# The Anatomy of a Secure Java Web App Using Apache Fortress

September 24, 2018

ApacheCon NA, Montréal

# Objective

Think about how we should be securing web apps.

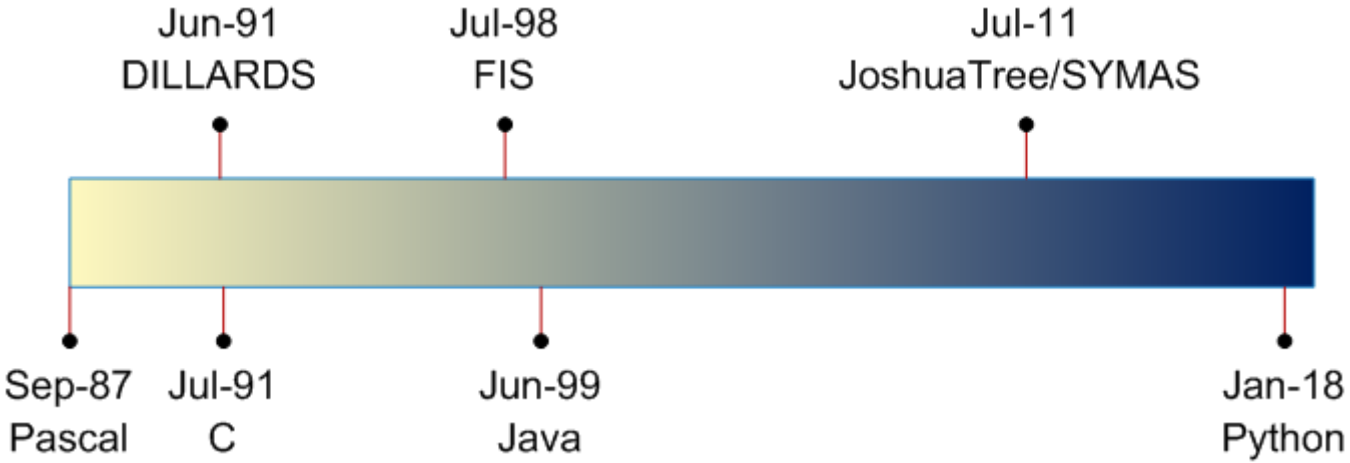
(If we pulled out all stops)

# Intro

**Shawn McKinney**

https://github.com/shawnmckinney

Code Monkey




- symas Software Architect
- PMC Apache Directory Project
- OpenLDAP™ PROJECT Engineering Team

# Agenda

1. Have a quick look at OWASP Vulnerability Scanning and Java Remote Code Execution Vulnerability
2. End-to-End Security w/ Apache Fortress Samples
3. Talk about RBAC, ABAC and how they can work together.

# Recommendation

Listen and absorb *conceptually*. Slides are published and have the *details*.

https://iamfortress.files.wordpress.com/2018/09/anatomy-secure-web-app-acna-2018-v5.pdf

# What's The Problem

- Equifax Breach
  - 143 million Americans' personal info, including names, addresses, dates of birth and SSNs compromised.
  - Only a veneer of security in place.

# https://cwiki.apache.org/confluence/display/WW/S2-045

## Summary

Possible Remote Code Execution when performing file upload based on Jakarta Multipart parser.

| | |
|---|---|
| **Who should read this** | All Struts 2 developers and users |
| **Impact of vulnerability** | Possible RCE when performing file upload based on Jakarta Multipart parser |
| **Maximum security rating** | Critical |
| **Recommendation** | Upgrade to Struts 2.3.32 or Struts 2.5.10.1 |
| **Affected Software** | Struts 2.3.5 - Struts 2.3.31, Struts 2.5 - Struts 2.5.10 |
| **Reporter** | Nike Zheng <nike dot zheng at dbappsecurity dot com dot cn> |
| **CVE Identifier** | CVE-2017-5638 |

# The Exploit

*"The Jakarta Multipart parser in Apache Struts 2 2.3.x before 2.3.32 and 2.5.x before 2.5.10.1 mishandles file upload, which allows remote attackers to execute arbitrary commands via a #cmd= string in a crafted Content-Type HTTP header, as exploited in the wild in March 2017."*

https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-5638

# The Solution

Ensure all appropriate patches have been applied.

# How do we ensure that our software is free of vulnerabilities?

# The Solution (Take 1)

Perform software vulnerability scans.

https://www.owasp.org/index.php/OWASP_Dependency_Check

# OWASP Vulnerability Scanning

Add to your Maven pom.xml file:

```
<plugin>
        <groupId>org.owasp</groupId>
        <artifactId>dependency-check-maven</artifactId>
        <version>3.3.1</version>
        <configuration>
        <failBuildOnAnyVulnerability>true</failBuildOnAnyVulnerability>

    <suppressionFile>${project.basedir}…/suppression.xml</suppressionFile>
        </configuration>
</plugin>
```

# False Positives

```
[INFO] BUILD FAILURE
[INFO] ------------------------------------------------------------------------
[ERROR] Failed to execute goal org.owasp:dependency-check-maven:3.3.1:check
        (default) on project fortress-core:
[ERROR]
[ERROR] One or more dependencies were identified with vulnerabilities:
[ERROR]
[ERROR] accelerator-api-1.0-RC41.jar: CVE-2006-5779, CVE-2002-1508, CVE-2009-3767,
        CVE-2013-4449, CVE-2011-4079, CVE-2017-14159, CVE-2002-1378, CVE-2002-
        0045, CVE-2002-1379, CVE-2006-6493, CVE-2007-6698, CVE-2012-1164, CVE-2017-
        9287, CVE-2005-4442, CVE-2015-3276, CVE-2017-17740, CVE-2005-2069, CVE-
        2012-2668, CVE-2015-6908
[ERROR]
[ERROR] See the dependency-check report for more details.
[ERROR] -> [Help 1]
[ERROR] For more information about the errors and possible solutions, please read the following articles:
[ERROR] [Help 1] http://cwiki.apache.org/confluence/display/MAVEN/MojoFailureException
smckinn@ubuntu:~/GIT/fortressDev/directory-fortress-core$ mvn install -Powasp
```

# Suppress False Positives

```
<suppressions xmlns="https://jeremylong.github.io/DependencyCheck/dependency-
    suppression.1.1.xsd">
  <!-- Suppress OWASP warnings about openldap serverside vulnerabilities. -->
  <suppress>
    <notes><![CDATA[
   file name: accelerator-api-1.0-RC41.jar
   ]]></notes>
    <gav regex="true">^org\.openldap:accelerator-api:.*$</gav>
    <cpe>cpe:/a:openldap:openldap</cpe>
  </suppress>
  …
</suppressions>
```

# How do we ensure that our software is free of vulnerabilities yet to be detected?

*It practically can't be done.*

# So Now What?

*"Security best practices dictate that this user have as little privilege as possible on the server itself, since security vulnerabilities in web applications and web servers are so commonly exploited."*

https://www.wired.com/story/equifax-breach-no-excuse/

# The Solution (Take 2)

Practice the principle of least privilege.

# Principle of least privilege

From Wikipedia, the free encyclopedia

https://en.wikipedia.org/wiki/Principle_of_least_privilege

*Not to be confused with Rule of least power.*

In information security, computer science, and other fields, the **principle of least privilege** (also known as the **principle of minimal privilege** or the **principle of least authority**) requires that in a particular abstraction layer of a computing environment, every module (such as a process, a user, or a program, depending on the subject) must be able to access only the information and resources that are necessary for its legitimate purpose.[1][2]

# Java Object Serialization Exploit

```
public class BadCode
  implements java.io.Serializable…
{…
 private void
  readObject(java.io.ObjectInputStream in)
 {
   in.defaultReadObject();
   Runtime.getRuntime().exec( cmd );
```

Java's remote code execution exploit occurs when a rogue object is read from an input resource and deserialized.

# Employ a Runtime Java Security Policy

```
grant codeBase "file:${catalina.home}/webapps/my-web-app-1/-" {
    permission java.net.SocketPermission "localhost", "resolve";
    permission java.net.SocketPermission "127.0.0.1:32768", "connect,resolve";
    permission java.lang.reflect.ReflectPermission "suppressAccessChecks";
    permission java.io.SerializablePermission "enableSubclassImplementation";
    permission java.io.FilePermission "…/resources/", "execute";
    …
    };
```

^ use w/ caution

# Example # 1

```
Begin serial exploit test....
Input: duke moscone center
Serialized data is saved in myObject.ser
BadCode will now run hacker script
user.home=/home/myuser
execute hacker command…
Exception in thread "main"
  java.security.AccessControlException:
  access denied ("java.io.FilePermission"
  "…/hacker-script.sh" "execute")
```

https://github.com/shawnmckinney/serial-exploit-sample

symas

# Not a Perfect Solution

```
grant codeBase "file:${catalina.home}/webapps/my-web-app-1/-" {
   permission java.net.SocketPermission "localhost", "resolve";
   permission java.io.FilePermission "…/resources/good-scripts*", "execute";
   permission java.net.SocketPermission "127.0.0.1:32768", "connect,resolve";
   permission java.lang.reflect.ReflectPermission "suppressAccessChecks";
   permission java.io.SerializablePermission "enableSubclassImplementation";
   permission java.lang.reflect.ReflectPermission "suppressAccessChecks";
   };
```

| Permission Target Name | What the Permission Allows | Risks of Allowing this Permission |
|---|---|---|
| suppressAccessChecks | ability to access fields and invoke methods in a class. Note that this includes not only public, but protected and private fields and methods as well. | This is dangerous in that information (possibly confidential) and methods normally unavailable would be accessible to malicious code. |

# One day maybe…

Beyond Java 8

- Modularization
- Improved encapsulation
- Finer control over package access.

# Meanwhile

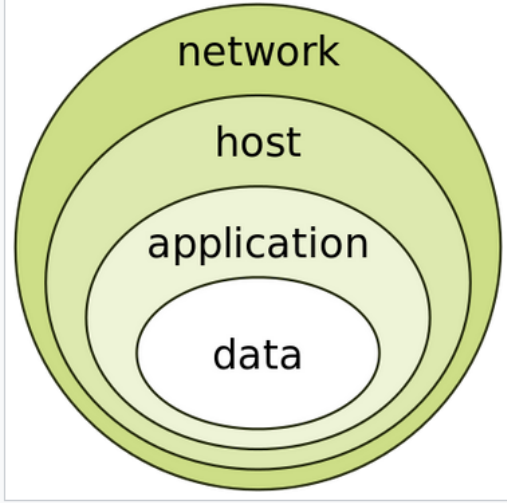
What should we do?

