ApacheCon

Understanding Tomcat Security

Anil Saldhana

Project Lead

JBoss Security and Identity Management

Red Hat Inc

EU 2007

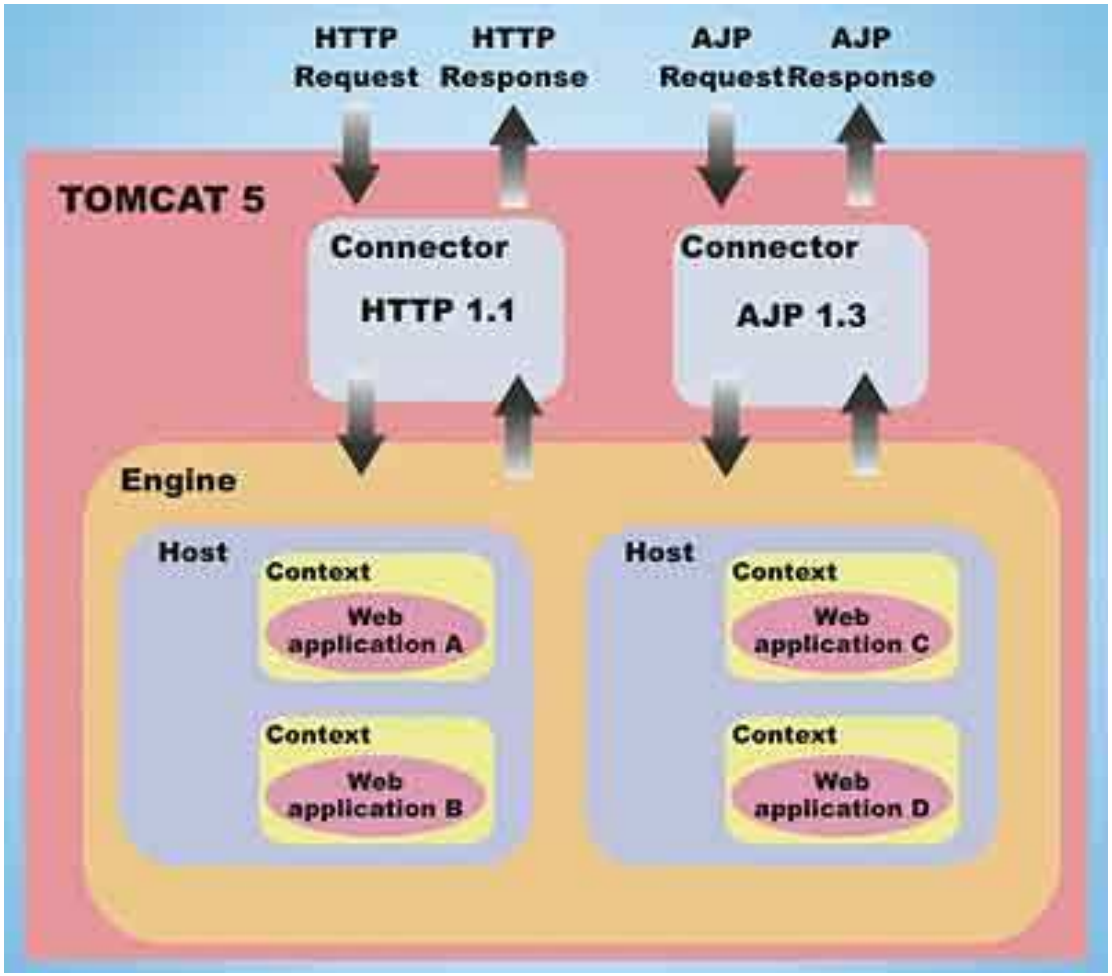
Speaker Introduction

- Apache Web Services Program Management Committee.
- Apache Scout Project Lead.
- JCP – JSR 196 Expert Group
- Oasis Technical Committees (XACML,SAML,PKI)
- W3C
- Lead, JBoss Security and Identity Management

Agenda

- Tomcat Architecture
- Tomcat Authenticators/Valves and Realms
- Tomcat Standard Valves
- Tomcat Standard Realms
- Writing custom Authenticators and realms
- Examples of use cases
- Demo, Q&A

Tomcat Architecture



- Source: 'Tomcat5', Sing Li, (http://www.vsj.co.uk/java/display.asp?id=319)

Tomcat Valves and Realms

- Valve: component that can be inserted into the request processing pipeline.
- Realm: represents a 3-tuple <username,password,roles> for users.
- Valves and Realms can be applied to an engine, host or context.

