opentext™

# kisa49

5/31/24

ikkb7

# Executive Summary

## Issues Overview

On May 31, 2024, a source code review was performed over the WebGoat code base. 964 files, 195,117 LOC (Executable) were scanned and reviewed for defects that could lead to potential security vulnerabilities. A total of 847 reviewed findings were uncovered during the analysis.

| Issues by 49 | |
|---|---|
| <none> | 223 |
| 07.02. API | 3 |
| 05.05. | 2 |
| 04.03. | 11 |
| 04.01. , 06.02. | 50 |
| 04.01. | 31 |
| 02.15. | 1 |
| 02.11. | 2 |
| 02.09. | 2 |
| 02.08. | 19 |
| 02.06. | 52 |
| 02.05. | 17 |
| 02.04. | 1 |
| 02.03. | 15 |
| 01.15. , 07.01. DNS lookup | 1 |
| 01.12. | 2 |
| 01.11. | 298 |
| 01.08. XML | 3 |
| 01.07. URL | 16 |
| 01.06. | 16 |
| 01.05. | 1 |
| 01.04. | 40 |
| 01.03. | 41 |

## Recommendations and Conclusions

The Issues Category section provides Fortify recommendations for addressing issues at a generic level. The recommendations for specific fixes can be extrapolated from those generic recommendations by the development group.

ot Copyright 2024 Open Text.

# Project Summary

## Code Base Summary

Code location: C:/Users/ikkb7/WebGoat
Number of Files: 964
Lines of Code: 195117
Build Label: <No Build Label>

## Scan Information

Scan time: 23:09
SCA Engine version: 24.2.0.0150
Machine Name: kbkim-surface4
Username running scan: ikkb7

## Results Certification

Results Certification Valid

Details:

Results Signature:

SCA Analysis Results has Valid signature

Rules Signature:

There were no custom rules used in this scan

## Attack Surface

Attack Surface:
Command Line Arguments:
null.MavenWrapperDownloader.main
org.owasp.webgoat.lessons.challenges.challenge7.MD5.main
org.owasp.webgoat.lessons.challenges.challenge7.PasswordResetLink.main
org.owasp.webgoat.server.StartWebGoat.main

Environment Variables:
null.null.null
java.lang.System.getenv
org.owasp.webgoat.container.WebGoat.pluginTargetDirectory
org.owasp.webgoat.lessons.passwordreset.ResetLinkAssignmentForgotPassword.ResetLinkAssignmentForgotPassword
org.owasp.webgoat.lessons.passwordreset.SimpleMailAssignment.SimpleMailAssignment
org.owasp.webgoat.lessons.pathtraversal.ProfileUpload.ProfileUpload
org.owasp.webgoat.lessons.pathtraversal.ProfileUploadFix.ProfileUploadFix
org.owasp.webgoat.lessons.pathtraversal.ProfileUploadRemoveUserInput.ProfileUploadRemoveUserInput
org.owasp.webgoat.lessons.pathtraversal.ProfileUploadRetrieval.ProfileUploadRetrieval
org.owasp.webgoat.lessons.pathtraversal.ProfileZipSlip.ProfileZipSlip
org.owasp.webgoat.lessons.webwolfintroduction.MailAssignment.MailAssignment
org.owasp.webgoat.lessons.xxe.BlindSendFileAssignment.BlindSendFileAssignment
org.springframework.core.env.PropertyResolver.getProperty

ot Copyright 2024 Open Text.

File System:
java.io.FileInputStream.FileInputStream
java.io.FileInputStream.FileInputStream
java.nio.file.Files.readAllBytes
java.util.zip.ZipFile.entries
org.springframework.core.io.ResourceLoader.getResource
org.springframework.util.FileCopyUtils.copyToByteArray

Network:
org.springframework.core.io.ResourceLoader.getResource

Private Information:
null.null.null
null.null.null
java.lang.System.getenv
java.util.Properties.getProperty
org.springframework.boot.autoconfigure.jdbc.DataSourceProperties.getPassword
org.springframework.core.env.PropertyResolver.getProperty

Java Properties:
java.lang.System.getProperty
java.util.Properties.load

Serialized Data:
null.null.null
java.io.ObjectInputStream.readObject

Stream:
java.io.InputStream.read
org.springframework.util.FileCopyUtils.copyToByteArray

System Information:
null.null.null
java.io.File.listFiles
java.lang.System.getProperty
java.lang.System.getProperty
java.lang.System.getProperty
java.lang.Throwable.getMessage
java.net.InetAddress.getLocalHost
org.apache.commons.lang3.exception.ExceptionUtils.getStackTrace
org.springframework.core.NestedRuntimeException.getMessage

Web:
null.null.null
null.null.prompt
null.~JS_Generic.val
javax.servlet.ServletRequest.getContentType
javax.servlet.http.HttpServletRequest.getMethod
org.springframework.web.servlet.LocaleResolver.resolveLocale

ot Copyright 2024 Open Text.

Web Service:
org.springframework.core.io.ResourceLoader.getResource

## Filter Set Summary

Current Enabled Filter Set:
Security Auditor View

Filter Set Details:

Folder Filters:
If [fortify priority order] contains critical Then set folder to Critical
If [fortify priority order] contains high Then set folder to High
If [fortify priority order] contains medium Then set folder to Medium
If [fortify priority order] contains low Then set folder to Low

## Audit Guide Summary

Audit guide not enabled

Copyright 2024 Open Text.

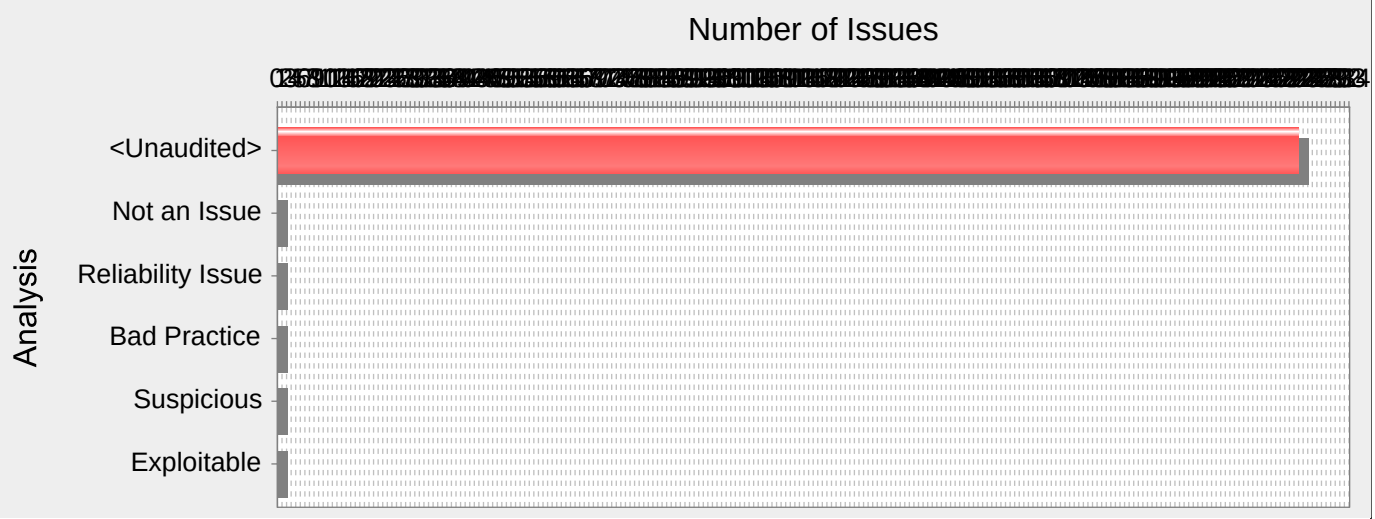
# Results Outline

## Overall number of results

The scan found 847 issues.

## Vulnerability Examples by Category

### 49: <none> (223 Issues)




**Abstract:**

println() .

**Explanation:**

1: Java .

```
public class MyClass
...
System.out.println("hello world");
...
}
```

Java System.out.println() .

. , , ID, , . .

" " . , System.out System.err .

**Recommendations:**

System.out System.err Java .

2: , Example 1 “hello world” log4j .

```
import org.apache.log4j.Logger;
import org.apache.log4j.BasicConfigurator;

public class MyClass {
private final static Logger logger =
Logger.getLogger(MyClass.class);

...
BasicConfigurator.configure();
logger.info("hello world");
...
}
```

**Tips:**

1. Fortify Static Code Analyzer System.out System.err main() (main() ) Poor Logging Practice: Use of a System Output Stream .

Copyright 2024 Open Text.

## MissingFunctionACUsers.java, line 90 (Mass Assignment: Insecure Binder Configuration)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | API Abuse | | |
| Abstract: | HTTP . | | |

Sink: MissingFunctionACUsers.java:90 Function: addUser()

```
88      @PostMapping(path = {"access-control/users", "access-control/users-admin-fix"}, consumes = "application/json", produces = "application/json")
89      @ResponseBody
90      public User addUser(@RequestBody User newUser) {
91          try {
92              userRepository.save(newUser);
```

## AuthBypass.html, line 37 (Hidden Field)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | AuthBypass.html 37 . | | |

Sink: AuthBypass.html:37

```
35                      <input type="hidden" name="jsEnabled" value="1" />
36                      <input type="hidden" name="verifyMethod" value="SEC_QUESTIONS" />
37                      <input type="hidden" name="userId" value="12309746" />
38
39                      <input name="submit" value="Submit" type="submit"/>
```

## LabelAndHintIntegrationTest.java, line 134 (Poor Logging Practice: Use of a System Output Stream)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | println() . | | |

Sink: LabelAndHintIntegrationTest.java:134 FunctionCall: println()

```
132                 }
133                 if (!jsonPath.getString(ESCAPE_JSON_PATH_CHAR+key+ESCAPE_JSON_PATH_CHAR).equals(propsLang.get(key))) {
134                     System.out.println("key: " + key + " in (" +lang+") has incorrect translation in label service");
135 System.out.println("actual:"+jsonPath.getString(ESCAPE_JSON_PATH_CHAR+key+ESCAPE_JSON_PATH_CHAR));
136                     System.out.println("expected: "+propsLang.getProperty(key));
```

## SqlInjectionLesson8.java, line 66 (SQL Injection)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | SqlInjectionLesson8.java 66 injectableQueryConfidentiality() SQL . SQL . | | |

Source: SqlInjectionLesson8.java:54 completed(0)

```
52      @PostMapping("/SqlInjection/attack8")
53      @ResponseBody
54      public AttackResult completed(@RequestParam String name, @RequestParam String auth_tan) {
55          return injectableQueryConfidentiality(name, auth_tan);
56      }
```

Sink: SqlInjectionLesson8.java:66 java.sql.Statement.executeQuery()

```
64                  Statement statement = connection.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_UPDATABLE);
65                  log(connection, query);
66                  ResultSet results = statement.executeQuery(query);
67
68                  if (results.getStatement() != null) {
```

## webwolf-login.html, line 37 (Privacy Violation: Autocomplete)

| Fortify Priority: | High | Folder | High |
|---|---|---|---|
| Kingdom: | Security Features | | |
| Abstract: | webwolf-login.html 37 . | | |

Sink: webwolf-login.html:37

```
35                          </div>
36                          <div class="form-group">
37                              <input type="password" name="password" id="password"
class="form-control input-lg"
38                                     placeholder="Password WebGoat" required="true"/>
39                          </div>
```

## AuthBypass.html, line 36 (Hidden Field)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | AuthBypass.html 36 . | | |

Sink: AuthBypass.html:36

```
34
35                      <input type="hidden" name="jsEnabled" value="1" />
36                      <input type="hidden" name="verifyMethod" value="SEC_QUESTIONS" />
37                      <input type="hidden" name="userId" value="12309746" />
```

## HtmlTampering.html, line 132 (Hidden Field)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | HtmlTampering.html 132 . | | |

Sink: HtmlTampering.html:132

```
130                                     </div>
131                                 </td>
132                                 <input id="Total" name="Total" type="HIDDEN"
value="2999.99"/>
133                             </tr>
134                             </tbody>
```

## ResetLinkAssignment.java, line 92 (Password Management: Password in Comment)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Security Features | | |
| Abstract: | . | | |

Sink: ResetLinkAssignment.java:92 Comment()

```
90              model.addAttribute("form", form);
91              modelAndView.addObject("form", form);
92              modelAndView.setViewName("password_reset"); //Display html page for
changing password
93          } else {
94           modelAndView.setViewName("password_link_not_found");
```

## AuthBypass.html, line 35 (Hidden Field)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | AuthBypass.html 35 . | | |

Sink: AuthBypass.html:35

```
33                      <input name="secQuestion1" value="" type="TEXT" /><br/><br />
34
35                      <input type="hidden" name="jsEnabled" value="1" />
36                      <input type="hidden" name="verifyMethod" value="SEC_QUESTIONS" />
37                      <input type="hidden" name="userId" value="12309746" />
```

## WebSecurityConfig.java, line 72 (Password Management: Password in Comment)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|

ot Copyright 2024 Open Text.

| | |
|---|---|
| Kingdom: | Security Features |
| Abstract: | . |
| Sink: | WebSecurityConfig.java:72 Comment() |

```
70      @Autowired
71      public void configureGlobal(AuthenticationManagerBuilder auth) throws Exception {
72          auth.userDetailsService(userDetailsService);
//.passwordEncoder(bCryptPasswordEncoder());
73      }
```

## EncodingAssignment.java, line 56 (Privacy Violation)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Security Features | | |
| Abstract: | EncodingAssignment.java getBasicAuth() . . | | |
| Source: | EncodingAssignment.java:42 Read password() | | |

```
40
41   public static String getBasicAuth(String username, String password) {
42       return
Base64.getEncoder().encodeToString(username.concat(":").concat(password).getBytes());
43      }
44
```

| | |
|---|---|
| Sink: | EncodingAssignment.java:56 Return() |

```
54     request.getSession().setAttribute("basicAuth", basicAuth);
55    }
56    return "Authorization: Basic ".concat(basicAuth);
57      }
58
```

## SSRF.html, line 40 (Hidden Field)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | SSRF.html 40 . | | |
| Sink: | SSRF.html:40 | | |

```
38                  <table>
39                      <tr>
40                          <td><input type="hidden" id="url2" name="url"
value="images/cat.png"/></td>
41
42                          <td><input
```

## webwolfPasswordReset.html, line 14 (Hidden Field)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | webwolfPasswordReset.html 14 . | | |
| Sink: | webwolfPasswordReset.html:14 | | |

```
12              <form role="form" method="GET" th:action="${webwolfUrl}">
13                  <h2 class="sign_up_title">Reset your password</h2>
14                  <input type="hidden" name="uniqueCode" th:value="${uniqueCode}"/>
15                  <div class="form-group">
16                      <label for="password" class="control-label">Password</label>
```

## SqlInjectionAdvanced.html, line 89 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Security Features | | |
| Abstract: | SqlInjectionAdvanced.html 89 . | | |
| Sink: | SqlInjectionAdvanced.html:89 | | |

```
87                                          </div>
88                                          <div class="form-group">
89                                              <input type="password"
name="password_login" id="password4" tabindex="2"
```

```
90                                                          class="form-control"
                placeholder="Password"/>
91                                                  </div>
```

## AsciiDoctorTemplateResolver.java, line 150 (Log Forging (debug))

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | AsciiDoctorTemplateResolver.java determineLanguage() 150 . . | | |
| **Source:** | AsciiDoctorTemplateResolver.java:148<br>javax.servlet.http.HttpServletRequest.getHeader() | | |

```
146                 return browserLocale.getLanguage();
147             } else {
148                 String langHeader = request.getHeader(Headers.ACCEPT_LANGUAGE_STRING);
149                 if (null != langHeader) {
150                     log.debug("browser locale {}", langHeader);
```

**Sink:** AsciiDoctorTemplateResolver.java:150 org.slf4j.Logger.debug()

```
148                 String langHeader = request.getHeader(Headers.ACCEPT_LANGUAGE_STRING);
149                 if (null != langHeader) {
150                     log.debug("browser locale {}", langHeader);
151                     return langHeader.substring(0,2);
152                 } else {
```

## WebSecurityConfig.java, line 51 (HTML5: Missing Content Security Policy)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | CSP(Content Security Policy) . | | |
| **Sink:** | WebSecurityConfig.java:51 Function: configure() | | |

```
49
50      @Override
51      protected void configure(HttpSecurity http) throws Exception {
52
            ExpressionUrlAuthorizationConfigurer<HttpSecurity>.ExpressionInterceptUrlRegistry
            security = http
53                  .authorizeRequests()
```

## MissingFunctionACUsers.java, line 67 (Trust Boundary Violation)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | MissingFunctionACUsers.java listUsers() . . | | |
| **Source:** | MissingAccessControlUserRepository.java:23<br>org.springframework.jdbc.core.namedparam.NamedParameterJdbcTemplate.query() | | |

```
21
22      public List<User> findAllUsers() {
23          return jdbcTemplate.query("select username, password, admin from
            access_control_users", mapper);
24      }
```

**Sink:** MissingFunctionACUsers.java:67<br>org.springframework.web.servlet.ModelAndView.addObject()

```
65                  displayUsers.add(new DisplayUser(user, PASSWORD_SALT_SIMPLE));
66              }
67              model.addObject("allUsers", displayUsers);
68
69              return model;
```

## SqlInjectionLesson2.java, line 62 (SQL Injection)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | SqlInjectionLesson2.java 62 injectableQuery() SQL . SQL . | | |

ot Copyright 2024 Open Text.

Source: SqlInjectionLesson2.java:55 completed(0)

```
53      @PostMapping("/SqlInjection/attack2")
54      @ResponseBody
55      public AttackResult completed(@RequestParam String query) {
56          return injectableQuery(query);
57      }
```

Sink: SqlInjectionLesson2.java:62 java.sql.Statement.executeQuery()

```
60          try (var connection = dataSource.getConnection()) {
61              Statement statement = connection.createStatement(TYPE_SCROLL_INSENSITIVE,
CONCUR_READ_ONLY);
62              ResultSet results = statement.executeQuery(query);
63              StringBuilder output = new StringBuilder();
```

## Servers.java, line 72 (SQL Injection)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Input Validation and Representation | | |

Abstract: Servers.java 72 sort() SQL . SQL .

Source: Servers.java:68 sort(0)

```
66      @GetMapping(produces = MediaType.APPLICATION_JSON_VALUE)
67      @ResponseBody
68      public List<Server> sort(@RequestParam String column) throws Exception {
69          List<Server> servers = new ArrayList<>();
```

Sink: Servers.java:72 java.sql.Connection.prepareStatement()

```
70
71          try (var connection = dataSource.getConnection()) {
72              try (var statement = connection.prepareStatement("select id, hostname, ip,
mac, status, description from SERVERS where status <> 'out of order' order by " +
column)) {
73                  try (var rs = statement.executeQuery()) {
74                      while (rs.next()) {
```

## login.html, line 39 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Security Features | | |

Abstract: login.html 39 .

Sink: login.html:39

```
37                  <div class="form-group">
38                      <label for="exampleInputPassword1"
th:text="#{password}">Password</label>
39                      <input type="password" class="form-control"
id="exampleInputPassword1" th:placeholder="#{password}"
40                             name='password' />
41                  </div>
```

## LabelAndHintIntegrationTest.java, line 137 (Poor Logging Practice: Use of a System Output Stream)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |

Abstract: println() .

