Web based Security for GLAST

Easily integrate centralized security into your SLAC web pages and web applications

5/27/2005

Matthew D. Langston
GLAST, Stanford Linear Accelerator Center

Outline

- This week
  - The problem
    - GLAST web security requirements
    - Goals
  - The `glast-profile` project
  - **How to use**
    - Secure JSP web pages now
      - Parts of JSP pages
      - Entire directories of web pages
- Next week
  - Access Control Lists (ACLs – remember VMS?)
  - Restricting data from Oracle queries
    - Filter rows based on column data
  - **How it works**
    - Secure java things (objects, methods arguments, return values)
    - Java code
- If there is interest
  - Calling and using the security framework from Perl and Python

GLAST Web Security Requirements

- We want a single security framework
  - We currently have three (that I know of)
  - Require
    - Login
    - Logout
    - Lockdown
    - Auditing
    - Permission assignments
- Can’t be complicated
  - No programming
    - Declarative security (configuration files)
    - `if (role == “ADMINISTRATOR”) { … }` **NO!**
  - Eeasy to use from Dreamweaver
- Must be complete
  - Applicable to
    - Web pages (sections of JSP pages as well as entire directories of pages)
    - Java objects, methods, arguments, return values
    - Oracle queries, etc.

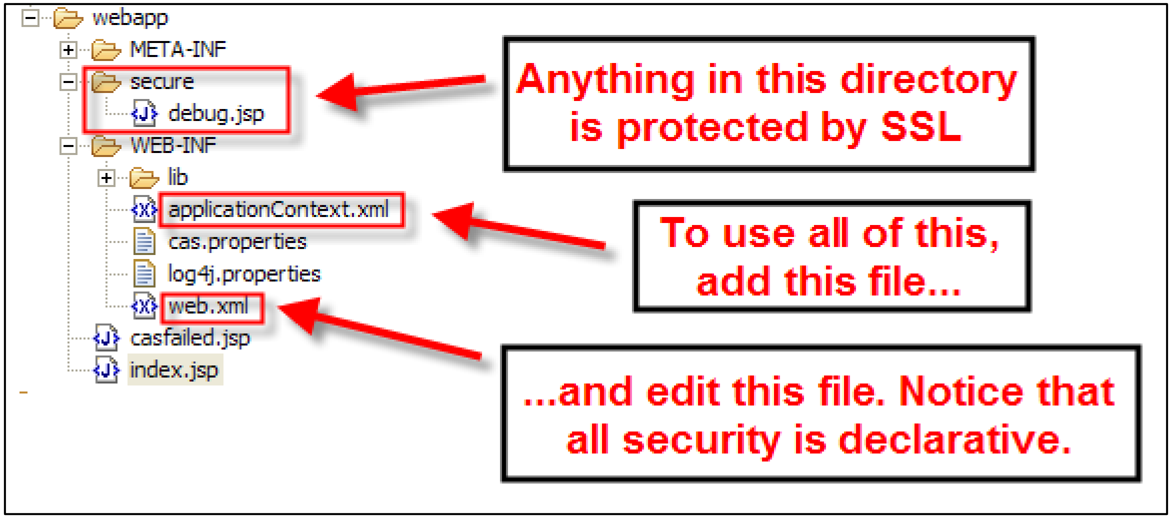
Goals

- Integrate with SLAC accounts
  - But not limited to SLAC accounts (JIRA/Confluence)
- Central repository of GLAST user profile information
  - Username and password at a minimum
  - Other user details
    - email address (“Reply to this email” for verifying accounts)
    - ICQ, AOL, MSN, etc… (Pat’s database?)
    - Full name, home institution, address etc. (Karen’s database?)
- Web-based
    - Easy to use from JSP pages
    - Developers can register their applications and create roles for them
    - Users can edit (only) their information
      - System uses its own framework to protect itself
- Provide login for JIRA and Confluence
- Integrate with Perl and Python

