Outline

- Server side
  - Dependencies
  - Installing it
  - Configuring it
- Client side coding
- Browser setup

Note

- Use the recent edg-java-security (e.g. from the CVS HEAD if you are building spitfire…
- Probably I'll tag one..

Deps: Secure Tomcat

- You need bog-standard Tomcat (preferably version 4.0.6 RPM version available from the EDG website)
- You need edg-java-security, preferably installed via RPM
- You should edit your server.xml in Tomcat to add a secure container pointing to $CATALINA_HOME/webapps-secure
- Want to move to 4.1.18 asap.

Deps: A database to use

- Initially, we assume MySQL
- Install standard MySQL
  - Latest 3.* or 4.*
- The edg-spitfire-config module will need the DBA role (‘root’) and password to set up its default tables.

Install Spitfire

- Can install via RPM or tarball
  - Much easier to install via RPM
- This installs into /opt/edg/share/webapps/edg-spitfire-server
- The config files live in WEB-INF/
  - They are tokenized, with the values to be filled in from either edg-spitfire-config or LCFG

LCFG

edg-spitfire-config

- Currently, this does LCFG's job
- Install this edg-spitfire-config, preferably by RPM
- This installs a script in
  - /opt/edg/sbin/edg-spitfire-configure.sh
- And a config file in
  - /opt/edg/etc/edg-spitfire/spitfireDB.conf

spitfireDB.conf file

- Must be edited a bit before running the configure script
  - DBA password
  - Names for your roles and their passwords
  - Some initial security role mappings
- Running the script will setup spitfire's config files and will create the matching roles and tables in the database.

What does it set up?

- MySQL: creates your users in the ‘mysql:user’ table
- poolman.xml: writes these usernames into the connection pool configuration so spitfire knows what names to use
  - Would like to drop poolman
- Creates GRID database and populates examples tables

How does it setup the authzManager?

- Authorization manager used with dbmaps
  - Initial role maps are added to the mapping tables in the database
- spitfireAuthzManager.xml: contains the setup information for the authzManager.
  - The script adds the relevant database connection names into this config file so the authzManager can look-up the DN to role mappings in the database.
- I won't discuss the authorization policy that is implemented…

Authz Question?

- Question for Friday….
- Are we happy with the current setup in Spitfire?
  - Default roles
  - Extra roles if you need them
  - Roles based on DN or regexp of DN
  - A bit complex..
  - Would like to be able to use the new Authz GUI too

Client side coding

- Currently, only the java client is available
- The RPM installs edg-spitfire-client-java.jar in /opt/edg/lib/edg-spitfire-client-java/

- Examples client programs are in the tarball
  - ./edg-spitfire-client-java-2.1.0/samples
  - Should the RPM contains the examples too?

What you need client side

- A few JARS
  - The Axis libs, log4j
  - Edg-java-security-trustmanager-client
  - Edg-java-security-authorization-client
  - Bouncycastle

  - (and edg-spitfire-client-java JAR)

Client config stuff

- One Axis file
  - client-config.wsdd
- Tells the client to pre-process the request and add ‘role’ and ‘policy’ SOAP headers to the request, based on the java system properties:
  - ‘edg-security.role’ and ‘edg-security.policy’

Coding is simple…

- import org.edg.data.spitfire.service.*;
- import org.edg.data.spitfire.service.base.*;

- SpitfireBaseServiceLocator sf = new SpitfireBaseServiceLocator();
- SpitfireBase sfBase = sf.getSpitfireBase(new URL(endpoint));

- int rowsInserted = sfBase.insert("GRID", "repcat", attribs, values);

- You specify either the ‘base’, ‘admin’ or ‘info’ API… {see javadoc}

Running it

- java -classpath "$cp" -Daxis.socketFactory=org.edg.security.trustmanager.\
  axis.AXISSocketFactoryFactory
  -DsslCAFiles=$certdir/*.0
  -DgridProxyFile=$proxy
  -Daxis.ClientConfigFile=client-config.wsdd
  -Dedg-security.role=$role
  -Dedg-security.policy=$policy“
  ProgramClass CommandLineParameters

edg-spitfire-browser

- This is setup to use the same authorization tables as the main spitfire service
- The edg-spitfire-config script will configure this too, if it finds it.
- Point web browser at the example page…