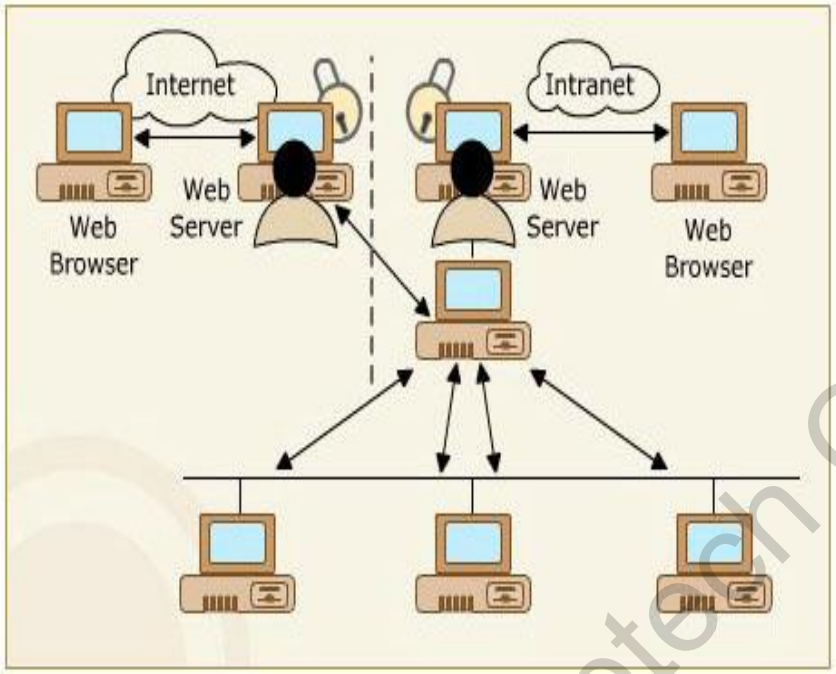
Session: 15

Securing Java Web Applications

Objectives

- Explain the necessity for and features of securing Web applications
- Describe Jakarta EE Security API
- Describe the HTTP basic, digest, client, and form-based authentication method of ensuring security
- Explain how to configure users in Tomcat
- Explain how to specify authentication mechanisms using web.xml
- Describe the seven steps to implement declarative security
- Explain the concept and five steps to implement programmatic security
- Describe the HttpServletRequest methods for identifying users
- Explain the use of SSL certification

