ENTERPRISE SECURITY
WITH KEYCLOAK
From the Intranet to Mobile

By Divya Mehra and Stian Thorgersen
AGENDA

FROM CHALLENGES TO SOLUTION
CHALLENGES
THE OLD WAY

- Securing monolithic web app relatively easy
- Username and password form
- Credentials verified against table in DB
- HTTP Session stores security context
IT'S NOT JUST A FORM AND A TABLE ANYMORE

- Enterprise software has changed
- No longer one or two apps inside firewall
- Now we have many separate systems
- Exposed to mobile users and partners
THE NEW WAY?

- Multiple apps
- Multiple variants of each app
- Multiple services
- Multiple user dbs
- Multiple logins
- Outside firewall
AUTHENTICATION

- Passwords not sufficient
- Users create bad passwords (123456 and password)
- Passwords policies help, but no guarantee
- Users reuse passwords
- Passwords can be lost
- Secure storage is required
- Need two-factor authentication
APP TYPES

Have to deal with many app, variants & programming languages

- Client-side and server-side web
- Mobile (native and hybrid)
- APIs/Services
- ...
MOBILE

- Users don't want to login frequently
- Don't store username and password on phone
- What if device is lost?
- Sessions and cookies aren't ideal
- Requires public services
SINGLE SIGN-ON

- Not as trivial as it may seem
- Single Sign-Out can be even harder
- Need Remote Sign-Out
MANAGE

- Apps
- Services
- Users
- Devices
- Permissions
- Sessions and logs

and.. Ideally manage everything from one console
SELF SERVICE

- Users can manage their own accounts
- Recover password
- Update profile
- Enable two-factor authentication
- Manage sessions
- Account history

and.. Ideally manage everything from one console
INTEGRATION

- Third party apps
- Existing Infrastructure
- New Infrastructure after acquisition
- External users
- Social networks
VULNERABILITIES

- Broken Authentication and Session Management is #2 on Open Web Application Security Project (OWASP) Top Ten list
- Recommendation is to not implement your own!
PROTOCOLS

- OpenID Connect
- SAML 2.0
OPENID CONNECT

- Built on OAuth 2.0
- RESTful
- JSON
- Easy to use
- Less mature - final spec released last year
SAML 2.0

- XML
- Harder to use and understand
- Mature - 1.0 was adopted as an OASIS standard in 2002
TOKENS

- Decouples authentication
- Cross-domain
- Stateless
- Only sent when needed
- Standards based
AUTHENTICATION

- Authenticate with Keycloak
- Login forms provided by Keycloak
- Two-factor authentication
- Requires SSL
- Passwords are salted and hashed with PBKDF2
  - Iterations configurable
<button onclick="keycloak.login()">Login</button>
Welcome App

WELCOME

LOGIN
Login to Keycloak realm
Hello
STIAN THORGERSEN
LOGOUT
APP INTEGRATION

- Keycloak Client Adapters
- Keycloak Proxy
- OpenID Connect Resource Provider library
- SAML Service Provider library
CLIENT ADAPTERS

- JBoss EAP & WildFly
- JBoss Fuse
- JBoss BRMS
- JavaScript
- NodeJS
- Mobile (Apache Cordova and Native)
- Spring
- Tomcat, Jetty
- More coming (contributions welcome!)
EXAMPLE

- Simple example to demonstrate features
- Two HTML5 applications
- RESTful services deployed to WildFly
ADMIN CONSOLE

Configure and manage everything from one console
- Including settings, applications, services, users, permissions and sessions
### Admin Console - Realm settings

#### Settings

<table>
<thead>
<tr>
<th>Section</th>
<th>Subsection</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Login</strong></td>
<td>User registration</td>
<td>OFF</td>
</tr>
<tr>
<td></td>
<td>Edit username</td>
<td>OFF</td>
</tr>
<tr>
<td></td>
<td>Forget password</td>
<td>OFF</td>
</tr>
<tr>
<td></td>
<td>Remember Me</td>
<td>OFF</td>
</tr>
<tr>
<td></td>
<td>Verify email</td>
<td>OFF</td>
</tr>
<tr>
<td></td>
<td>Require SSL</td>
<td>OFF</td>
</tr>
</tbody>
</table>

