Grouper UI custom authentication example

Grouper UI authentication

The Grouper UI will check request attribute REMOTE_USER, or request.getUserPrincipal(), or request.getRemoteUser(), so it should work with common SSO solutions (e.g. Shib, Cosign, etc)

See debug information in logs in log4j.properties

```
log4j.logger.edu.internet2.middleware.grouper.ui.GrouperUiFilter = DEBUG
```

Here is an example of doing something else... in the web.xml declare your filter:

```
<filter>  <filter-name>Your Filter</filter-name>  <filter-class>com.path.whatever.YourFilter</filter-class> </filter>
```

Note, this part Im not sure about... I think you can just protect everything (*), though you could pick and choose URL patterns like the existing config does... Note, this part has to be above the existing filter mappings in the web.xml so it is outside the other filters.

```
<filter-mapping>  <filter-name>Your Filter</filter-name>  <url-pattern>* </url-pattern>  </filter-mapping>
```

Make a Java class called that, and do this: here is a simple example that gets the username from a cookie (note, you should use encryption or a lookup for a token for this design)

```
/*  * @author mchyzer  * $Id: WebsecFilter.java,v 1.3 2009-11-25 20:01:26 mchyzer Exp $  */ package com.path.whatever; import java.io.IOException; import javax.servlet.Filter; import javax.servlet.FilterChain; import javax.servlet.FilterConfig; import javax.servlet.ServletException; import javax.servlet.ServletRequest; import javax.servlet.ServletResponse; import javax.servlet.http.Cookie; import javax.servlet.http.HttpServletRequest; /**  *  */ public class YourFilter implements Filter {  /**   * @see javax.servlet.Filter#destroy()   */  public void destroy() {  } /**   * @see javax.servlet.Filter#doFilter(javax.servlet.ServletRequest, javax.servlet.ServletResponse, javax.servlet.FilterChain)   */  public void doFilter(ServletRequest req, ServletResponse res, FilterChain filterChain)  throws IOException, ServletException {  HttpServletRequest request  = (HttpServletRequest)req;  String username = cookieValue(request, "someCookieName");  //do some decryption or something?  :)  request.setAttribute("REMOTE_USER", username);  filterChain.doFilter(request, res);  }  /**   * get a cookie value by name, null if not there   * @param httpServletRequest   * @param name   * @return the cookie value or null if not there   */  public static String cookieValue(HttpServletRequest httpServletRequest, String name) {  Cookie cookie = findCookie(httpServletRequest, name);  return cookie == null ? null : cookie.getValue();  }  /**   * find a cookie or null if cant find   * @param httpServletRequest   * @param name   * @return the cookie or null if not found   */  public static Cookie findCookie(HttpServletRequest httpServletRequest, String name) {  //no nulls  if (name != null) {  Cookie[] cookies = httpServletRequest.getCookies();  //go through all cookies and find the cookie by name  int cookiesLength = cookies == null ? 0 : cookies.length;  for (int i=0;i<cookiesLength;i++) {  if (name.equals(cookies[i].getName())) {  return cookies[i];  }  }  }  return null;  }  /**   * @see javax.servlet.Filter#init(javax.servlet.FilterConfig)   */  public void init(FilterConfig arg0) throws ServletException {  } }
```

Previous documentation

Gary already has an example of custom Grouper UI authentication, the Yale CAS auth. I configured the grouper UI to work with Penn's single signon, and I thought another example committed to grouper UI cvs would be useful for people (not because you would use Penn's SSO, but because you might integrate with the UI similarly). Here are the steps I used to get it to work:

1. Add a new eclipse project (mine is in the same as UI in contrib, but you will probably keep in your local source control), this will depend on the grouper-ui project. Here is what my dir looks like

2. There is a build.xml which will be linked from the grouper-ui build.xml. I kept things simple by not having any build config params, you might have a build.properties...

3. You need a custom web.something.xml which will merge into the web.core.xml. In my case, it is web.penn.xml, and it again is simplified, I just protect all .do resources (even public ones, Im ok having everything protected I think), but the URL we publish will probably be a protected page (e.g. grouper /home.do), and not the public one (grouper/).

4. That filter we added to the web.xml will do two things, first it will redirect the user to our single signon login screen if there is no detected user. Second, it wraps the HttpSevletRequest object so that any calls to getRemoteUser() (which is what grouper-ui uses to get the logged in user) will get the user from the token passed in from single signon.

5. The request wrapper caches the user identity, but makes sure the token from single signon doesnt change (e.g. if a user logs in after another user didnt log off). If there is a mismatch, it kills the session and cookies which should allow the user to login again. If the identity is in session cache, use it (since it is expensive to decode a token). If not, then decode the token (if there wasnt a token, the user will be redirected to the login page by the filter above). To decode the token, we use our jar which calls a command line program which has the security associated with it. Then cache the result.