Introduction

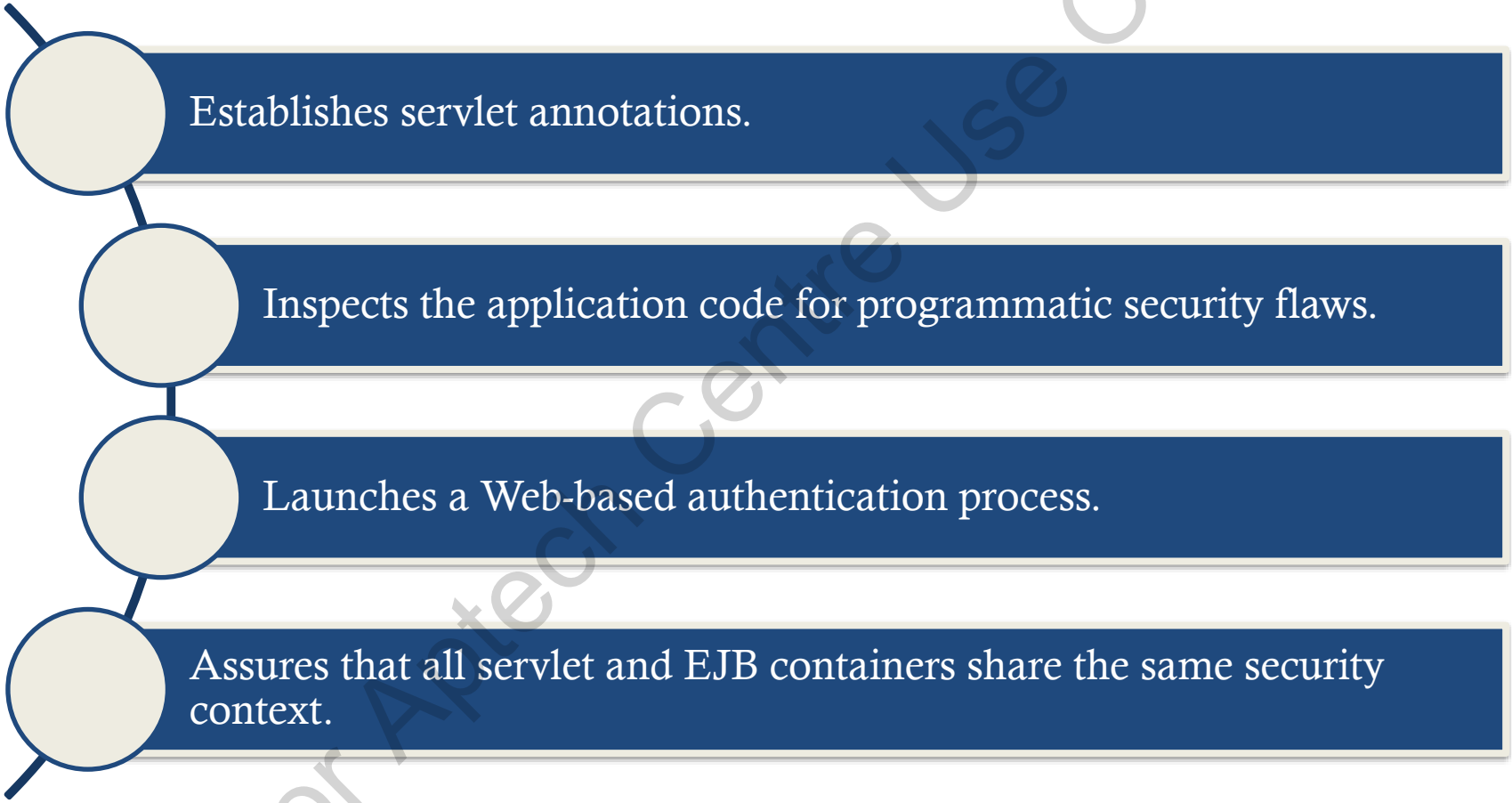


Unauthorized Access on the Internet

- HTTP basic authentication method
- HTTP digest authentication method
- Form-based authentication method
- HTTPS client authentication method