## Defense in depth [ edit ]

https://en.wikipedia.org/wiki/Information_security

*Main article: Defense in depth (computing)*

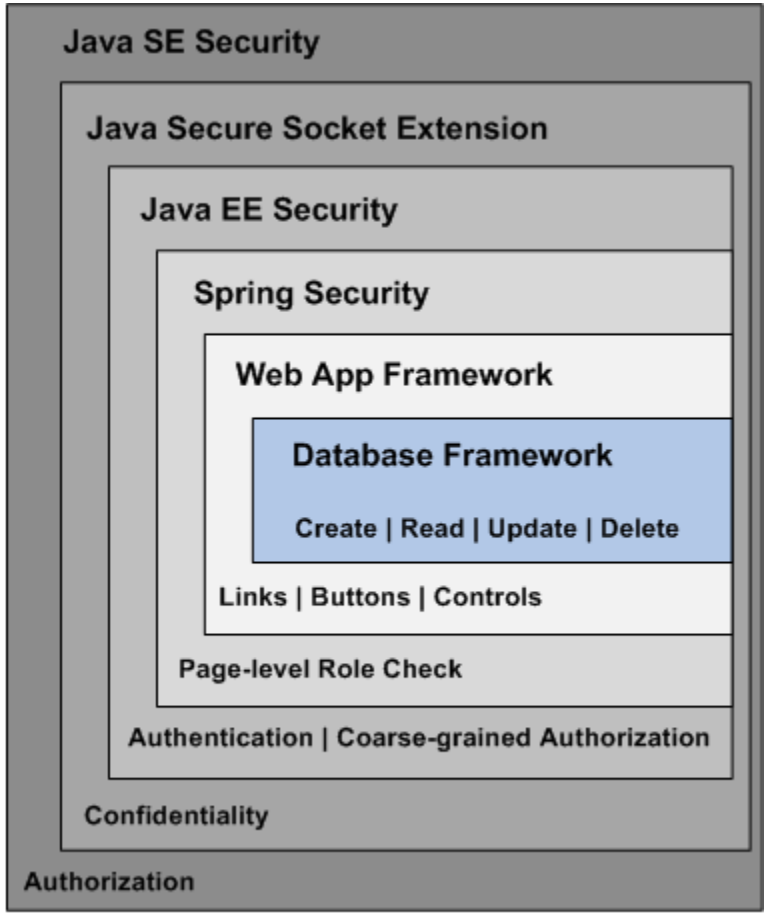
Information security must protect information throughout the life span of the information, from the initial creation of the information on through to the final disposal of the information. The information must be protected while in motion and while at rest. During its lifetime, information may pass through many different information processing systems and through many different parts of information processing systems. There are many different ways the information and information systems can be threatened. To fully protect the information during its lifetime, each component of the information processing system must have its own protection mechanisms. The building up, layering on and overlapping of security measures is called **defense in depth**. In contrast to a metal chain, which is famously only as strong as its weakest link, the defense-in-depth aims at a structure where, should one defensive measure fail, other measures will continue to provide protection.




The onion model of defense in depth

# Java Web Security Layers

1. Java SE Security
2. Java Secure Socket Extension (JSSE)
3. Java EE Security
4. Spring Security
5. Web App Framework
6. Database Framework



# Each with a specific purpose

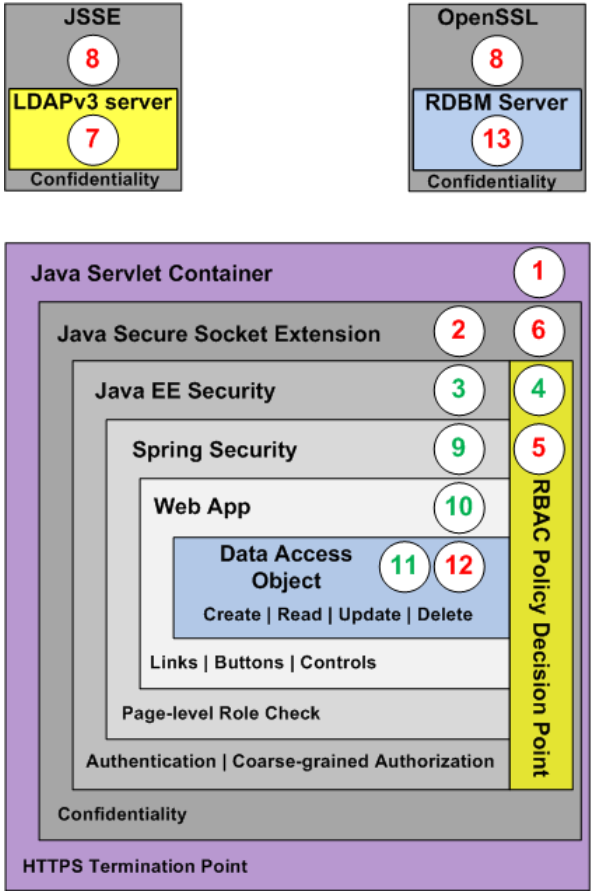
1. Java SE Security ---------- principle of least privilege
2. JSSE ---------- private conversations
3. Java EE Security ---------- deadbolt on front door
4. Spring Security ---------- locks on room doors
5. Web App Framework - locks on equipment in rooms
6. Database Functions ---- content filtering

# Example #2

Apache

Fortress

Demo




HTTP JDBC LDAPv3

Confidentiality Authorization

1. HTTPS server
2. HTTPS private key
3. Java EE AuthN & AuthZ
4. RBAC Policy Decision Point
5. LDAP SSL client
6. SSL public key
7. LDAP SSL server
8. SSL private key
9. Spring AuthZ
10. Web App AuthZ
11. DAO AuthZ
12. JDBC SSL client
13. Database SSL server

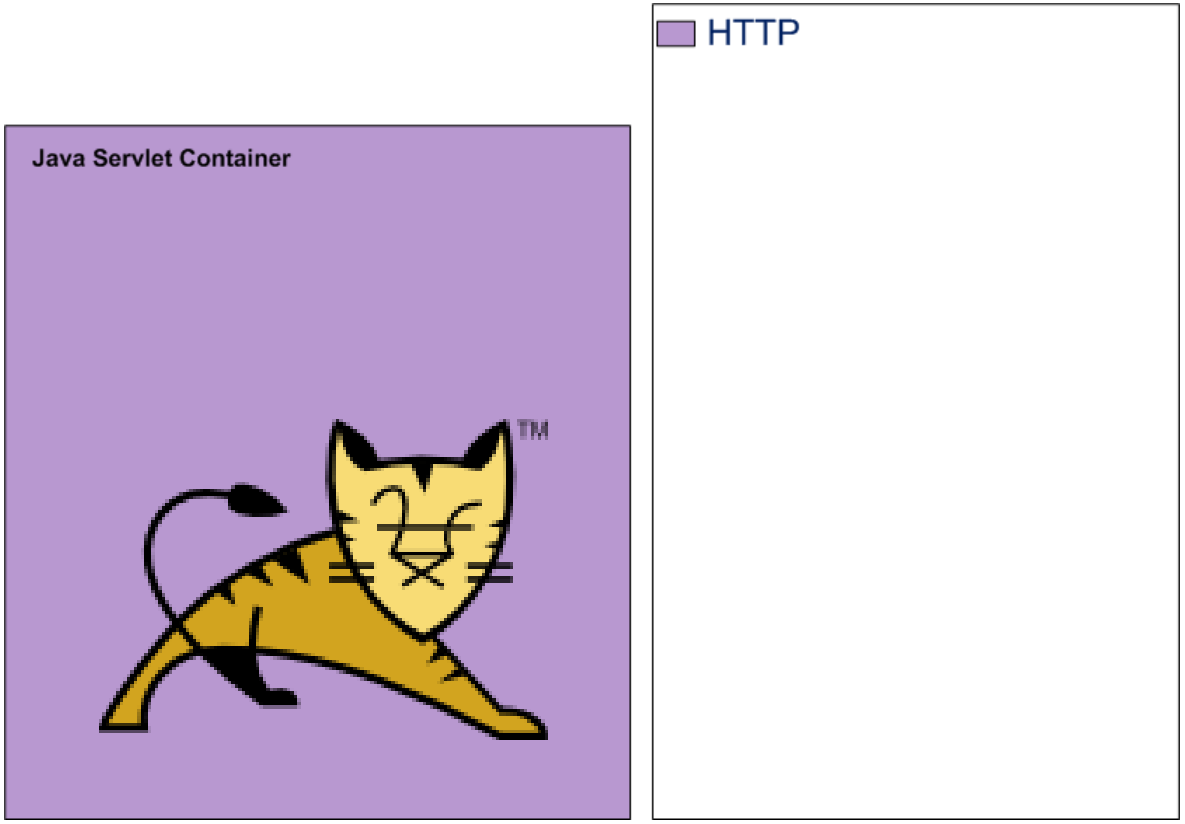
https://github.com/shawnmckinney/apache-fortress-demo

# Two Areas of Control

1. JavaSE, JSSE, JavaEE and Spring Declarative controls

2. Programmatic AuthZ controls in the Web and DB layers

# Start with Tomcat Servlet Container

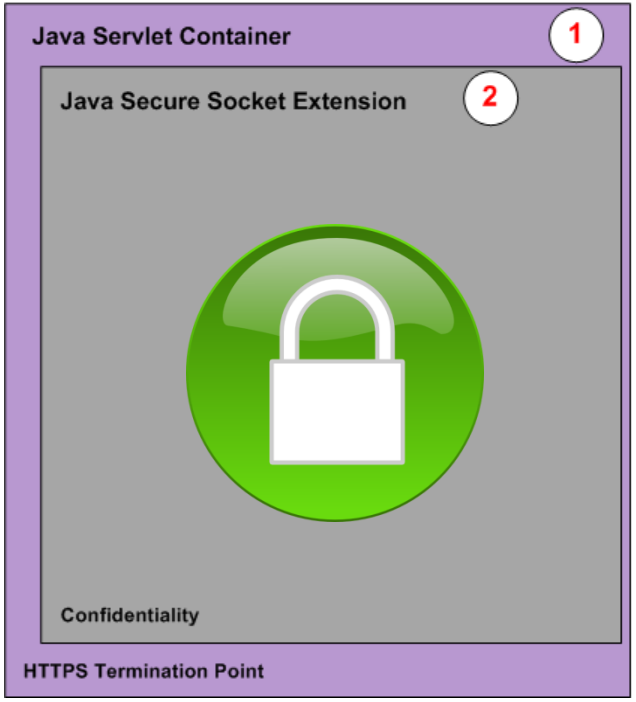


# 1 & 2. Enable HTTPS

ssssh!!!

1. Update the Server.xml

2. Add private key




HTTP

Confidentiality

1. HTTPS server
2. HTTPS private key

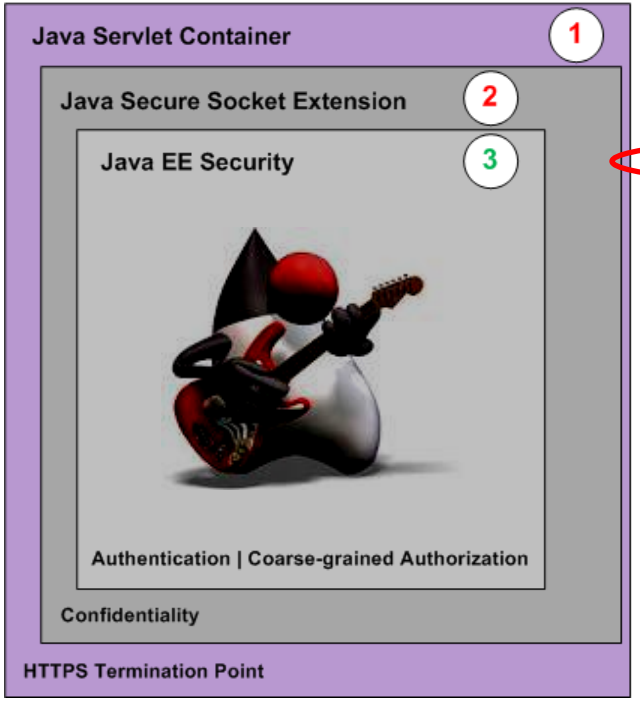
# 3. Enable Java EE Security the deadbolt