Sink: LabelAndHintIntegrationTest.java:137 FunctionCall: println()

```
135
System.out.println("actual:"+jsonPath.getString(ESCAPE_JSON_PATH_CHAR+key+ESCAPE_JSON_
PATH_CHAR));
136                 System.out.println("expected: "+propsLang.getProperty(key));
137                 System.out.println();
138                 Assertions.fail();
139             }
```

## SqlInjectionLesson5a.java, line 62 (SQL Injection)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Input Validation and Representation | | |

ot Copyright 2024 Open Text.

| | |
|---|---|
| **Abstract:** | SqlInjectionLesson5a.java 62 injectableQuery() SQL . SQL . |

**Source:** SqlInjectionLesson5a.java:53 completed(0)

```
51      @PostMapping("/SqlInjection/assignment5a")
52      @ResponseBody
53      public AttackResult completed(@RequestParam String account, @RequestParam String
operator, @RequestParam String injection) {
54          return injectableQuery(account + " " + operator + " " + injection);
55      }
```

**Sink:** SqlInjectionLesson5a.java:62 java.sql.Statement.executeQuery()

```
60                  query = "SELECT * FROM user_data WHERE first_name = 'John' and last_name =
'" + accountName + "'";
61                  try (Statement statement =
connection.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_UPDATABLE)) {
62                      ResultSet results = statement.executeQuery(query);
63
64                      if ((results != null) && (results.first())) {
```

## Challenge1.html, line 30 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Security Features | | |
| **Abstract:** | Challenge1.html 30 . | | |

**Sink:** Challenge1.html:30

```
28                          <div class="form-group">
29                              <label for="exampleInputPassword1"
th:text="#{password}">Password</label>
30                              <input type="password" class="form-control"
id="exampleInputPassword1"
31                                  placeholder="Password"
32                                  name='password'/>
```

## SqlInjectionChallengeLogin.java, line 50 (Access Control: Database)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Security Features | | |
| **Abstract:** | access control SqlInjectionChallengeLogin.java login() 50 SQL . | | |

**Source:** SqlInjectionChallengeLogin.java:46 login(1)

```
44      @PostMapping("/SqlInjectionAdvanced/challenge_Login")
45      @ResponseBody
46      public AttackResult login(@RequestParam String username_login, @RequestParam
String password_login) throws Exception {
47          try (var connection = dataSource.getConnection()) {
48              var statement = connection.prepareStatement("select password from
sql_challenge_users where userid = ? and password =******
```

**Sink:** SqlInjectionChallengeLogin.java:50 java.sql.PreparedStatement.setString()

```
48              var statement = connection.prepareStatement("select password from
sql_challenge_users where userid = ? and password =******
49              statement.setString(1, username_login);
50              statement.setString(2, password_login);
51              var resultSet = statement.executeQuery();
```

## AsciiDoctorTemplateResolver.java, line 145 (Privacy Violation)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Security Features | | |
| **Abstract:** | AsciiDoctorTemplateResolver.java determineLanguage() . . | | |

**Source:** EncodingAssignment.java:53 Read password()

```
51    if (basicAuth == null) {
52     String password =****** Random().nextInt(HashingAssignment.SECRETS.length)];
53     basicAuth = getBasicAuth(username, password);
54     request.getSession().setAttribute("basicAuth", basicAuth);
55    }
```

**Sink:** AsciiDoctorTemplateResolver.java:145 org.slf4j.Logger.debug()

ot Copyright 2024 Open Text.

```
143                 Locale browserLocale = (Locale)
request.getSession().getAttribute(SessionLocaleResolver.LOCALE_SESSION_ATTRIBUTE_NAME)
;
144                 if (null != browserLocale) {
145                     log.debug("browser locale {}", browserLocale);
146                     return browserLocale.getLanguage();
147                 } else {
```

## SigningAssignment.java, line 59 (Privacy Violation)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Security Features | | |
| **Abstract:** | SigningAssignment.java getPrivateKey() . . | | |

**Source:** SigningAssignment.java:56 Read privateKey()

```
54    KeyPair keyPair = CryptoUtil.generateKeyPair();
55    privateKey = CryptoUtil.getPrivateKeyInPEM(keyPair);
56    request.getSession().setAttribute("privateKeyString", privateKey);
57    request.getSession().setAttribute("keyPair", keyPair);
58   }
```

**Sink:** SigningAssignment.java:59 Return privateKey()

```
57    request.getSession().setAttribute("keyPair", keyPair);
58   }
59   return privateKey;
60    }
61
```

## SqlInjectionAdvanced.html, line 133 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Security Features | | |
| **Abstract:** | SqlInjectionAdvanced.html 133 . | | |

**Sink:** SqlInjectionAdvanced.html:133

```
131                                                 </div>
132                                                 <div class="form-group">
133                                                     <input type="password"
name="confirm_password_reg" id="confirm-password"
134                                                            tabindex="2" class="form-control"
placeholder="Confirm Password"/>
135                                                 </div>
```

## CrossSiteScriptingMitigation.html, line 30 (Hidden Field)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | CrossSiteScriptingMitigation.html 30 . | | |

**Sink:** CrossSiteScriptingMitigation.html:30

```
28    <script th:src="@{/lesson_js/assignment3.js}" type="text/javascript" charset="utf-
8"></script>
29    </div>
30    <input type="hidden" name="editor"/>
31    <div class="input-group" style="position: absolute; top: 365px;">
32     <button class="btn btn-primary" type="submit">Submit</button>
```

## jquery.form.js, line 931 (Password Management: Password in Comment)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Security Features | | |
| **Abstract:** | . | | |

**Sink:** jquery.form.js:931 Comment()

```
929 };
930
931 /**
932  * formToArray() gathers form element data into an array of objects that can
933  * be passed to any of the following ajax functions: $.get, $.post, or load.
```

ot Copyright 2024 Open Text.

## SqlInjectionLesson8.java, line 138 (SQL Injection)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | SqlInjectionLesson8.java 138 log() SQL . SQL . | | |

**Source:** SqlInjectionLesson8.java:54 completed(0)

```
52      @PostMapping("/SqlInjection/attack8")
53      @ResponseBody
54      public AttackResult completed(@RequestParam String name, @RequestParam String auth_tan) {
55          return injectableQueryConfidentiality(name, auth_tan);
56      }
```

**Sink:** SqlInjectionLesson8.java:138 java.sql.Statement.executeUpdate()

```
136         try {
137             Statement statement = connection.createStatement(TYPE_SCROLL_SENSITIVE, CONCUR_UPDATABLE);
138             statement.executeUpdate(logQuery);
139         } catch (SQLException e) {
140             System.err.println(e.getMessage());
```

## WebSecurityConfig.java, line 57 (Spring Security Misconfiguration: Default Permit)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Security Features | | |
| **Abstract:** | Spring Security . | | |

**Sink:** WebSecurityConfig.java:57 FunctionCall: permitAll()

```
55                  .antMatchers(HttpMethod.GET, "/mail/**", "/requests/**").authenticated()
56                  .antMatchers("/files").authenticated()
57                  .anyRequest().permitAll();
58          security.and().csrf().disable().formLogin()
59                  .loginPage("/login").failureUrl("/login?error=true");
```

## JWTFinalEndpoint.java, line 94 (SQL Injection)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | JWTFinalEndpoint.java 94 resolveSigningKeyBytes() SQL . SQL . | | |

**Sink:** JWTFinalEndpoint.java:94 executeQuery()

```
92                      final String kid = (String) header.get("kid");
93                      try (var connection = dataSource.getConnection()) {
94                          ResultSet rs = connection.createStatement().executeQuery("SELECT key FROM jwt_keys WHERE id = '" + kid + "'");
95                          while (rs.next()) {
96                              return TextCodec.BASE64.decode(rs.getString(1));
```

## AuthBypass.html, line 56 (Hidden Field)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | AuthBypass.html 56 . | | |

**Sink:** AuthBypass.html:56

```
54          <input name="newPasswordConfirm" value="" type="password" /><br/><br />
55
56          <input type="hidden" name="userId" value="12309746" />
57
58          <input name="submit" value="Submit" type="submit"/>
```

## InsecureLogin.html, line 31 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Security Features | | |

ot Copyright 2024 Open Text.

| | |
|---|---|
| **Abstract:** | InsecureLogin.html 31 . |
| **Sink:** | InsecureLogin.html:31 |

```
29
30                      <input type="text" value="" name="username" placeholder="username"/>
31                      <input type="password" value="" name="password" placeholder="password"
/>
32                      <input type="submit" value="Submit" />
```

## Challenge6.html, line 81 (Password Management: Insecure Submission)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Security Features | | |
| **Abstract:** | Challenge6.html 81 HTTP GET , . | | |
| **Sink:** | Challenge6.html:81 | | |

```
79                                          </div>
80                                          <div class="form-group">
81                                              <input type="password"
name="confirm_password_reg" id="confirm-password"
82                                                     tabindex="2" class="form-control"
placeholder="Confirm Password"/>
83                                          </div>
```

## Assignment1Test.java, line 91 (Password Management: Password in Comment)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Security Features | | |
| **Abstract:** | . | | |
| **Sink:** | Assignment1Test.java:91 Comment() | | |

```
89
90      //    @Test
91      //    public void correctPasswordXForwardHeaderWrong() throws Exception {
92      //        mockMvc.perform(MockMvcRequestBuilders.post("/challenge/1")
93      //                .header("X-Forwarded-For", "127.0.1.2")
```

## FileServer.java, line 118 (Trust Boundary Violation)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | FileServer.java getFiles() . . | | |
| **Source:** | FileServer.java:118 Read this.server() | | |

```
116
117             modelAndView.addObject("files", uploadedFiles);
118             modelAndView.addObject("webwolf_url", "http://" + server + ":" + port);
119             return modelAndView;
120         }
```

| | |
|---|---|
| **Sink:** | FileServer.java:118<br>org.springframework.web.servlet.ModelAndView.addObject() |

```
116
117             modelAndView.addObject("files", uploadedFiles);
118             modelAndView.addObject("webwolf_url", "http://" + server + ":" + port);
119             return modelAndView;
120         }
```

## Assignment1Test.java, line 82 (Password Management: Password in Comment)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Security Features | | |
| **Abstract:** | . | | |
| **Sink:** | Assignment1Test.java:82 Comment() | | |

```
80
81      //    @Test
82      //    public void correctPasswordXForwardHeaderMissing() throws Exception {
83      //        mockMvc.perform(MockMvcRequestBuilders.post("/challenge/1")
```

Copyright 2024 Open Text.

```
84                  //                  .param("username", "admin")
```

## SqlInjectionChallenge.java, line 65 (SQL Injection)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Input Validation and Representation | | |

**Abstract:** SqlInjectionChallenge.java 65 registerNewUser() SQL . SQL .

**Source:** SqlInjectionChallenge.java:56 registerNewUser(0)

```
54      //assignment path is bounded to class so we use different http method :-)
55      @ResponseBody
56      public AttackResult registerNewUser(@RequestParam String username_reg,
@RequestParam String email_reg, @RequestParam String password_reg) throws Exception {
57          AttackResult attackResult = checkArguments(username_reg, email_reg,
password_reg);
```

**Sink:** SqlInjectionChallenge.java:65 java.sql.Statement.executeQuery()

```
63                  String checkUserQuery = "select userid from sql_challenge_users where
userid = '" + username_reg + "'";
64                  Statement statement = connection.createStatement();
65                  ResultSet resultSet = statement.executeQuery(checkUserQuery);
66
67                  if (resultSet.next()) {
```

## SqlInjectionMitigations.html, line 51 (Hidden Field)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |

**Abstract:** SqlInjectionMitigations.html 51 .

**Sink:** SqlInjectionMitigations.html:51

```
49                  <script th:src="@{/lesson_js/assignment10b.js}" type="text/javascript"
charset="utf-8"></script>
50              </div>
51              <input type="hidden" name="editor"/>
52              <div class="input-group" style="position: absolute; top: 310px;">
53                  <button class="btn btn-primary" type="submit">Submit</button>
```

## AuthBypass.html, line 51 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Security Features | | |

**Abstract:** AuthBypass.html 51 .

**Sink:** AuthBypass.html:51

```
49
50                  <p>Password:</p>
51                  <input name="newPassword" value="" type="password" /><br/>
52
53                  <p>Confirm Password:</p>
```

## SqlInjectionLesson6a.java, line 53 (Password Management: Password in Comment)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Security Features | | |

**Abstract:** .

**Sink:** SqlInjectionLesson6a.java:53 Comment()

```
51      public AttackResult completed(@RequestParam(value="userid_6a")  String userId) {
52          return injectableQuery(userId);
53          // The answer: Smith' union select userid,user_name, password,cookie,cookie,
cookie,userid from user_system_data --
54      }
```

## registration.html, line 41 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Security Features | | |

**Abstract:** registration.html 41 .

ot Copyright 2024 Open Text.

| | |
|---|---|
| Sink: | registration.html:41 |

```
39                                  <label for="password" class="col-sm-2 control-label"
th:text="#{password}">Password</label>
40                                  <div class="col-sm-4">
41                                      <input type="password" class="form-control" id="password"
th:placeholder="#{password}"
42                                             name='password' th:value="*{password}"/>
43                                  </div>
```

## PasswordReset.html, line 202 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Security Features | | |
| Abstract: | PasswordReset.html 202 . | | |
| Sink: | PasswordReset.html:202 | | |

```
200                 </i>
201             </span>
202                                             <input class="form-control"
placeholder="Password" name="password"
203                                                    type="password"
204                                                    value="" required=""/>
```

## login.html, line 39 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Security Features | | |
| Abstract: | login.html 39 . | | |
| Sink: | login.html:39 | | |

```
37                      <div class="form-group">
38                          <label for="exampleInputPassword1"
th:text="#{password}">Password</label>
39                          <input type="password" class="form-control"
id="exampleInputPassword1" th:placeholder="#{password}"
40                                 name='password' />
41                      </div>
```

## Requests.java, line 76 (Trust Boundary Violation)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | Requests.java get() . . | | |
| Source: | Requests.java:53 Requests(0) | | |

```
51  */
52 @Controller
53 @RequiredArgsConstructor
54 @Slf4j
55 @RequestMapping(value = "/requests")
```

| | |
|---|---|
| Sink: | Requests.java:76 org.springframework.web.servlet.ModelAndView.addObject() |

```
74              .filter(t -> allowedTrace(t, user))
75                    .map(t -> new Tracert(t.getTimestamp(), path(t),
toJsonString(t))).collect(toList());
76          model.addObject("traces", traces);
77
78          return model;
```

## SqlInjectionAdvanced.html, line 133 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Security Features | | |
| Abstract: | SqlInjectionAdvanced.html 133 . | | |
| Sink: | SqlInjectionAdvanced.html:133 | | |

```
131                                             </div>
132                                             <div class="form-group">
```

```
133                                                     <input type="password"
name="confirm_password_reg" id="confirm-password"
134                                                     tabindex="2" class="form-control"
placeholder="Confirm Password"/>
135                                                 </div>
```

## IDOR.html, line 30 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Security Features | | |
| **Abstract:** | IDOR.html 30 . | | |

**Sink:** IDOR.html:30

```
28                          <td>user/pass</td>
29                          <td>user:<input name="username" value="" type="TEXT" /></td>
30                          <td>pass:<input name="password" value="" type="password" /></td>
31                          <td>
32                              <input
```

## BypassRestrictions.html, line 108 (Hidden Field)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | BypassRestrictions.html 108 . | | |

**Sink:** BypassRestrictions.html:108

```
106                         <textarea cols="25" name="field7" rows="1">301-604-4882</textarea>
107                     </div>
108                     <input type="hidden" value="" name="error"/>
109                     <p>
110                         <button type="submit" class="btn btn-primary">Submit</button>
```

## Challenge6.html, line 81 (Password Management: Insecure Submission)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Security Features | | |
| **Abstract:** | Challenge6.html 81 HTTP GET , . | | |

**Sink:** Challenge6.html:81

```
79                                                  </div>
80                                                  <div class="form-group">
81                                                      <input type="password"
name="confirm_password_reg" id="confirm-password"
82                                                      tabindex="2" class="form-control"
placeholder="Confirm Password"/>
83                                                  </div>
```

## password_reset.html, line 15 (Hidden Field)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | password_reset.html 15 . | | |

**Sink:** password_reset.html:15

```
13                      <h2 class="sign_up_title">Reset your password</h2>
14                          <div class="form-group"
th:classappend="${#fields.hasErrors('password')}? 'has-error'">
15                              <input type="hidden" name="resetLink" th:field="*{resetLink}"
/>
16                              <label for="password" class="control-label"
th:text="#{password}">Password</label>
17                              <input type="password" class="form-control" id="password"
placeholder="Password"
```

## SSRF.html, line 18 (Hidden Field)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | SSRF.html 18 . | | |

Copyright 2024 Open Text.

| Sink: | SSRF.html:18 |
|---|---|

```
16                          <table>
17                              <tr>
18                                  <td><input type="hidden" id="url1" name="url"
value="images/tom.png"/></td>
19
20                                  <td><input
```

## ClientSideFiltering.html, line 22 (Hidden Field)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | ClientSideFiltering.html 22 . | | |

| Sink: | ClientSideFiltering.html:22 |
|---|---|

```
20              <script th:src="@{/lesson_js/clientSideFiltering.js}"
21                      language="JavaScript"></script>
22              <input id="userID" value="101" name="userID" type="HIDDEN"/>
23              <div id="lesson_wrapper">
24                  <div id="lesson_header"></div>
```

## PasswordReset.html, line 48 (HTML5: Form Validation Turned Off)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Input Validation and Representation | | |
| Abstract: | HTML5 . | | |

| Sink: | PasswordReset.html:48 |
|---|---|

```
46                              </div>
47                          </form>
48                          <form class="attack-form" accept-charset="UNKNOWN"
novalidate="novalidate"
49                                method="POST"
50                                action="/WebGoat/PasswordReset/simple-mail">
```

## PathTraversal.html, line 49 (Privacy Violation: Autocomplete)

| Fortify Priority: | High | Folder | High |
|---|---|---|---|
| Kingdom: | Security Features | | |
| Abstract: | PathTraversal.html 49 . | | |

| Sink: | PathTraversal.html:49 |
|---|---|

```
47                          <div class="form-group">
48                              <label>Password:</label>
49                              <input class="form-control" type="password" id="password"
name="password" required
50                                   placeholder="Enter Password" value="test"/>
51                              <span class="Error"></span>
```

## ChromeDevTools.html, line 57 (Hidden Field)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | ChromeDevTools.html 57 . | | |

| Sink: | ChromeDevTools.html:57 |
|---|---|

```
55                          });
56                      </script>
57                      <input type="hidden" name="networkNum" id="networkNum" value="foo" />
58                      <table>
59                          <tr>
```

## PathTraversal.html, line 103 (Privacy Violation: Autocomplete)

| Fortify Priority: | High | Folder | High |
|---|---|---|---|
| Kingdom: | Security Features | | |
| Abstract: | PathTraversal.html 103 . | | |

| | |
|---|---|
| **Sink:** | PathTraversal.html:103 |

```
101                         <div class="form-group">
102                             <label>Password:</label>
103                             <input class="form-control" type="password" id="passwordFix"
name="password" required
104                                    placeholder="Enter Password" value="test"/>
105                             <span class="Error"></span>
```

## SqlInjectionLesson6a.java, line 67 (SQL Injection)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | SqlInjectionLesson6a.java 67 injectableQuery() SQL . SQL . | | |
| **Source:** | SqlInjectionLesson6a.java:51 completed(0) | | |

```
49      @PostMapping("/SqlInjectionAdvanced/attack6a")
50      @ResponseBody
51      public AttackResult completed(@RequestParam(value="userid_6a") String userId) {
52          return injectableQuery(userId);
53          // The answer: Smith' union select userid,user_name, password,cookie,cookie,
cookie,userid from user_system_data --
```

| | |
|---|---|
| **Sink:** | SqlInjectionLesson6a.java:67 java.sql.Statement.executeQuery() |

```
65              try (Statement statement =
connection.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
66                      ResultSet.CONCUR_READ_ONLY)) {
67                  ResultSet results = statement.executeQuery(query);
68
69                  if ((results != null) && results.first()) {
```

## Challenge6.html, line 38 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Security Features | | |
| **Abstract:** | Challenge6.html 38 . | | |
| **Sink:** | Challenge6.html:38 | | |

```
36                                              </div>
37                                              <div class="form-group">
38                                                  <input type="password"
name="password_login" id="password4" tabindex="2"
39                                                         class="form-control"
placeholder="Password"/>
40                                              </div>
```

## AuthBypass.html, line 36 (Hidden Field)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | AuthBypass.html 36 . | | |
| **Sink:** | AuthBypass.html:36 | | |

```
34
35                      <input type="hidden" name="jsEnabled" value="1" />
36                      <input type="hidden" name="verifyMethod" value="SEC_QUESTIONS" />
37                      <input type="hidden" name="userId" value="12309746" />
```

## webwolf-login.html, line 37 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Security Features | | |
| **Abstract:** | webwolf-login.html 37 . | | |
| **Sink:** | webwolf-login.html:37 | | |

```
35                      </div>
36                      <div class="form-group">
37                          <input type="password" name="password" id="password"
class="form-control input-lg"
38                                 placeholder="Password WebGoat" required="true"/>
```

ot Copyright 2024 Open Text.

```
39                                  </div>
```

## webwolfPasswordReset.html, line 17 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Security Features | | |
| **Abstract:** | webwolfPasswordReset.html 17 . | | |

**Sink:** webwolfPasswordReset.html:17

```
15                          <div class="form-group">
16                              <label for="password" class="control-label">Password</label>
17                              <input type="password" class="form-control" id="password"
placeholder="Password"
18                                      name='password'/>
19                          </div>
```

## CSRF.html, line 100 (Hidden Field)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | CSRF.html 100 . | | |

**Sink:** CSRF.html:100

```
98                                                  type="text"/>
99                                              <input class="form-control" id="reviewStars"
name="stars" type="text"/>
100                                             <input type="hidden" name="validateReq"
value="2aa14227b9a13d0bede0388a7fba9aa9"/>
101                                             <input type="submit" name="submit" value="Submit
review"/>
102                                         </form>
```