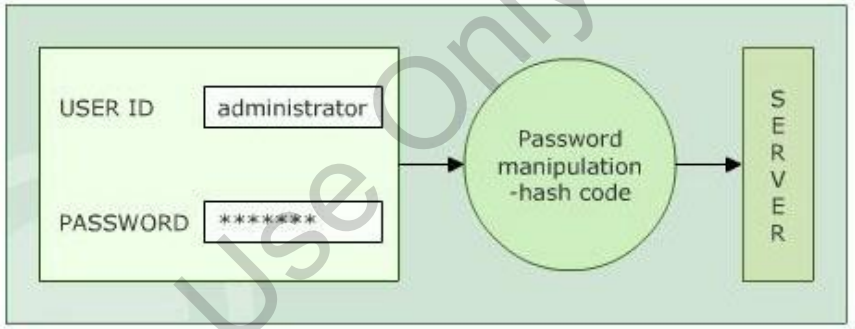
Pillars of Security/Security Mechanisms

Jakarta EE Security API

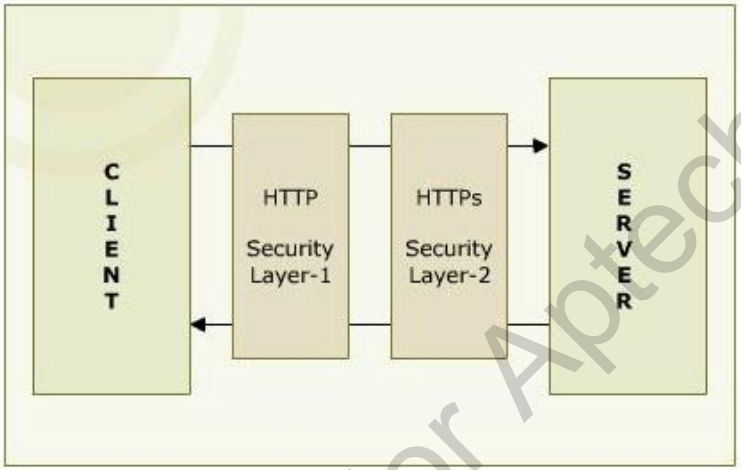


Authentication Mechanisms

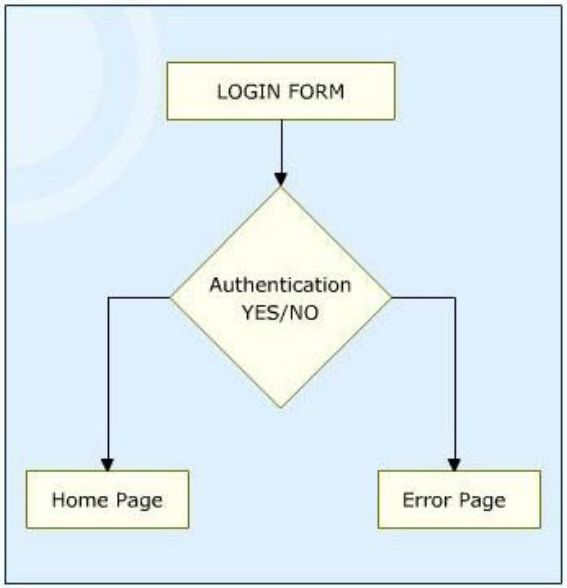
HTTP Basic Authentication



HTTP Digest Authentication



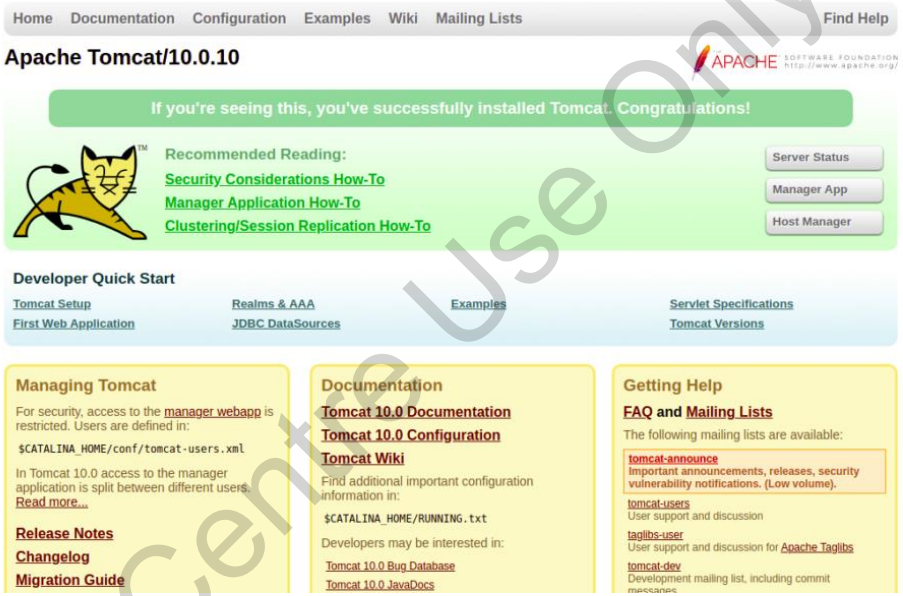
HTTP Client Authentication



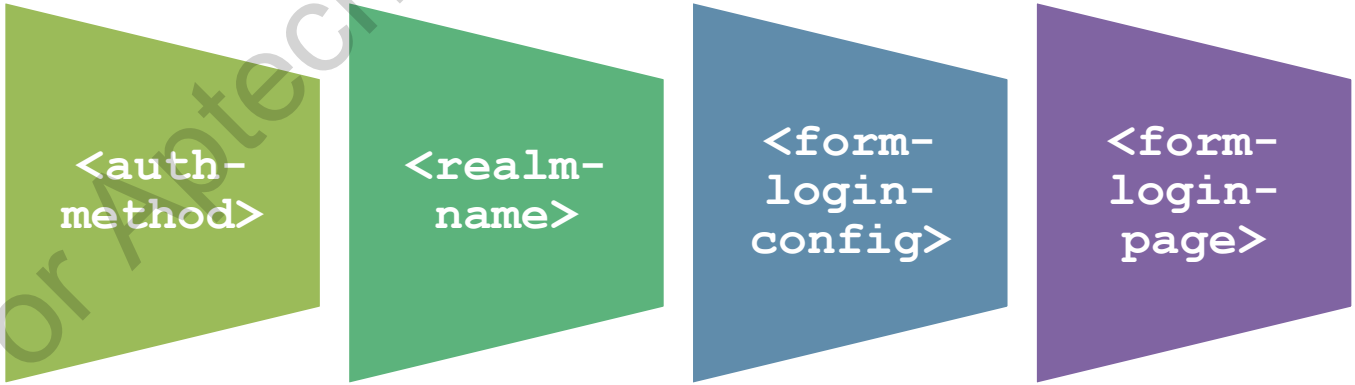
Form-based Authentication

Authentication and web.xml

Home Page of Tomcat Web Server



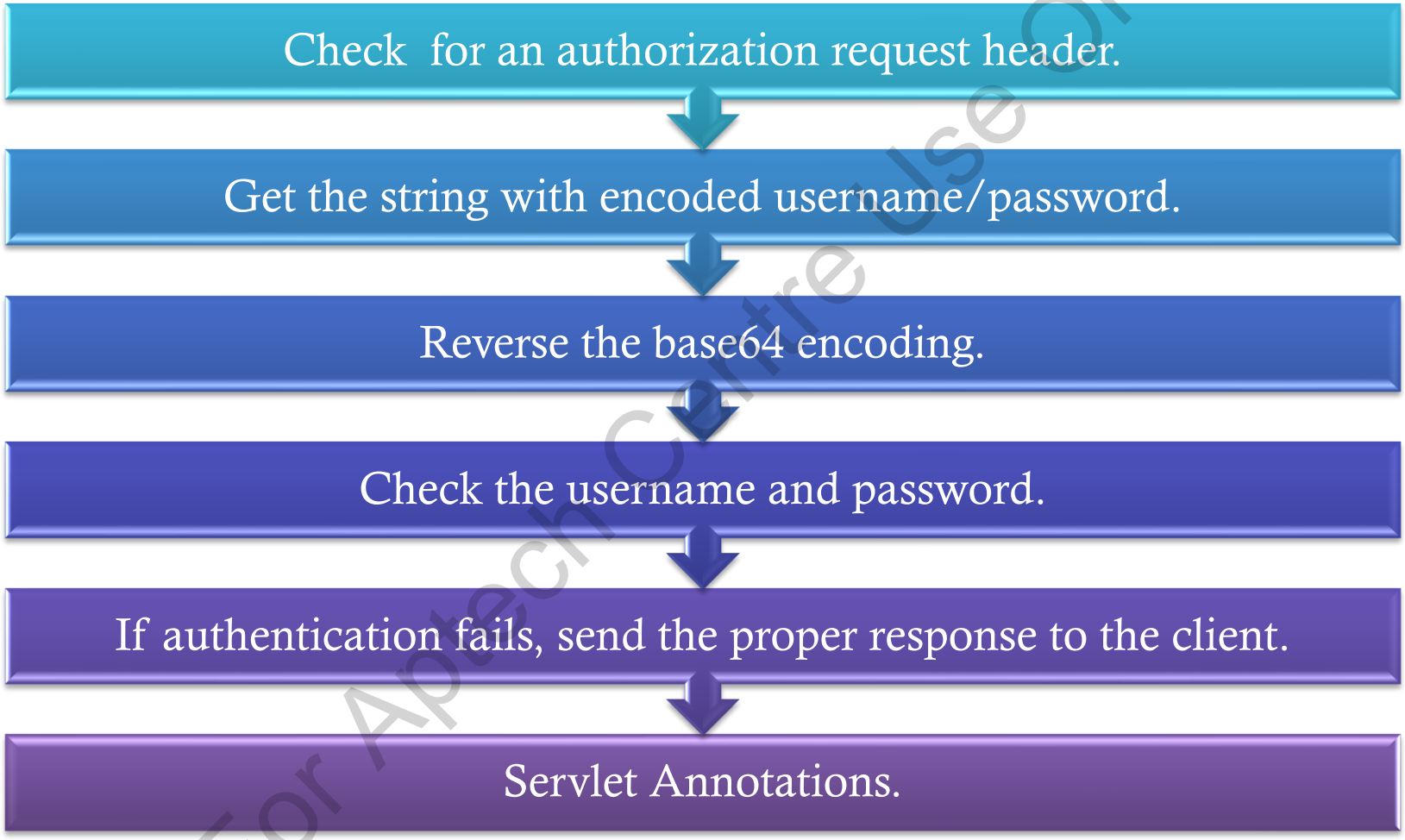
Authentication Mechanisms



Declarative Security

Advantages	• Ignores the programming constraints. • Requires little change in security model. • Easily maintainable.
Disadvantages	• Provides access to all or denies. • Gives access only if the password matches. • Cannot use both form-based and basic authentication for the same page.

Programmatic Security



ServerAuthModule Interface

- Initialize()
- getSupportedMessageType
- ValidateRequest()
- SecureResponse()
