RBAC AUTHORIZATION
WITH APACHE DIRECTORY SERVER AND FORTRESS

Shawn McKinney
Emmanuel Lécharny
Introducing Fortress

- Accepted as a **Directory** sub-project in October 2014.
- **IP** clearance on its way
- Code injection in **Git** repository last week
- ~90 000 SLOCs
Introducing Fortress

RBAC Access Control & Management Apps

- **Core** - Software Development Kit
- **Realm** - Policy Enforcement Plug-in for Java EE Apps & Servers
- **Web** - Administrative GUI for Management of Policies
- **REST** - XML/HTTP Policy Server
Fortress Features

- **ANSI Role-Based Access Control**
- **ARBAC02 Model Delegated Administration**
- **IETF Password Policies**
- **LDAPv3 Interoperability**
- **Multi-tenant Data and Object Model**
- **Audit Trail (OpenLDAP only)**
- **Works with any LDAP v3 compliant server**
Introducing ApacheDS

LDAP & Kerberos Server, and more...

- **ApacheDS** - LDAP Server
- **Studio** - LDAP Browser
- **LDAP API** - A new Java LDAP API
- **Mavibot** - MVCC Btree
- **eSCIMO** - SCIM 2.0 Implementation
Introducing ApacheDS

- Get accepted in Apache Incubator in 2003
- Became a TLP in 2005
- 37 releases since then
- Around 700,000 SLocs
ApacheDS Features

- **LDAP server in Java**
- **Kerberos server built-in**
- **Embeddable**
- **Password Policy support**
- **Multi-master replication (RFC 4533)**
- **X500 Authorization**
- **Multi-platform (Windows, Linux, Mac OSX)**
Early History

- Introduced in 1992 (David Ferraiolo and Richard Kuhn)
- Meant to address critical shortcomings of DAC
- Integrity was lacking as the requirement for data and process to be modified only in authorized ways by authorized users.
Middle Years – ‘Towards a Unified Standard’

- **2000**, 'The NIST Model for a Role-Based Access Control: Towards a Unified Standard' (Sandhu, Ferraiolo, Kuhn).

- **RBAC Formal Model**

- **Basis for the Standard to follow.**

- **Functional Specs written in z-notation**
In 2004 ANSI formalized RBAC into a Standard.

ANSI INCITS 359
ANSI RBAC INCITS 359

RBAC 0 - Users, Roles, Perms Objects, Operations
RBAC 1 - Hierarchical Roles
RBAC 2 - Static Separation of Duties
RBAC 3 - Dynamic Separation of Duties
ANSI RBAC INCITS 359
ANSI RBAC Functional Model

Three standard interfaces:

- **Administrative** - CRUD
- **Review** - policy interrogation
- **System** - policy enforcement
**ANSI RBAC Object Model**

- **User** – human or machine entity
- **Role** – a job function within an organization
- **Object** – maps to system resources
- **Operation** – executable image of program
- **Permission** – approval to perform an Operation on one or more Objects
- **Session** – contains set of activated roles for User
Delegated Admin:

- **Manage** the RBAC system
- **Used for controlling** what the administrators can do
ARBAC 02
ARBAC Functional Model

Three standard interfaces:

- **Administrative** - CRUD
- **Review** - policy interrogation
- **System** - policy enforcement
ARBAC OBJECT MODEL

Four basic elements:

- **Admin Role** - an administrator
- **Admin Permission** - approval to perform an RBAC administrative operation
- **User Org Unit** - maps to an org chart of people
- **Perm Org Unit** - maps to a graph of IT applications and systems
DEMO FORTRESS

Java Secure Socket Extension ➔ JSSE

Java EE Security ➔ JAVA EE SECURITY

Spring Security ➔ SPRING SECURITY

Web App ➔ WEB APP FRAMEWORK

Database Functions ➔ DATABASE FUNCTIONS
## Demo Security Policy

<table>
<thead>
<tr>
<th></th>
<th>P1 C123</th>
<th>P1 C456</th>
<th>P1 C789</th>
<th>P2 C123</th>
<th>P2 C456</th>
<th>P2 C789</th>
<th>P3 C123</th>
<th>P3 C456</th>
<th>P3 C789</th>
</tr>
</thead>
<tbody>
<tr>
<td>Super User *</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>Power User</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>User1</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>User1 123</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>User1 123</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

* denotes Dynamic Separation of Duties is NOT enforced
WHERE TO GET DEMO CODE

Download from here:

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

Follow instructions in the README to generate tutorial document
Future...

- ApacheDS in Fortress
  - In-Memory
  - All in One
  - Fast
  - SyncRepl
FUTURE...

- FORTRESS IN APACHEDS
  - All in one
  - Fast
  - Lightweight

OPENLDAP ALREADY AS IT: https://symas.com/products/symas-enforcement- foundry/suite/accelerator/
Contacts

HTTP://DIRECTORY.APACHE.ORG

HTTP://WWW.SYMAS.COM

Shawn McKinney,
twitter@shawnmckinney
smckinney@apache.org

Emmanuel Lécharny
twitter@elecharny
elecharny@apache.org