a. Update web.xml

b. Drop the proxy jar

c. Add context.xml

d. Add fortress to pom.xml




HTTP

Confidentiality Authorization

1. HTTPS server
2. HTTPS private key
3. Java EE AuthN & AuthZ

# Current Specs for Java EE Security

1. JSR-196 – JASPIC - AuthN
2. JSR-115 – JAAC - AuthZ
3. JSR-375 – JavaEE Security API

# What is a Realm?

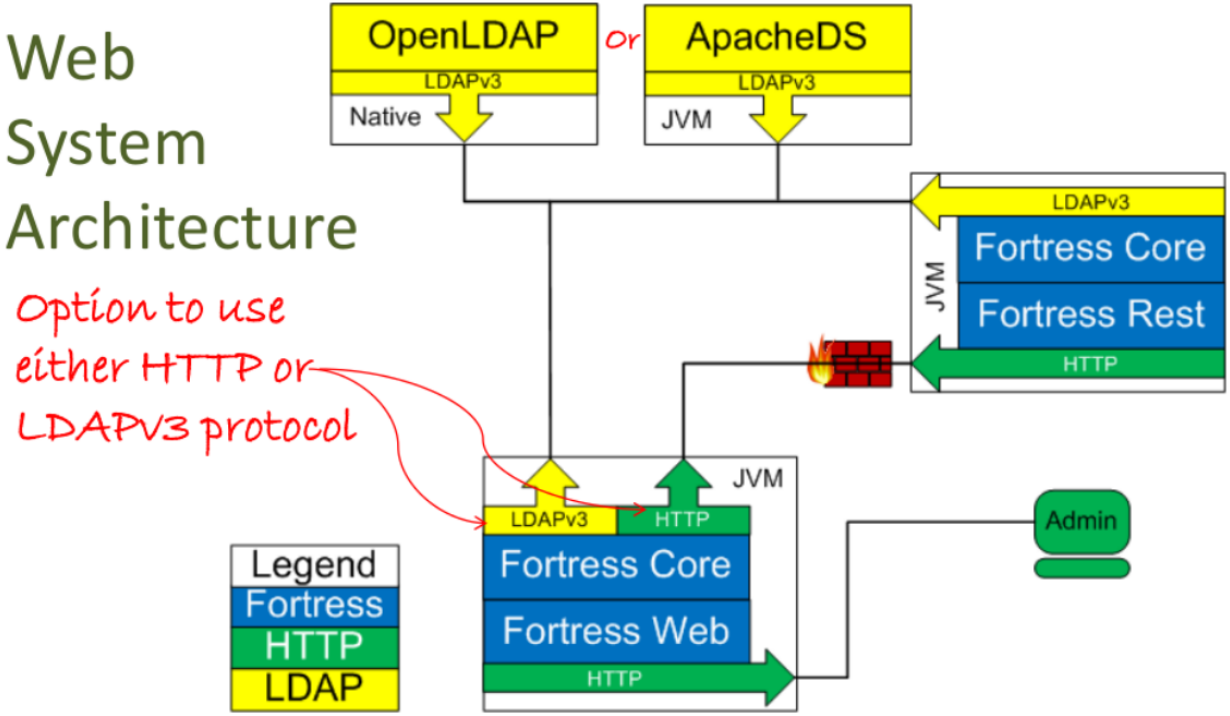
*A Realm is a "database" of usernames and passwords that identify valid users of a web application (or set of web applications), plus an enumeration of the list of roles associated with each valid user.*

https://tomcat.apache.org/tomcat-9.0-doc/realm-howto.html

# Apache Fortress™

## Access Management SDK and Web Components

**A standards-based access management system, written in Java, supports ANSI INCITS 359 RBAC and more.**

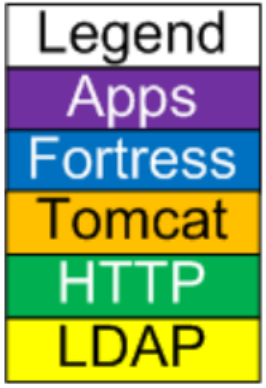


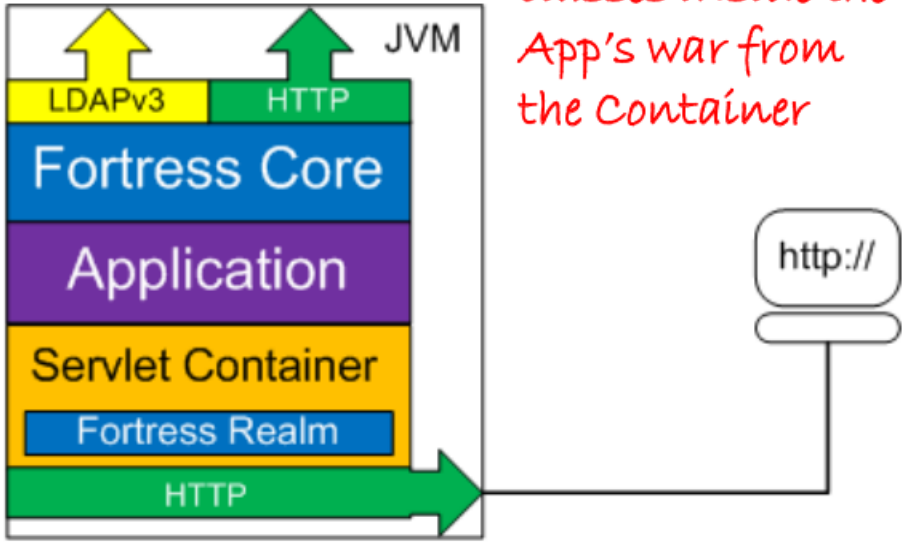
# Apache Fortress Context Realm

## Realm Context System Architecture

Shares the RBAC session (activated roles) created inside the Realm with the App.

Isolates Fortress classes inside the App's war from the Container



# Add Fortress Realm Dependency

Add Fortress Dependency to web app's pom.xml:

```
<dependency>
  <groupId>org.apache.directory.fortress</groupId>
  <artifactId>
```

fortress-realm-impl

```
[root@IL1SCOLSP102 lib]# pwd
/usr/local/tomcat7/webapps/apache-fortress-demo/WEB-INF/lib
[root@IL1SCOLSP102 lib]# ls -l fortress*
-rw-r--r-- 1 root root 502112 Aug 30 06:55 fortress-core-1.0-RC41-SNAPSHOT.jar
-rw-r--r-- 1 root root  22005 Aug 29 12:20 fortress-realm-impl-1.0-RC41-SNAPSHOT.jar
-rw-r--r-- 1 root root 789927 Aug 29 12:40 fortress-web-1.0-RC41-SNAPSHOT-classes.jar
[root@IL1SCOLSP102 lib]#
```

# Enable Fortress Context Realm

Add **context.xml** to META-INF folder:

```
<Context reloadable="true">
 <Realm className=
  "org.apache.directory.fortress.realm.tomcat.Tc7AccessMgrProxy"
  defaultRoles="ROLE_PAGE1,ROLE_PAGE2,
  ROLE_PAGE3,…"
  contextId="tenant314"
/>
</Context>
```

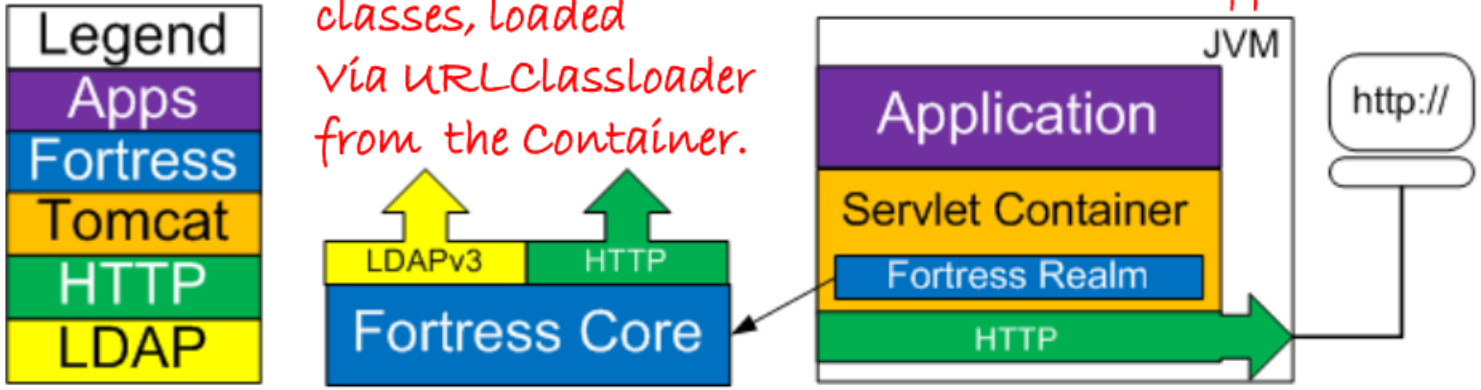
https://github.com/shawnmckinney/apache-fortress-demo/blob/master/src/main/resources/META-INF/context.xml

# Or use Host Realm, doesn't expose the App to Fortress

## Realm Host System Architecture

Shares the RBAC session (activated roles) created inside the Realm with the App.

Isolates Fortress classes, loaded via URLClassloader from the Container.




But, then the App can't use the programmatic APIs.

# Enable Fortress Tomcat Realm

Drop the Fortress Realm Proxy Jar in Tomcat's lib folder:

```
[root@IL1SCOLSP102 lib]# pwd
/usr/local/tomcat7/lib
[root@IL1SCOLSP102 lib]# ls -l fortress*
-rw-r--r-- 1 root root 15119 Aug 29 12:37 fortress-realm-proxy-1.0-RC41-SNAPSHOT.jar
[root@IL1SCOLSP102 lib]#
```

# Enable Fortress Tomcat Realm

Add to App's Web.xml

```
<security-constraint>
  <display-name>My Project Security Constraint</display-name>
  <web-resource-collection>
    <web-resource-name>Protected Area</web-resource-name>
    <url-pattern>/wicket/*</url-pattern>
  </web-resource-collection>
  <auth-constraint>
    <role-name>DEMO2_USER</role-name>
  </auth-constraint>
</security-constraint>
<login-config>
  <auth-method>FORM</auth-method>
  <realm-name>MySecurityRealm</realm-name>
  <form-login-config>
<form-login-page>/login/login.html</form-login-page>
```

1. Java EE container protects this URL Automatically.

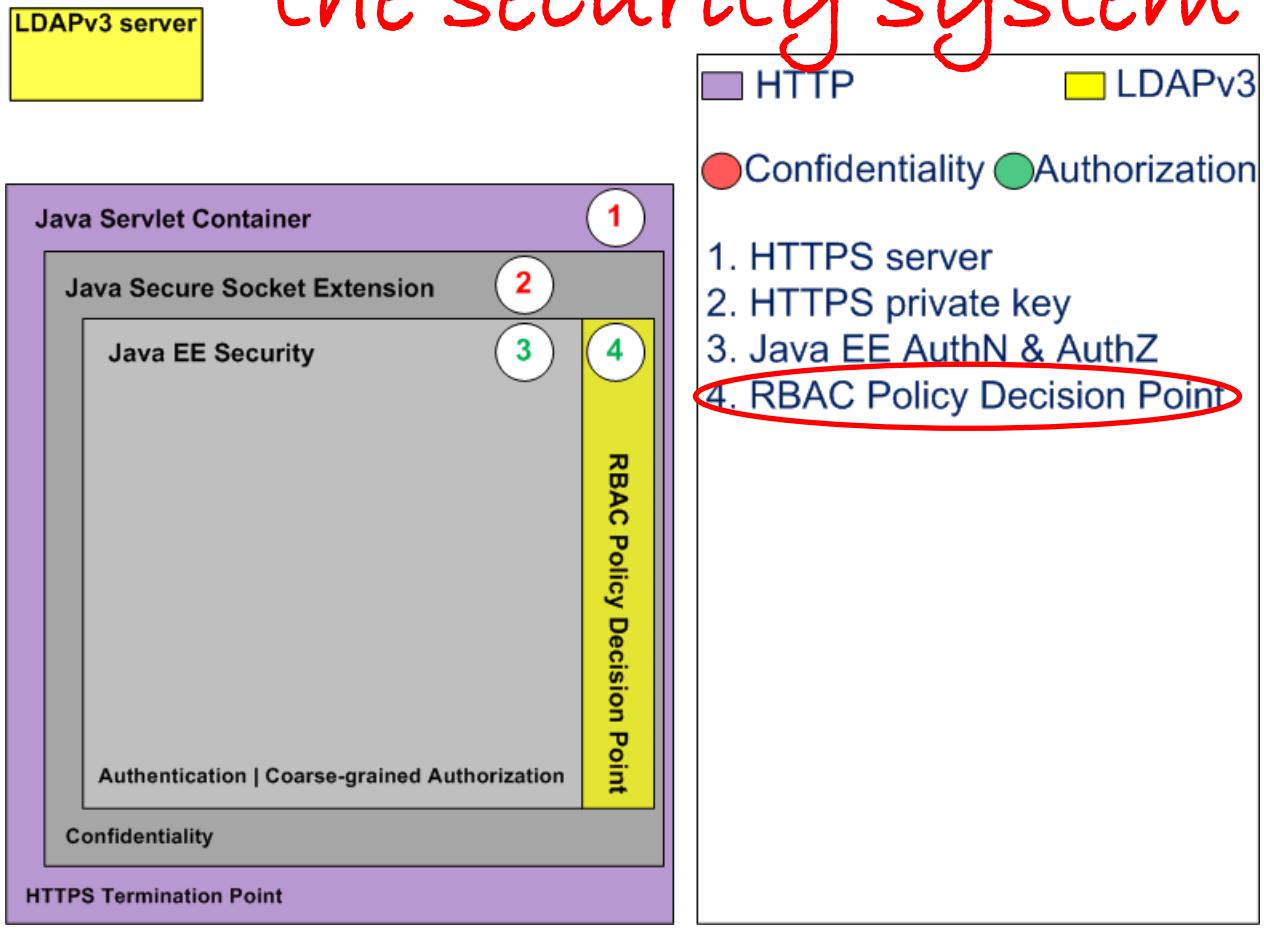
2. All users must have this role to gain entry.