## PathTraversal.html, line 258 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Security Features | | |
| **Abstract:** | PathTraversal.html 258 . | | |

**Sink:** PathTraversal.html:258

```
256                         <div class="form-group">
257                             <label>Password:</label>
258                             <input class="form-control" type="password" id="passwordZipSlip"
name="password" required
259                                     placeholder="Enter Password" value="test"/>
260                             <span class="Error"></span>
```

## SSRF.html, line 18 (Hidden Field)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | SSRF.html 18 . | | |

**Sink:** SSRF.html:18

```
16                          <table>
17                              <tr>
18                                  <td><input type="hidden" id="url1" name="url"
value="images/tom.png"/></td>
19
20                                  <td><input
```

## clientSideFiltering.js, line 38 (Privacy Violation)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Security Features | | |
| **Abstract:** | clientSideFiltering.js 38 . . | | |

**Source:** clientSideFiltering.js:31 Read SSN()

```
29                  html = html + '<td>' + result[i].FirstName + '</td>';
30                  html = html + '<td>' + result[i].LastName + '</td>';
```

Copyright 2024 Open Text.

```
31                          html = html + '<td>' + result[i].SSN + '</td>';
32                          html = html + '<td>' + result[i].Salary + '</td>';
33                          html = html + '</tr>';
```

**Sink:** clientSideFiltering.js:38 Assignment to newdiv.innerHTML()

```
36
37                  var newdiv = document.createElement("div");
38                  newdiv.innerHTML = html;
39                  var container = document.getElementById("hiddenEmployeeRecords");
40                  container.appendChild(newdiv);
```

## CrossSiteScriptingMitigation.html, line 30 (Hidden Field)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |

**Abstract:** CrossSiteScriptingMitigation.html 30 .

**Sink:** CrossSiteScriptingMitigation.html:30

```
28          <script th:src="@{/lesson_js/assignment3.js}" type="text/javascript" charset="utf-8"></script>
29        </div>
30        <input type="hidden" name="editor"/>
31        <div class="input-group" style="position: absolute; top: 365px;">
32         <button class="btn btn-primary" type="submit">Submit</button>
```

## VulnerableTaskHolder.java, line 64 (Denial of Service)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Input Validation and Representation | | |

**Abstract:** VulnerableTaskHolder.java 64 , readLine() .

**Sink:** VulnerableTaskHolder.java:64 readLine()

```
62                                      new InputStreamReader(p.getInputStream()));
63                  String line = null;
64                  while ((line = in.readLine()) != null) {
65                      log.info(line);
66                  }
```

## SqlInjectionLesson5b.java, line 68 (Access Control: Database)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Security Features | | |

**Abstract:** access control SqlInjectionLesson5b.java injectableQuery() 68 SQL .

**Source:** SqlInjectionLesson5b.java:51 completed(1)

```
49      @PostMapping("/SqlInjection/assignment5b")
50      @ResponseBody
51      public AttackResult completed(@RequestParam String userid, @RequestParam String login_count, HttpServletRequest request) throws IOException {
52          return injectableQuery(login_count, userid);
53      }
```

**Sink:** SqlInjectionLesson5b.java:68 java.sql.PreparedStatement.setInt()

```
66                  }
67
68                  query.setInt(1, count);
69                  //String query = "SELECT * FROM user_data WHERE Login_Count = " + login_count + " and userid = " + accountName, ;
70                  try {
```

## ChromeDevTools.html, line 78 (Hidden Field)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |

**Abstract:** ChromeDevTools.html 78 .

**Sink:** ChromeDevTools.html:78

```
76                      </tr>
77                  </table>
```

```
78                          <input type="hidden" name="network_num" id="networkNumCopy" value="foo" />
79                      </form>
80                      <div class="attack-feedback"></div>
```

## IDOREditOtherProfiile.java, line 45 (Mass Assignment: Insecure Binder Configuration)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | API Abuse | | |

**Abstract:** HTTP .

**Sink:** IDOREditOtherProfiile.java:45 Function: completed()

```
43      @PutMapping(path = "/IDOR/profile/{userId}", consumes = "application/json")
44      @ResponseBody
45      public AttackResult completed(@PathVariable("userId") String userId, @RequestBody
UserProfile userSubmittedProfile) {
46
47          String authUserId = (String) userSessionData.getValue("idor-authenticated-
user-id");
```

## SqlInjectionLesson3.java, line 65 (SQL Injection)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Input Validation and Representation | | |

**Abstract:** SqlInjectionLesson3.java 65 injectableQuery() SQL . SQL .

**Source:** SqlInjectionLesson3.java:56 completed(0)

```
54      @PostMapping("/SqlInjection/attack3")
55      @ResponseBody
56      public AttackResult completed(@RequestParam String query) {
57          return injectableQuery(query);
58      }
```

**Sink:** SqlInjectionLesson3.java:65 java.sql.Statement.executeUpdate()

```
63                  Statement checkStatement =
connection.createStatement(TYPE_SCROLL_INSENSITIVE,
64                          CONCUR_READ_ONLY);
65                  statement.executeUpdate(query);
66                  ResultSet results = checkStatement.executeQuery("SELECT * FROM
employees WHERE last_name='Barnett';");
67                  StringBuilder output = new StringBuilder();
```

## PasswordReset.html, line 24 (HTML5: Form Validation Turned Off)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Input Validation and Representation | | |

**Abstract:** HTML5 .

**Sink:** PasswordReset.html:24

```
22
23
24                      <form class="attack-form" accept-charset="UNKNOWN"
novalidate="novalidate"
25                              method="POST"
26                              action="/WebGoat/PasswordReset/simple-mail/reset">
```

## WebSecurityConfig.java, line 56 (Spring Security Misconfiguration: Incorrect Request Matcher Type)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Security Features | | |

**Abstract:** Spring Security .

**Sink:** WebSecurityConfig.java:56 FunctionCall: antMatchers()

```
54                  .antMatchers("/css/**", "/images/**", "/js/**", "/fonts/**",
"/webjars/**", "/home").permitAll()
55                  .antMatchers(HttpMethod.GET, "/mail/**",
"/requests/**").authenticated()
56                  .antMatchers("/files").authenticated()
57                  .anyRequest().permitAll();
58          security.and().csrf().disable().formLogin()
```

Copyright 2024 Open Text.

## StartupMessage.java, line 22 (Log Forging)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | StartupMessage.java onStartup() 22 . . | | |

**Source:** StartupMessage.java:26 org.springframework.core.env.PropertyResolver.getProperty()

```
24          if (event.getApplicationContext().getApplicationName().contains("WebGoat")) {
25              port =
event.getApplicationContext().getEnvironment().getProperty("server.port");
26              address =
event.getApplicationContext().getEnvironment().getProperty("server.address");
27          }
28      }
```

**Sink:** StartupMessage.java:22 org.slf4j.Logger.info()

```
20      void onStartup(ApplicationReadyEvent event) {
21          if (StringUtils.hasText(port) &&
!StringUtils.hasText(System.getProperty("running.in.docker"))) {
22              log.info("Please browse to http://{}:{}/WebGoat to get started...",
address, port);
23          }
24          if (event.getApplicationContext().getApplicationName().contains("WebGoat")) {
```

## CSRF.html, line 21 (Hidden Field)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | CSRF.html 21 . | | |

**Sink:** CSRF.html:21

```
19              successCallback=""
20              action="/WebGoat/csrf/basic-get-flag">
21              <input name="csrf" type="hidden" value="false"/>
22              <input type="submit" name="submit"/>
```

## Challenge7.html, line 49 (Hidden Field)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | Challenge7.html 49 . | | |

**Sink:** Challenge7.html:49

```
47                                          </div>
48
49                                          <input type="hidden" class="hide" name="token"
id="token" value=""/>
50                                      </form>
```

## HttpBasics.html, line 64 (Hidden Field)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | HttpBasics.html 64 . | | |

**Sink:** HttpBasics.html:64

```
62          webgoat.customjs.assignRandomVal();
63          </script>
64          <input type="hidden" name="magic_num" id="magic_num" value="foo" />
65          <table>
66           <tr>
```

## webwolfPasswordReset.html, line 17 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Security Features | | |
| **Abstract:** | webwolfPasswordReset.html 17 . | | |

ot Copyright 2024 Open Text.

| | |
|---|---|
| **Sink:** | webwolfPasswordReset.html:17 |

```
15                          <div class="form-group">
16                              <label for="password" class="control-label">Password</label>
17                              <input type="password" class="form-control" id="password"
placeholder="Password"
18                                     name='password'/>
19                          </div>
```

## PasswordReset.html, line 48 (HTML5: Form Validation Turned Off)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | HTML5 . | | |
| **Sink:** | PasswordReset.html:48 | | |

```
46                                  </div>
47                              </form>
48                              <form class="attack-form" accept-charset="UNKNOWN"
novalidate="novalidate"
49                                    method="POST"
50                                    action="/WebGoat/PasswordReset/simple-mail">
```

## password_reset.html, line 17 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Security Features | | |
| **Abstract:** | password_reset.html 17 . | | |
| **Sink:** | password_reset.html:17 | | |

```
15                                  <input type="hidden" name="resetLink" th:field="*{resetLink}"
/>
16                                  <label for="password" class="control-label"
th:text="#{password}">Password</label>
17                                  <input type="password" class="form-control" id="password"
placeholder="Password"
18                                         name='password' th:value="*{password}"/>
19                                  <span th:if="${#fields.hasErrors('password')}"
th:errors="*{password}">Password error</span>
```

## ClientSideFiltering.html, line 88 (Hidden Field)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | ClientSideFiltering.html 88 . | | |
| **Sink:** | ClientSideFiltering.html:88 | | |

```
86                              action="/WebGoat/clientSideFiltering/getItForFree">
87
88                          <input id="discount" type="hidden" value="0"/>
89                          <div class="row">
```

## CommentsCache.java, line 102 (XML Entity Expansion Injection)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Medium | **Folder** | Medium |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | CommentsCache.java:102 XML DTD(Document Type Definition) . XML injection . | | |
| **Source:** | BlindSendFileAssignment.java:80 addComment(0) | | |

```
78      @PostMapping(path = "xxe/blind", consumes = ALL_VALUE, produces =
APPLICATION_JSON_VALUE)
79      @ResponseBody
80      public AttackResult addComment(@RequestBody String commentStr) {
81          var fileContentsForUser =
userToFileContents.getOrDefault(getWebSession().getUser(), "");
```

| | |
|---|---|
| **Sink:** | CommentsCache.java:102 javax.xml.stream.XMLInputFactory.createXMLStreamReader() |

```
100         }
101
```

```
102                     var xsr = xif.createXMLStreamReader(new StringReader(xml));
103
104                     var unmarshaller = jc.createUnmarshaller();
```

## Ping.java, line 51 (Log Forging (debug))

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | Ping.java logRequest() 51 . . | | |

Source: Ping.java:49 logRequest(0)

```
47      @RequestMapping(method = RequestMethod.GET)
48      @ResponseBody
49      public String logRequest(@RequestHeader("User-Agent") String userAgent,
@RequestParam(required = false) String text) {
50          String logLine = String.format("%s %s %s", "GET", userAgent, text);
51          log.debug(logLine);
```

Sink: Ping.java:51 org.slf4j.Logger.debug()

```
49      public String logRequest(@RequestHeader("User-Agent") String userAgent,
@RequestParam(required = false) String text) {
50          String logLine = String.format("%s %s %s", "GET", userAgent, text);
51          log.debug(logLine);
52          File logFile = new File(webGoatHomeDirectory, "/XXE/log" +
webSession.getUserName() + ".txt");
53          try {
```

## CrossSiteScriptingMitigation.html, line 50 (Hidden Field)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | CrossSiteScriptingMitigation.html 50 . | | |

Sink: CrossSiteScriptingMitigation.html:50

```
48      <script th:src="@{/lesson_js/assignment4.js}" type="text/javascript" charset="utf-
8"></script>
49    </div>
50    <input type="hidden" name="editor2"/>
51    <div class="input-group" style="position: absolute; top: 365px;">
52     <button class="btn btn-primary" type="submit">Submit</button>
```

## Assignment1Test.java, line 95 (Password Management: Password in Comment)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Security Features | | |
| Abstract: | . | | |

Sink: Assignment1Test.java:95 Comment()

```
93      //                .header("X-Forwarded-For", "127.0.1.2")
94      //                .param("username", "admin")
95      //                .param("password", SolutionConstants.PASSWORD))
96      //                .andExpect(jsonPath("$.feedback",
CoreMatchers.is(messages.getMessage("ip.address.unknown"))))
97      //                .andExpect(jsonPath("$.lessonCompleted", CoreMatchers.is(false)));
```

## Challenge7.html, line 49 (Hidden Field)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | Challenge7.html 49 . | | |

Sink: Challenge7.html:49

```
47                                              </div>
48
49                                              <input type="hidden" class="hide" name="token"
id="token" value=""/>
50                                          </form>
```

## SqlInjectionLesson6a.java, line 67 (SQL Injection)

| Fortify Priority: | Critical | Folder | Critical |
|---|---|---|---|
| Kingdom: | Input Validation and Representation | | |
| Abstract: | SqlInjectionLesson6a.java 67 injectableQuery() SQL . SQL . | | |

Source: SqlOnlyInputValidationOnKeywords.java:48 attack(0)

```
46      @PostMapping("/SqlOnlyInputValidationOnKeywords/attack")
47      @ResponseBody
48      public AttackResult
attack(@RequestParam("userid_sql_only_input_validation_on_keywords") String userId) {
49          userId = userId.toUpperCase().replace("FROM", "").replace("SELECT", "");
50          if (userId.contains(" ")) {
```

Sink: SqlInjectionLesson6a.java:67 java.sql.Statement.executeQuery()

```
65              try (Statement statement =
connection.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
66                      ResultSet.CONCUR_READ_ONLY)) {
67                  ResultSet results = statement.executeQuery(query);
68
69                  if ((results != null) && results.first()) {
```

## hijackform.html, line 16 (Privacy Violation: Autocomplete)

| Fortify Priority: | High | Folder | High |
|---|---|---|---|
| Kingdom: | Security Features | | |
| Abstract: | hijackform.html 16 . | | |

Sink: hijackform.html:16

```
14      <div class="form-group input-group">
15       <span class="input-group-addon"><i
16        class="glyphicon glyphicon-lock"></i></span> <input class="form-control"
17        placeholder="Password" name="password" type="password" />
18      </div>
```

## ChromeDevTools.html, line 57 (Hidden Field)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | ChromeDevTools.html 57 . | | |

Sink: ChromeDevTools.html:57

```
55              });
56          </script>
57          <input type="hidden" name="networkNum" id="networkNum" value="foo" />
58          <table>
59              <tr>
```

## SqlInjectionMitigations.html, line 51 (Hidden Field)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | SqlInjectionMitigations.html 51 . | | |

Sink: SqlInjectionMitigations.html:51

```
49              <script th:src="@{/lesson_js/assignment10b.js}" type="text/javascript"
charset="utf-8"></script>
50          </div>
51          <input type="hidden" name="editor"/>
52          <div class="input-group" style="position: absolute; top: 310px;">
53              <button class="btn btn-primary" type="submit">Submit</button>
```

## CSRF.html, line 21 (Hidden Field)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | CSRF.html 21 . | | |

Sink: CSRF.html:21

```
19          successCallback=""
```

Copyright 2024 Open Text.

```
20                          action="/WebGoat/csrf/basic-get-flag">
21                      <input name="csrf" type="hidden" value="false"/>
22                      <input type="submit" name="submit"/>
```

## SqlInjectionAdvanced.html, line 129 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Security Features | | |
| Abstract: | SqlInjectionAdvanced.html 129 . | | |

Sink: SqlInjectionAdvanced.html:129

```
127                                                 </div>
128                                                 <div class="form-group">
129                                                     <input type="password" name="password_reg"
id="password" tabindex="2"
130                                                            class="form-control"
placeholder="Password"/>
131                                                 </div>
```

## InsecureLogin.html, line 31 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Security Features | | |
| Abstract: | InsecureLogin.html 31 . | | |

Sink: InsecureLogin.html:31

```
29
30                      <input type="text" value="" name="username" placeholder="username"/>
31                      <input type="password" value="" name="password" placeholder="password"
/>
32                      <input type="submit" value="Submit" />
```

## LogSpoofing.html, line 22 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Security Features | | |
| Abstract: | LogSpoofing.html 22 . | | |

Sink: LogSpoofing.html:22

```
20
21                  <input type="text" value="" name="username" placeholder="username"/>
22                  <input type="password" value="" name="password" placeholder="password"/>
23                  <input type="submit" value="Submit"/>
```

## SecurePasswordsAssignment.java, line 74 (Privacy Violation)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Security Features | | |
| Abstract: | SecurePasswordsAssignment.java completed() . . | | |

Source: SecurePasswordsAssignment.java:51 Read password()

```
49
50              output.append("<b>Your Password: *******</b></br>");
51              output.append("<b>Length: </b>" + password.length() + "</br>");
52              output.append("<b>Estimated guesses needed to crack your password: </b>" +
df.format(strength.getGuesses()) + "</br>");
53              output.append("<div style=\"float: left;padding-right: 10px;\"><b>Score: </b>"
+ strength.getScore() + "/4 </div>");
```

Sink: SecurePasswordsAssignment.java:74 Return()

```
72
73              if (strength.getScore() >= 4)
74                  return success(this).feedback("securepassword-
success").output(output.toString()).build();
75              else
76                  return failed(this).feedback("securepassword-
failed").output(output.toString()).build();
```

## Challenge6.html, line 81 (Privacy Violation: Autocomplete)

| Fortify Priority: | High | Folder | High |
|---|---|---|---|
| Kingdom: | Security Features | | |
| Abstract: | Challenge6.html 81 . | | |

Sink: Challenge6.html:81

```
79                                                  </div>
80                                                  <div class="form-group">
81                                                      <input type="password"
name="confirm_password_reg" id="confirm-password"
82                                                             tabindex="2" class="form-control"
placeholder="Confirm Password"/>
83                                                  </div>
```

## registration.html, line 50 (Privacy Violation: Autocomplete)

| Fortify Priority: | High | Folder | High |
|---|---|---|---|
| Kingdom: | Security Features | | |
| Abstract: | registration.html 50 . | | |

Sink: registration.html:50

```
48                                      password</label>
49                                  <div class="col-sm-4">
50                                      <input type="password" class="form-control"
id="matchingPassword" th:placeholder="#{password}"
51                                             name='matchingPassword'
th:value="*{matchingPassword}"/>
52                                  </div>
```

## PathTraversal.html, line 160 (Privacy Violation: Autocomplete)

| Fortify Priority: | High | Folder | High |
|---|---|---|---|
| Kingdom: | Security Features | | |
| Abstract: | PathTraversal.html 160 . | | |

Sink: PathTraversal.html:160

```
158                     <div class="form-group">
159                         <label>Password:</label>
160                         <input class="form-control" type="password"
id="passwordRemoveUserInput" name="password" required
161                                placeholder="Enter Password" value="test"/>
162                         <span class="Error"></span>
```

## SqlInjectionChallengeLogin.java, line 49 (Access Control: Database)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Security Features | | |
| Abstract: | access control SqlInjectionChallengeLogin.java login() 49 SQL . | | |

Source: SqlInjectionChallengeLogin.java:46 login(0)

```
44      @PostMapping("/SqlInjectionAdvanced/challenge_Login")
45      @ResponseBody
46      public AttackResult login(@RequestParam String username_login, @RequestParam
String password_login) throws Exception {
47          try (var connection = dataSource.getConnection()) {
48              var statement = connection.prepareStatement("select password from
sql_challenge_users where userid = ? and password =******
```

Sink: SqlInjectionChallengeLogin.java:49 java.sql.PreparedStatement.setString()

```
47          try (var connection = dataSource.getConnection()) {
48              var statement = connection.prepareStatement("select password from
sql_challenge_users where userid = ? and password =******
49              statement.setString(1, username_login);
50              statement.setString(2, password_login);
51              var resultSet = statement.executeQuery();
```

## Challenge6.html, line 77 (Privacy Violation: Autocomplete)

| Fortify Priority: | High | Folder | High |
|---|---|---|---|
| Kingdom: | Security Features | | |

| | |
|---|---|
| **Abstract:** | Challenge6.html 77 . |
| **Sink:** | Challenge6.html:77 |

```
75                                                  </div>
76                                                  <div class="form-group">
77                                                      <input type="password" name="password_reg"
id="password" tabindex="2"
78                                                             class="form-control"
placeholder="Password"/>
79                                                  </div>
```