**Additional note:**
- External request for Require SSL.
## Admin Console - Clients

![Admin Console - Clients](image)

### Clients

<table>
<thead>
<tr>
<th>Client ID</th>
<th>Enabled</th>
<th>Base URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>app</td>
<td>true</td>
<td><a href="http://localhost:8080/app">http://localhost:8080/app</a></td>
</tr>
<tr>
<td>realm-management</td>
<td>true</td>
<td>Not defined</td>
</tr>
<tr>
<td>security-admin-console</td>
<td>true</td>
<td>/auth/admin/summit/console/index.html</td>
</tr>
<tr>
<td>services</td>
<td>true</td>
<td><a href="http://localhost:8080/services">http://localhost:8080/services</a></td>
</tr>
<tr>
<td>broker</td>
<td>true</td>
<td>Not defined</td>
</tr>
<tr>
<td>app2</td>
<td>true</td>
<td><a href="http://localhost:8080/app2">http://localhost:8080/app2</a></td>
</tr>
<tr>
<td>account</td>
<td>true</td>
<td>/auth/realm/summit/account</td>
</tr>
</tbody>
</table>
Admin Console - Client settings

App

Settings | Roles | Mappers | Scope | Revocation | Sessions | Installation
---|---|---|---|---|---|---
Client ID | app
Name
Enabled | ON
Consent Required | OFF
Direct Grants Only | OFF
Client Protocol | openid-connect
Access Type | public
Valid Redirect URIs
| http://localhost:8080/app/*
Base URL | http://localhost:8080/app
Admin URL
Web Origins | http://localhost:8080
Admin Console - User settings

### Stian

<table>
<thead>
<tr>
<th>Attributes</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>f4fd759-25aa-49a4-a711-fbcba4ab3deee</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Username</td>
<td>stian</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Email</td>
<td><a href="mailto:stian@redhat.com">stian@redhat.com</a></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>First Name</td>
<td>Stian</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Last Name</td>
<td>Thorgeren</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Enabled</td>
<td>ON</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Email verified</td>
<td>OFF</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

> Contact Information

<table>
<thead>
<tr>
<th>Actions</th>
<th>Select an action...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete</td>
<td>Delete</td>
</tr>
</tbody>
</table>
Admin Console - User role mappings

[Image of Keycloak Admin Console showing role mappings for a user named Stian]
ACCOUNT MANAGEMENT

A console for users to manage their own account
Account Management - Profile

Edit Account

- **Username**: stian
- **Email**: stian@redhat.com
- **First Name**: Stian
- **Last Name**: Thorgersen

Fields:
- Street
- City or Locality
- State, Province, or Region
- Zip or Postal code
- Country

Actions:
- Cancel
- Save
Account Management - Password

Change Password

- Password
- New Password
- Confirmation

All fields required

[Cancel] [Save]
### Account Management - Applications

#### Applications

<table>
<thead>
<tr>
<th>Application</th>
<th>Available Permissions</th>
<th>Granted Permissions</th>
<th>Granted Personal Info</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message App</td>
<td>Administrator privileges, User privileges, View profile in <code>Account</code>, Manage account in <code>Account</code></td>
<td>Full Access</td>
<td>Full Access</td>
<td></td>
</tr>
<tr>
<td>Welcome App</td>
<td>Administrator privileges, User privileges, View profile in <code>Account</code>, Manage account in <code>Account</code></td>
<td>Full Access</td>
<td>Full Access</td>
<td></td>
</tr>
<tr>
<td>Account</td>
<td>View profile in <code>Account</code>, Manage account in <code>Account</code></td>
<td>Full Access</td>
<td>Full Access</td>
<td></td>
</tr>
</tbody>
</table>
## Account Log