3. Route un-authN requests to my form.

https://github.com/shawnmckinney/apache-fortress-demo/blob/master/src/main/webapp/WEB-INF/web.xml

# 4. Setup Policy Decision Point

## the security system



# Intro to the RBAC Standard

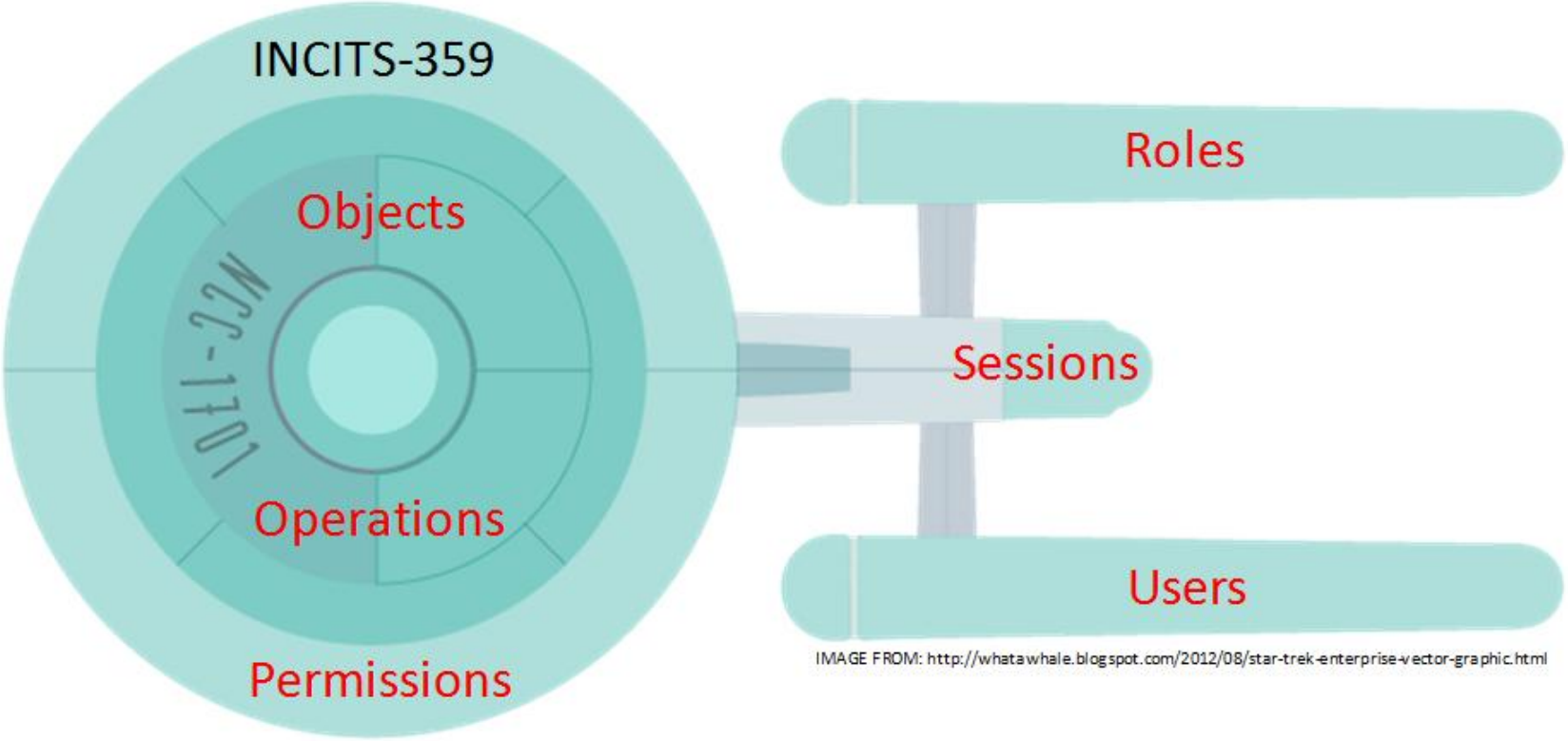



IMAGE FROM: http://whatawhale.blogspot.com/2012/08/star-trek-enterprise-vector-graphic.html

# Use ANSI RBAC INCITS 359 Specification

**RBAC0**:

- Users, Roles, Perms, Sessions

**RBAC1**:

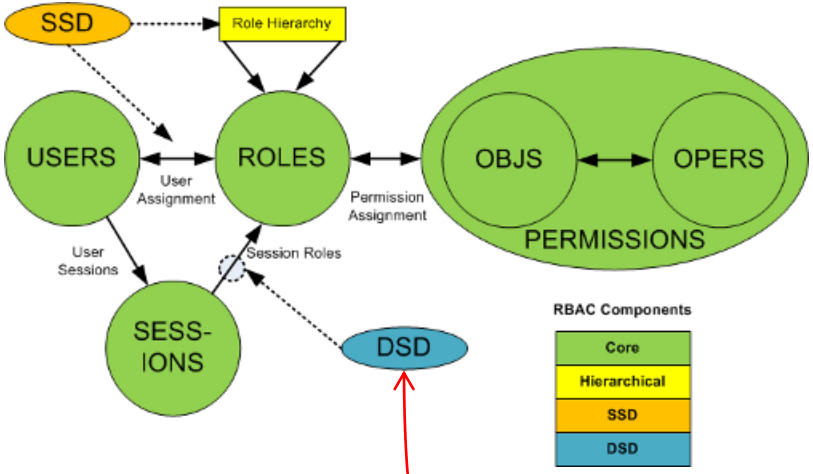
- Hierarchical Roles

**RBAC2**:

- Static Separation of Duties

**RBAC3**:

- Dynamic Separation of Duties




Today we demo this

# Early Years

- The Role-Based Access Control model was formally introduced in 1992 by David Ferraiolo and Richard Kuhn of National Institute of Standards and Technology.

- Their model, already in use for some time, was meant to address critical shortcomings of the Discretionary Access Control.  DAC was not meeting the needs of non-DoD organizations.

- In particular integrity was lacking, defined by them, as the requirement for data and process to be modified only in authorized ways by authorized users.

# Middle Years

- Eight years later, in 2000, they teamed with Ravi Sandhu and produced another influential paper entitled ‘The NIST Model for a Role-Based Access Control: Towards a Unified Standard’.

- Later the team released the RBAC formal model. One that laid out in discrete terms how these types of systems were to work. The specifications, written in Z-notation, left no ambiguity whatsoever.

- This model formed the basis for the standard that followed:
  - ANSI INCITS 359

# Current Years

- INCITS 359-2012 RBAC also known as Core.
- INCITS 494-2012 RBAC Policy Enhanced allows attribute modifiers on permissions specifically to provide support for fine-grained authorization.

# Use RBAC Object Model

Six basic elements:

1. **User** – human or machine entity
2. **Role** – a job function within an organization
3. **Object** – maps to system resources
4. **Operation** – executable image of program
5. **Permission** – approval to perform an Operation on one or more Objects
6. **Session** – contains set of activated roles for User

# Use RBAC Functional Model

APIs form three standard interfaces:

Management and Config processes

1. Admin – Add, Update, Delete
2. Review – Read, Search
3. System – Access Control

Demo runtime processes

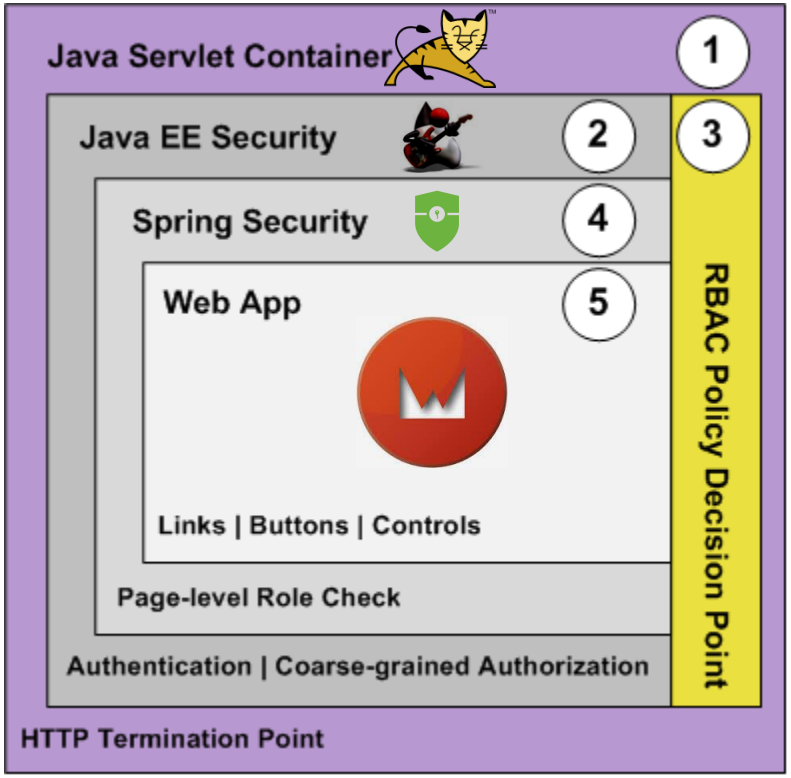
# Use RBAC Functional Model

System Manager APIs:

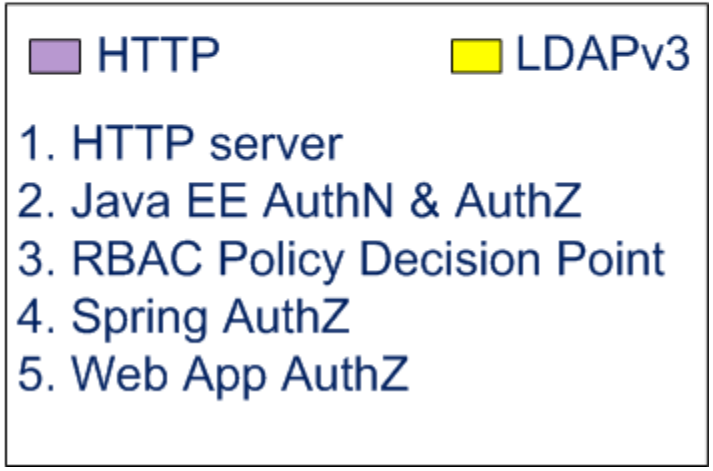
http://directory.apache.org/fortress/gen-docs/latest/apidocs/org/apache/directory/fortress/core/impl/AccessMgrImpl.html