The `glast-profile` project

- Java project
  - `.jar` file for other Java projects to use (web-based or not)
  - `.war` file for web front-end
    - Register applications
    - Define roles
    - Edit user information
- CVS
  - Module name is `glast-profile`
- Currently backed by MySQL database
  - Oracle just a matter of Configuration
- Maven, Spring, Acegi Security, Hibernate, CAS

How to Use: Directory Layout



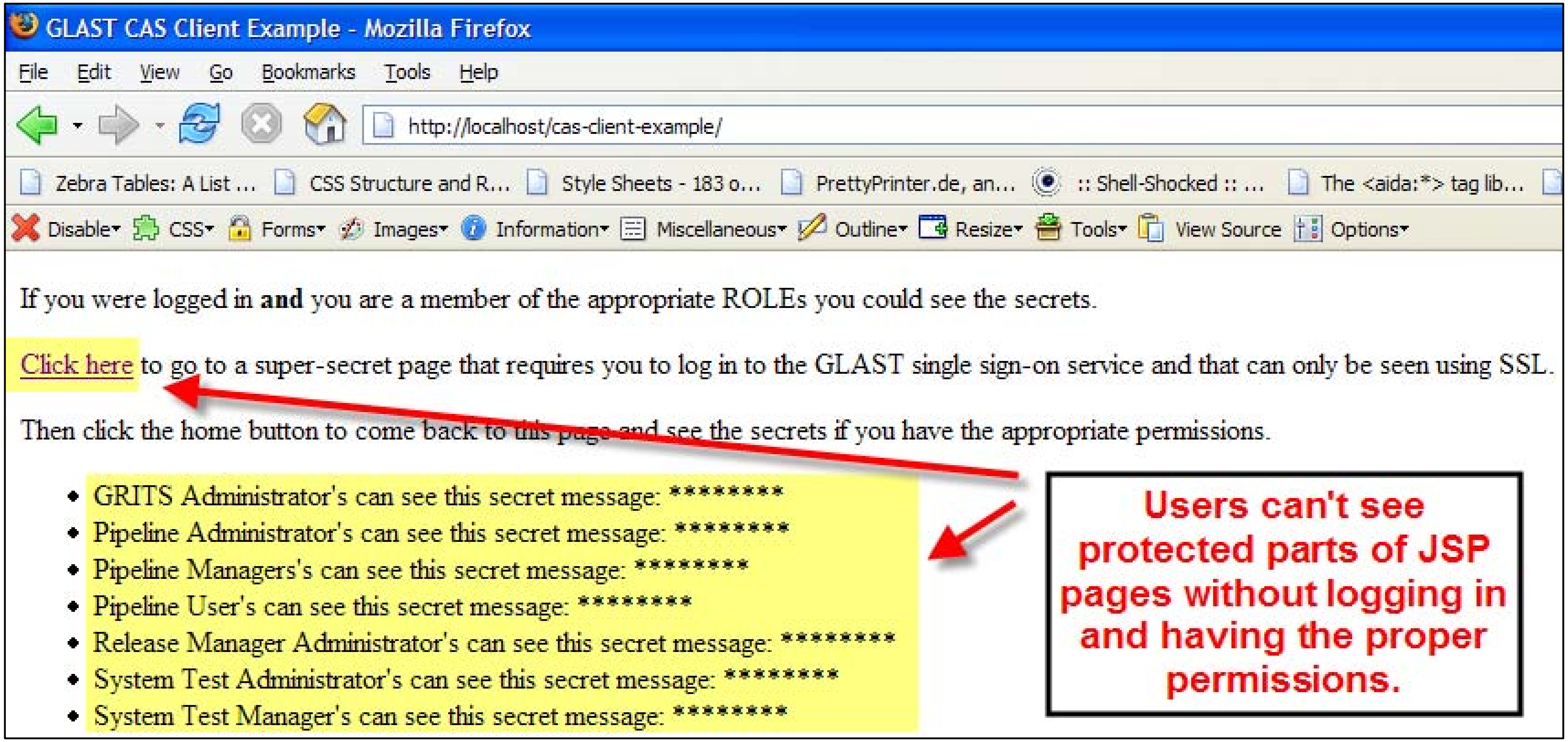
```
webapp
  META-INF
  secure
    debug.jsp
  WEB-INF
    lib
    applicationContext.xml
    cas.properties
    log4j.properties
    web.xml
  casfailed.jsp
  index.jsp
```

Anything in this directory is protected by SSL

To use all of this, add this file...