## CommentsCache.java, line 102 (XML Entity Expansion Injection)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Medium | **Folder** | Medium |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | CommentsCache.java:102 XML DTD(Document Type Definition) . XML injection . | | |
| **Source:** | SimpleXXE.java:68 createNewComment(1) | | |

```
66      @PostMapping(path = "xxe/simple", consumes = ALL_VALUE, produces =
APPLICATION_JSON_VALUE)
67      @ResponseBody
68      public AttackResult createNewComment(HttpServletRequest request, @RequestBody
String commentStr) {
69              String error = "";
70              try {
```

| | |
|---|---|
| **Sink:** | CommentsCache.java:102 javax.xml.stream.XMLInputFactory.createXMLStreamReader() |

```
100             }
101
102             var xsr = xif.createXMLStreamReader(new StringReader(xml));
103
104             var unmarshaller = jc.createUnmarshaller();
```

## FileServer.java, line 103 (Trust Boundary Violation)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | FileServer.java getFiles() . . | | |
| **Source:** | FileServer.java:103 javax.servlet.ServletRequest.getParameter() | | |

```
101             File changeIndicatorFile = new File(destinationDir, user.getUsername() +
"_changed");
102             if (changeIndicatorFile.exists()) {
103                 modelAndView.addObject("uploadSuccess",
request.getParameter("uploadSuccess"));
104             }
105             changeIndicatorFile.delete();
```

| | |
|---|---|
| **Sink:** | FileServer.java:103 org.springframework.web.servlet.ModelAndView.addObject() |

```
101             File changeIndicatorFile = new File(destinationDir, user.getUsername() +
"_changed");
102             if (changeIndicatorFile.exists()) {
103                 modelAndView.addObject("uploadSuccess",
request.getParameter("uploadSuccess"));
104             }
105             changeIndicatorFile.delete();
```

## webwolfPasswordReset.html, line 17 (Password Management: Insecure Submission)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Security Features | | |
| **Abstract:** | webwolfPasswordReset.html 17 HTTP GET , . | | |
| **Sink:** | webwolfPasswordReset.html:17 | | |

```
15                  <div class="form-group">
16                      <label for="password" class="control-label">Password</label>
17                      <input type="password" class="form-control" id="password"
placeholder="Password"
```

```
18                                      name='password'/>
19                              </div>
```

## Challenge6.html, line 77 (Password Management: Insecure Submission)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Security Features | | |
| **Abstract:** | Challenge6.html 77 HTTP GET , . | | |

**Sink:** Challenge6.html:77

```
75                                                  </div>
76                                                  <div class="form-group">
77                                                      <input type="password" name="password_reg"
                id="password" tabindex="2"
78                                                             class="form-control"
                placeholder="Password"/>
79                                                  </div>
```

## SqlInjectionLesson4.java, line 63 (SQL Injection)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | SqlInjectionLesson4.java 63 injectableQuery() SQL . SQL . | | |

**Source:** SqlInjectionLesson4.java:56 completed(0)

```
54      @PostMapping("/SqlInjection/attack4")
55      @ResponseBody
56      public AttackResult completed(@RequestParam String query) {
57          return injectableQuery(query);
58      }
```

**Sink:** SqlInjectionLesson4.java:63 java.sql.Statement.executeUpdate()

```
61          try (Connection connection = dataSource.getConnection()) {
62              try (Statement statement =
                connection.createStatement(TYPE_SCROLL_INSENSITIVE, CONCUR_READ_ONLY)) {
63                  statement.executeUpdate(query);
64                  connection.commit();
65                  ResultSet results = statement.executeQuery("SELECT phone from
                employees;");
```

## ClientSideFiltering.html, line 14 (Hidden Field)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | ClientSideFiltering.html 14 . | | |

**Sink:** ClientSideFiltering.html:14

```
12      <div class="attack-container">
13          <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
            hidden="true"></i></div>
14          <input type="hidden" id="user_id" value="102"/>
15          <!-- using attack-form class on your form, will allow your request to be
            ajaxified and stay within the display framework for webgoat -->
16          <form class="attack-form" accept-charset="UNKNOWN" method="POST" name="form"
```

## password_reset.html, line 17 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Security Features | | |
| **Abstract:** | password_reset.html 17 . | | |

**Sink:** password_reset.html:17

```
15                          <input type="hidden" name="resetLink" th:field="*{resetLink}"
            />
16                          <label for="password" class="control-label"
            th:text="#{password}">Password</label>
17                          <input type="password" class="form-control" id="password"
            placeholder="Password"
18                                 name='password' th:value="*{password}"/>
19                          <span th:if="${#fields.hasErrors('password')}"
            th:errors="*{password}">Password error</span>
```

## PathTraversal.html, line 258 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Security Features | | |
| Abstract: | PathTraversal.html 258 . | | |

Sink: PathTraversal.html:258

```
256                     <div class="form-group">
257                         <label>Password:</label>
258                         <input class="form-control" type="password" id="passwordZipSlip"
name="password" required
259                                placeholder="Enter Password" value="test"/>
260                         <span class="Error"></span>
```

## EncodingAssignment.java, line 56 (Privacy Violation)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Security Features | | |
| Abstract: | EncodingAssignment.java getBasicAuth() . . | | |

Source: EncodingAssignment.java:53 Read password()

```
51    if (basicAuth == null) {
52     String password =****** Random().nextInt(HashingAssignment.SECRETS.length)];
53     basicAuth = getBasicAuth(username, password);
54     request.getSession().setAttribute("basicAuth", basicAuth);
55    }
```

Sink: EncodingAssignment.java:56 Return()

```
54     request.getSession().setAttribute("basicAuth", basicAuth);
55    }
56    return "Authorization: Basic ".concat(basicAuth);
57      }
58
```

## SqlInjectionLesson5.java, line 73 (SQL Injection)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | SqlInjectionLesson5.java 73 injectableQuery() SQL . SQL . | | |

Source: SqlInjectionLesson5.java:65 completed(0)

```
63      @PostMapping("/SqlInjection/attack5")
64      @ResponseBody
65      public AttackResult completed(String query) {
66          createUser();
67          return injectableQuery(query);
```

Sink: SqlInjectionLesson5.java:73 java.sql.Statement.executeQuery()

```
71          try (Connection connection = dataSource.getConnection()) {
72              try (Statement statement =
connection.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_UPDATABLE)) {
73                  statement.executeQuery(query);
74                  if (checkSolution(connection)) {
75                      return success(this).build();
```

## jquery.form.js, line 931 (Password Management: Password in Comment)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Security Features | | |
| Abstract: | . | | |

Sink: jquery.form.js:931 Comment()

```
929 };
930
931 /**
932  * formToArray() gathers form element data into an array of objects that can
933  * be passed to any of the following ajax functions: $.get, $.post, or load.
```

ot Copyright 2024 Open Text.

## SqlInjectionLesson13.java, line 56 (Access Control: Database)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Security Features | | |
| Abstract: | access control SqlInjectionLesson13.java completed() 56 SQL . | | |

Source: SqlInjectionLesson13.java:53 completed(0)

```
51      @PostMapping("/SqlInjectionMitigations/attack12a")
52      @ResponseBody
53      public AttackResult completed(@RequestParam String ip) {
54          try (Connection connection = dataSource.getConnection();
55               PreparedStatement preparedStatement = connection.prepareStatement("select
ip from servers where ip = ? and hostname = ?")) {
```

Sink: SqlInjectionLesson13.java:56 java.sql.PreparedStatement.setString()

```
54          try (Connection connection = dataSource.getConnection();
55               PreparedStatement preparedStatement = connection.prepareStatement("select
ip from servers where ip = ? and hostname = ?")) {
56              preparedStatement.setString(1, ip);
57              preparedStatement.setString(2, "webgoat-prd");
58              ResultSet resultSet = preparedStatement.executeQuery();
```

## UserValidator.java, line 27 (Access Control: Database)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Security Features | | |
| Abstract: | access control UserValidator.java validate() 27 SQL . | | |

Source: RegistrationController.java:35 registration(0)

```
33
34      @PostMapping("/register.mvc")
35      public String registration(@ModelAttribute("userForm") @Valid UserForm userForm,
BindingResult bindingResult, HttpServletRequest request) throws ServletException {
36          userValidator.validate(userForm, bindingResult);
```

Sink: UserValidator.java:27 org.owasp.webgoat.container.users.UserRepository.findByUsername()

```
25          UserForm userForm = (UserForm) o;
26
27          if (userRepository.findByUsername(userForm.getUsername()) != null) {
28              errors.rejectValue("username", "username.duplicate");
29          }
```

## webwolfPasswordReset.html, line 14 (Hidden Field)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | webwolfPasswordReset.html 14 . | | |

Sink: webwolfPasswordReset.html:14

```
12              <form role="form" method="GET" th:action="${webwolfUrl}">
13                  <h2 class="sign_up_title">Reset your password</h2>
14                  <input type="hidden" name="uniqueCode" th:value="${uniqueCode}"/>
15                  <div class="form-group">
16                      <label for="password" class="control-label">Password</label>
```

## WebWolfRedirect.java, line 19 (Trust Boundary Violation)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | WebWolfRedirect.java openWebWolf() . . | | |

Source: WebWolfRedirect.java:17 org.springframework.core.env.PropertyResolver.getProperty()

```
15      @GetMapping("/WebWolf")
16      public ModelAndView openWebWolf() {
17          var url = applicationContext.getEnvironment().getProperty("webwolf.url");
18
```

ot Copyright 2024 Open Text.

```
19                  return new ModelAndView("redirect:" + url + "/home");
```

Sink: WebWolfRedirect.java:19 org.springframework.web.servlet.ModelAndView.ModelAndView()

```
17          var url = applicationContext.getEnvironment().getProperty("webwolf.url");
18
19          return new ModelAndView("redirect:" + url + "/home");
20      }
21  }
```

## SqlInjectionLesson8.java, line 66 (SQL Injection)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | SqlInjectionLesson8.java 66 injectableQueryConfidentiality() SQL . SQL . | | |

Source: SqlInjectionLesson8.java:54 completed(1)

```
52      @PostMapping("/SqlInjection/attack8")
53      @ResponseBody
54      public AttackResult completed(@RequestParam String name, @RequestParam String auth_tan) {
55          return injectableQueryConfidentiality(name, auth_tan);
56      }
```

Sink: SqlInjectionLesson8.java:66 java.sql.Statement.executeQuery()

```
64                  Statement statement = connection.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_UPDATABLE);
65                  log(connection, query);
66                  ResultSet results = statement.executeQuery(query);
67
68                  if (results.getStatement() != null) {
```

## Challenge6.html, line 81 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Security Features | | |
| Abstract: | Challenge6.html 81 . | | |

Sink: Challenge6.html:81

```
79                                          </div>
80                                          <div class="form-group">
81                                              <input type="password" name="confirm_password_reg" id="confirm-password"
82                                                     tabindex="2" class="form-control" placeholder="Confirm Password"/>
83                                          </div>
```

## Challenge5.html, line 34 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Security Features | | |
| Abstract: | Challenge5.html 34 . | | |

Sink: Challenge5.html:34

```
32                                          </div>
33                                          <div class="form-group">
34                                              <input type="password" name="password_login" id="password4" tabindex="2"
35                                                     class="form-control" placeholder="Password"/>
36                                          </div>
```

## LogBleedingTask.java, line 50 (Privacy Violation)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Security Features | | |
| Abstract: | LogBleedingTask.java generatePassword() . . | | |

Copyright 2024 Open Text.

| | |
|---|---|
| Source: | LogBleedingTask.java:50 Read this.password() |

```
48                  public void generatePassword(){
49                      password =******
50                      log.info("Password for admin: {}",
Base64.getEncoder().encodeToString(password.getBytes(StandardCharsets.UTF_8)));
51                  }
```

| | |
|---|---|
| Sink: | LogBleedingTask.java:50 org.slf4j.Logger.info() |

```
48                  public void generatePassword(){
49                      password =******
50                      log.info("Password for admin: {}",
Base64.getEncoder().encodeToString(password.getBytes(StandardCharsets.UTF_8)));
51                  }
```

## PasswordReset.html, line 24 (HTML5: Form Validation Turned Off)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | HTML5 . | | |
| Sink: | PasswordReset.html:24 | | |

```
22
23
24                          <form class="attack-form" accept-charset="UNKNOWN"
novalidate="novalidate"
25                                  method="POST"
26                                  action="/WebGoat/PasswordReset/simple-mail/reset">
```

## ace.js, line 1740 (HTML5: Overly Permissive Message Posting Policy)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | ace.js 1740 . | | |
| Sink: | ace.js:1740 FunctionPointerCall: postMessage() | | |

```
1738
1739            exports.addListener(win, "message", listener);
1740            win.postMessage(messageName, "*");
1741        };
1742    }
```

## PasswordReset.html, line 202 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Security Features | | |
| Abstract: | PasswordReset.html 202 . | | |
| Sink: | PasswordReset.html:202 | | |

```
200                 </i>
201               </span>
202                                                 <input class="form-control"
placeholder="Password" name="password"
203                                                        type="password"
204                                                        value="" required=""/>
```

## SqlInjectionLesson8.java, line 140 (Poor Logging Practice: Use of a System Output Stream)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | println() . | | |
| Sink: | SqlInjectionLesson8.java:140 FunctionCall: println() | | |

```
138                 statement.executeUpdate(logQuery);
139             } catch (SQLException e) {
140                 System.err.println(e.getMessage());
141             }
142         }
```

Copyright 2024 Open Text.

## SqlInjectionLesson6a.java, line 67 (SQL Injection)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | SqlInjectionLesson6a.java 67 injectableQuery() SQL . SQL . | | |

**Source:** SqlOnlyInputValidation.java:48 attack(0)

```
46      @PostMapping("/SqlOnlyInputValidation/attack")
47      @ResponseBody
48      public AttackResult attack(@RequestParam("userid_sql_only_input_validation")
String userId) {
49          if (userId.contains(" ")) {
50              return failed(this).feedback("SqlOnlyInputValidation-failed").build();
```

**Sink:** SqlInjectionLesson6a.java:67 java.sql.Statement.executeQuery()

```
65              try (Statement statement =
connection.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
66                      ResultSet.CONCUR_READ_ONLY)) {
67                  ResultSet results = statement.executeQuery(query);
68
69                  if ((results != null) && results.first()) {
```

## PathTraversal.html, line 49 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Security Features | | |
| **Abstract:** | PathTraversal.html 49 . | | |

**Sink:** PathTraversal.html:49

```
47                  <div class="form-group">
48                      <label>Password:</label>
49                      <input class="form-control" type="password" id="password"
name="password" required
50                             placeholder="Enter Password" value="test"/>
51                      <span class="Error"></span>
```

## CrossSiteScriptingMitigation.html, line 50 (Hidden Field)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | CrossSiteScriptingMitigation.html 50 . | | |

**Sink:** CrossSiteScriptingMitigation.html:50

```
48      <script th:src="@{/lesson_js/assignment4.js}" type="text/javascript" charset="utf-
8"></script>
49    </div>
50    <input type="hidden" name="editor2"/>
51    <div class="input-group" style="position: absolute; top: 365px;">
52     <button class="btn btn-primary" type="submit">Submit</button>
```

## spoofcookieform.html, line 16 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Security Features | | |
| **Abstract:** | spoofcookieform.html 16 . | | |

**Sink:** spoofcookieform.html:16

```
14      <div class="form-group input-group">
15       <span class="input-group-addon"><i
16        class="glyphicon glyphicon-lock"></i></span> <input class="form-control"
17        placeholder="Password" name="password" type="password"
18        id="spoof_password" />
```

## ResetLinkAssignment.java, line 91 (Trust Boundary Violation)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | ResetLinkAssignment.java resetPassword() . . | | |

ot Copyright 2024 Open Text.

| | |
|---|---|
| Source: | ResetLinkAssignment.java:85 resetPassword(0) |

```
83
84                  @GetMapping("/PasswordReset/reset/reset-password/{link}")
85                  public ModelAndView resetPassword(@PathVariable(value = "link") String link, Model
model) {
86                   ModelAndView modelAndView = new ModelAndView();
87                      if (ResetLinkAssignment.resetLinks.contains(link)) {
```

| | |
|---|---|
| Sink: | ResetLinkAssignment.java:91 org.springframework.web.servlet.ModelAndView.addObject() |

```
89                          form.setResetLink(link);
90                          model.addAttribute("form", form);
91                          modelAndView.addObject("form", form);
92                          modelAndView.setViewName("password_reset"); //Display html page for
changing password
93                      } else {
```

## SqlInjectionLesson9.java, line 77 (Poor Logging Practice: Use of a System Output Stream)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | println() . | | |
| Sink: | SqlInjectionLesson9.java:77 FunctionCall: println() | | |

```
75                      }
76                  } catch (SQLException e) {
77                      System.err.println(e.getMessage());
78                      return failed(this).output("<br><span class='feedback-negative'>" +
e.getMessage() + "</span>").build();
79                  }
```

## registration.html, line 30 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Security Features | | |
| Abstract: | registration.html 30 . | | |
| Sink: | registration.html:30 | | |

```
28                      <label for="password" class="col-sm-2 control-label"
th:text="#{password}">Password</label>
29                      <div class="col-sm-4">
30                          <input type="password" class="form-control" id="password"
placeholder="Password"
31                                 name='password' th:value="*{password}"/>
32                      </div>
```

## LandingAssignment.java, line 63 (Trust Boundary Violation)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | LandingAssignment.java openPasswordReset() . . | | |
| Source: | LandingAssignment.java:63 Read this.landingPageUrl() | | |

```
61          URI uri = new URI(request.getRequestURL().toString());
62          ModelAndView modelAndView = new ModelAndView();
63          modelAndView.addObject("webwolfUrl", landingPageUrl);
64          modelAndView.addObject("uniqueCode",
StringUtils.reverse(getWebSession().getUserName()));
```

| | |
|---|---|
| Sink: | LandingAssignment.java:63 org.springframework.web.servlet.ModelAndView.addObject() |

```
61          URI uri = new URI(request.getRequestURL().toString());
62          ModelAndView modelAndView = new ModelAndView();
63          modelAndView.addObject("webwolfUrl", landingPageUrl);
64          modelAndView.addObject("uniqueCode",
StringUtils.reverse(getWebSession().getUserName()));
```

## SecurePasswordsAssignment.java, line 65 (Password Management: Password in Comment)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |

| Kingdom: | Security Features |
|---|---|
| Abstract: | . |
| Sink: | SecurePasswordsAssignment.java:65 Comment() |

```
63                  output.append("<b>Warning: </b>" + strength.getFeedback().getWarning() +
"</br>");
64          // possible feedback:
https://github.com/dropbox/zxcvbn/blob/master/src/feedback.coffee
65          // maybe ask user to try also weak passwords to see and understand feedback?
66          if (strength.getFeedback().getSuggestions().size() != 0) {
67              output.append("<b>Suggestions:</b></br><ul>");
```

## HtmlTampering.html, line 132 (Hidden Field)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | HtmlTampering.html 132 . | | |
| Sink: | HtmlTampering.html:132 | | |

```
130                                         </div>
131                                     </td>
132                                     <input id="Total" name="Total" type="HIDDEN"
value="2999.99"/>
133                                 </tr>
134                                 </tbody>
```

## SqlInjectionLesson10.java, line 100 (Poor Logging Practice: Use of a System Output Stream)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | println() . | | |
| Sink: | SqlInjectionLesson10.java:100 FunctionCall: println() | | |

```
98                      return false;
99                  } else {
100                     System.err.println(e.getMessage());
101                     return false;
102                 }
```

## FileServer.java, line 77 (Log Forging (debug))

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Input Validation and Representation | | |
| Abstract: | FileServer.java importFile() 77 . . | | |
| Source: | FileServer.java:72 importFile(0) | | |

```
70
71      @PostMapping(value = "/fileupload")
72      public ModelAndView importFile(@RequestParam("file") MultipartFile myFile) throws
IOException {
73          var user = (WebGoatUser)
SecurityContextHolder.getContext().getAuthentication().getPrincipal();
74          var destinationDir = new File(fileLocation, user.getUsername());
```

| Sink: | FileServer.java:77 org.slf4j.Logger.debug() |
|---|---|

```
75          destinationDir.mkdirs();
76          myFile.transferTo(new File(destinationDir, myFile.getOriginalFilename()));
77          log.debug("File saved to {}", new File(destinationDir,
myFile.getOriginalFilename()));
78
79          return new ModelAndView(
```

## registration.html, line 39 (Privacy Violation: Autocomplete)

| Fortify Priority: | High | Folder | High |
|---|---|---|---|
| Kingdom: | Security Features | | |
| Abstract: | registration.html 39 . | | |
| Sink: | registration.html:39 | | |

```
37                                  password</label>
38                              <div class="col-sm-4">
39                                  <input type="password" class="form-control" id="matchingPassword"
placeholder="Password"
40                                         name='matchingPassword' th:value="*{matchingPassword}"/>
41                              </div>
```