1. createSession – authenticate, activate roles
2. checkAccess – permission check
3. sessionPermissions – all perms active for user
4. sessionRoles – return all roles active
5. addActiveRole – add new role to session
6. dropActiveRole – remove role from session

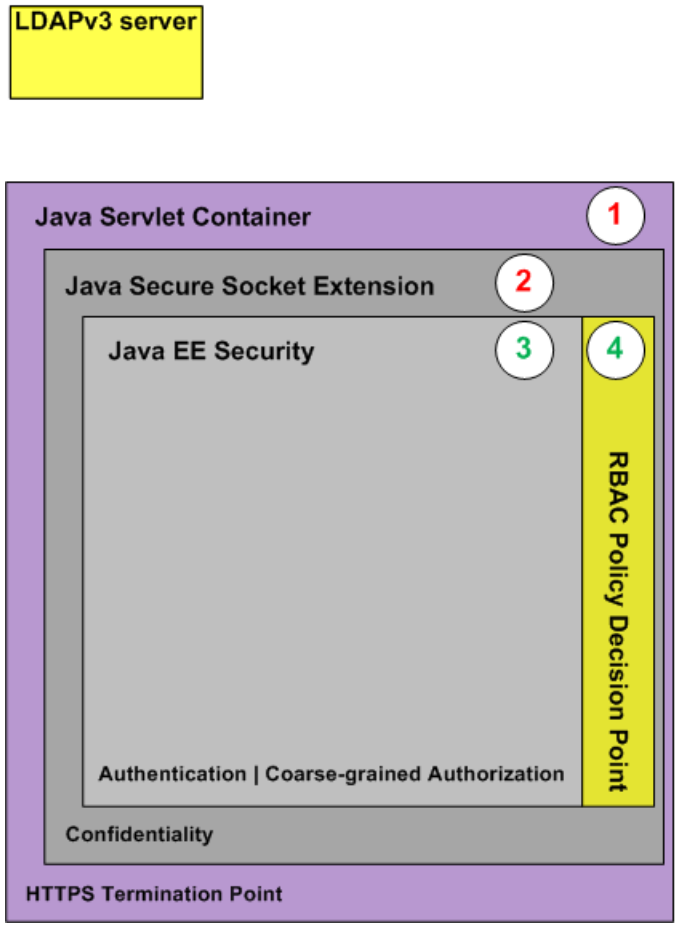
# Example #3 : Role Engineering Sample

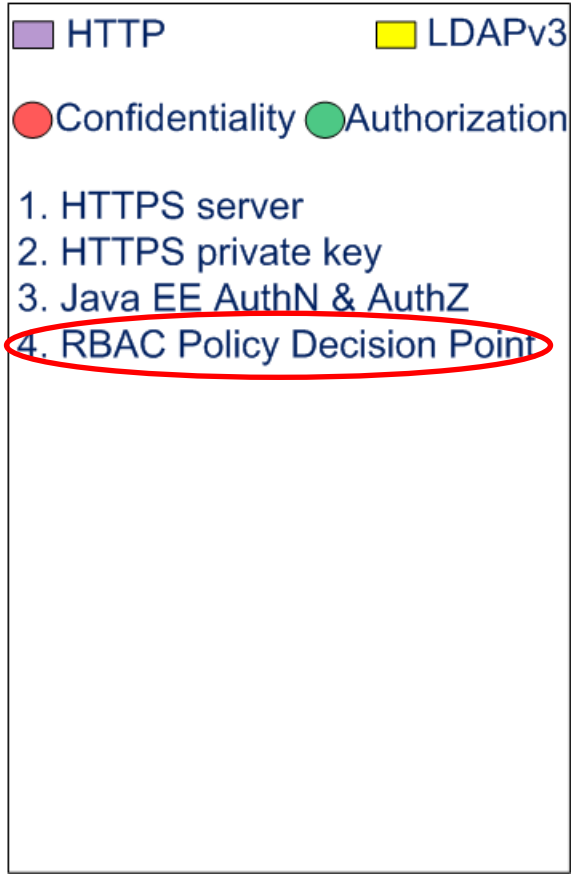



https://github.com/shawnmckinney/role-engineering-sample



# 4. Back to Installing a policy decision point

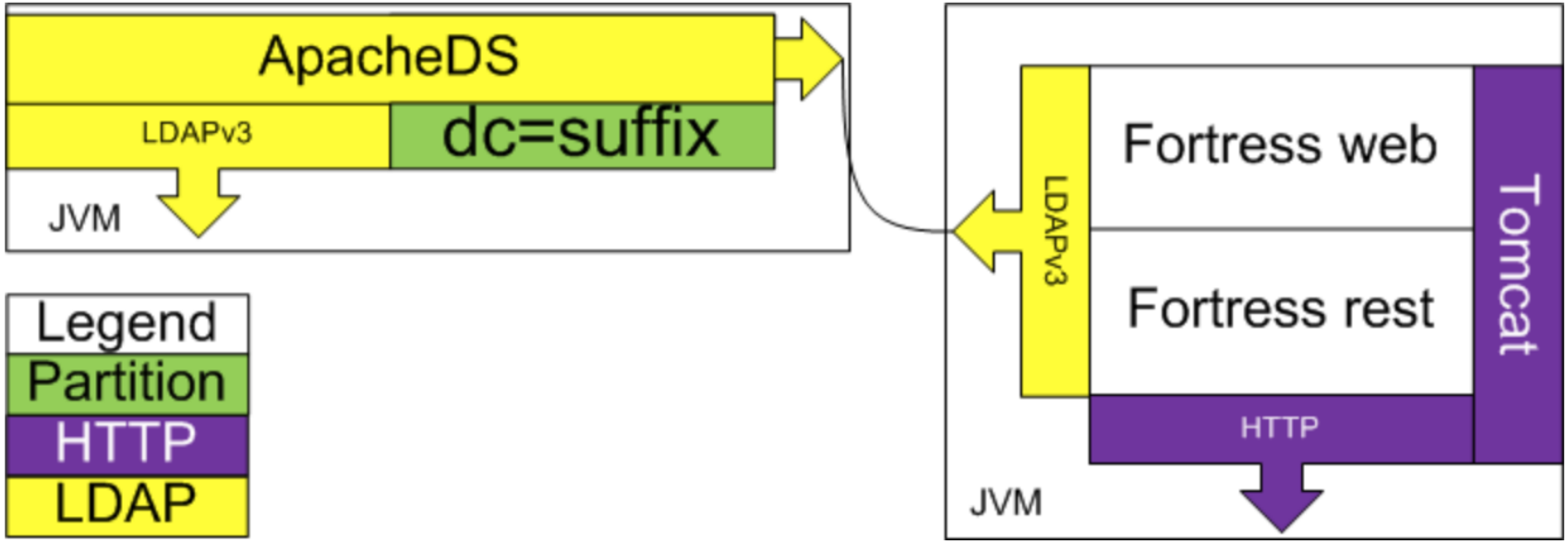


Use

# ApacheDS & Fortress QUICKSTART

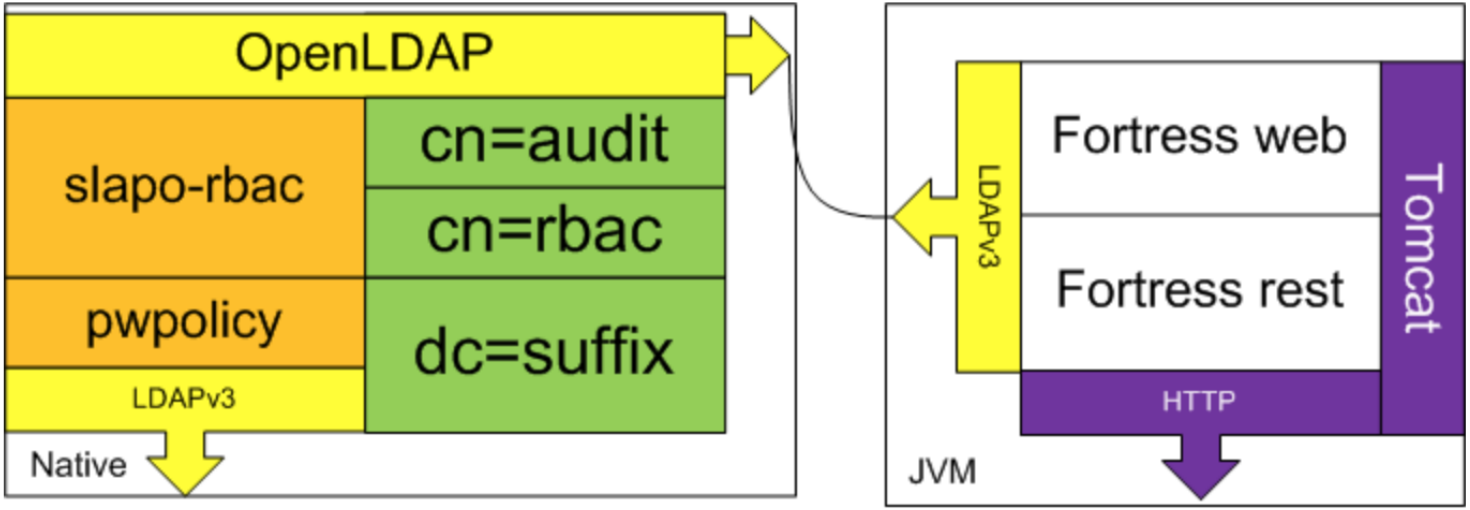
Apache Fortress 2.0.0-RC1-SNAPSHOT and ApacheDS Quickstart System Architecture



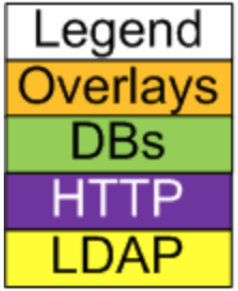

https://github.com/apache/directory-fortress-core/blob/master/README-QUICKSTART-APACHEDS.md

# Or OpenLDAP & Fortress QUICKSTART

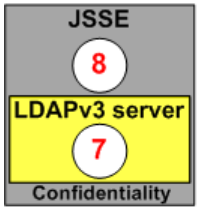
Apache Fortress 2.0.0-RC2 and OpenLDAP Quickstart System Architecture



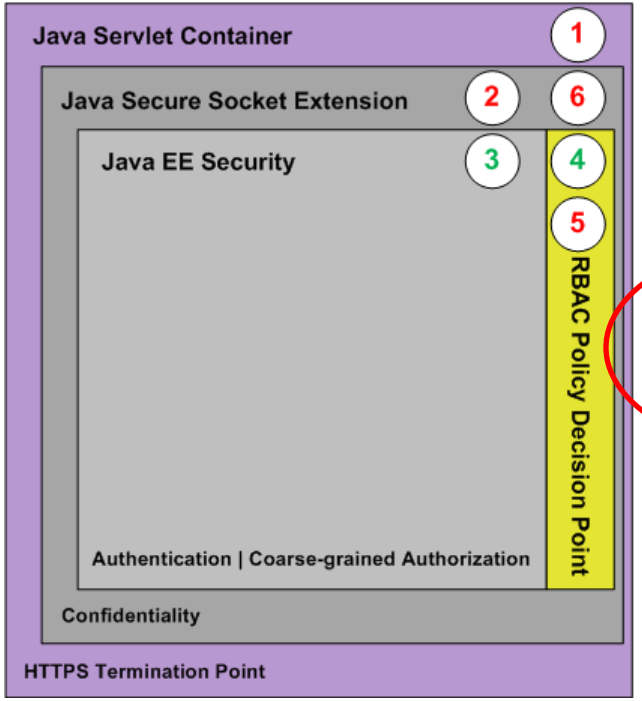

https://github.com/apache/directory-fortress-core/blob/master/README-QUICKSTART-SLAPD.md