Tomcat Standard Valves

- Remote Address Filter
- Remote Host Filter
- Request Dumper Valve
- Single Sign On Valve
- Access Log Valve

Tomcat Standard Realms

- Memory Based Realm
- JDBC Database Realm
- Data source Database Realm
- JNDI Directory Realm
- User Database Realm
- JAAS Realm

Tomcat Standard Authenticators

- Valves that deal with container authentication
  - FORM Authenticator (FORM Auth)
  - BASIC Authenticator (BASIC Auth)
  - SSL Authenticator (CLIENT-CERT Auth)
  - DIGEST Authenticator (DIGEST Auth)

Tomcat Request Processing

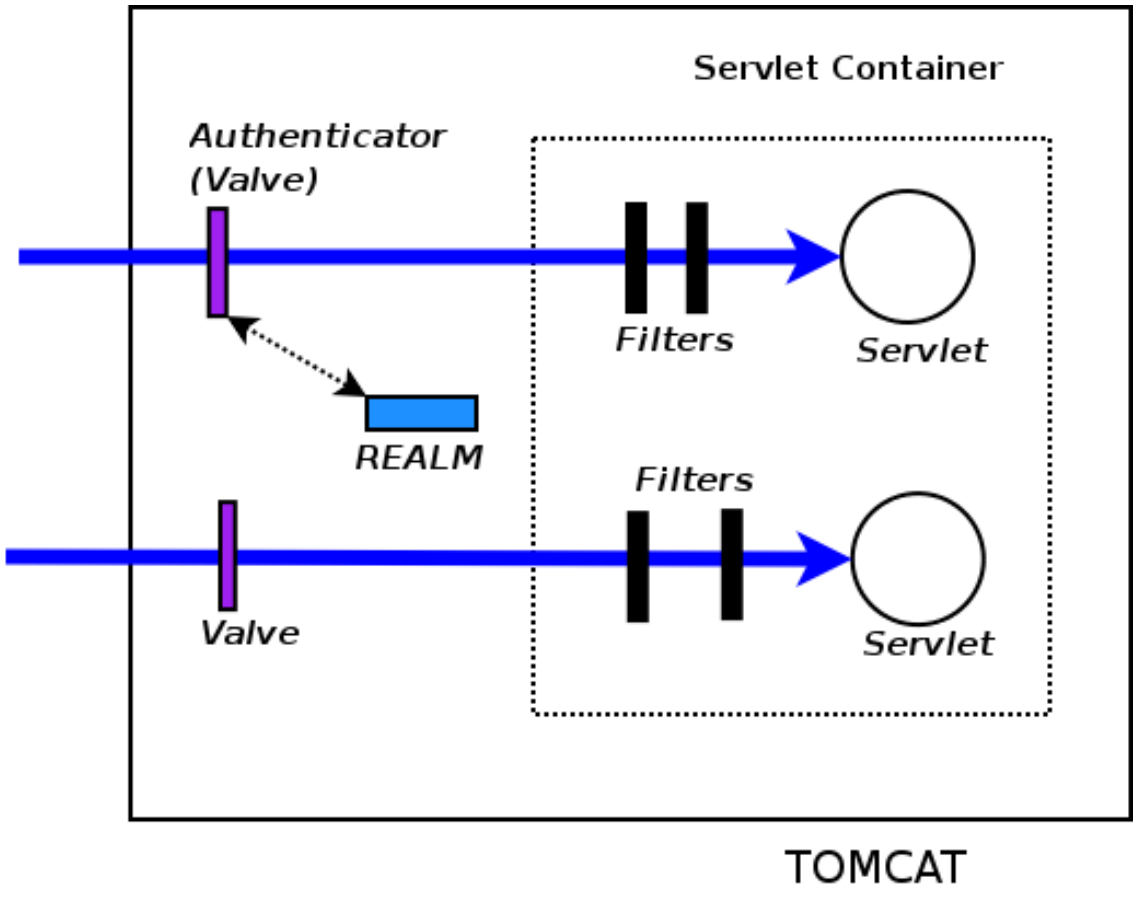
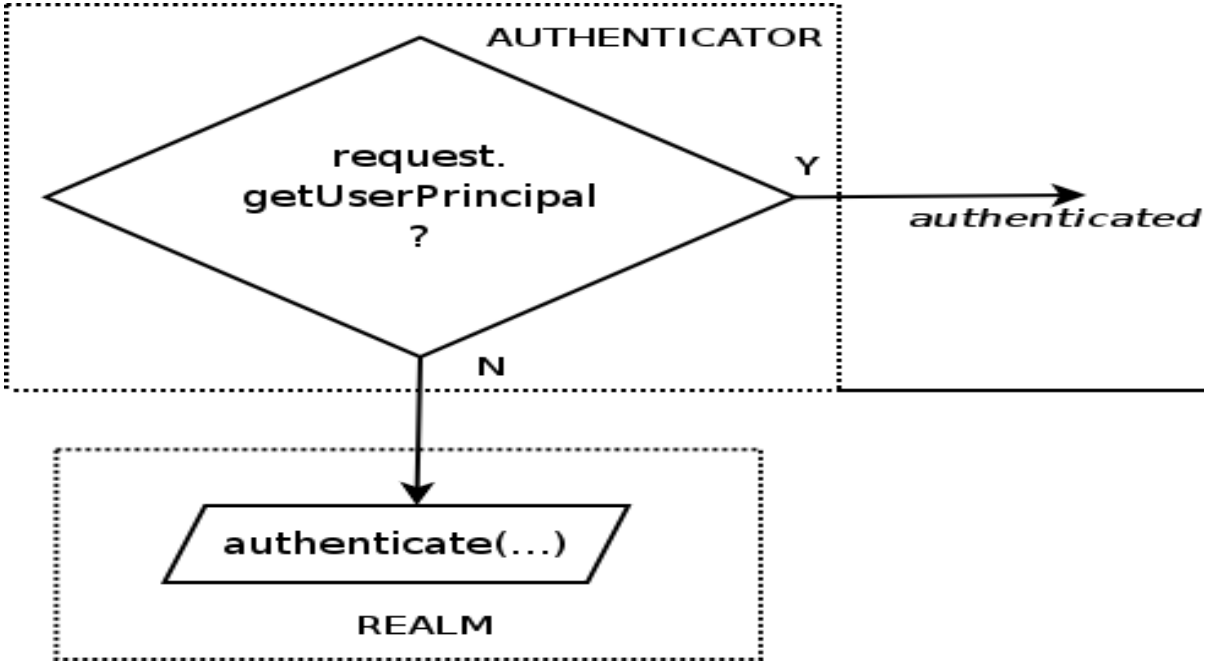