## AuthBypass.html, line 54 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Security Features | | |
| **Abstract:** | AuthBypass.html 54 . | | |

**Sink:** AuthBypass.html:54

```
52
53                              <p>Confirm Password:</p>
54                              <input name="newPasswordConfirm" value="" type="password" /><br/><br
/>
55
56                              <input type="hidden" name="userId" value="12309746" />
```

## ace.js, line 1740 (HTML5: Overly Permissive Message Posting Policy)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | ace.js 1740 . | | |

**Sink:** ace.js:1740 FunctionPointerCall: postMessage()

```
1738
1739            exports.addListener(win, "message", listener);
1740            win.postMessage(messageName, "*");
1741        };
1742    }
```

## registration.html, line 41 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Security Features | | |
| **Abstract:** | registration.html 41 . | | |

**Sink:** registration.html:41

```
39                              <label for="password" class="col-sm-2 control-label"
th:text="#{password}">Password</label>
40                              <div class="col-sm-4">
41                                  <input type="password" class="form-control" id="password"
th:placeholder="#{password}"
42                                         name='password' th:value="*{password}"/>
43                              </div>
```

## ClientSideFiltering.html, line 88 (Hidden Field)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | ClientSideFiltering.html 88 . | | |

**Sink:** ClientSideFiltering.html:88

```
86                          action="/WebGoat/clientSideFiltering/getItForFree">
87
88                              <input id="discount" type="hidden" value="0"/>
89                              <div class="row">
```

## ClientSideFiltering.html, line 14 (Hidden Field)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | ClientSideFiltering.html 14 . | | |

**Sink:** ClientSideFiltering.html:14

```
12      <div class="attack-container">
```

ot Copyright 2024 Open Text.

```
13                  <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
14                  <input type="hidden" id="user_id" value="102"/>
15                  <!-- using attack-form class on your form, will allow your request to be
ajaxified and stay within the display framework for webgoat -->
16                  <form class="attack-form" accept-charset="UNKNOWN" method="POST" name="form"
```

## AuthBypass.html, line 35 (Hidden Field)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | AuthBypass.html 35 . | | |

**Sink:** AuthBypass.html:35

```
33                      <input name="secQuestion1" value="" type="TEXT" /><br/><br />
34
35                      <input type="hidden" name="jsEnabled" value="1" />
36                      <input type="hidden" name="verifyMethod" value="SEC_QUESTIONS" />
37                      <input type="hidden" name="userId" value="12309746" />
```

## Assignment1Test.java, line 85 (Password Management: Password in Comment)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Security Features | | |
| **Abstract:** | . | | |

**Sink:** Assignment1Test.java:85 Comment()

```
83  //      mockMvc.perform(MockMvcRequestBuilders.post("/challenge/1")
84  //              .param("username", "admin")
85  //              .param("password", SolutionConstants.PASSWORD))
86  //              .andExpect(jsonPath("$.feedback",
CoreMatchers.is(messages.getMessage("ip.address.unknown"))))
87  //              .andExpect(jsonPath("$.lessonCompleted", CoreMatchers.is(false)));
```

## password_reset.html, line 15 (Hidden Field)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | password_reset.html 15 . | | |

**Sink:** password_reset.html:15

```
13                  <h2 class="sign_up_title">Reset your password</h2>
14                      <div class="form-group"
th:classappend="${#fields.hasErrors('password')}? 'has-error'">
15                          <input type="hidden" name="resetLink" th:field="*{resetLink}"
/>
16                          <label for="password" class="control-label"
th:text="#{password}">Password</label>
17                          <input type="password" class="form-control" id="password"
placeholder="Password"
```

## BypassRestrictions.html, line 108 (Hidden Field)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | BypassRestrictions.html 108 . | | |

**Sink:** BypassRestrictions.html:108

```
106                 <textarea cols="25" name="field7" rows="1">301-604-4882</textarea>
107             </div>
108             <input type="hidden" value="" name="error"/>
109             <p>
110                 <button type="submit" class="btn btn-primary">Submit</button>
```

## AsciiDoctorTemplateResolver.java, line 145 (Privacy Violation)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Security Features | | |
| **Abstract:** | AsciiDoctorTemplateResolver.java determineLanguage() . . | | |

ot Copyright 2024 Open Text.

| | |
|---|---|
| Source: | SigningAssignment.java:56 Read privateKey() |

```
54    KeyPair keyPair = CryptoUtil.generateKeyPair();
55    privateKey = CryptoUtil.getPrivateKeyInPEM(keyPair);
56    request.getSession().setAttribute("privateKeyString", privateKey);
57    request.getSession().setAttribute("keyPair", keyPair);
58   }
```

| | |
|---|---|
| Sink: | AsciiDoctorTemplateResolver.java:145 org.slf4j.Logger.debug() |

```
143         Locale browserLocale = (Locale) request.getSession().getAttribute(SessionLocaleResolver.LOCALE_SESSION_ATTRIBUTE_NAME);
144         if (null != browserLocale) {
145             log.debug("browser locale {}", browserLocale);
146             return browserLocale.getLanguage();
147         } else {
```

## SecurePasswordsAssignment.java, line 76 (Privacy Violation)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Security Features | | |
| Abstract: | SecurePasswordsAssignment.java completed() . . | | |
| Source: | SecurePasswordsAssignment.java:51 Read password() | | |

```
49
50          output.append("<b>Your Password: *******</b></br>");
51          output.append("<b>Length: </b>" + password.length() + "</br>");
52          output.append("<b>Estimated guesses needed to crack your password: </b>" + df.format(strength.getGuesses()) + "</br>");
53          output.append("<div style=\"float: left;padding-right: 10px;\"><b>Score: </b>" + strength.getScore() + "/4 </div>");
```

| | |
|---|---|
| Sink: | SecurePasswordsAssignment.java:76 Return() |

```
74              return success(this).feedback("securepassword-success").output(output.toString()).build();
75          else
76              return failed(this).feedback("securepassword-failed").output(output.toString()).build();
77      }
```

## jquery.form.js, line 931 (Password Management: Password in Comment)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Security Features | | |
| Abstract: | . | | |
| Sink: | jquery.form.js:931 Comment() | | |

```
929 };
930
931 /**
932  * formToArray() gathers form element data into an array of objects that can
933  * be passed to any of the following ajax functions: $.get, $.post, or load.
```

## registration.html, line 50 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Security Features | | |
| Abstract: | registration.html 50 . | | |
| Sink: | registration.html:50 | | |

```
48                                  password</label>
49                              <div class="col-sm-4">
50                                  <input type="password" class="form-control" id="matchingPassword" th:placeholder="#{password}"
51                                         name='matchingPassword' th:value="*{matchingPassword}"/>
52                              </div>
```

## CommentsCache.java, line 102 (XML Entity Expansion Injection)

| | | | |
|---|---|---|---|
| Fortify Priority: | Medium | Folder | Medium |

| | |
|---|---|
| Kingdom: | Input Validation and Representation |
| Abstract: | CommentsCache.java:102 XML DTD(Document Type Definition) . XML injection . |
| Source: | ContentTypeAssignment.java:60 createNewUser(1) |

```
58      @PostMapping(path = "xxe/content-type")
59      @ResponseBody
60      public AttackResult createNewUser(HttpServletRequest request, @RequestBody String commentStr, @RequestHeader("Content-Type") String contentType) throws Exception {
61          AttackResult attackResult = failed(this).build();
```

| | |
|---|---|
| Sink: | CommentsCache.java:102 javax.xml.stream.XMLInputFactory.createXMLStreamReader() |

```
100         }
101
102         var xsr = xif.createXMLStreamReader(new StringReader(xml));
103
104         var unmarshaller = jc.createUnmarshaller();
```

## ResetLinkAssignmentForgotPassword.java, line 40 (Password Management: Password in Comment)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Security Features | | |
| Abstract: | . | | |
| Sink: | ResetLinkAssignmentForgotPassword.java:40 Comment() | | |

```
38  import java.util.UUID;
39
40  /**
41   * Part of the password reset assignment. Used to send the e-mail.
42   *
```

## Challenge1.html, line 30 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Security Features | | |
| Abstract: | Challenge1.html 30 . | | |
| Sink: | Challenge1.html:30 | | |

```
28                              <div class="form-group">
29                                  <label for="exampleInputPassword1" th:text="#{password}">Password</label>
30                                  <input type="password" class="form-control" id="exampleInputPassword1"
31                                          placeholder="Password"
32                                          name='password'/>
```

## AuthBypass.html, line 54 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Security Features | | |
| Abstract: | AuthBypass.html 54 . | | |
| Sink: | AuthBypass.html:54 | | |

```
52
53                      <p>Confirm Password:</p>
54                      <input name="newPasswordConfirm" value="" type="password" /><br/><br />
55
56                      <input type="hidden" name="userId" value="12309746" />
```

## SqlInjectionAdvanced.html, line 129 (Password Management: Insecure Submission)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Security Features | | |
| Abstract: | SqlInjectionAdvanced.html 129 HTTP GET , . | | |
| Sink: | SqlInjectionAdvanced.html:129 | | |

```
127                                 </div>
```

```
128                                                 <div class="form-group">
129                                                     <input type="password" name="password_reg"
                    id="password" tabindex="2"
130                                                            class="form-control"
                    placeholder="Password"/>
131                                                 </div>
```

## AuthBypass.html, line 56 (Hidden Field)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | AuthBypass.html 56 . | | |

**Sink:** AuthBypass.html:56

```
54                          <input name="newPasswordConfirm" value="" type="password" /><br/><br
                    />
55
56                          <input type="hidden" name="userId" value="12309746" />
57
58                          <input name="submit" value="Submit" type="submit"/>
```

## UserService.java, line 52 (SQL Injection)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | UserService.java 52 createLessonsForUser() SQL . SQL . | | |

**Source:** RegistrationController.java:35 registration(0)

```
33
34                      @PostMapping("/register.mvc")
35                      public String registration(@ModelAttribute("userForm") @Valid UserForm userForm,
                    BindingResult bindingResult, HttpServletRequest request) throws ServletException {
36                          userValidator.validate(userForm, bindingResult);
```

**Sink:** UserService.java:52 org.springframework.jdbc.core.JdbcTemplate.execute()

```
50
51                      private void createLessonsForUser(WebGoatUser webGoatUser) {
52                          jdbcTemplate.execute("CREATE SCHEMA \"" + webGoatUser.getUsername() + "\"
                    authorization dba");
53                          flywayLessons.apply(webGoatUser.getUsername()).migrate();
54                      }
```

## ChromeDevTools.html, line 78 (Hidden Field)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | ChromeDevTools.html 78 . | | |

**Sink:** ChromeDevTools.html:78

```
76                              </tr>
77                          </table>
78                          <input type="hidden" name="network_num" id="networkNumCopy" value="foo" />
79                      </form>
80                      <div class="attack-feedback"></div>
```

## AuthBypass.html, line 51 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Security Features | | |
| **Abstract:** | AuthBypass.html 51 . | | |

**Sink:** AuthBypass.html:51

```
49
50                          <p>Password:</p>
51                          <input name="newPassword" value="" type="password" /><br/>
52
53                          <p>Confirm Password:</p>
```

## LabelAndHintIntegrationTest.java, line 136 (Poor Logging Practice: Use of a System Output Stream)

ot Copyright 2024 Open Text.

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | println() . | | |

Sink: LabelAndHintIntegrationTest.java:136 FunctionCall: println()

```
134                     System.out.println("key: " + key + " in (" +lang+") has incorrect
translation in label service");
135
System.out.println("actual:"+jsonPath.getString(ESCAPE_JSON_PATH_CHAR+key+ESCAPE_JSON_
PATH_CHAR));
136                     System.out.println("expected: "+propsLang.getProperty(key));
137                     System.out.println();
138                     Assertions.fail();
```

## AsciiDoctorTemplateResolver.java, line 145 (Privacy Violation)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Security Features | | |
| Abstract: | AsciiDoctorTemplateResolver.java determineLanguage() . . | | |

Source: EncodingAssignment.java:42 Read password()

```
40
41                  public static String getBasicAuth(String username, String password) {
42                      return
Base64.getEncoder().encodeToString(username.concat(":").concat(password).getBytes());
43                  }
44
```

Sink: AsciiDoctorTemplateResolver.java:145 org.slf4j.Logger.debug()

```
143             Locale browserLocale = (Locale)
request.getSession().getAttribute(SessionLocaleResolver.LOCALE_SESSION_ATTRIBUTE_NAME)
;
144             if (null != browserLocale) {
145                 log.debug("browser locale {}", browserLocale);
146                 return browserLocale.getLanguage();
147             } else {
```

## SqlInjectionLesson10.java, line 63 (SQL Injection)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | SqlInjectionLesson10.java 63 injectableQueryAvailability() SQL . SQL . | | |

Source: SqlInjectionLesson10.java:52 completed(0)

```
50      @PostMapping("/SqlInjection/attack10")
51      @ResponseBody
52      public AttackResult completed(@RequestParam String action_string) {
53          return injectableQueryAvailability(action_string);
54      }
```

Sink: SqlInjectionLesson10.java:63 java.sql.Statement.executeQuery()

```
61              try {
62                  Statement statement =
connection.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_READ_ONLY);
63                  ResultSet results = statement.executeQuery(query);
64
65                  if (results.getStatement() != null) {
```

## SigningAssignment.java, line 59 (Privacy Violation)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Security Features | | |
| Abstract: | SigningAssignment.java getPrivateKey() . . | | |

Source: SigningAssignment.java:59 Read privateKey()

```
57        request.getSession().setAttribute("keyPair", keyPair);
58      }
59      return privateKey;
```

```
60                  }
61
```

Sink: SigningAssignment.java:59 Return privateKey()

```
57                request.getSession().setAttribute("keyPair", keyPair);
58              }
59              return privateKey;
60                }
61
```

## MailboxController.java, line 53 (Trust Boundary Violation)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | MailboxController.java mail() . . | | |

Source: MailboxController.java:50 org.owasp.webgoat.webwolf.mailbox.MailboxRepository.findByRecipientOrderByTimeDesc()

```
48          UserDetails user = (UserDetails) SecurityContextHolder.getContext().getAuthentication().getPrincipal();
49          ModelAndView modelAndView = new ModelAndView();
50          List<Email> emails = mailboxRepository.findByRecipientOrderByTimeDesc(user.getUsername());
51          if (emails != null && !emails.isEmpty()) {
52              modelAndView.addObject("total", emails.size());
```

Sink: MailboxController.java:53 org.springframework.web.servlet.ModelAndView.addObject()

```
51          if (emails != null && !emails.isEmpty()) {
52              modelAndView.addObject("total", emails.size());
53              modelAndView.addObject("emails", emails);
54          }
55          modelAndView.setViewName("mailbox");
```

## PasswordReset.html, line 63 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Security Features | | |
| Abstract: | PasswordReset.html 63 . | | |

Sink: PasswordReset.html:63

```
61                                  <div class="form-group input-group">
62                                      <span class="input-group-addon"><i class="glyphicon glyphicon-lock"></i></span>
63                                      <input class="form-control" placeholder="Password" name="password"
64                                             type="password" value=""/>
65                                  </div>
```

## Salaries.java, line 107 (Log Forging)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | Salaries.java invoke() 107 . . | | |

Source: Salaries.java:76 Read this.webGoatHomeDirectory()

```
74      public List<Map<String, Object>> invoke() {
75          NodeList nodes = null;
76          File d = new File(webGoatHomeDirectory, "ClientSideFiltering/employees.xml");
77          XPathFactory factory = XPathFactory.newInstance();
78          XPath path = factory.newXPath();
```

Sink: Salaries.java:107 org.slf4j.Logger.error()

```
105             log.error("Unable to parse xml", e);
106         } catch (IOException e) {
107             log.error("Unable to read employees.xml at location: '{}'", d);
108         }
109         return json;
```

## Ping.java, line 51 (Log Forging (debug))

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | Ping.java logRequest() 51 . . | | |

Source: Ping.java:49 logRequest(1)

```
47      @RequestMapping(method = RequestMethod.GET)
48      @ResponseBody
49      public String logRequest(@RequestHeader("User-Agent") String userAgent,
@RequestParam(required = false) String text) {
50          String logLine = String.format("%s %s %s", "GET", userAgent, text);
51          log.debug(logLine);
```

Sink: Ping.java:51 org.slf4j.Logger.debug()

```
49      public String logRequest(@RequestHeader("User-Agent") String userAgent,
@RequestParam(required = false) String text) {
50          String logLine = String.format("%s %s %s", "GET", userAgent, text);
51          log.debug(logLine);
52          File logFile = new File(webGoatHomeDirectory, "/XXE/log" +
webSession.getUserName() + ".txt");
53          try {
```

## spoofcookieform.html, line 16 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Security Features | | |
| Abstract: | spoofcookieform.html 16 . | | |

Sink: spoofcookieform.html:16

```
14      <div class="form-group input-group">
15       <span class="input-group-addon"><i
16        class="glyphicon glyphicon-lock"></i></span> <input class="form-control"
17        placeholder="Password" name="password" type="password"
18        id="spoof_password" />
```

## PathTraversal.html, line 103 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Security Features | | |
| Abstract: | PathTraversal.html 103 . | | |

Sink: PathTraversal.html:103

```
101                 <div class="form-group">
102                     <label>Password:</label>
103                     <input class="form-control" type="password" id="passwordFix"
name="password" required
104                            placeholder="Enter Password" value="test"/>
105                     <span class="Error"></span>
```

## FileServer.java, line 118 (Trust Boundary Violation)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | FileServer.java getFiles() . . | | |

Source: FileServer.java:118 Read this.port()

```
116
117         modelAndView.addObject("files", uploadedFiles);
118         modelAndView.addObject("webwolf_url", "http://" + server + ":" + port);
119         return modelAndView;
120     }
```

Sink: FileServer.java:118 org.springframework.web.servlet.ModelAndView.addObject()

```
116
117         modelAndView.addObject("files", uploadedFiles);
118         modelAndView.addObject("webwolf_url", "http://" + server + ":" + port);
119         return modelAndView;
```

ot Copyright 2024 Open Text.

```
120                 }
```

## LabelAndHintIntegrationTest.java, line 135 (Poor Logging Practice: Use of a System Output Stream)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |

**Abstract:** println() .

**Sink:** LabelAndHintIntegrationTest.java:135 FunctionCall: println()

```
133                 if (!jsonPath.getString(ESCAPE_JSON_PATH_CHAR+key+ESCAPE_JSON_PATH_CHAR).equals(propsLang.get(key))) {
134                     System.out.println("key: " + key + " in (" +lang+") has incorrect translation in label service");
135                     System.out.println("actual:"+jsonPath.getString(ESCAPE_JSON_PATH_CHAR+key+ESCAPE_JSON_PATH_CHAR));
136                     System.out.println("expected: "+propsLang.getProperty(key));
137                     System.out.println();
```

## registration.html, line 30 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Security Features | | |

**Abstract:** registration.html 30 .

**Sink:** registration.html:30

```
28                      <label for="password" class="col-sm-2 control-label" th:text="#{password}">Password</label>
29                      <div class="col-sm-4">
30                          <input type="password" class="form-control" id="password" placeholder="Password"
31                                 name='password' th:value="*{password}"/>
32                      </div>
```

## SqlInjectionLesson9.java, line 84 (Poor Logging Practice: Use of a System Output Stream)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |

**Abstract:** println() .

**Sink:** SqlInjectionLesson9.java:84 FunctionCall: println()

```
82
83          } catch (Exception e) {
84              System.err.println(e.getMessage());
85              return failed(this).output("<br><span class='feedback-negative'>" + e.getMessage() + "</span>").build();
86          }
```

## LabelAndHintIntegrationTest.java, line 130 (Poor Logging Practice: Use of a System Output Stream)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |

**Abstract:** println() .

**Sink:** LabelAndHintIntegrationTest.java:130 FunctionCall: println()

```
128         for (String key: propsLang.stringPropertyNames()) {
129             if (!propsDefault.containsKey(key)) {
130                 System.err.println("key: " + key + " in (" +lang+") is missing from default properties");
131                 Assertions.fail();
132             }
```

## SqlInjectionChallenge.java, line 75 (Access Control: Database)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Security Features | | |

**Abstract:** access control SqlInjectionChallenge.java registerNewUser() 75 SQL .

**Source:** SqlInjectionChallenge.java:56 registerNewUser(0)

ot Copyright 2024 Open Text.

```
54                      //assignment path is bounded to class so we use different http method :-)
55                      @ResponseBody
56                      public AttackResult registerNewUser(@RequestParam String username_reg,
@RequestParam String email_reg, @RequestParam String password_reg) throws Exception {
57                          AttackResult attackResult = checkArguments(username_reg, email_reg,
password_reg);
```

Sink: SqlInjectionChallenge.java:75 java.sql.PreparedStatement.setString()

```
73                              } else {
74                                  PreparedStatement preparedStatement =
connection.prepareStatement("INSERT INTO sql_challenge_users VALUES (?, ?, ?)");
75                                  preparedStatement.setString(1, username_reg);
76                                  preparedStatement.setString(2, email_reg);
77                                  preparedStatement.setString(3, password_reg);
```

## SqlInjectionLesson8.java, line 138 (SQL Injection)

| Fortify Priority: | Critical | Folder | Critical |
|---|---|---|---|
| Kingdom: | Input Validation and Representation | | |
| Abstract: | SqlInjectionLesson8.java 138 log() SQL . SQL . | | |

Source: SqlInjectionLesson9.java:55 completed(1)

```
53      @PostMapping("/SqlInjection/attack9")
54      @ResponseBody
55      public AttackResult completed(@RequestParam String name, @RequestParam String
auth_tan) {
56          return injectableQueryIntegrity(name, auth_tan);
57      }
```

Sink: SqlInjectionLesson8.java:138 java.sql.Statement.executeUpdate()

```
136             try {
137                 Statement statement = connection.createStatement(TYPE_SCROLL_SENSITIVE,
CONCUR_UPDATABLE);
138                 statement.executeUpdate(logQuery);
139             } catch (SQLException e) {
140                 System.err.println(e.getMessage());
```

## Requests.java, line 76 (Trust Boundary Violation)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | Requests.java get() . . | | |

Source: Requests.java:53 Requests(1)

```
51  */
52 @Controller
53 @RequiredArgsConstructor
54 @Slf4j
55 @RequestMapping(value = "/requests")
```

Sink: Requests.java:76 org.springframework.web.servlet.ModelAndView.addObject()

```
74              .filter(t -> allowedTrace(t, user))
75                      .map(t -> new Tracert(t.getTimestamp(), path(t),
toJsonString(t))).collect(toList());
76              model.addObject("traces", traces);
77
78              return model;
```

## LogSpoofing.html, line 44 (Privacy Violation: Autocomplete)

| Fortify Priority: | High | Folder | High |
|---|---|---|---|
| Kingdom: | Security Features | | |
| Abstract: | LogSpoofing.html 44 . | | |

Sink: LogSpoofing.html:44

```
42
43                  <input type="text" value="" name="username" placeholder="username"/>
44                  <input type="password" value="" name="password" placeholder="password"/>
45                  <input type="submit" value="Submit"/>
```

ot Copyright 2024 Open Text.