# 5 – 8 Enable LDAP SSL




confidentiality




HTTP LDAPv3

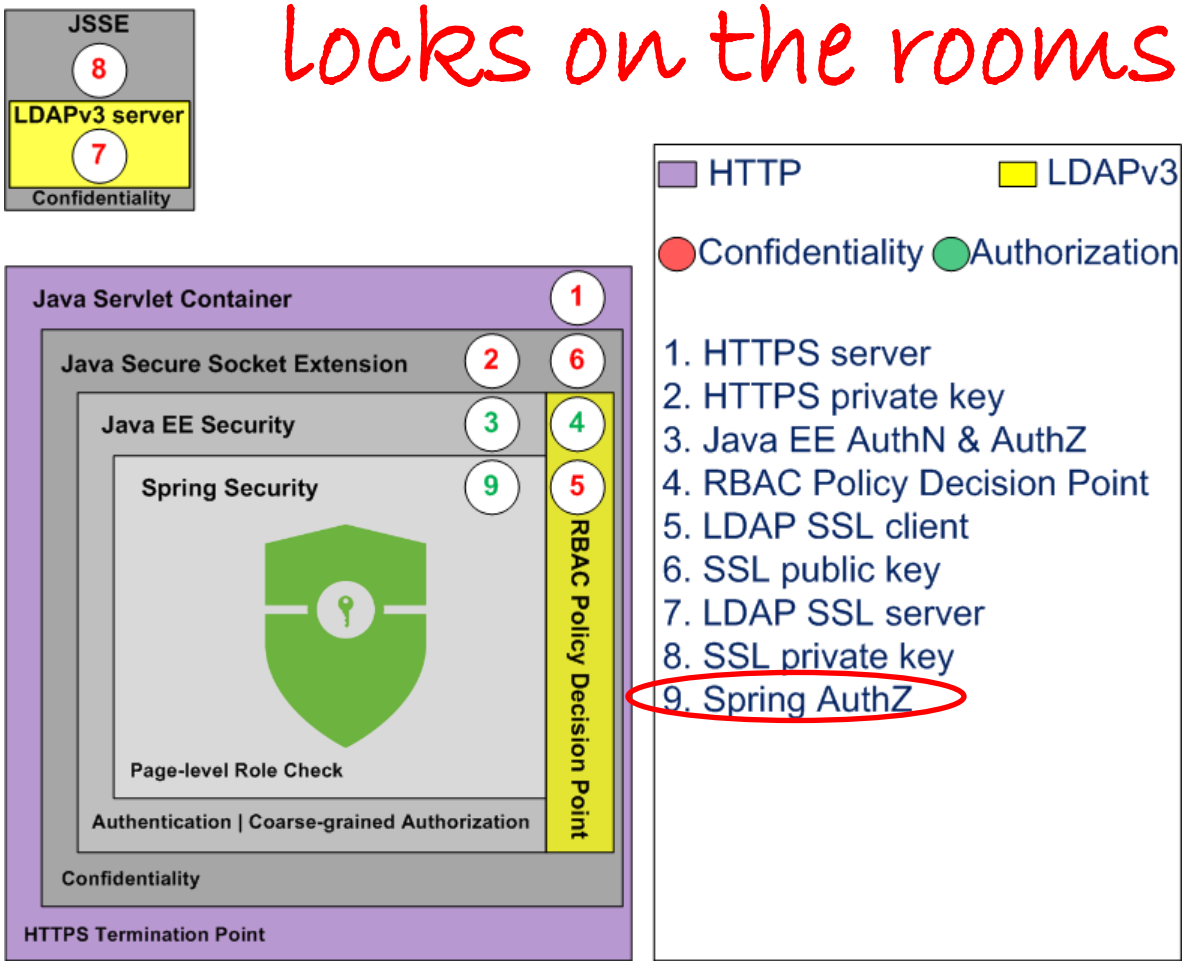
Confidentiality Authorization

1. HTTPS server
2. HTTPS private key
3. Java EE AuthN & AuthZ
4. RBAC Policy Decision Point
5. LDAP SSL client
6. SSL public key
7. LDAP SSL server
8. SSL private key

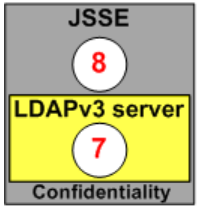
# 9. Enable Spring Security

a. Authorization

b. Role mapping



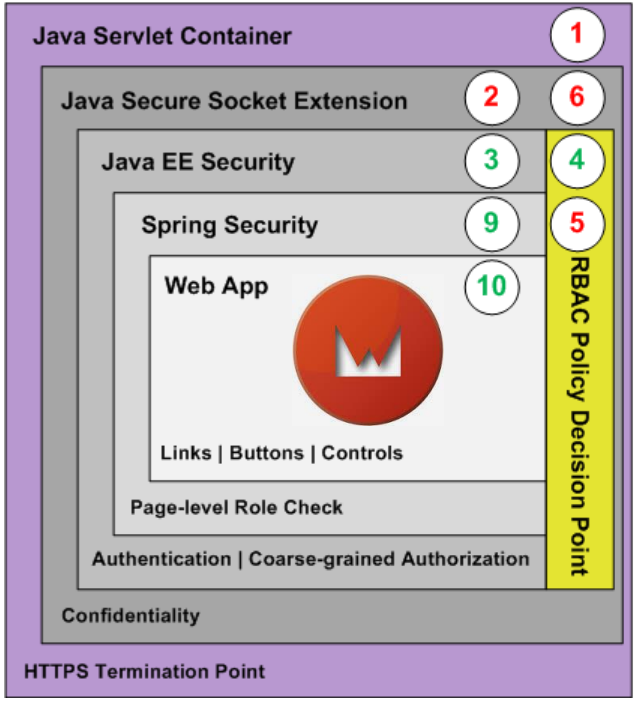
# 10. Web App Authorization




locks on equipment

Add fine-grained checks:

a. Page links
b. Buttons
c. Other controls



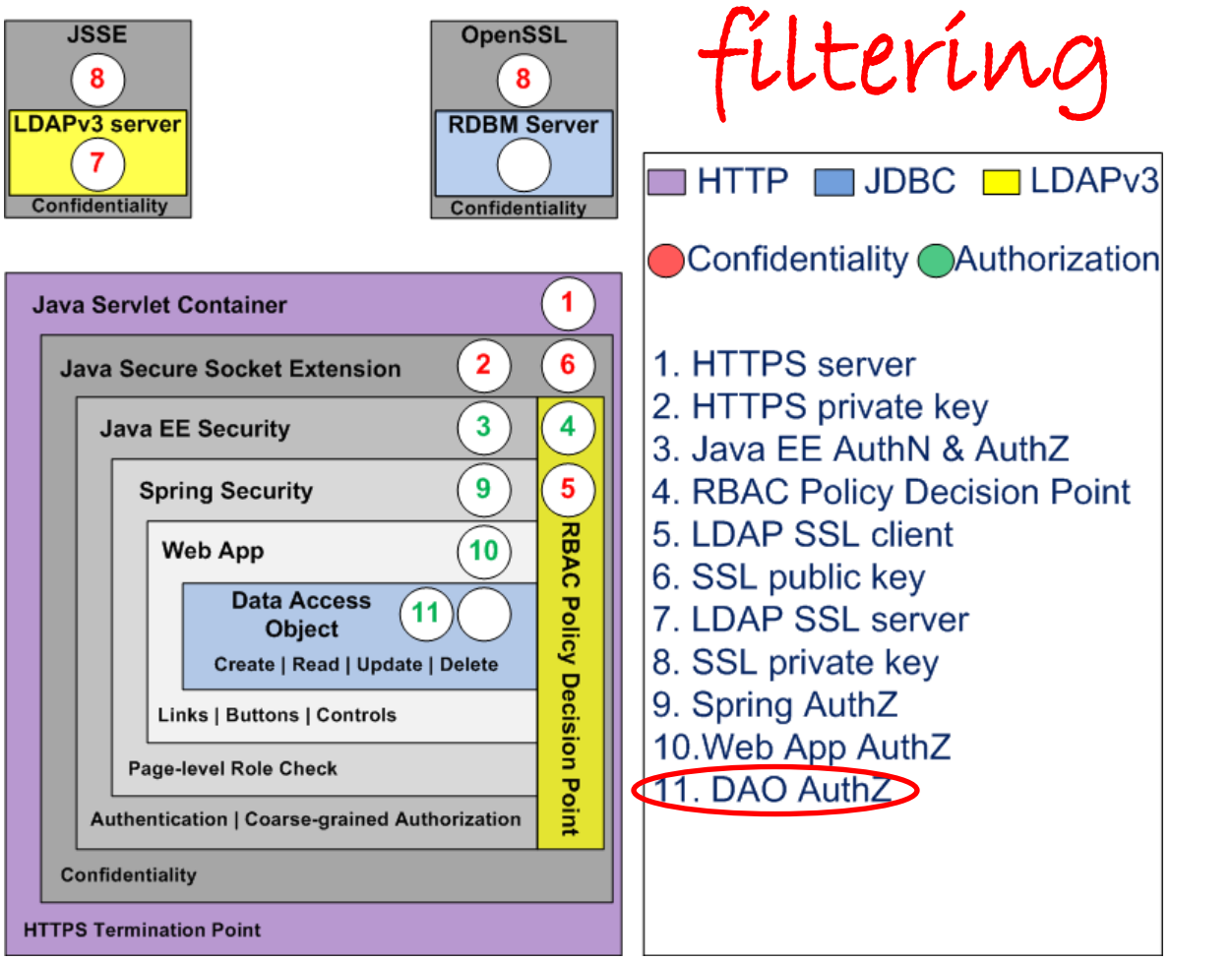

HTTP
LDAPv3
Confidentiality
Authorization

1. HTTPS server
2. HTTPS private key
3. Java EE AuthN & AuthZ
4. RBAC Policy Decision Point
5. LDAP SSL client
6. SSL public key
7. LDAP SSL server
8. SSL private key
9. Spring AuthZ
10. Web App AuthZ

# 11. DAO Authorization

Add fine-grained Checks to:

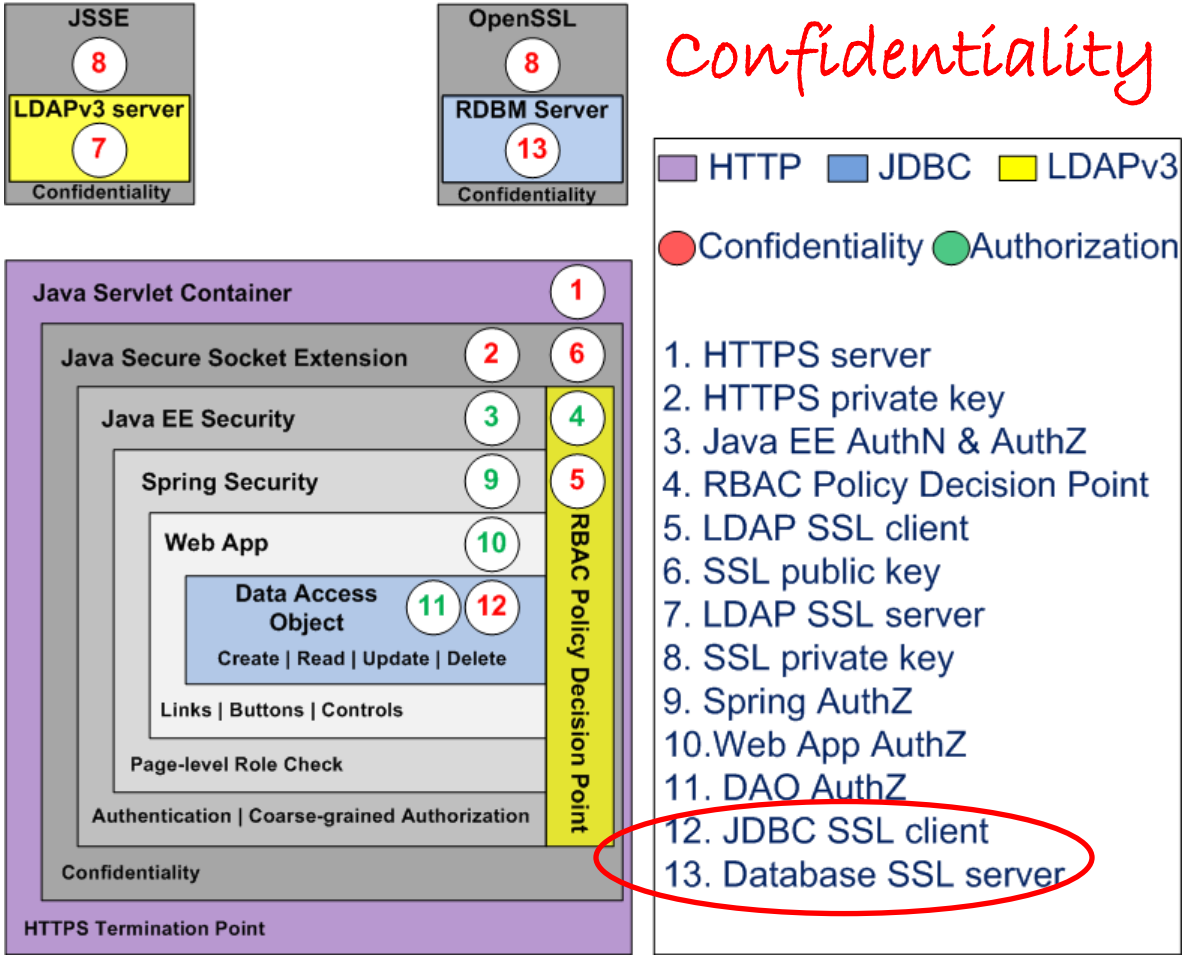
a. Create
b. Read
c. Update
d. Delete



# 12, 13. Enable DB SSL

12. Client
 a. public key
 b. config

13. Server
 a. private key
 b. config



# Apache Fortress Demo

- Three Pages and Three Customers
- One role for every page to customer combo
- Users may be assigned to one or more roles
- One and only one role may be activated

| Pages | Customer 123 | Customer 456 | Customer 789 |
|---|---|---|---|
| Page One | PAGE1_123 | PAGE1_456 | PAGE1_789 |
| Page Two | PAGE2_123 | PAGE2_456 | PAGE2_789 |
| Page Three | PAGE3_123 | PAGE3_456 | PAGE3_789 |

| User123 | Customer 123 | Customer 456 | Customer 789 |
|---|---|---|---|
| Page1 | True | False | False |
| Page2 | True | False | False |
| Page3 | True | False | False |

| User1 | Customer 123 | Customer 456 | Customer 789 |
|---|---|---|---|
| Page1 | True | True | True |
| Page2 | False | False | False |
| Page3 | False | False | False |

| User1_123 | Customer 123 | Customer 456 | Customer 789 |
|---|---|---|---|
| Page1 | True | False | False |
| Page2 | False | False | False |
| Page3 | False | False | False |

| User456 | Customer 123 | Customer 456 | Customer 789 |
|---|---|---|---|
| Page1 | False | True | False |
| Page2 | False | True | False |
| Page3 | False | True | False |

| User2 | Customer 123 | Customer 456 | Customer 789 |
|---|---|---|---|
| Page1 | False | False | False |
| Page2 | True | True | True |
| Page3 | False | False | False |

| User2_123 | Customer 123 | Customer 456 | Customer 789 |
|---|---|---|---|
| Page1 | False | True | False |
| Page2 | False | False | False |
| Page3 | False | False | False |

# RBAC Demo


http://www.wright-brothers.org/Information_Desk/Help_with_Homework/Wright_Photos/Wright_Photos_images/1902_Glider_Flying.jpg

# Testing

- Verify security functionality via automation.
- Beware of regressions. Can go unnoticed for weeks, months, years.

https://github.com/shawnmckinney/apache-fortress-demo/.../ApacheFortressDemoSeleniumITCase.java

# Apache Fortress Demo

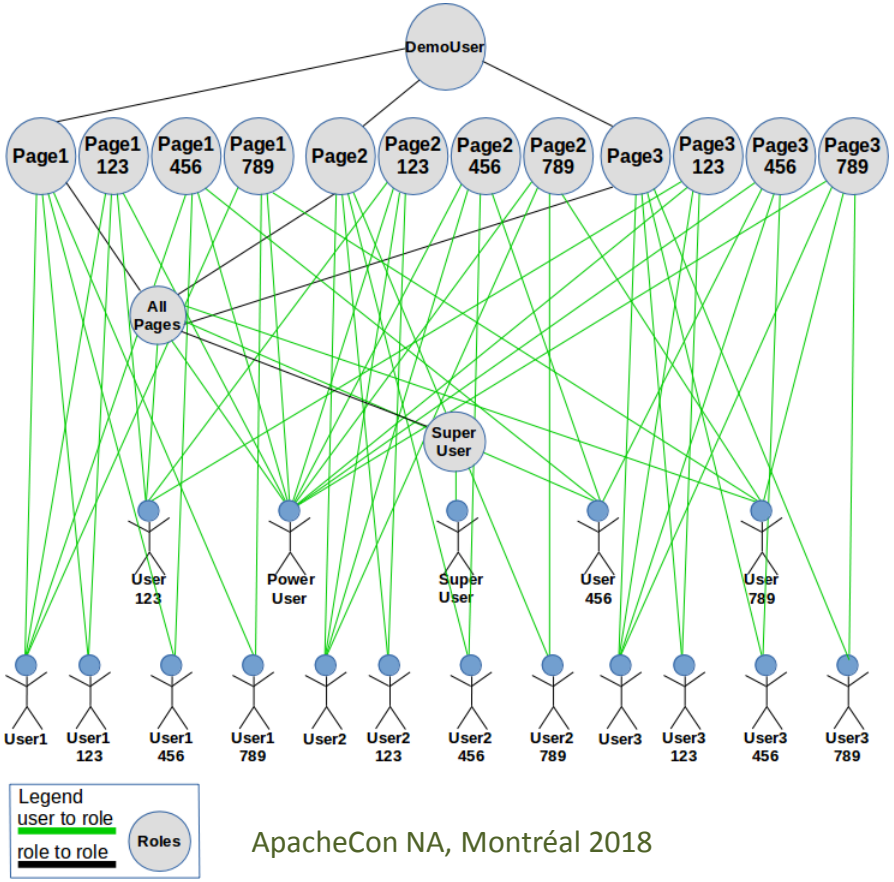
- https://github.com/shawnmckinney/apache-fortress-demo

| User Foo | Customer 123 | Customer 456 | Customer 789 |
|---|---|---|---|
| Page1 | False | True | True |
| Page2 | True | False | False |
| Page3 | True | False | False |

# We still have a problem…

- ou=Roles (17)
  - cn=PAGE1_123
  - cn=PAGE1_456
  - cn=PAGE1_789
  - cn=PAGE2_123
  - cn=PAGE2_456
  - cn=PAGE2_789
  - cn=PAGE3_123
  - cn=PAGE3_456
  - cn=PAGE3_789

# Roles Have Exploded



# Cartesian Product

$A \times B = \{(a,b) \mid a \in A \text{ and } b \in B\}$

- A : role
- B : relationships

# Number of Roles = sizeof(A) * sizeof(B)

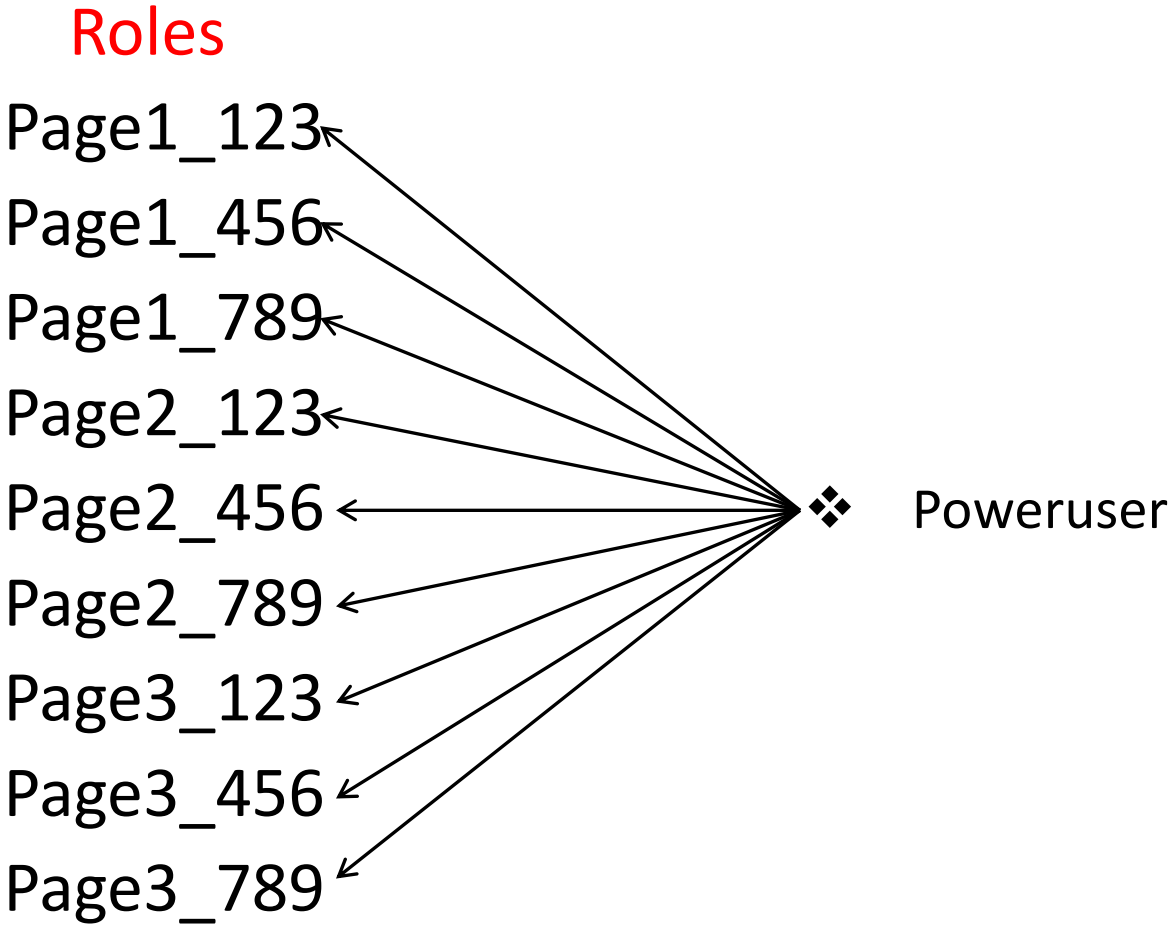
| Roles (A) | | Relationships (B) |
|---|---|---|
| Page1 | | Customer 123 |
| Page2 | * | Customer 456 |
| Page3 | | Customer 789 |

=>

Roles

1. Page1-123
2. Page1-456
3. Page1-789
4. Page2-123
5. Page2-456
6. Page2-789
7. Page3-123
8. Page3-456
9. Page3-789

# RBAC only


Roles
Page1_123
Page1_456
Page1_789
Page2_123
Page2_456
Page2_789
Page3_123
Page3_456
Page3_789
Poweruser

# The Solution

Use attributes to constrain under what conditions roles may be activated.