<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
<th>IP</th>
<th>Client</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jun 17, 2015 1:25:20 PM</td>
<td>login</td>
<td>127.0.0.1</td>
<td>account</td>
<td>auth_method = form, username = stan</td>
</tr>
<tr>
<td>Jun 17, 2015 1:20:44 PM</td>
<td>login</td>
<td>127.0.0.1</td>
<td>account</td>
<td>auth_method = form, username = stan</td>
</tr>
<tr>
<td>Jun 17, 2015 9:10:05 AM</td>
<td>login</td>
<td>127.0.0.1</td>
<td>app2</td>
<td>auth_method = form, username = stan</td>
</tr>
<tr>
<td>Jun 17, 2015 8:57:25 AM</td>
<td>logout</td>
<td>127.0.0.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jun 17, 2015 8:57:21 AM</td>
<td>login</td>
<td>127.0.0.1</td>
<td>app2</td>
<td>auth_method = form, username = stan</td>
</tr>
</tbody>
</table>
SINGLE SIGN-ON

- Web SSO
- Enterprise/Desktop SSO Bridge (Kerberos)
- Single Sign-Out
- Remote Sign-Out
THEMES

- Brand login pages and account management to integrate with your corporate brand
- HTML templates for more than just styling
Admin Console - Configure theme
Login - Summit theme
LOGIN FLOWS

- Required actions
- Recover password
- Two factor authentication
- Registration
Login - Extra features enabled

LOG IN TO SUMMIT

Username or email
Password

Remember me
Forgot Password?

Log in  Cancel

New user? Register
Login - Configure two factor authentication

MOBILE AUTHENTICATOR SETUP

You need to set up Mobile Authenticator to activate your account.

1. Install FreeOTP or Google Authenticator on your mobile. Both applications are available in Google Play and Apple App Store.
2. Open the application and scan the barcode or enter the key:

![QR code image]

G5QW IT2U OB3U KVKF KNGE CTRT MIZW C6S

3. Enter the one-time code provided by the application and click Submit to finish the setup.
<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Email</td>
<td><a href="mailto:stian@redhat.com">stian@redhat.com</a></td>
</tr>
<tr>
<td>First name</td>
<td>Stian</td>
</tr>
<tr>
<td>Last name</td>
<td>Thorgersen</td>
</tr>
</tbody>
</table>

You need to update your user profile to activate your account.
PASSWORD POLICIES

- Set required complexity for passwords
- Prevent reuse of old passwords
- Require regular updating of passwords
- Set hashing intervals
Admin Console - Password policies

### Settings

#### Realm Credentials Settings

<table>
<thead>
<tr>
<th>Required User Credentials</th>
</tr>
</thead>
<tbody>
<tr>
<td>x.password</td>
</tr>
</tbody>
</table>

#### Realm Password Policy

<table>
<thead>
<tr>
<th>Policy Type</th>
<th>Policy Value</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hash Iterations</td>
<td>10000</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Force Expired Password Change</td>
<td>365</td>
<td></td>
</tr>
<tr>
<td>Password History</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>
Login - invalid password update

Invalid password: must not be equal to any of last 3 passwords.
USER FEDERATION

- Sync users with external directories
- Read-only or read-write
Admin Console - Add LDAP user federation

Add LDAP User Federation Provider

Required Settings

- **Console display name**: defaults to id
- **Priority**: 0
- **Edit mode**: 
- **Sync Registrations**: OFF
- **Vendor**: Active Directory
- **Username LDAP attribute**: cn
- **RDN LDAP attribute**: cn
- **UUID LDAP attribute**: objectGUID
- **User Object Classes**: person, organizationalPerson, user
- **Connection URL**: ldap