## Assignment5.java, line 59 (SQL Injection)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | Assignment5.java 59 login() SQL . SQL . | | |

**Source:** Assignment5.java:51 login(1)

```
49      @PostMapping("/challenge/5")
50      @ResponseBody
51      public AttackResult login(@RequestParam String username_login, @RequestParam
String password_login) throws Exception {
52          if (!StringUtils.hasText(username_login) ||
!StringUtils.hasText(password_login)) {
53              return failed(this).feedback("required4").build();
```

**Sink:** Assignment5.java:59 java.sql.Connection.prepareStatement()

```
57          }
58          try (var connection = dataSource.getConnection()) {
59              PreparedStatement statement = connection.prepareStatement("select password
from challenge_users where userid = '" + username_login + "' and password =****** +
password_login + "'");
60              ResultSet resultSet = statement.executeQuery();
```

## clientSideFiltering.js, line 38 (Privacy Violation)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Security Features | | |
| **Abstract:** | clientSideFiltering.js 38 . . | | |

**Source:** clientSideFiltering.js:31 Read SSN()

```
29              html = html + '<td>' + result[i].FirstName + '</td>';
30              html = html + '<td>' + result[i].LastName + '</td>';
31              html = html + '<td>' + result[i].SSN + '</td>';
32              html = html + '<td>' + result[i].Salary + '</td>';
33              html = html + '</tr>';
```

**Sink:** clientSideFiltering.js:38 Assignment to newdiv.innerHTML()

```
36
37          var newdiv = document.createElement("div");
38          newdiv.innerHTML = html;
39          var container = document.getElementById("hiddenEmployeeRecords");
40          container.appendChild(newdiv);
```

## CrossSiteScriptingLesson4.java, line 47 (Log Forging (debug))

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | CrossSiteScriptingLesson4.java completed() 47 . . | | |

**Source:** CrossSiteScriptingLesson4.java:44 completed(0)

```
42      @PostMapping("/CrossSiteScripting/attack4")
43      @ResponseBody
44      public AttackResult completed(@RequestParam String editor2) {
45
46          String editor = editor2.replaceAll("\\<.*?>", "");
```

**Sink:** CrossSiteScriptingLesson4.java:47 org.slf4j.Logger.debug()

```
45
46          String editor = editor2.replaceAll("\\<.*?>", "");
47          log.debug(editor);
48
49          if ((editor.contains("Policy.getInstance(\"antisamy-slashdot.xml\"") ||
editor.contains(".scan(newComment, \"antisamy-slashdot.xml\"") ||
editor.contains(".scan(newComment, new File(\"antisamy-slashdot.xml\")")) &&
```

## SqlInjectionChallenge.java, line 76 (Access Control: Database)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Security Features | | |

ot Copyright 2024 Open Text.

| | |
|---|---|
| **Abstract:** | access control SqlInjectionChallenge.java registerNewUser() 76 SQL . |
| **Source:** | SqlInjectionChallenge.java:56 registerNewUser(1) |

```
54      //assignment path is bounded to class so we use different http method :-)
55      @ResponseBody
56      public AttackResult registerNewUser(@RequestParam String username_reg,
@RequestParam String email_reg, @RequestParam String password_reg) throws Exception {
57          AttackResult attackResult = checkArguments(username_reg, email_reg,
password_reg);
```

| | |
|---|---|
| **Sink:** | SqlInjectionChallenge.java:76 java.sql.PreparedStatement.setString() |

```
74                      PreparedStatement preparedStatement =
connection.prepareStatement("INSERT INTO sql_challenge_users VALUES (?, ?, ?)");
75                      preparedStatement.setString(1, username_reg);
76                      preparedStatement.setString(2, email_reg);
77                      preparedStatement.setString(3, password_reg);
78                      preparedStatement.execute();
```

## ResetLinkAssignment.java, line 107 (Mass Assignment: Insecure Binder Configuration)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | API Abuse | | |
| **Abstract:** | HTTP . | | |

| | |
|---|---|
| **Sink:** | ResetLinkAssignment.java:107 Function: changePassword() |

```
105
106     @PostMapping("/PasswordReset/reset/change-password")
107     public ModelAndView changePassword(@ModelAttribute("form") PasswordChangeForm
form, BindingResult bindingResult) {
108      ModelAndView modelAndView = new ModelAndView();
109         if (!org.springframework.util.StringUtils.hasText(form.getPassword())) {
```

## Email.java, line 45 (Mass Assignment: Request Parameters Bound into Persisted Objects)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | API Abuse | | |
| **Abstract:** | Email.java . . | | |

| | |
|---|---|
| **Sink:** | Email.java:45 Class: Email() |

```
43  @Entity
44  @NoArgsConstructor
45  public class Email implements Serializable {
46
47      @Id
```

## Challenge5.html, line 34 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Security Features | | |
| **Abstract:** | Challenge5.html 34 . | | |

| | |
|---|---|
| **Sink:** | Challenge5.html:34 |

```
32                                              </div>
33                                              <div class="form-group">
34                                                  <input type="password"
name="password_login" id="password4" tabindex="2"
35                                                         class="form-control"
placeholder="Password"/>
36                                              </div>
```

## SSRF.html, line 40 (Hidden Field)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | SSRF.html 40 . | | |

| | |
|---|---|
| **Sink:** | SSRF.html:40 |

```
38                  <table>
39                      <tr>
```

ot Copyright 2024 Open Text.

```
40                                      <td><input type="hidden" id="url2" name="url"
value="images/cat.png"/></td>
41
42                                      <td><input
```

## LessonConnectionInvocationHandler.java, line 30 (SQL Injection)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | LessonConnectionInvocationHandler.java 30 invoke() SQL . SQL . | | |
| **Sink:** | LessonConnectionInvocationHandler.java:30 execute() | | |

```
28          if (authentication != null && authentication.getPrincipal() instanceof
WebGoatUser user) {
29              try (var statement = targetConnection.createStatement()) {
30                  statement.execute("SET SCHEMA \"" + user.getUsername() + "\"");
31              }
32          }
```

## password_reset.html, line 12 (HTML5: Form Validation Turned Off)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | HTML5 . | | |
| **Sink:** | password_reset.html:12 | | |

```
10      <div class="row">
11          <div class="col-xs-12 col-sm-8 col-md-6 col-sm-offset-2 col-md-offset-3">
12              <form role="form" method="POST"
action="/WebGoat/PasswordReset/reset/change-password" th:object="${form}"
novalidate="novalidate">
13                  <h2 class="sign_up_title">Reset your password</h2>
14                      <div class="form-group"
th:classappend="${#fields.hasErrors('password')}? 'has-error'">
```

## Challenge6.html, line 77 (Password Management: Insecure Submission)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Security Features | | |
| **Abstract:** | Challenge6.html 77 HTTP GET , . | | |
| **Sink:** | Challenge6.html:77 | | |

```
75                                              </div>
76                                              <div class="form-group">
77                                                  <input type="password" name="password_reg"
id="password" tabindex="2"
78                                                         class="form-control"
placeholder="Password"/>
79                                              </div>
```

## SqlInjectionLesson8.java, line 138 (SQL Injection)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | SqlInjectionLesson8.java 138 log() SQL . SQL . | | |
| **Source:** | SqlInjectionLesson8.java:54 completed(1) | | |

```
52      @PostMapping("/SqlInjection/attack8")
53      @ResponseBody
54      public AttackResult completed(@RequestParam String name, @RequestParam String
auth_tan) {
55          return injectableQueryConfidentiality(name, auth_tan);
56      }
```

**Sink:** SqlInjectionLesson8.java:138 java.sql.Statement.executeUpdate()

```
136         try {
137             Statement statement = connection.createStatement(TYPE_SCROLL_SENSITIVE,
CONCUR_UPDATABLE);
138             statement.executeUpdate(logQuery);
139         } catch (SQLException e) {
```

ot Copyright 2024 Open Text.

```
140                         System.err.println(e.getMessage());
```

## CSRF.html, line 100 (Hidden Field)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | CSRF.html 100 . | | |

**Sink:** CSRF.html:100

```
98                                                  type="text"/>
99                                              <input class="form-control" id="reviewStars"
name="stars" type="text"/>
100                                             <input type="hidden" name="validateReq"
value="2aa14227b9a13d0bede0388a7fba9aa9"/>
101                                             <input type="submit" name="submit" value="Submit
review"/>
102                                         </form>
```

## Assignment5.java, line 59 (SQL Injection)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | Assignment5.java 59 login() SQL . SQL . | | |

**Source:** Assignment5.java:51 login(0)

```
49      @PostMapping("/challenge/5")
50      @ResponseBody
51      public AttackResult login(@RequestParam String username_login, @RequestParam
String password_login) throws Exception {
52          if (!StringUtils.hasText(username_login) ||
!StringUtils.hasText(password_login)) {
53              return failed(this).feedback("required4").build();
```

**Sink:** Assignment5.java:59 java.sql.Connection.prepareStatement()

```
57          }
58          try (var connection = dataSource.getConnection()) {
59              PreparedStatement statement = connection.prepareStatement("select password
from challenge_users where userid = '" + username_login + "' and password =****** +
password_login + "'");
60              ResultSet resultSet = statement.executeQuery();
```

## PasswordReset.html, line 63 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Security Features | | |
| **Abstract:** | PasswordReset.html 63 . | | |

**Sink:** PasswordReset.html:63

```
61                                          <div class="form-group input-group">
62                                              <span class="input-group-addon"><i
class="glyphicon glyphicon-lock"></i></span>
63                                              <input class="form-control" placeholder="Password"
name="password"
64                                                  type="password" value=""/>
65                                          </div>
```

## SqlInjectionAdvanced.html, line 129 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Security Features | | |
| **Abstract:** | SqlInjectionAdvanced.html 129 . | | |

**Sink:** SqlInjectionAdvanced.html:129

```
127                                             </div>
128                                             <div class="form-group">
129                                                 <input type="password" name="password_reg"
id="password" tabindex="2"
130                                                         class="form-control"
placeholder="Password"/>
131                                             </div>
```

| HttpBasics.html, line 64 (Hidden Field) | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | HttpBasics.html 64 . | | |
| Sink: | HttpBasics.html:64 | | |

```
62          webgoat.customjs.assignRandomVal();
63        </script>
64        <input type="hidden" name="magic_num" id="magic_num" value="foo" />
65        <table>
66         <tr>
```

| SqlInjectionAdvanced.html, line 129 (Password Management: Insecure Submission) | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Security Features | | |
| Abstract: | SqlInjectionAdvanced.html 129 HTTP GET , . | | |
| Sink: | SqlInjectionAdvanced.html:129 | | |

```
127                                    </div>
128                                    <div class="form-group">
129                                        <input type="password" name="password_reg"
id="password" tabindex="2"
130                                               class="form-control"
placeholder="Password"/>
131                                    </div>
```

| SqlInjectionLesson5a.java, line 62 (SQL Injection) | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | SqlInjectionLesson5a.java 62 injectableQuery() SQL . SQL . | | |
| Source: | SqlInjectionLesson5a.java:53 completed(2) | | |

```
51      @PostMapping("/SqlInjection/assignment5a")
52      @ResponseBody
53      public AttackResult completed(@RequestParam String account, @RequestParam String
operator, @RequestParam String injection) {
54          return injectableQuery(account + " " + operator + " " + injection);
55      }
```

| Sink: | SqlInjectionLesson5a.java:62 java.sql.Statement.executeQuery() |
|---|---|

```
60              query = "SELECT * FROM user_data WHERE first_name = 'John' and last_name =
'" + accountName + "'";
61              try (Statement statement =
connection.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_UPDATABLE)) {
62                  ResultSet results = statement.executeQuery(query);
63
64                  if ((results != null) && (results.first())) {
```

| Challenge6.html, line 77 (Privacy Violation: Autocomplete) | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Security Features | | |
| Abstract: | Challenge6.html 77 . | | |
| Sink: | Challenge6.html:77 | | |

```
75                                    </div>
76                                    <div class="form-group">
77                                        <input type="password" name="password_reg"
id="password" tabindex="2"
78                                               class="form-control"
placeholder="Password"/>
79                                    </div>
```

| SqlInjectionLesson9.java, line 66 (SQL Injection) | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |

| | |
|---|---|
| Kingdom: | Input Validation and Representation |
| Abstract: | SqlInjectionLesson9.java 66 injectableQueryIntegrity() SQL . SQL . |
| Source: | SqlInjectionLesson9.java:55 completed(1) |

```
53      @PostMapping("/SqlInjection/attack9")
54      @ResponseBody
55      public AttackResult completed(@RequestParam String name, @RequestParam String
auth_tan) {
56          return injectableQueryIntegrity(name, auth_tan);
57      }
```

Sink: SqlInjectionLesson9.java:66 java.sql.Statement.executeQuery()

```
64                  Statement statement =
connection.createStatement(TYPE_SCROLL_SENSITIVE, CONCUR_UPDATABLE);
65                  SqlInjectionLesson8.log(connection, query);
66                  ResultSet results = statement.executeQuery(query);
67                  var test = results.getRow() != 0;
68                  if (results.getStatement() != null) {
```

## LogSpoofing.html, line 44 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Security Features | | |
| Abstract: | LogSpoofing.html 44 . | | |

Sink: LogSpoofing.html:44

```
42
43              <input type="text" value="" name="username" placeholder="username"/>
44              <input type="password" value="" name="password" placeholder="password"/>
45              <input type="submit" value="Submit"/>
```

## SigningAssignment.java, line 73 (Log Forging)

| | | | |
|---|---|---|---|
| Fortify Priority: | Medium | Folder | Medium |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | SigningAssignment.java completed() 73 . . | | |

Source: SigningAssignment.java:64 completed(1)

```
62      @PostMapping("/crypto/signing/verify")
63      @ResponseBody
64      public AttackResult completed(HttpServletRequest request, @RequestParam String
modulus, @RequestParam String signature) {
65
66        String tempModulus = modulus;/* used to validate the modulus of the public key but
might need to be corrected */
```

Sink: SigningAssignment.java:73 org.slf4j.Logger.warn()

```
71        }
72        if
(!DatatypeConverter.printHexBinary(rsaPubKey.getModulus().toByteArray()).equals(tempMo
dulus.toUpperCase())) {
73          log.warn("modulus {} incorrect", modulus);
74          return failed(this).feedback("crypto-signing.modulusnotok").build();
75        }
```

## SqlInjectionChallenge.java, line 77 (Access Control: Database)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Security Features | | |
| Abstract: | access control SqlInjectionChallenge.java registerNewUser() 77 SQL . | | |

Source: SqlInjectionChallenge.java:56 registerNewUser(2)

```
54      //assignment path is bounded to class so we use different http method :-)
55      @ResponseBody
56      public AttackResult registerNewUser(@RequestParam String username_reg,
@RequestParam String email_reg, @RequestParam String password_reg) throws Exception {
57          AttackResult attackResult = checkArguments(username_reg, email_reg,
password_reg);
```

Sink: SqlInjectionChallenge.java:77 java.sql.PreparedStatement.setString()

```
75                  preparedStatement.setString(1, username_reg);
```

```
76                                  preparedStatement.setString(2, email_reg);
77                                  preparedStatement.setString(3, password_reg);
78                                  preparedStatement.execute();
79                                  attackResult =
success(this).feedback("user.created").feedbackArgs(username_reg).build();
```

## SqlInjectionLesson9.java, line 66 (SQL Injection)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | SqlInjectionLesson9.java 66 injectableQueryIntegrity() SQL . SQL . | | |
| **Source:** | SqlInjectionLesson9.java:55 completed(0) | | |

```
53      @PostMapping("/SqlInjection/attack9")
54      @ResponseBody
55      public AttackResult completed(@RequestParam String name, @RequestParam String
auth_tan) {
56          return injectableQueryIntegrity(name, auth_tan);
57      }
```

**Sink:** SqlInjectionLesson9.java:66 java.sql.Statement.executeQuery()

```
64                  Statement statement =
connection.createStatement(TYPE_SCROLL_SENSITIVE, CONCUR_UPDATABLE);
65                  SqlInjectionLesson8.log(connection, query);
66                  ResultSet results = statement.executeQuery(query);
67                  var test = results.getRow() != 0;
68                  if (results.getStatement() != null) {
```

## SqlInjectionAdvanced.html, line 133 (Password Management: Insecure Submission)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Security Features | | |
| **Abstract:** | SqlInjectionAdvanced.html 133 HTTP GET , . | | |
| **Sink:** | SqlInjectionAdvanced.html:133 | | |

```
131                                             </div>
132                                             <div class="form-group">
133                                                 <input type="password"
name="confirm_password_reg" id="confirm-password"
134                                                        tabindex="2" class="form-control"
placeholder="Confirm Password"/>
135                                             </div>
```

## IDOR.html, line 30 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Security Features | | |
| **Abstract:** | IDOR.html 30 . | | |
| **Sink:** | IDOR.html:30 | | |

```
28                          <td>user/pass</td>
29                          <td>user:<input name="username" value="" type="TEXT" /></td>
30                          <td>pass:<input name="password" value="" type="password" /></td>
31                          <td>
32                              <input
```

## WebSecurityConfig.java, line 60 (HTML5: Missing Content Security Policy)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | CSP(Content Security Policy) . | | |
| **Sink:** | WebSecurityConfig.java:60 Function: configure() | | |

```
58
59      @Override
60      protected void configure(HttpSecurity http) throws Exception {
61
ExpressionUrlAuthorizationConfigurer<HttpSecurity>.ExpressionInterceptUrlRegistry
security = http
62                  .authorizeRequests()
```

## ClientSideFiltering.html, line 22 (Hidden Field)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | ClientSideFiltering.html 22 . | | |

Sink: ClientSideFiltering.html:22

```
20                  <script th:src="@{/lesson_js/clientSideFiltering.js}"
21                          language="JavaScript"></script>
22                  <input id="userID" value="101" name="userID" type="HIDDEN"/>
23                  <div id="lesson_wrapper">
24                      <div id="lesson_header"></div>
```

## hijackform.html, line 16 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Security Features | | |
| Abstract: | hijackform.html 16 . | | |

Sink: hijackform.html:16

```
14          <div class="form-group input-group">
15           <span class="input-group-addon"><i
16            class="glyphicon glyphicon-lock"></i></span> <input class="form-control"
17            placeholder="Password" name="password" type="password" />
18          </div>
```

## SqlInjectionAdvanced.html, line 89 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Security Features | | |
| Abstract: | SqlInjectionAdvanced.html 89 . | | |

Sink: SqlInjectionAdvanced.html:89

```
87                                          </div>
88                                          <div class="form-group">
89                                              <input type="password"
name="password_login" id="password4" tabindex="2"
90                                                     class="form-control"
placeholder="Password"/>
91                                          </div>
```

## Challenge6.html, line 38 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Security Features | | |
| Abstract: | Challenge6.html 38 . | | |

Sink: Challenge6.html:38

```
36                                          </div>
37                                          <div class="form-group">
38                                              <input type="password"
name="password_login" id="password4" tabindex="2"
39                                                     class="form-control"
placeholder="Password"/>
40                                          </div>
```

## AuthBypass.html, line 37 (Hidden Field)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | AuthBypass.html 37 . | | |

Sink: AuthBypass.html:37

```
35                      <input type="hidden" name="jsEnabled" value="1" />
36                      <input type="hidden" name="verifyMethod" value="SEC_QUESTIONS" />
37                      <input type="hidden" name="userId" value="12309746" />
38
39                      <input name="submit" value="Submit" type="submit"/>
```

ot Copyright 2024 Open Text.