Image created using Open Source 'Dia'

Tomcat Request Processing



- Filter information from the request and pass to the **realm** for authentication/authorization

Writing Custom Authenticators

- ***org.apache.catalina.Authenticator*** interface
  - Marker Interface
- Extend the abstract class ***org.apache.catalina.authenticator.AuthenticatorBase***
  - Extends ValveBase
  - authenticate(Request,Response,LoginConfig)

Writing Custom Authenticators

```
//Will expand the basic FORM authentication to include auth based on request headers
public class CustomAuthenticator extends FormAuthenticator
{
  public boolean authenticate(Request request, Response response, LoginConfig config)
  throws IOException
  {
     if(request.getUserPrincipal() == null)
     {
        Realm realm = context.getRealm();
        //Pick the user name and password from the request headers
        if(username == null || pass ==null)  return super.authenticate(….);
        boolean authenticated = realm.authenticate(username,pass);
        if(authenticated == false)  return false;
        //Set the username/password on the session  and set the principal in request
        session.setNote(Constants.SESS_USERNAME_NOTE, username);
        session.setNote(Constants.SESS_PASSWORD_NOTE, password);
        request.setUserPrincipal(principal);
        register(request, response, principal, Constants.FORM_METHOD, username, pass);
     }
     return true;
  }
}
```

Writing Custom Realms

- ***org.apache.catalina.Realm*** interface
  - authenticate methods
  - hasResourcePermission
  - hasRole
  - hasUserDataPermission
- Extend the abstract class ***org.apache.catalina.realm.RealmBase***

Writing Custom Realms

```
import org.apache.catalina.realm.RealmBase;

public class ExtensibleRealm extends RealmBase
{
  public Principal authenticate(String username, String pass)
  {
     return new GenericPrincipal(this, username, pass, roles); //roles is a List
  }

  public boolean hasResourcePermission(Request request, Response response,
      SecurityConstraint[] sc, Context context) throws ....
  {
     if(request.getUserPrincipal() == null) return false;  //Not Authenticated
     // I am free to use any logic to do access control on the request
  }
}
```

Install Custom Valves/Realms

- Create a ***META-INF/context.xml*** in your WAR

```
<Context ....>
   <Valve ..../>
   <Realm .../>
</Context>
```

- Example:

```
<Context>
  <Realm className="org.test.ExtensibleRealm" debug="99"/>
</Context>
```

- Note: Context, Realm,Valve start in Caps.

Examples of Use Cases

- Perimeter Authentication
  - Authentication is performed by external system
    - generates a token (SAML for example)
    - redirects to tomcat for authorization
  - Tomcat uses custom authenticator to verify token
    - No token, fall back to regular authentication
  - Authenticator picks username/credential to pass to the realm for authorization
    - Realm provides a Tomcat GenericPrincipal instance
    - Principal is placed in the request

Examples of Use Cases

- Fine Grained Authorization
  - Container Authentication is fine
  - Need extra checks made for resources
    - This portion of the web application is accessible if your role is Manager and the time is between 9am-5pm
    - Employee should view only his payroll information.
  - Custom Realm can make use of the hasRole & hasResourcePermission to do the checks.
  - Make use of OASIS XACML?

Demo – Q&A