Test connection
## User Federation

<table>
<thead>
<tr>
<th>ID</th>
<th>Provider Name</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>3e49a3ae-9f6d-49d4-9d5c-b49c103c4d65</td>
<td>Ldap</td>
<td>0</td>
</tr>
<tr>
<td>e76f4a3-384f-4d2f-a81c-61beb5ceed08</td>
<td>Kerberos</td>
<td>0</td>
</tr>
</tbody>
</table>
IDENTITY BROKERING

- Allow external users to sign-on
- Supports sign-on with social networks
Admin Console - Add SAML Identity Provider

Identity Provider: SAML

- Alias: saml
- Enabled: On
- Authenticate By Default: Off
- Store Tokens: Off
- Stored Tokens Readable: Off
- Update Profile on First Login: Off
- Trust email: Off
- GUI order: 

SAML Config:

- Single Sign-On Service URL: 

## Admin Console - Identity Providers

### Identity Providers Summit

<table>
<thead>
<tr>
<th>Name</th>
<th>Provider</th>
<th>Enabled</th>
<th>GUI order</th>
</tr>
</thead>
<tbody>
<tr>
<td>saml</td>
<td>saml</td>
<td>true</td>
<td></td>
</tr>
<tr>
<td>oidc</td>
<td>oidc</td>
<td>true</td>
<td></td>
</tr>
<tr>
<td>github</td>
<td>github</td>
<td>true</td>
<td></td>
</tr>
<tr>
<td>twitter</td>
<td>twitter</td>
<td>true</td>
<td></td>
</tr>
<tr>
<td>facebook</td>
<td>facebook</td>
<td>true</td>
<td></td>
</tr>
<tr>
<td>google</td>
<td>google</td>
<td>true</td>
<td></td>
</tr>
<tr>
<td>linkedin</td>
<td>linkedin</td>
<td>true</td>
<td></td>
</tr>
<tr>
<td>stackoverflow</td>
<td>stackoverflow</td>
<td>true</td>
<td></td>
</tr>
</tbody>
</table>
MAPPERS

- Customize tokens
- Map claims and attributes from external tokens
- Map attributes and groups from LDAP
Admin Console - Token mappers

![Admin Console screenshot]

A table showing token mappers:

<table>
<thead>
<tr>
<th>Name</th>
<th>Category</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>email</td>
<td>Token mapper</td>
<td>User Property</td>
</tr>
<tr>
<td>given name</td>
<td>Token mapper</td>
<td>User Property</td>
</tr>
<tr>
<td>full name</td>
<td>Token mapper</td>
<td>User's full name</td>
</tr>
<tr>
<td>username</td>
<td>Token mapper</td>
<td>User Property</td>
</tr>
<tr>
<td>family name</td>
<td>Token mapper</td>
<td>User Property</td>
</tr>
</tbody>
</table>
### Keycloak-oidc Mappers

<table>
<thead>
<tr>
<th>Name</th>
<th>Category</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>external-user-to-user</td>
<td>Role Importer</td>
<td>External Role to Role</td>
</tr>
</tbody>
</table>
Admin Console - LDAP mappers

<table>
<thead>
<tr>
<th>Name</th>
<th>Category</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>creation date</td>
<td>Attribute Mapper</td>
<td>User Attribute</td>
</tr>
<tr>
<td>email</td>
<td>Attribute Mapper</td>
<td>User Attribute</td>
</tr>
<tr>
<td>first name</td>
<td>Attribute Mapper</td>
<td>User Attribute</td>
</tr>
<tr>
<td>last name</td>
<td>Attribute Mapper</td>
<td>User Attribute</td>
</tr>
<tr>
<td>modify date</td>
<td>Attribute Mapper</td>
<td>User Attribute</td>
</tr>
<tr>
<td>username</td>
<td>Attribute Mapper</td>
<td>User Attribute</td>
</tr>
</tbody>
</table>
VULNERABILITIES

- Standard Protocols
- Built-in Brute Force protection
- Integrate with Intrusion Detection
- Protected against known attacks
- Patches