## SqlInjectionLesson5b.java, line 58 (SQL Injection)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | SqlInjectionLesson5b.java 58 injectableQuery() SQL . SQL . | | |

**Source:** SqlInjectionLesson5b.java:51 completed(0)

```
49      @PostMapping("/SqlInjection/assignment5b")
50      @ResponseBody
51      public AttackResult completed(@RequestParam String userid, @RequestParam String
login_count, HttpServletRequest request) throws IOException {
52          return injectableQuery(login_count, userid);
53      }
```

**Sink:** SqlInjectionLesson5b.java:58 java.sql.Connection.prepareStatement()

```
56          String queryString = "SELECT * From user_data WHERE Login_Count = ? and
userid= " + accountName;
57          try (Connection connection = dataSource.getConnection()) {
58              PreparedStatement query = connection.prepareStatement(queryString,
ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_READ_ONLY);
59
60              int count = 0;
```

## ResetLinkAssignment.java, line 90 (Trust Boundary Violation)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | ResetLinkAssignment.java resetPassword() . . | | |

**Source:** ResetLinkAssignment.java:85 resetPassword(0)

```
83
84      @GetMapping("/PasswordReset/reset/reset-password/{link}")
85      public ModelAndView resetPassword(@PathVariable(value = "link") String link, Model
model) {
86       ModelAndView modelAndView = new ModelAndView();
87          if (ResetLinkAssignment.resetLinks.contains(link)) {
```

**Sink:** ResetLinkAssignment.java:90 org.springframework.ui.Model.addAttribute()

```
88              PasswordChangeForm form = new PasswordChangeForm();
89              form.setResetLink(link);
90              model.addAttribute("form", form);
91              modelAndView.addObject("form", form);
92              modelAndView.setViewName("password_reset"); //Display html page for
changing password
```

## PathTraversal.html, line 160 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Security Features | | |
| **Abstract:** | PathTraversal.html 160 . | | |

**Sink:** PathTraversal.html:160

```
158                 <div class="form-group">
159                     <label>Password:</label>
160                     <input class="form-control" type="password"
id="passwordRemoveUserInput" name="password" required
161                            placeholder="Enter Password" value="test"/>
162                     <span class="Error"></span>
```

## SqlInjectionLesson5a.java, line 62 (SQL Injection)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | SqlInjectionLesson5a.java 62 injectableQuery() SQL . SQL . | | |

**Source:** SqlInjectionLesson5a.java:53 completed(1)

```
51      @PostMapping("/SqlInjection/assignment5a")
52      @ResponseBody
```

```
53                      public AttackResult completed(@RequestParam String account, @RequestParam String
operator, @RequestParam String injection) {
54                          return injectableQuery(account + " " + operator + " " + injection);
55                      }
```

Sink: SqlInjectionLesson5a.java:62 java.sql.Statement.executeQuery()

```
60                              query = "SELECT * FROM user_data WHERE first_name = 'John' and last_name =
'" + accountName + "'";
61                              try (Statement statement =
connection.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_UPDATABLE)) {
62                                  ResultSet results = statement.executeQuery(query);
63
64                                  if ((results != null) && (results.first())) {
```

## LogSpoofing.html, line 22 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Security Features | | |
| Abstract: | LogSpoofing.html 22 . | | |

Sink: LogSpoofing.html:22

```
20
21                      <input type="text" value="" name="username" placeholder="username"/>
22                      <input type="password" value="" name="password" placeholder="password"/>
23                      <input type="submit" value="Submit"/>
```

## jquery.form.js, line 931 (Password Management: Password in Comment)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Security Features | | |
| Abstract: | . | | |

Sink: jquery.form.js:931 Comment()

```
929 };
930
931 /**
932  * formToArray() gathers form element data into an array of objects that can
933  * be passed to any of the following ajax functions: $.get, $.post, or load.
```

## SqlInjectionAdvanced.html, line 133 (Password Management: Insecure Submission)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Security Features | | |
| Abstract: | SqlInjectionAdvanced.html 133 HTTP GET , . | | |

Sink: SqlInjectionAdvanced.html:133

```
131                                                 </div>
132                                                 <div class="form-group">
133                                                     <input type="password"
name="confirm_password_reg" id="confirm-password"
134                                                            tabindex="2" class="form-control"
placeholder="Confirm Password"/>
135                                                 </div>
```

## SqlInjectionLesson8.java, line 138 (SQL Injection)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | SqlInjectionLesson8.java 138 log() SQL . SQL . | | |

Source: SqlInjectionLesson9.java:55 completed(0)

```
53      @PostMapping("/SqlInjection/attack9")
54      @ResponseBody
55      public AttackResult completed(@RequestParam String name, @RequestParam String
auth_tan) {
56          return injectableQueryIntegrity(name, auth_tan);
57      }
```

Sink: SqlInjectionLesson8.java:138 java.sql.Statement.executeUpdate()

ot Copyright 2024 Open Text.

```
136                 try {
137                     Statement statement = connection.createStatement(TYPE_SCROLL_SENSITIVE,
CONCUR_UPDATABLE);
138                     statement.executeUpdate(logQuery);
139                 } catch (SQLException e) {
140                     System.err.println(e.getMessage());
```

## webwolfPasswordReset.html, line 17 (Password Management: Insecure Submission)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Security Features | | |
| Abstract: | webwolfPasswordReset.html 17 HTTP GET , . | | |

Sink: webwolfPasswordReset.html:17

```
15                          <div class="form-group">
16                              <label for="password" class="control-label">Password</label>
17                              <input type="password" class="form-control" id="password"
placeholder="Password"
18                                     name='password'/>
19                          </div>
```

## registration.html, line 39 (Privacy Violation: Autocomplete)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Security Features | | |
| Abstract: | registration.html 39 . | | |

Sink: registration.html:39

```
37                              password</label>
38                          <div class="col-sm-4">
39                              <input type="password" class="form-control" id="matchingPassword"
placeholder="Password"
40                                     name='matchingPassword' th:value="*{matchingPassword}"/>
41                          </div>
```

## password_reset.html, line 12 (HTML5: Form Validation Turned Off)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | HTML5 . | | |

Sink: password_reset.html:12

```
10      <div class="row">
11          <div class="col-xs-12 col-sm-8 col-md-6 col-sm-offset-2 col-md-offset-3">
12              <form role="form" method="POST"
action="/WebGoat/PasswordReset/reset/change-password" th:object="${form}"
novalidate="novalidate">
13                  <h2 class="sign_up_title">Reset your password</h2>
14                  <div class="form-group"
th:classappend="${#fields.hasErrors('password')}? 'has-error'">
```

## ProgressRaceConditionIntegrationTest.java, line 50 (Poor Logging Practice: Use of a System Output Stream)

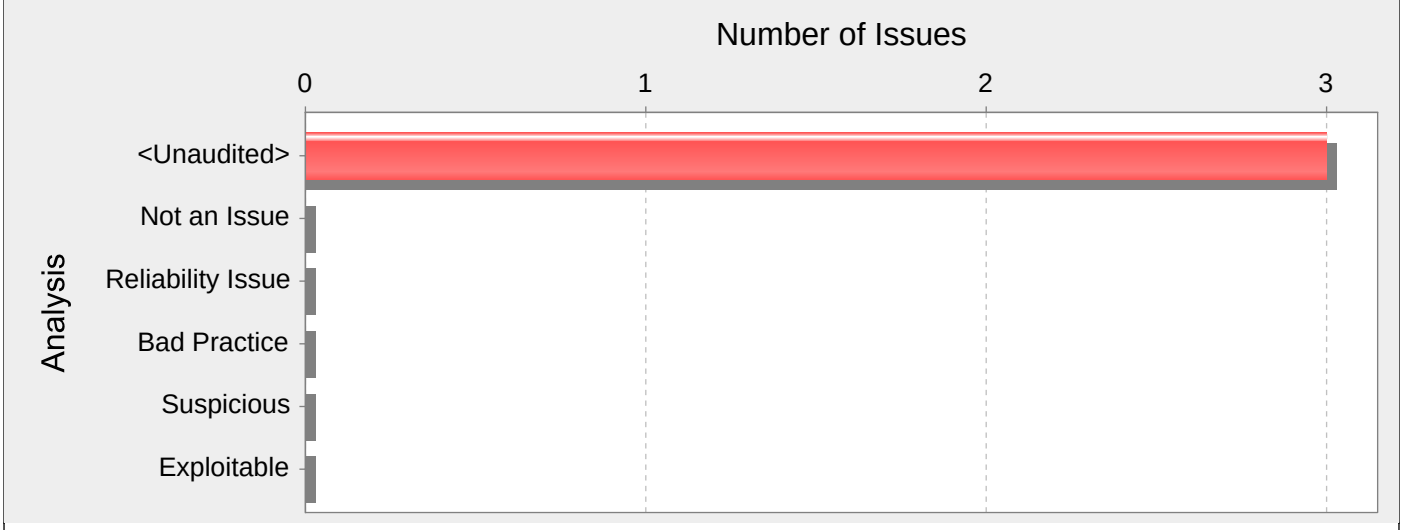
| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | println() . | | |

Sink: ProgressRaceConditionIntegrationTest.java:50 FunctionCall: println()

```
48                  }
49              }).count();
50              System.err.println("counted status 500: "+countStatusCode500);
51              Assertions.assertThat(countStatusCode500).isLessThanOrEqualTo((NUMBER_OF_CALLS
- (NUMBER_OF_CALLS/NUMBER_OF_PARALLEL_THREADS)));
52      }
```

ot Copyright 2024 Open Text.

## 49: 07.02. API (3 Issues)




**Abstract:**
MavenWrapperDownloader.java 89 , main() exit() . .

**Explanation:**
. leftover debug code Non-J2EE .

**Recommendations:**
. J2EE . .

**Tips:**
1. J2EE exit() halt() . exit() halt() Non-J2EE .

2. J2EE Java J2EE . Audit Guide .

### MavenWrapperDownloader.java, line 89 (J2EE Bad Practices: JVM Termination)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Time and State | | |
| **Abstract:** | MavenWrapperDownloader.java 89 , main() exit() . . | | |

**Sink:** MavenWrapperDownloader.java:89 exit()

```
87                  downloadFileFromURL(url, outputFile);
88                  System.out.println("Done");
89                  System.exit(0);
90              } catch (Throwable e) {
91                  System.out.println("- Error downloading");
```

### MavenWrapperDownloader.java, line 93 (J2EE Bad Practices: JVM Termination)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Time and State | | |
| **Abstract:** | MavenWrapperDownloader.java 93 , main() exit() . . | | |

**Sink:** MavenWrapperDownloader.java:93 exit()

```
91                  System.out.println("- Error downloading");
92                  e.printStackTrace();
93                  System.exit(1);
94              }
95          }
```

### PasswordResetLink.java, line 36 (J2EE Bad Practices: JVM Termination)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Time and State | | |
| **Abstract:** | PasswordResetLink.java 36 , main() exit() . . | | |

**Sink:** PasswordResetLink.java:36 exit()

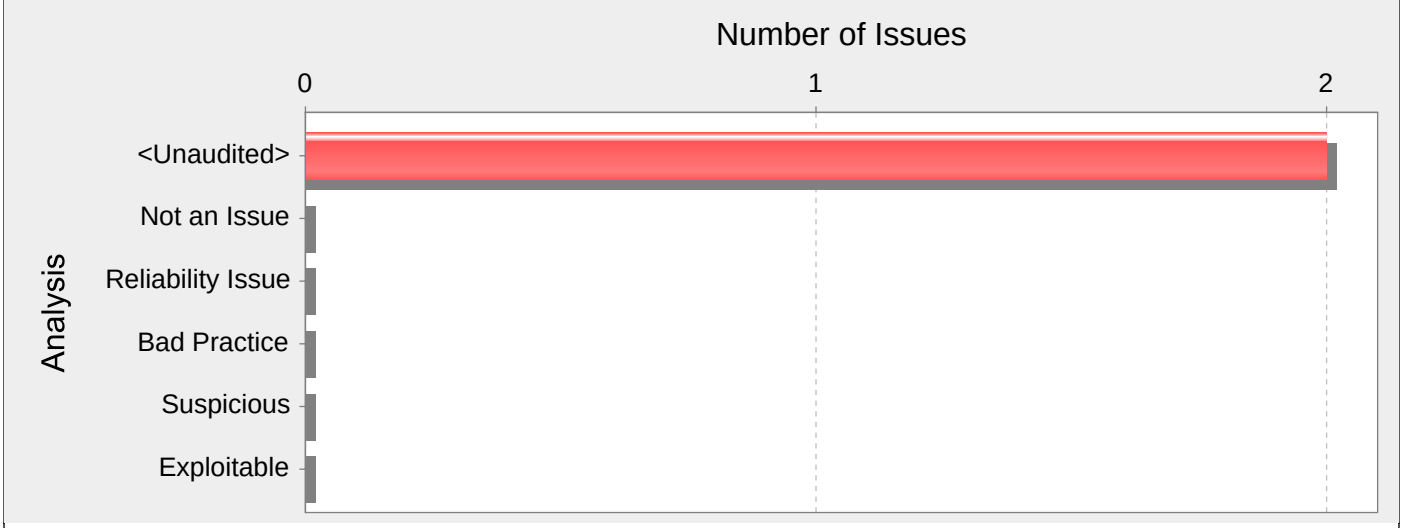
```
34              if (args == null || args.length != 2) {
35                  System.out.println("Need a username and key");
```

ot Copyright 2024 Open Text.

```
36                              System.exit(1);
37                          }
38                          String username = args[0];
```

Copyright 2024 Open Text.

## 49: 05.05. (2 Issues)




**Abstract:**

, , DOS(Denial of Service) .

**Explanation:**

Java . Java . . . .

classpath serializable , . JDK JVM , , .

1: .

```
InputStream is = request.getInputStream();
ObjectInputStream ois = new ObjectInputStream(is);
MyObject obj = (MyObject) ois.readObject();
```

**Recommendations:**

. .

. Java . java.io.ObjectInputStream resolveClass(ObjectStreamClass desc) . .

Apache Commons IO(org.apache.commons.io.serialization.ValidatingObjectInputStream) . . .
. .

(: JMX, RMI, JMS, HTTP Invoker ) . .

-
-
-
-

Fortify Runtime ObjectInputStream . .

**Tips:**

1. ObjectInputStream DoS(Denial of Service) ObjectInputStream Medium .

### InsecureDeserializationTask.java, line 56 (Dynamic Code Evaluation: Unsafe Deserialization)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Medium | **Folder** | Medium |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | , , DOS(Denial of Service) . | | |

**Source:** InsecureDeserializationTask.java:46 completed(0)

```
44      @PostMapping("/InsecureDeserialization/task")
45      @ResponseBody
46      public AttackResult completed(@RequestParam String token) throws IOException {
47          String b64token;
48          long before;
```

**Sink:** InsecureDeserializationTask.java:56 java.io.ObjectInputStream.readObject()

```
54          try (ObjectInputStream ois = new ObjectInputStream(new
ByteArrayInputStream(Base64.getDecoder().decode(b64token)))) {
55              before = System.currentTimeMillis();
```

ot Copyright 2024 Open Text.

```
56                          Object o = ois.readObject();
57                          if (!(o instanceof VulnerableTaskHolder)) {
58                              if (o instanceof String) {
```

| VulnerableComponentsLesson.java, line 52 (Dynamic Code Evaluation: Unsafe XStream Deserialization) | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | VulnerableComponentsLesson.java 52 XStream XML . XML . | | |

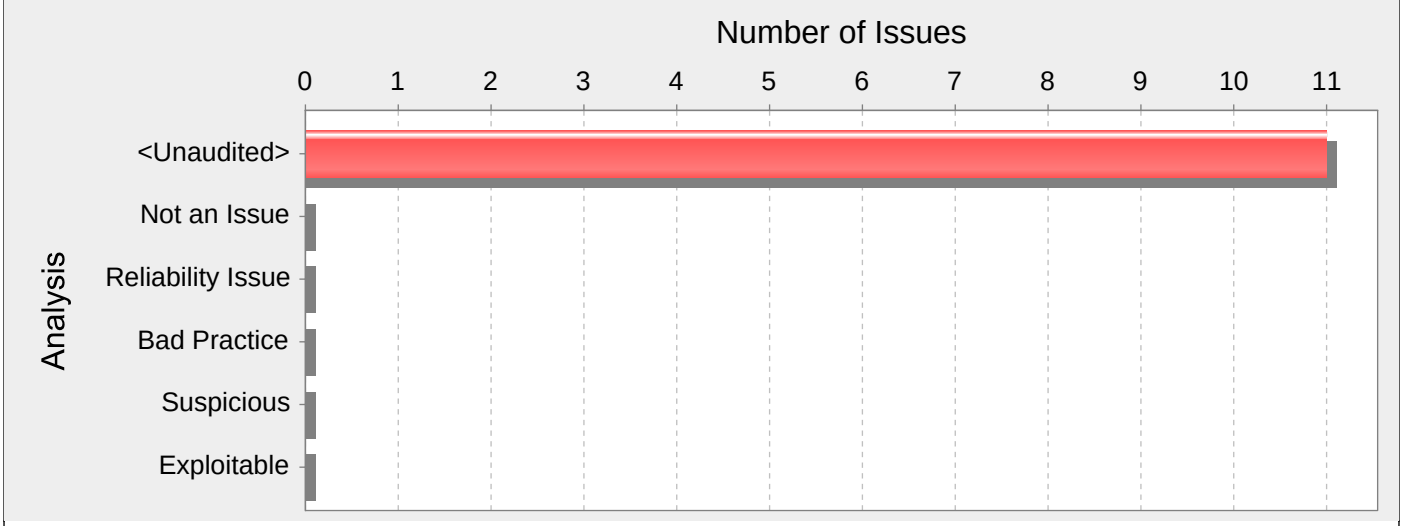
**Source:** VulnerableComponentsLesson.java:41 completed(0)

```
39      @PostMapping("/VulnerableComponents/attack1")
40      public @ResponseBody
41      AttackResult completed(@RequestParam String payload) {
42          XStream xstream = new XStream();
43          xstream.setClassLoader(Contact.class.getClassLoader());
```

**Sink:** VulnerableComponentsLesson.java:52 com.thoughtworks.xstream.XStream.fromXML()

```
50                          payload = payload.replace("+", "").replace("\r", "").replace("\n",
"").replace("> ", ">").replace(" <", "<");
51                          }
52                          contact = (Contact) xstream.fromXML(payload);
53                      } catch (Exception ex) {
54                          return failed(this).feedback("vulnerable-
components.close").output(ex.getMessage()).build();
```

## 49: 04.03. (11 Issues)




**Abstract:**

Salaries.java copyFiles() 63 mkdir() . .

**Explanation:**

Java java.io read() . Java . ( Java C . .) stream reader . . .

read() IO .

: . 1KB read() . .

```
FileInputStream fis;
byte[] byteArray = new byte[1024];
for (Iterator i=users.iterator(); i.hasNext();) {
String userName = (String) i.next();
String pFileName = PFILE_ROOT + "/" + userName;
FileInputStream fis = new FileInputStream(pFileName);
fis.read(byteArray); // the file is always 1k bytes
fis.close();
processPFile(userName, byteArray);
}
```

**Recommendations:**

```
FileInputStream fis;
byte[] byteArray = new byte[1024];
for (Iterator i=users.iterator(); i.hasNext();) {
String userName = (String) i.next();
String pFileName = PFILE_ROOT + "/" + userName;
fis = new FileInputStream(pFileName);
int bRead = 0;
while (bRead < 1024) {
int rd = fis.read(byteArray, bRead, 1024 - bRead);
if (rd == -1) {
throw new IOException("file is unusually small");
}
bRead += rd;
}
// could add check to see if file is too large here
fis.close();
processPFile(userName, byteArray);
}
```

: . file system race condition .