6. If you dont have an additional-build.xml file for grouper-ui (used to link additional build steps without editing the build.xml directly), then you can copy and example, and rename to additional-build.xml. Edit the contents to point to your build.xml for your auth mechanism (e.g. build.xml above). Note this build file must have a webapp and resources target.

7. In your build.properties for grouper-ui, specify where this additional-build.xml is, e.g.

```
#add an additonal build file to incorporate site specific changes
additional.build=additional-build.xml
```

8. For configuration settings, you can either use params in the web.xml (like Gary's example), or a config file (I used media.properties since I dont expect to have to change my settings, but if so I want a way without compiling). Note that my use of media.properties happens before the local.media.properties is considered since the user isnt logged yet, so again, it might not be a good example, up to you. You can look in the request wrapper for my example.

9. Remove the simple auth in the web.core.xml

```
  <security-constraint>
    <web-resource-collection>
      <web-resource-name>Tomcat login</web-resource-name>
      <url-pattern>/login.do</url-pattern>
    </web-resource-collection>
    <auth-constraint>
       <!-- NOTE:  This role is not present in the default users file -->
       <role-name>grouper_user</role-name>
    </auth-constraint>
  </security-constraint>

  <!-- Define the Login Configuration for this Application -->
  <login-config>
    <auth-method>BASIC</auth-method>
    <realm-name>Grouper Application</realm-name>
  </login-config>

  <!-- Security roles referenced by this web application -->
  <security-role>
    <description>
      The role that is required to log in to the Manager Application
    </description>
    <role-name>grouper_user</role-name>
  </security-role>
```

<!-- /* Style Definitions */ table.MsoNormalTable

Unknown macro: {mso-style-name}

-->True, we should add that capability... I assume you have a way so that nefarious users don’t set their cookie value to act as other users... :)

This is what you need to do:

1. in the web.xml declare your filter:

<filter>

  <filter-name>Your Filter</filter-name>

  <filter-class>com.path.whatever.YourFilter</filter-class>

 </filter>

Note, this part Im not sure about... I think you can just protect everything (*), though you could pick and choose URL patterns like the existing config does... Note, this part has to be above the existing filter mappings in the web.xml so it is outside the other filters.

```
<filter-mapping>
    <filter-name>Your Filter</filter-name>
    <url-pattern>* </url-pattern>
 </filter-mapping>
```

2. Make a Java class called that, and do this: here is a simple example

```
/*
 * @author mchyzer
 * $Id: WebsecFilter.java,v 1.3 2009-11-25 20:01:26 mchyzer Exp $
 */
package com.path.whatever;
import java.io.IOException;
import javax.servlet.Filter;
import javax.servlet.FilterChain;
import javax.servlet.FilterConfig;
import javax.servlet.ServletException;
import javax.servlet.ServletRequest;
import javax.servlet.ServletResponse;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServletRequest;
/**
 *
 */
public class YourFilter implements Filter {
  /**
   * @see javax.servlet.Filter#destroy()
   */
  public void destroy()
  Unknown macro: {   }
  /**
   * @see javax.servlet.Filter#doFilter(javax.servlet.ServletRequest, javax.servlet.ServletResponse, javax.servlet.FilterChain)
   */
  public void doFilter(ServletRequest req, ServletResponse res, FilterChain filterChain)
      throws IOException, ServletException
  Unknown macro: {   HttpServletRequest request = (HttpServletRequest)req;   String username = cookieValue(request,
"samaCookieName");   //do some decryption or something?  :)   request.setAttribute("REMOTE_USER", username);   filterChain.doFilter
(request, res);   }
  /**
   * get a cookie value by name, null if not there
   * @param httpServletRequest
   * @param name
```

```
   * @return the cookie value or null if not there
   */
  public static String cookieValue(HttpServletRequest httpServletRequest, String name)
    Unknown macro: {     Cookie cookie = findCookie(httpServletRequest, name);     return cookie == null ? null }
  /**
   * find a cookie or null if cant find
   * @param httpServletRequest
   * @param name
   * @return the cookie or null if not found
   */
  public static Cookie findCookie(HttpServletRequest httpServletRequest, String name) {
    //no nulls
    if (name != null) {
      Cookie[] cookies = httpServletRequest.getCookies();
      //go through all cookies and find the cookie by name
      int cookiesLength = cookies == null ? 0 : cookies.length;
      for (int i=0;i<cookiesLength;i++) {
        if (name.equals(cookies[i].getName()))
    Unknown macro: {          return cookies[i];        }
      }
    }
    return null;
  }

  /**
   * @see javax.servlet.Filter#init(javax.servlet.FilterConfig)
   */
  public void init(FilterConfig arg0) throws ServletException
}
```