...and edit this file. Notice that all security is declarative.

How to use



GLAST CAS Client Example - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://localhost/cas-client-example/
Zebra Tables: A List ... CSS Structure and R... Style Sheets - 183 o... PrettyPrinter.de, an... :: Shell-Shocked :: ... The <aida:*> tag lib...
Disable CSS Forms Images Information Miscellaneous Outline Resize Tools View Source Options

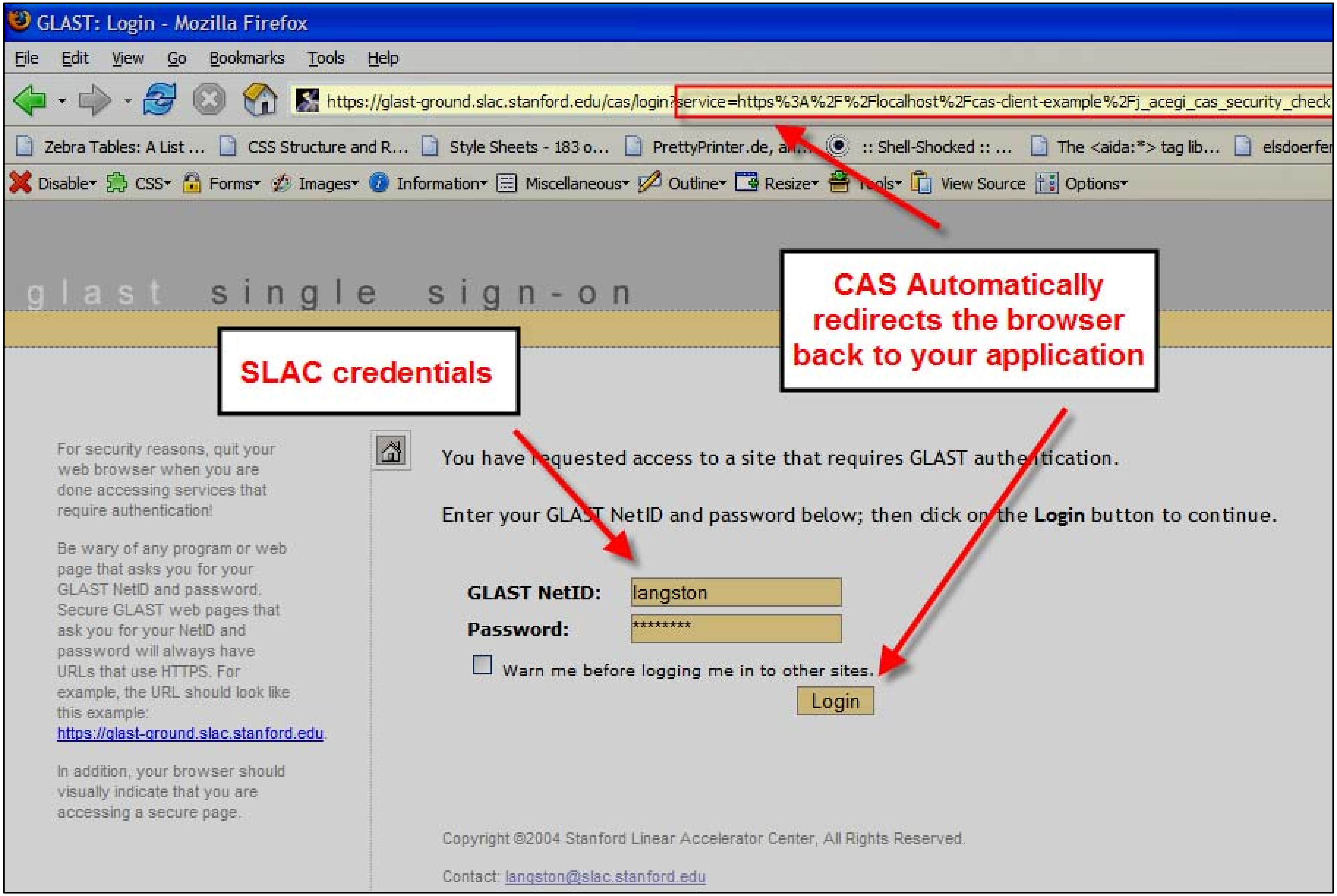
If you were logged in **and** you are a member of the appropriate ROLEs you could see the secrets.