# RBAC w/ ABAC

| Roles | Users | Constraints |
|---|---|---|
| Page1 | User1-123 | Page1 : 123 |
| | User1-456 | Page1 : 456 |
| | User1-789 | Page1 : 789 |
| Page2 | User2-123 | Page2 : 123 |
| | User2-456 | Page2 : 456 |
| | User2-789 | Page2 : 789 |
| Page3 | User3-123 | Page3 : 123 |
| | User3-456 | Page3 : 456 |
| | User3-789 | Page3 : 789 |

# RBAC w/ ABAC

## Roles

Page1

Page2

Page3

- User1
  - Page1 : 123, 456, 789, …
- User123
  - Page1 : 123
  - Page2 : 123,
  - Page3 : 123
- User2
  - Page2 : 123, 456, 789, …
- User456
  - Page1 : 456
  - Page2 : 456
  - Page3 : 456
- User3
  - Page3 : 123, 345, 789, …
- User789
  - Page1 : 789
  - Page2 : 789
  - Page3 : 789

# RBAC w/ ABAC

Roles

Page1

Page2

Page3

- Poweruser
  - Page1 : 123, 456, 789, …
  - Page2 : 123, 456, 789, …
  - Page3 : 123, 345, 789, …

# Under the Hood

https://appdevcloudworkshop.github.io/images/introduction/image16.png

# RBAC only

- ou=Roles (17)
  - cn=PAGE1_123
  - cn=PAGE1_456
  - cn=PAGE1_789
  - cn=PAGE2_123
  - cn=PAGE2_456
  - cn=PAGE2_789
  - cn=PAGE3_123
  - cn=PAGE3_456
  - cn=PAGE3_789

| uid | user123 |
|---|---|
| **uidNumber** | **1237** |
| description | Apache Fortress Demo User123 Access all Pages for Customer 123 |
| displayName | test2 |
| ftCstr | user123$0$0000$0000$20090101$20990101$none$none$1234567 |
| ftProps | initAttrArrays: |
| ftRA | PAGE1_123 |
| ftRA | PAGE2_123 |
| ftRA | PAGE3_123 |
| ftRC | PAGE1_123$0$0000$0000$none$none$none$none$all |
| ftRC | PAGE2_123$0$0000$0000$none$none$none$none$all |
| ftRC | PAGE3_123$0$0000$0000$none$none$none$none$all |

# RBAC w/ ABAC

- ou=Roles (23)
  - cn=ABAC_PAGE1
  - cn=ABAC_PAGE2
  - cn=ABAC_PAGE3

| uid | user123 |
|---|---|
| **uidNumber** | **1254** |
| description | Apache Fortress Demo User123 Access all Pages for Customer 123 |
| displayName | test2 |
| ftCstr | user123$0$$$$$$$ |
| ftProps | initAttrArrays: |
| ftRA | ABAC_PAGE1 |
| ftRA | ABAC_PAGE2 |
| ftRA | ABAC_PAGE3 |
| ftRC | ABAC_PAGE1$0$$$$$$$ |
| ftRC | abac_page1$type$USER$customer$123$ |
| ftRC | ABAC_PAGE2$0$$$$$$$ |
| ftRC | abac_page2$type$USER$customer$123$ |
| ftRC | ABAC_PAGE3$0$$$$$$$ |
| ftRC | abac_page3$type$USER$customer$123$ |

symas

# Role Constraints

```
<roleconstraint role="PAGE1"
  key="customer" … />
<roleconstraint role="PAGE2"
  key="customer" … />
<roleconstraint role="PAGE3"
  key="customer" … />
```

https://github.com/shawnmckinney/fortress-abac-demo/blob/master/src/main/resources/fortress-abac-demo-load-policy.xml

# User-Role Constraints

```
<roleconstraint userId="User123" role="PAGE1"
  key="customer" value="123" … />

<roleconstraint userId="User123" role="PAGE2"
  key="customer" value="123" … />

<roleconstraint userId="User123" role="PAGE3"
  key="customer" value="123" … />
```

https://github.com/shawnmckinney/fortress-abac-demo/blob/master/src/main/resources/fortress-abac-demo-load-policy.xml

# Code Sample

```
// Nothing new here:
User user = new User("curly");

// This is new:
RoleConstraint constraint = new RoleConstraint( );

// In practice we're not gonna pass hard-coded key-values in here:
constraint.setKey( "customer" );
constraint.setValue( "123" );

// This is just boilerplate goop:
List<RoleConstraint> constraints = new ArrayList();
constraints.add( constraint );

try
{
   // Create the RBAC session with ABAC constraint -- customer=123, asserted:
   Session session = accessMgr.createSession( user, constraints );
   ...
}
```

https://github.com/shawnmckinney/fortress-abac-demo/blob/master/src/main/java/com/mycompany/MyBasePage.java

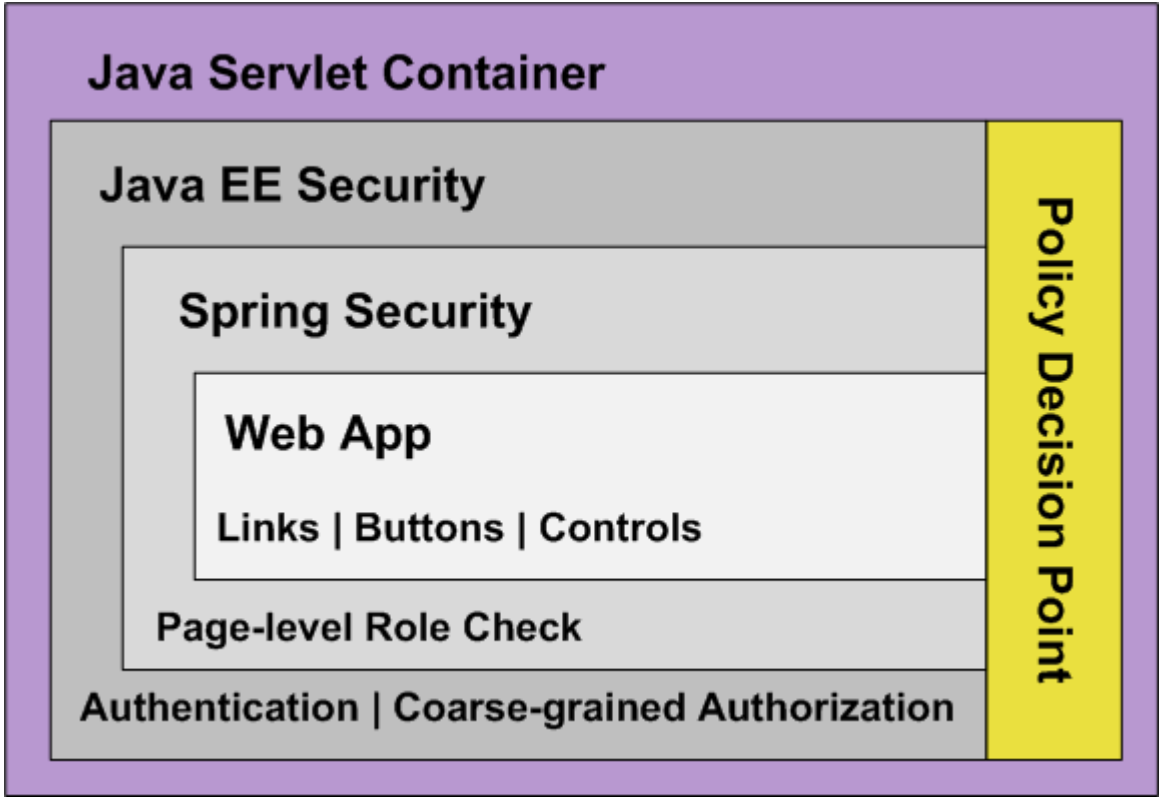
# Example #4

Apache

Fortress

ABAC

Demo




https://github.com/shawnmckinney/fortress-abac-demo

# ABAC Demo

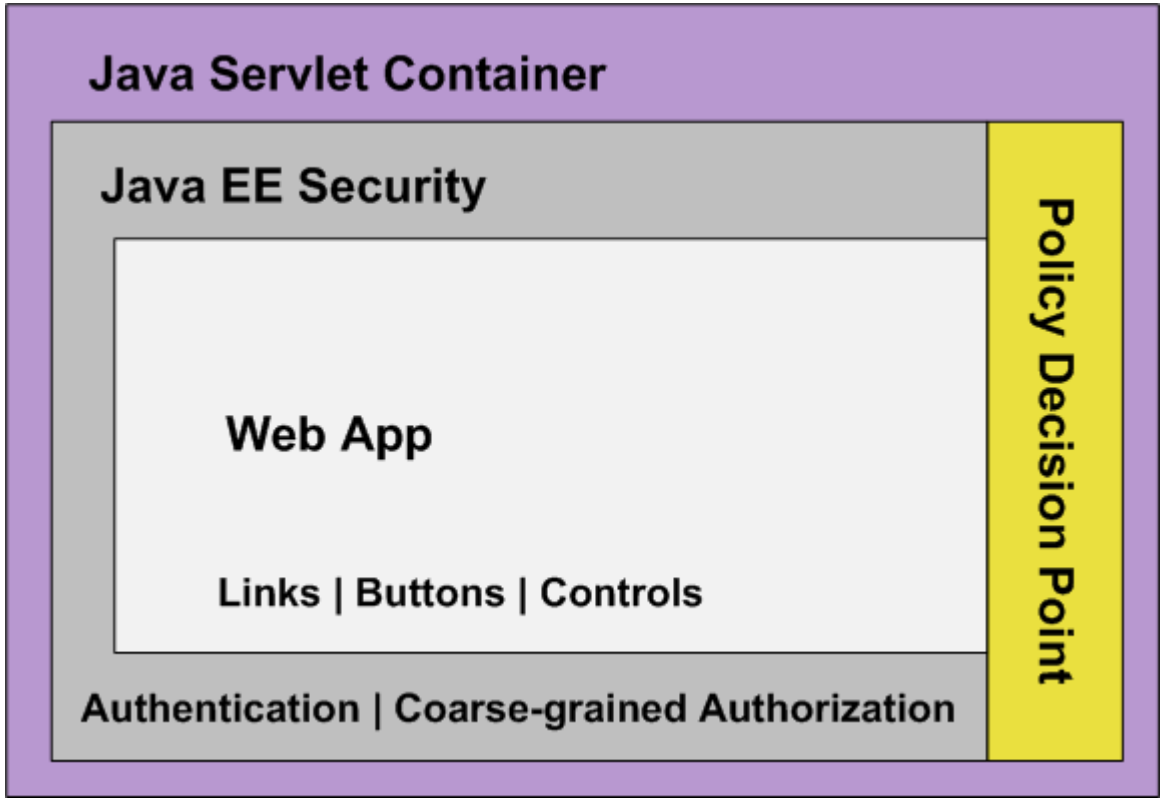

is is the first time you've seen this Stop error screen,
art your computer. If this screen appears again, follow
e steps:
to make sure any new hardware or software is properly installed.
is is a new installation, ask your hardware or software manufactu
any windows updates you might need.
oblems continue, disable or remove any newly
ftware. Disable BIOS memory options such as
u need to use Safe Mode to remov
computer, press F8 to select Ad
ct Safe Mode.
nical information:
STOP: 0x00000050 (0xFD3094C2,0x00000
SPCMDCON.SYS - Address FBFE7617 base at

# Example #5

RBAC

ABAC

Demo




https://github.com/shawnmckinney/rbac-abac-sample

# Closing Thoughts

1. Never allow users more than they need to do their jobs
   - *Principle of Least Privilege*
2. Apply security controls across many layers
   - *Defense in Depth*
3. RBAC may be combined with ABAC
   - *Fine-grained Authorization*

# Examples

1. https://github.com/shawnmckinney/serial-exploit-sample
2. https://github.com/shawnmckinney/apache-fortress-demo
3. https://github.com/shawnmckinney/role-engineering-sample
4. https://github.com/shawnmckinney/fortress-abac-demo
5. https://github.com/shawnmckinney/rbac-abac-sample

Bonus:

6. https://github.com/shawnmckinney/fortress-saml-demo

# Contact Info

Twitter: @shawnmckinney

Website: http://symas.com

Email: smckinney@apache.org

Blog: https://iamfortress.net

Project: https://directory.apache.org/fortress