Spring Security

Introduction

Fedora security configuration via spring was introduced in Fedora 3.5. In general, security in Fedora is provided by a series of servlet filters. Each filter provides some security-related purpose, such as policy enforcement, authentication, or ssl redirection. The set of active security-related filters and their individual configuration settings is determined by Spring, using configuration files present in `FEDORA_HOME/server/config/spring/web/`. This method of configuring security replaces the pre-3.5 technique of specifying security-related servlet filters directly in `web.xml`.

Important files and directories

- **`FEDORA_HOME/server/config/spring/web/`**: Any spring xml file present in this directory will be loaded into the web application context. The web application context in 3.5 is distinct from the fedora server context, so all beans relevant to servlet filters must be present in `FEDORA_HOME/server/config/spring/web/`.
- **`FEDORA_HOME/server/config/spring/web/security.xml`**: Default spring bean xml file containing security configuration. The contents of this file are initially determined by the fedora installer.
- **`FEDORA_HOME/server/config/spring/web/web.properties`**: Contains properties and values that are substituted into any spring beans at runtime. All spring beans defined in the web application context are pre-processed with a PropertyPlaceholderConfigurer.

Unimportant files and directories

- **`FEDORA_HOME/server/config/spring/web/alt`**: Directory containing additional or alternate configuration files that are *not* active by default. This directory is ignored by spring. Nothing in this directory will be parsed into the web application context.

- **`FEDORA_HOME/server/config/spring/web/alt/security-complicated.xml`**: The existing `security.xml` file tries to emulate the classic behaviour of Fedora with regard to SSL requirements on certain resources. In the process of doing so, the list of resources considered "access" or "management" for purposes of requiring SSL seemed inconsistent with their documentation as access or management methods, especially when considering the REST API. This file defines security behaviour that can be considered "more correct", but may not match the current expectations/assumptions of fedora clients and thus is disabled by default.

Configuration settings

As mentioned earlier, security in Fedora is achieved through the action of servlet filters. A single DelegatingFilterProxy filter for security is defined in `web.xml`. This filter is configured to delegate to a FilterChainProxy, which forms the core of spring security in Fedora. This FilterChainProxy can be configured to apply any number of servlet filters based upon certain criteria such as URL path. Configuring security in Fedora, then, is a matter of choosing which servlet filters are applied to which resource URLs. Each servlet filter is itself a bean which is instantiated through Spring, and may have its own configuration and collaborators defined through spring beans. The result is a flexible

Authentication

There is a recommended choice for authentication with Fedora: FESL. There is also an experimental facility available for "upstream authentication", which is meant to allow integration with SSO services.

FESL

Upstream Auth

The upstream authentication filter is available as "org.fcrepo.security.http.AuthZHttpRequestAttributesFilter". It enables the use of HTTP request headers and authentication status in the Fedora access policy machinery. It requires two pieces of configuration: a header-name that contains the trusted name of the authenticated principal, and a list of header-names that contain attributes of interest. An example configuration:

```
 <bean id="upstreamAuthFilter" class="org.fcrepo.security.http.AuthZHttpRequestAttributesFilter">
    <property name="principalHeader" value="name"/>    <property name="names" value="age weight height"/>
</bean>
```

Assuming incoming requests are actually populated with these headers by some upstream process, and assuming that incoming requests are actually authenticated, this configuration will provide XACML policies with a subject identified by HTTP header "name" and additional attributes drawn from headers "age", "weight", and "height". In a future release of Fedora, it is possible that more flexibility will be provided to this apparatus to cover more configurations (e.g. using environment variables *or* HTTP headers).

Policy Enforcement

SSL

Advanced Customization