Click here to go to a super-secret page that requires you to log in to the GLAST single sign-on service and that can only be seen using SSL.

Then click the home button to come back to this page and see the secrets if you have the appropriate permissions.

- GRITS Administrator's can see this secret message: *********
- Pipeline Administrator's can see this secret message: *********
- Pipeline Managers's can see this secret message: *********
- Pipeline User's can see this secret message: *********
- Release Manager Administrator's can see this secret message: *********
- System Test Administrator's can see this secret message: *********
- System Test Manager's can see this secret message: *********

Users can't see protected parts of JSP pages without logging in and having the proper permissions.



https://glast-ground.slac.stanford.edu/cas/login?service=https%3A%2F%2Flocalhost%2Fcas-client-example%2Fj_acegi_cas_security_check

CAS Automatically redirects the browser back to your application

SLAC credentials

For security reasons, quit your web browser when you are done accessing services that require authentication!

Be wary of any program or web page that asks you for your GLAST NetID and password. Secure GLAST web pages that ask you for your NetID and password will always have URLs that use HTTPS. For example, the URL should look like this example: https://glast-ground.slac.stanford.edu

In addition, your browser should visually indicate that you are accessing a secure page.

You have requested access to a site that requires GLAST authentication.

Enter your GLAST NetID and password below; then click on the **Login** button to continue.

