a 302 redirect)
- Application A validates the SAML token (how?) COME_BACK
- A Security Context is created for the User A for Application A.
- The user can now access paths within Application A that he is allowed to access.
- If the user in the same authenticated browser session wishes to access Application B, Application B will also provide a redirect to the IDP.
- Since the user's browser has an authenticated session against the IDP application, the user will be issued a SAML token and redirected to the original path.
- A Security Context is created for the User on Application B.
- The user uses applications as needed.
- The Global Logout feature is used to log out of all application.
- Browser is closed.

Recommendations

- Use SSL for all network communication.
- Logging out of one application should trigger a call to the Global Logout feature of PicketLink IDP.

Notes

- Centralizes authentication of web applications.
- IDP application security configuration can be tweaked to implement the desired authentication.
  - LdapLoginModule for connection to LDAP/AD, for example.
  - SPENGO Login Module and TomCat Authenticator for SSL to the IDP(log in once to the workstation).
  - Can be used with any JAAS Login Module that is compatible with JBoss.

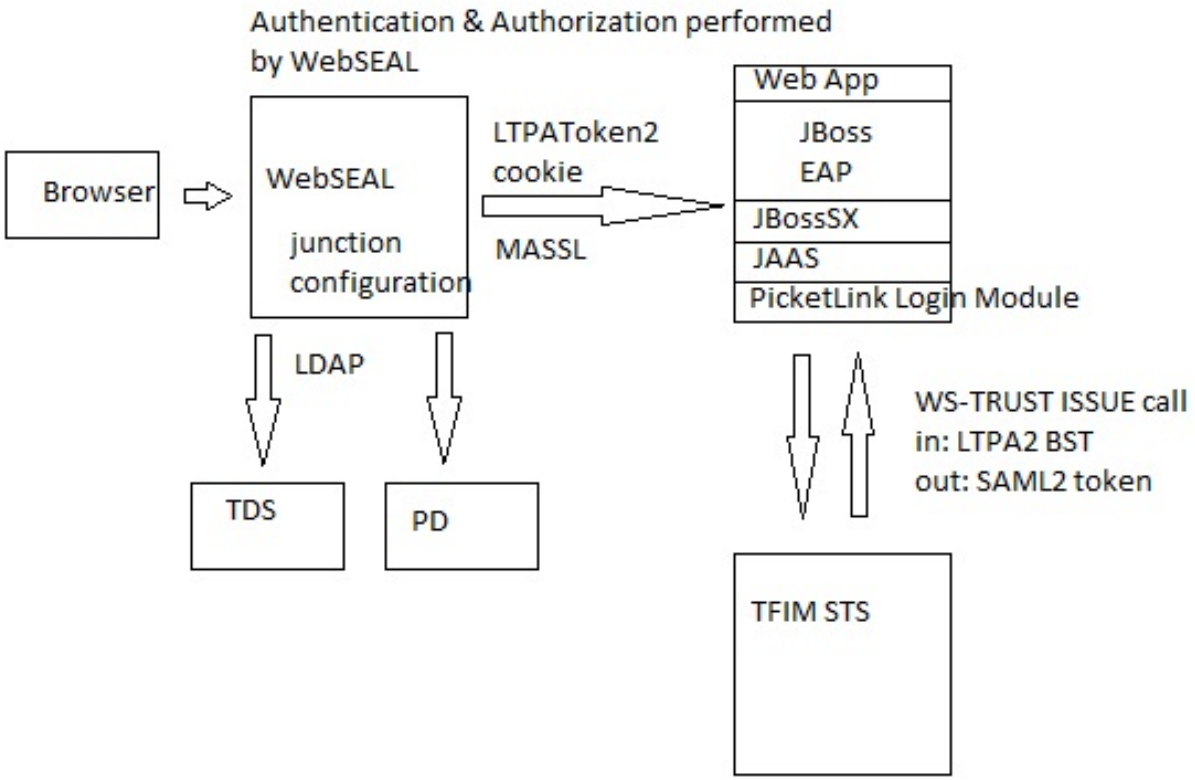
Use Case 2: Integration with a Reverse-Proxy

- A Reverse Proxy handles the authentication and authorization of web traffic within an organization.
  - Can integrate with just about any web application.
  - Sits between users and web application tiers.
  - Abstracts authentication and authorization away from application server tier.
- Can pass the authenticated identity to the web application.
  - Web application tier can use this to build its own native representation of a user's security session (ie, JBoss JAAS Subject in our case).
  - Doing this securely can be complex.
- An example of a reverse proxy would be IBM WebSphere ® Tivoli Access Manager ® (TAMeb/WebSEAL).
- This was originally implemented with PicketLink 1.x on JBoss EAP 5.1.0.
  - Some PicketLink components may not be battle hardened migrated in JBoss EAP 6.x and PicketLink 2.x.

Use of IBM TFIM STS

- PicketLink implements a client API for making WS-Trust calls.
- PicketLink provides a JAAS Login Module called SAML2STSIssuingLoginModule.
  - Can make calls to a WS-Trust compliant Security Token Service(STS).
- SAML2STSIssuingLoginModule can be configured to pull a token out of an HTTP Header or Cookie.
  - WebSEAL has the ability to pass several types of tokens.
  - Used ivcred HTTP Header and LTPAToken2 cookie(contains an LTPAv2 token).
    - Both are proprietary IBM formats.

Integration with a Reverse-Proxy (cont.)



Reverse Proxy Integration Notes

- Want secure communication between WebSEAL® and JBoss EAP
  - Secure Identity Propagation
  - Limiting who can connect to the Application Server
  - Transport layer security (Mutually Authenticated SSL)
- This system allows JBoss to build a JAAS Subject based upon the contents of an LTPAv2 token.
  - Effectively provides support for using LTPAv2 tokens in JBoss.

More Information

- If you are interested in knowing more about how to make this work, please contact me(RCBJ).
  - I have a white paper that was prepared on this topic.

Use Case 3: Integration with a third-party STS

- Actually, demonstrated with the Reverse-Proxy use-case.
  - PicketLink SAML2STSIssuingLoginModule was modified to communicate with the IBM Websphere TFIM STS.
    - Support for LTPA2 token and ivcred token passed from WebSEAL added to Login Module.
  - PicketLink is fully WS-Trust spec-compliant.
    - No changes needed for basic communication.
- These changes have not yet been battle hardened in EAP 6.0

Questions?

Thank You

- 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