- CleanSubject()

```
1  package com.authentication;
2
3  import java.util.Map;
12
13 public class AuthExample implements ServerAuthModule {
14
15     @Override
16     public void initialize(MessagePolicy arg0, MessagePolicy arg1,
17             CallbackHandler arg2, Map arg3) throws AuthException {
18     }
19
20     @Override
21     public AuthStatus validateRequest(MessageInfo arg0, Subject arg1,
22             Subject arg2) throws AuthException {
23         return null;
24     }
25     @Override
26     public void cleanSubject(MessageInfo arg0, Subject arg1)
27             throws AuthException {
28
29     }
30     @Override
31     public Class[] getSupportedMessageTypes() {
32         return null;
33     }
34     @Override
35     public AuthStatus secureResponse(MessageInfo arg0, Subject arg1)
36             throws AuthException {
37         return null;
38     }
39 }
40
```

Authentication Module

Summary

- A Web application is an application, which is accessed with the help of a Web browser.
- The reason of for the popularity of Web applications is the ability to maintain it them without changing the client computers. When you access the Web, hackers can get your information. So, to keep your information secret, it is necessary to secure Web applications.
- There are four authentication Mechanisms available. These are HTTP Basic Authentication, HTTP Digest Authentication, HTTPS Client Authentication, and Form-Based Authentication.
- To configure a user in Tomcat, first include the Tomcat 6.0 from Apache Software Foundation. When the browser loads a resource, which is secured by web.xml file, the browser responds in two ways.
- The browser challenges the user if you are using basic authentication or forwards the login page if you are using form-based authentication.
- The declarative security provides security to a resource with the help of server configuration.
- Programmatic security authenticates the users and grants access to the users.