GLAST NetID: langston

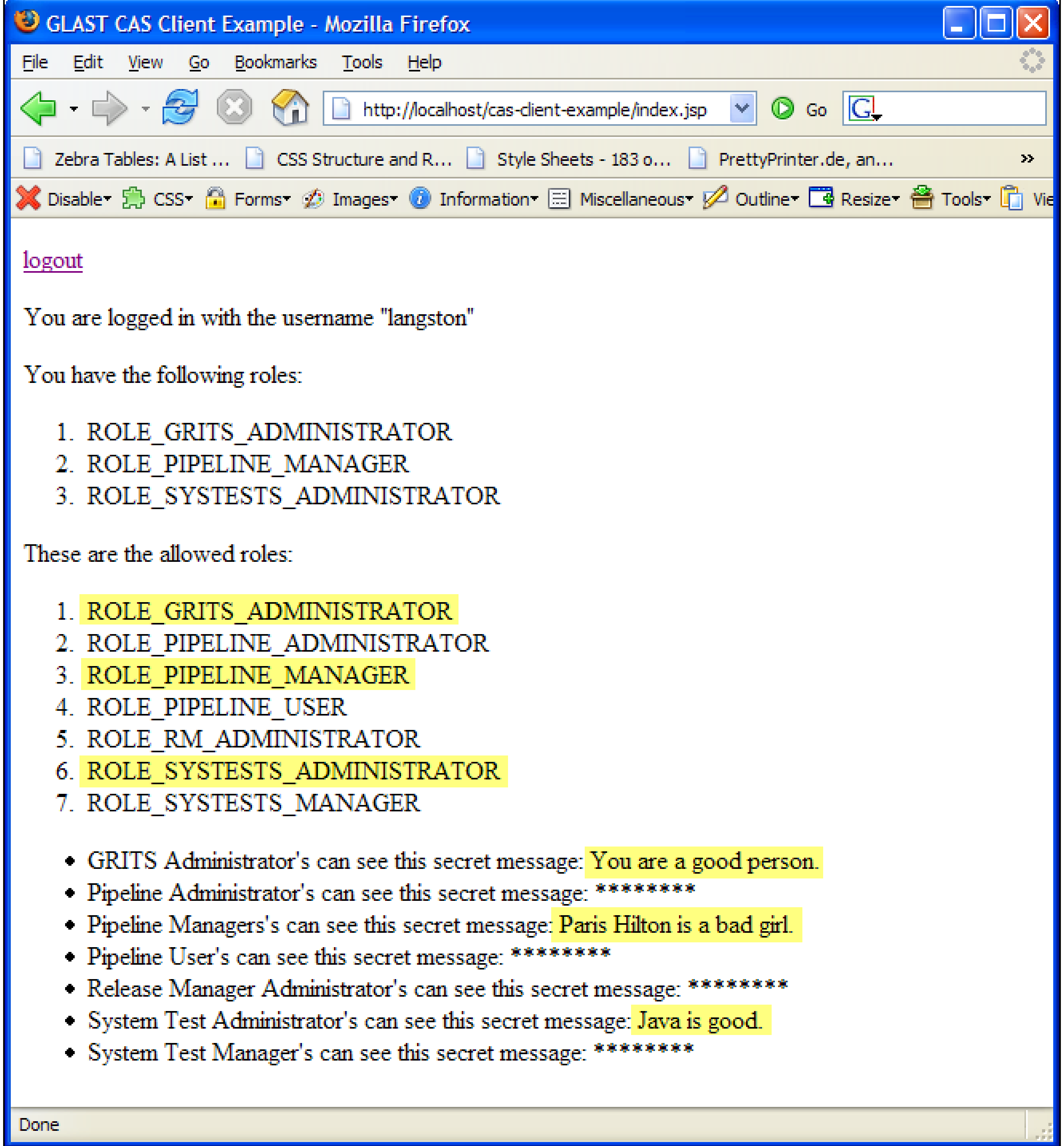
Password: ********

□ Warn me before logging me in to other sites.

Login

Copyright ©2004 Stanford Linear Accelerator Center, All Rights Reserved.

Contact: langston@slac.stanford.edu



logout

You are logged in with the username "langston"

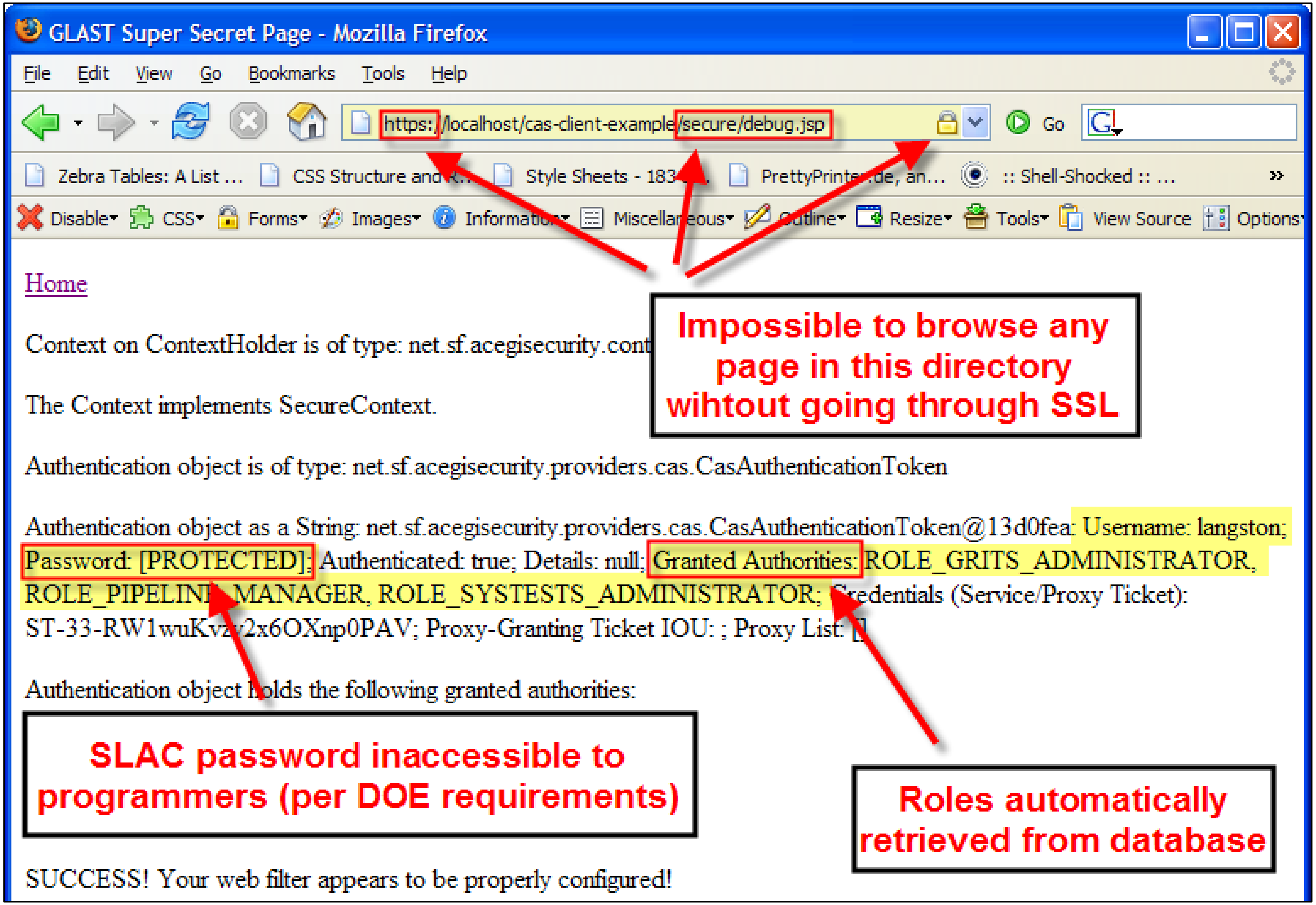
You have the following roles:

1. ROLE_GRITS_ADMINISTRATOR
2. ROLE_PIPELINE_MANAGER
3. ROLE_SYSTESTS_ADMINISTRATOR

These are the allowed roles:

1. ROLE_GRITS_ADMINISTRATOR
2. ROLE_PIPELINE_ADMINISTRATOR
3. ROLE_PIPELINE_MANAGER
4. ROLE_PIPELINE_USER
5. ROLE_RM_ADMINISTRATOR
6. ROLE_SYSTESTS_ADMINISTRATOR
7. ROLE_SYSTESTS_MANAGER

- GRITS Administrator's can see this secret message: You are a good person.
- Pipeline Administrator's can see this secret message: ********
- Pipeline Managers's can see this secret message: Paris Hilton is a bad girl.
- Pipeline User's can see this secret message: ********
- Release Manager Administrator's can see this secret message: ********
- System Test Administrator's can see this secret message: Java is good.
- System Test Manager's can see this secret message: ********


GLAST Super Secret Page - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
https://localhost/cas-client-example/secure/debug.jsp
Go
Zebra Tables: A List ... CSS Structure and R... Style Sheets - 183... PrettyPrinter.de, an... :: Shell-Shocked :: ...
Disable CSS Forms Images Information Miscellaneous Outline Resize Tools View Source Options

Home

Context on ContextHolder is of type: net.sf.acegisecurity.cont

The Context implements SecureContext.

Authentication object is of type: net.sf.acegisecurity.providers.cas.CasAuthenticationToken

Authentication object as a String: net.sf.acegisecurity.providers.cas.CasAuthenticationToken@13d0fea: Username: langston; Password: [PROTECTED]; Authenticated: true; Details: null; Granted Authorities: ROLE_GRITS_ADMINISTRATOR, ROLE_PIPELINE_MANAGER, ROLE_SYSTESTS_ADMINISTRATOR; Credentials (Service/Proxy Ticket): ST-33-RW1wuKvzy2x6OXnp0PAV; Proxy-Granting Ticket IOU: ; Proxy List: []

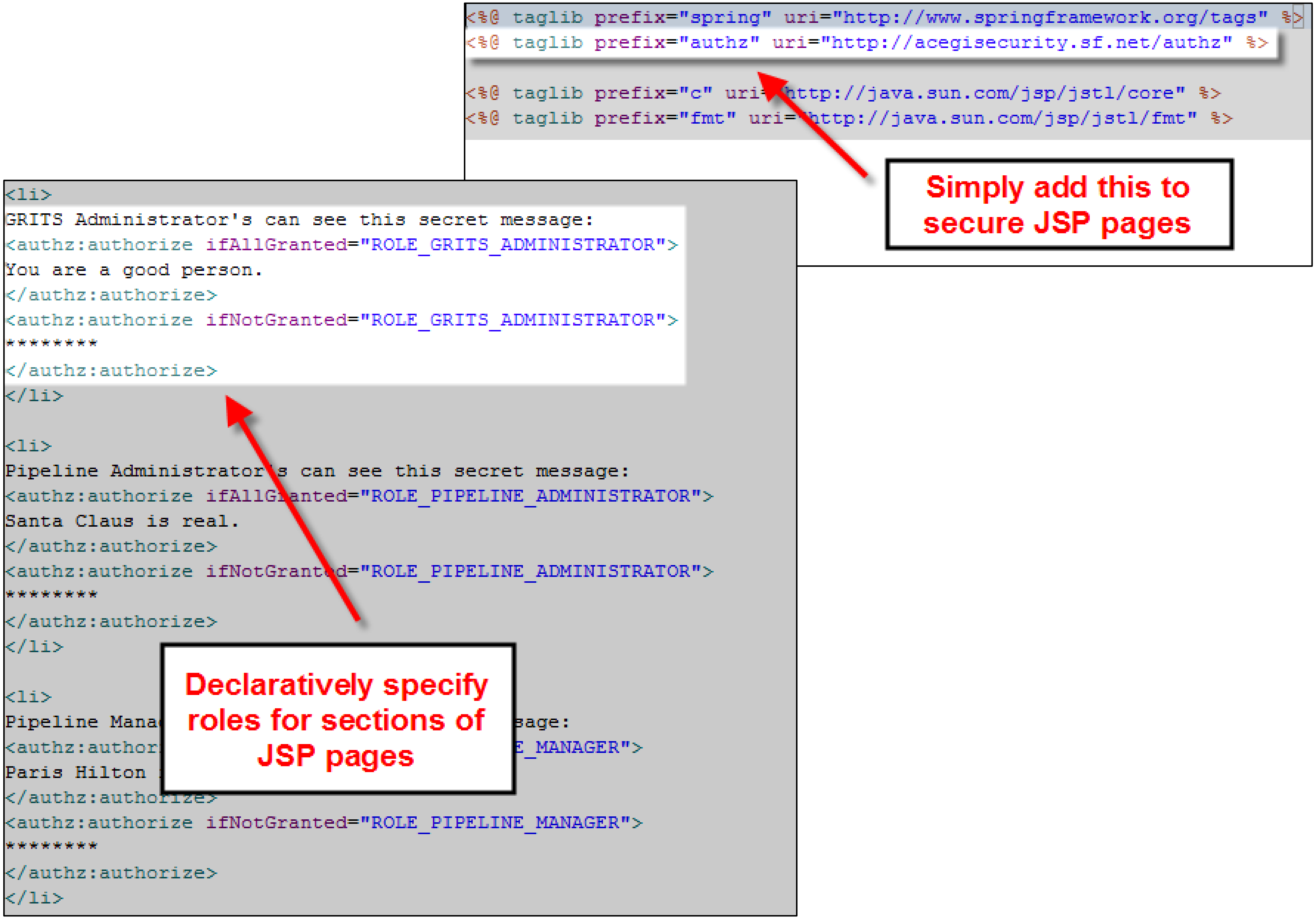
Authentication object holds the following granted authorities:

SUCCESS! Your web filter appears to be properly configured!

Impossible to browse any page in this directory wihtout going through SSL

SLAC password inaccessible to programmers (per DOE requirements)

Roles automatically retrieved from database



```
<%@ taglib prefix="spring" uri="http://www.springframework.org/tags" %>
<%@ taglib prefix="authz" uri="http://acegisecurity.sf.net/authz" %>

<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
```

Simply add this to secure JSP pages

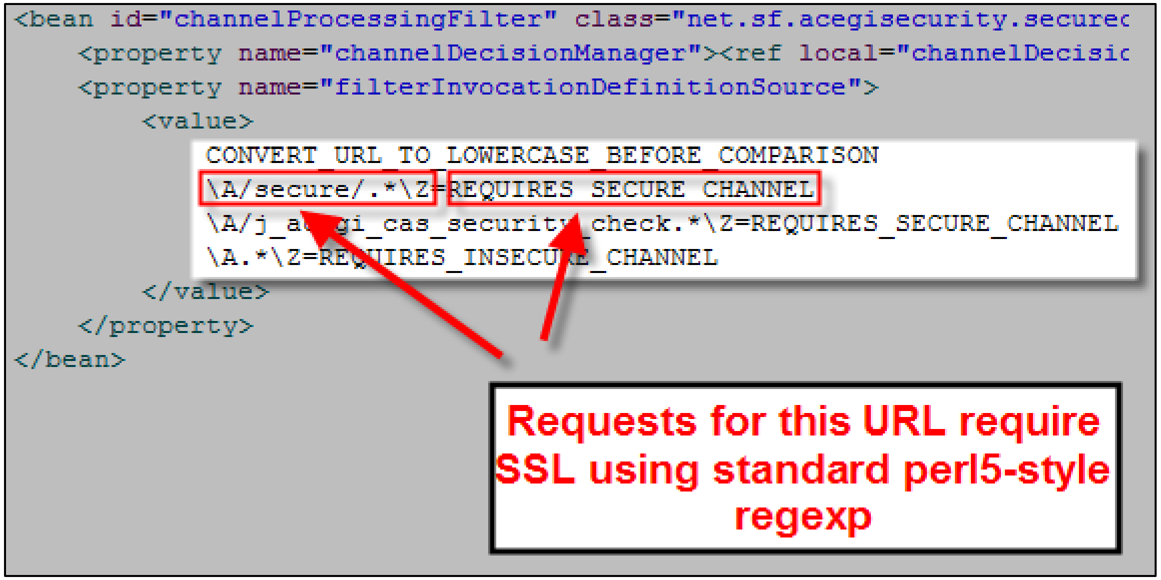
```
<li>
GRITS Administrator's can see this secret message:
<authz:authorize ifAllGranted="ROLE_GRITS_ADMINISTRATOR">
You are a good person.
</authz:authorize>
<authz:authorize ifNotGranted="ROLE_GRITS_ADMINISTRATOR">
********
</authz:authorize>
</li>

<li>
Pipeline Administrator's can see this secret message:
<authz:authorize ifAllGranted="ROLE_PIPELINE_ADMINISTRATOR">
Santa Claus is real.
</authz:authorize>
<authz:authorize ifNotGranted="ROLE_PIPELINE_ADMINISTRATOR">
********
</authz:authorize>
</li>

<li>
Pipeline Mana                         sage:
<authz:author                         E_MANAGER">
Paris Hilton
</authz:authorize>
<authz:authorize ifNotGranted="ROLE_PIPELINE_MANAGER">
********
</authz:authorize>
</li>
```

Declaratively specify roles for sections of JSP pages

Protecting Directories



```
<bean id="channelProcessingFilter" class="net.sf.acegisecurity.securec
    <property name="channelDecisionManager"><ref local="channelDecisic
    <property name="filterInvocationDefinitionSource">
        <value>
            CONVERT_URL_TO_LOWERCASE_BEFORE_COMPARISON
            \A/secure/.*\Z=REQUIRES_SECURE_CHANNEL
            \A/j_acegi_cas_security_check.*\Z=REQUIRES_SECURE_CHANNEL
            \A.*\Z=REQUIRES_INSECURE_CHANNEL
        </value>
    </property>
</bean>
```

Requests for this URL require SSL using standard perl5-style regexp

Conclusion