Copyright 2024 Open Text.

Tips:
1. " . ..." . . , , .

## SqlInjectionLesson2.java, line 68 (Denial of Service: StringBuilder)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Input Validation and Representation | | |

**Abstract:** SqlInjectionLesson2.java 68 append() (16) StringBuilder StringBuffer . JVM .

**Source:** SqlInjectionLesson2.java:55 completed(0)

```
53      @PostMapping("/SqlInjection/attack2")
54      @ResponseBody
55      public AttackResult completed(@RequestParam String query) {
56          return injectableQuery(query);
57      }
```

**Sink:** SqlInjectionLesson2.java:68 java.lang.StringBuilder.append()

```
66
67                  if (results.getString("department").equals("Marketing")) {
68                      output.append("<span class='feedback-positive'>" + query + "</span>");
69                      output.append(SqlInjectionLesson8.generateTable(results));
70                      return success(this).feedback("sql-
injection.2.success").output(output.toString()).build();
```

## ProfileUploadRetrieval.java, line 48 (Unchecked Return Value)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | API Abuse | | |

**Abstract:** ProfileUploadRetrieval.java ProfileUploadRetrieval() 48 mkdirs() . .

**Sink:** ProfileUploadRetrieval.java:48 mkdirs()

```
46      public ProfileUploadRetrieval(@Value("${webgoat.server.directory}") String
webGoatHomeDirectory) {
47          this.catPicturesDirectory = new File(webGoatHomeDirectory, "/PathTraversal/" +
"/cats");
48          this.catPicturesDirectory.mkdirs();
49      }
```

## SqlInjectionLesson3.java, line 71 (Denial of Service: StringBuilder)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Input Validation and Representation | | |

**Abstract:** SqlInjectionLesson3.java 71 append() (16) StringBuilder StringBuffer . JVM .

**Source:** SqlInjectionLesson3.java:56 completed(0)

```
54      @PostMapping("/SqlInjection/attack3")
55      @ResponseBody
56      public AttackResult completed(@RequestParam String query) {
57          return injectableQuery(query);
58      }
```

**Sink:** SqlInjectionLesson3.java:71 java.lang.StringBuilder.append()

```
69                  results.first();
70                  if (results.getString("department").equals("Sales")) {
71                      output.append("<span class='feedback-positive'>" + query +
"</span>");
72                      output.append(SqlInjectionLesson8.generateTable(results));
73                      return success(this).output(output.toString()).build();
```

## SqlInjectionLesson4.java, line 69 (Denial of Service: StringBuilder)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Input Validation and Representation | | |

**Abstract:** SqlInjectionLesson4.java 69 append() (16) StringBuilder StringBuffer . JVM .

**Source:** SqlInjectionLesson4.java:56 completed(0)

```
54                  @PostMapping("/SqlInjection/attack4")
55                  @ResponseBody
56                  public AttackResult completed(@RequestParam String query) {
57                      return injectableQuery(query);
58                  }
```

Sink: SqlInjectionLesson4.java:69 java.lang.StringBuilder.append()

```
67                          // user completes lesson if column phone exists
68                          if (results.first()) {
69                              output.append("<span class='feedback-positive'>" + query +
"</span>");
70                              return success(this).output(output.toString()).build();
71                          } else {
```

## BlindSendFileAssignment.java, line 69 (Unchecked Return Value)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | API Abuse | | |

Abstract: BlindSendFileAssignment.java createSecretFileWithRandomContents() 69 mkdirs() . .

Sink: BlindSendFileAssignment.java:69 mkdirs()

```
67          File targetDirectory = new File(webGoatHomeDirectory, "/XXE/" +
user.getUsername());
68          if (!targetDirectory.exists()) {
69              targetDirectory.mkdirs();
70          }
71          try {
```

## ProfileUploadBase.java, line 42 (Unchecked Return Value)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | API Abuse | | |

Abstract: ProfileUploadBase.java execute() 42 mkdirs() . .

Sink: ProfileUploadBase.java:42 mkdirs()

```
40
41          try {
42              uploadDirectory.mkdirs();
43              var uploadedFile = new File(uploadDirectory, fullName);
44              uploadedFile.createNewFile();
```

## MvcConfiguration.java, line 63 (Unchecked Return Value)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | API Abuse | | |

Abstract: MvcConfiguration.java createDirectory() 63 mkdirs() . .

Sink: MvcConfiguration.java:63 mkdirs()

```
61          File file = new File(fileLocation);
62          if (!file.exists()) {
63              file.mkdirs();
64          }
65      }
```

## FileServer.java, line 75 (Unchecked Return Value)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | API Abuse | | |

Abstract: FileServer.java importFile() 75 mkdirs() . .

Sink: FileServer.java:75 mkdirs()

```
73          var user = (WebGoatUser)
SecurityContextHolder.getContext().getAuthentication().getPrincipal();
74          var destinationDir = new File(fileLocation, user.getUsername());
75          destinationDir.mkdirs();
76          myFile.transferTo(new File(destinationDir, myFile.getOriginalFilename()));
77          log.debug("File saved to {}", new File(destinationDir,
myFile.getOriginalFilename()));
```

ot Copyright 2024 Open Text.

## CrossSiteScriptingLesson5a.java, line 66 (Denial of Service: StringBuilder)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | CrossSiteScriptingLesson5a.java 66 append() (16) StringBuilder StringBuffer . JVM . | | |

**Source:** CrossSiteScriptingLesson5a.java:54 completed(4)

```
52      public AttackResult completed(@RequestParam Integer QTY1,
53                                    @RequestParam Integer QTY2, @RequestParam Integer QTY3,
54                                    @RequestParam Integer QTY4, @RequestParam String field1,
55                                    @RequestParam String field2) {
```

**Sink:** CrossSiteScriptingLesson5a.java:66 java.lang.StringBuilder.append()

```
64          StringBuilder cart = new StringBuilder();
65          cart.append("Thank you for shopping at WebGoat. <br />Your support is appreciated<hr />");
66          cart.append("<p>We have charged credit card:" + field1 + "<br />");
67          cart.append("                                  ------------------- <br />");
68          cart.append("                                    $" + totalSale);
```

## FileServer.java, line 105 (Unchecked Return Value)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | API Abuse | | |
| **Abstract:** | FileServer.java getFiles() 105 delete() . . | | |

**Sink:** FileServer.java:105 delete()

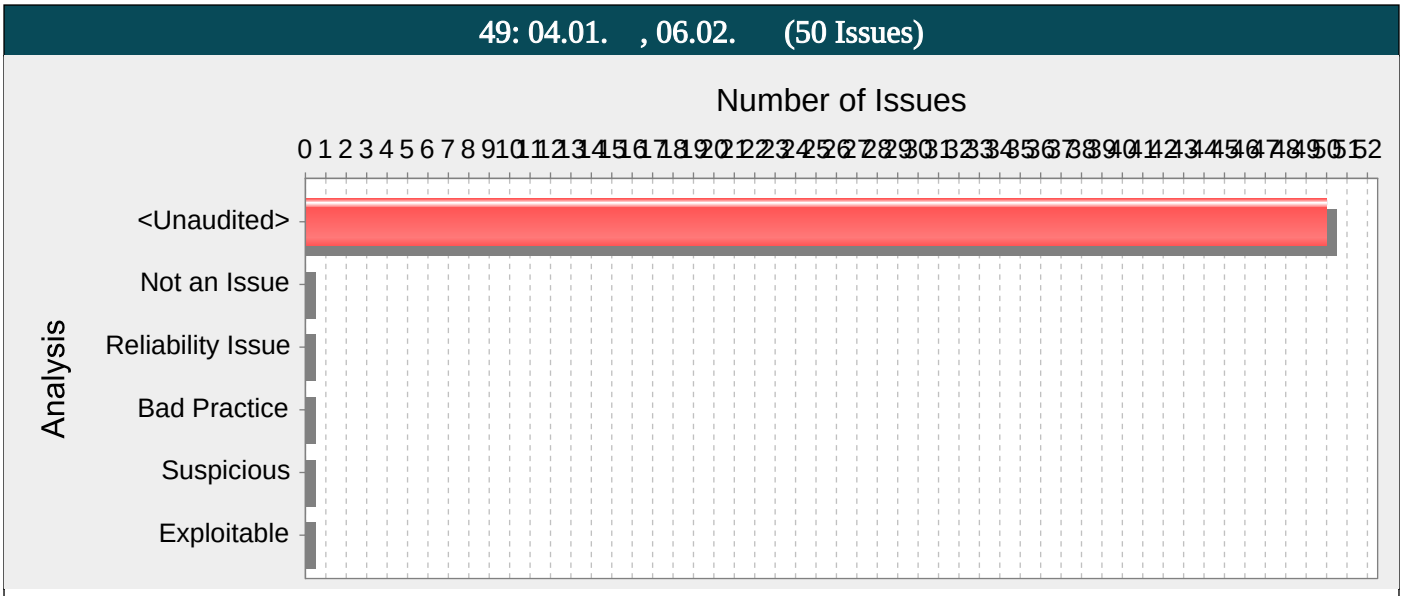
```
103             modelAndView.addObject("uploadSuccess", request.getParameter("uploadSuccess"));
104         }
105         changeIndicatorFile.delete();
106
107         var uploadedFiles = new ArrayList<>();
```

## Salaries.java, line 63 (Unchecked Return Value)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | API Abuse | | |
| **Abstract:** | Salaries.java copyFiles() 63 mkdir() . . | | |

**Sink:** Salaries.java:63 mkdir()

```
61          File targetDirectory = new File(webGoatHomeDirectory, "/ClientSideFiltering");
62          if (!targetDirectory.exists()) {
63              targetDirectory.mkdir();
64          }
65          try {
```




**Abstract:**

.

**Explanation:**

. , , . .

1: HTTP .

```
protected void doPost (HttpServletRequest req, HttpServletResponse res) throws IOException {
...
PrintWriter out = res.getWriter();
try {
...
} catch (Exception e) {
out.println(e.getMessage());
}
}
```

. . , SQL injection . . Example 1 , .

. . .

2: Android catch .

```
...
try {
...
} catch (Exception e) {
String exception = Log.getStackTraceString(e);
Intent i = new Intent();
i.setAction("SEND_EXCEPTION");
i.putExtra("exception", exception);
view.getContext().sendBroadcast(i);
}
...
```

. NFC( ) . NFC . NFC NFC , .

3: Android NFC . .

```
...
public static final String TAG = "NfcActivity";
private static final String DATA_SPLITTER = "__:DATA:__";
private static final String MIME_TYPE = "application/my.applications.mimetype";
...
```

Copyright 2024 Open Text.

TelephonyManager tm = (TelephonyManager)Context.getSystemService(Context.TELEPHONY_SERVICE);
String VERSION = tm.getDeviceSoftwareVersion();
...
NfcAdapter nfcAdapter = NfcAdapter.getDefaultAdapter(this);
if (nfcAdapter == null)
return;

String text = TAG + DATA_SPLITTER + VERSION;
NdefRecord record = new NdefRecord(NdefRecord.TNF_MIME_MEDIA,
MIME_TYPE.getBytes(), new byte[0], text.getBytes());
NdefRecord[] records = { record };
NdefMessage msg = new NdefMessage(records);
nfcAdapter.setNdefPushMessage(msg, this);
...

NDEF(NFC ) , URI . , MIME .

**Recommendations:**

. . . (: HTML ).

. , " " . .

4: catch , . .

...
try {
...
} catch (Exception e) {
String exception = Log.getStackTraceString(e);
Intent i = new Intent();
i.setAction("SEND_EXCEPTION");
i.putExtra("exception", exception);
LocalBroadcastManager.getInstance(view.getContext()).sendBroadcast(i);
}
...

Android NFC . , , .

**Tips:**

1. System information leak , IT . .

2. . , , Audit Guide .

## CSRFFeedback.java, line 71 (System Information Leak: External)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | . | | |

**Source:** CSRFFeedback.java:71 Read e()

```
69                  objectMapper.readValue(feedback.getBytes(), Map.class);
70              } catch (IOException e) {
71                  return failed(this).feedback(ExceptionUtils.getStackTrace(e)).build();
72              }
73              boolean correctCSRF = requestContainsWebGoatCookie(request.getCookies()) &&
request.getContentType().contains(MediaType.TEXT_PLAIN_VALUE);
```

**Sink:** CSRFFeedback.java:71 Return()

```
69                  objectMapper.readValue(feedback.getBytes(), Map.class);
70              } catch (IOException e) {
71                  return failed(this).feedback(ExceptionUtils.getStackTrace(e)).build();
72              }
73              boolean correctCSRF = requestContainsWebGoatCookie(request.getCookies()) &&
request.getContentType().contains(MediaType.TEXT_PLAIN_VALUE);
```

## ProfileUploadRetrieval.java, line 100 (System Information Leak: External)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | ProfileUploadRetrieval.java getProfilePicture() 100 location()() . location() . | | |

Source: ProfileUploadRetrieval.java:46 ProfileUploadRetrieval(0)

```
44                  private final File catPicturesDirectory;
45
46                  public ProfileUploadRetrieval(@Value("${webgoat.server.directory}") String
webGoatHomeDirectory) {
47                      this.catPicturesDirectory = new File(webGoatHomeDirectory, "/PathTraversal/" +
"/cats");
48                      this.catPicturesDirectory.mkdirs();
```

Sink: ProfileUploadRetrieval.java:100 org.springframework.http.ResponseEntity.HeadersBuilder.location()

```
98                      }
99                      return ResponseEntity.status(HttpStatus.NOT_FOUND)
100                             .location(new URI("/PathTraversal/random-picture?id=" +
catPicture.getName()))
101
.body(StringUtils.arrayToCommaDelimitedString(catPicture.getParentFile().listFiles()).
getBytes());
102                 } catch (IOException | URISyntaxException e) {
```

## ProfileUploadRetrieval.java, line 101 (System Information Leak: External)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | ProfileUploadRetrieval.java getProfilePicture() 101 body()() . body() . | | |

Source: ProfileUploadRetrieval.java:101 java.io.File.listFiles()

```
99                      return ResponseEntity.status(HttpStatus.NOT_FOUND)
100                             .location(new URI("/PathTraversal/random-picture?id=" +
catPicture.getName()))
101
.body(StringUtils.arrayToCommaDelimitedString(catPicture.getParentFile().listFiles()).
getBytes());
102                 } catch (IOException | URISyntaxException e) {
103                     log.error("Image not found", e);
```

Sink: ProfileUploadRetrieval.java:101 org.springframework.http.ResponseEntity.BodyBuilder.body()

```
99                      return ResponseEntity.status(HttpStatus.NOT_FOUND)
100                             .location(new URI("/PathTraversal/random-picture?id=" +
catPicture.getName()))
101
.body(StringUtils.arrayToCommaDelimitedString(catPicture.getParentFile().listFiles()).
getBytes());
102                 } catch (IOException | URISyntaxException e) {
103                     log.error("Image not found", e);
```

## SqlInjectionLesson3.java, line 57 (System Information Leak: External)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | . | | |

Source: SqlInjectionLesson3.java:82 java.lang.Throwable.getMessage()

```
80                      }
81                  } catch (Exception e) {
82                      return failed(this).output(this.getClass().getName() + " : " +
e.getMessage()).build();
83                  }
84              }
```

Sink: SqlInjectionLesson3.java:57 Return()

```
55              @ResponseBody
56              public AttackResult completed(@RequestParam String query) {
57                  return injectableQuery(query);
58              }
```

## SqlOnlyInputValidation.java, line 53 (System Information Leak: External)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | . | | |

Source: SqlInjectionLesson6a.java:92 java.lang.Throwable.getMessage()

```
90                      }
91                  } catch (SQLException sqle) {
92                      return failed(this).output(sqle.getMessage() + YOUR_QUERY_WAS +
query).build();
93                  }
94              } catch (Exception e) {
```

Sink: SqlOnlyInputValidation.java:53 Return()

```
51              }
52              AttackResult attackResult = lesson6a.injectableQuery(userId);
53              return new AttackResult(attackResult.isLessonCompleted(),
attackResult.getFeedback(), attackResult.getOutput(), getClass().getSimpleName(),
true);
54          }
55      }
```

## application-webgoat.properties, line 1 (System Information Leak: External)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | . | | |

Sink: application-webgoat.properties:1 server.error.include-stacktrace()

```
-1  server.error.include-stacktrace=always
0   server.error.path=/error.html
1   server.servlet.context-path=/WebGoat
```

## JWTSecretKeyEndpoint.java, line 92 (System Information Leak: External)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | . | | |

Source: JWTSecretKeyEndpoint.java:92 java.lang.Throwable.getMessage()

```
90                  }
91              } catch (Exception e) {
92                  return failed(this).feedback("jwt-invalid-
token").output(e.getMessage()).build();
93              }
94          }
```

Sink: JWTSecretKeyEndpoint.java:92 Return()

```
90                  }
91              } catch (Exception e) {
92                  return failed(this).feedback("jwt-invalid-
token").output(e.getMessage()).build();
93              }
94          }
```

## JWTVotesEndpoint.java, line 180 (System Information Leak: External)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | . | | |

Source: JWTVotesEndpoint.java:180 Read e()

```
178                 }
179             } catch (JwtException e) {
180                 return failed(this).feedback("jwt-invalid-
token").output(e.toString()).build();
181             }
182         }
```

| | |
|---|---|
| Sink: | JWTVotesEndpoint.java:180 Return() |

```
178                     }
179                 } catch (JwtException e) {
180                     return failed(this).feedback("jwt-invalid-
token").output(e.toString()).build();
181                 }
182             }
```

**ProfileUploadFix.java, line 31 (System Information Leak: External)**

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | . | | |
| Source: | ProfileUploadBase.java:53 java.lang.Throwable.getMessage() | | |

```
51
52              } catch (IOException e) {
53                  return failed(this).output(e.getMessage()).build();
54              }
55          }
```

| | |
|---|---|
| Sink: | ProfileUploadFix.java:31 Return() |

```
29                  @RequestParam("uploadedFileFix") MultipartFile file,
30                  @RequestParam(value = "fullNameFix", required = false) String fullName) {
31              return super.execute(file, fullName != null ? fullName.replace("../", "") :
"");
32          }
```

**SqlInjectionLesson5a.java, line 54 (System Information Leak: External)**

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | . | | |
| Source: | SqlInjectionLesson5a.java:81 java.lang.Throwable.getMessage() | | |

```
79                      }
80                  } catch (SQLException sqle) {
81                      return failed(this).output(sqle.getMessage() + "<br> Your query was: "
+ query).build();
82                  }
83              } catch (Exception e) {
```

| | |
|---|---|
| Sink: | SqlInjectionLesson5a.java:54 Return() |

```
52      @ResponseBody
53      public AttackResult completed(@RequestParam String account, @RequestParam String
operator, @RequestParam String injection) {
54          return injectableQuery(account + " " + operator + " " + injection);
55      }
```

**SqlInjectionLesson5a.java, line 54 (System Information Leak: External)**

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | . | | |
| Source: | SqlInjectionLesson5a.java:84 java.lang.Throwable.getMessage() | | |

```
82                  }
83              } catch (Exception e) {
84                  return failed(this).output(this.getClass().getName() + " : " +
e.getMessage() + "<br> Your query was: " + query).build();
85              }
86          }
```

| | |
|---|---|
| Sink: | SqlInjectionLesson5a.java:54 Return() |

```
52      @ResponseBody
53      public AttackResult completed(@RequestParam String account, @RequestParam String
operator, @RequestParam String injection) {
54          return injectableQuery(account + " " + operator + " " + injection);
55      }
```

ot Copyright 2024 Open Text.

## BlindSendFileAssignment.java, line 95 (System Information Leak: External)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | . | | |

Source: BlindSendFileAssignment.java:95 Read e()

```
93                  comments.addComment(comment, false);
94              } catch (Exception e) {
95                  return failed(this).output(e.toString()).build();
96              }
97              return failed(this).build();
```

Sink: BlindSendFileAssignment.java:95 Return()

```
93                  comments.addComment(comment, false);
94              } catch (Exception e) {
95                  return failed(this).output(e.toString()).build();
96              }
97              return failed(this).build();
```

## application-webwolf.properties, line 1 (System Information Leak: External)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | . | | |

Sink: application-webwolf.properties:1 server.error.include-stacktrace()

```
-1  server.error.include-stacktrace=always
0   server.error.path=/error.html
1   server.port=${webwolf.port:9090}
```

## SqlOnlyInputValidationOnKeywords.java, line 54 (System Information Leak: External)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | . | | |

Source: SqlInjectionLesson6a.java:95 java.lang.Throwable.getMessage()

```
93                  }
94              } catch (Exception e) {
95                  return failed(this).output(this.getClass().getName() + " : " +
e.getMessage() + YOUR_QUERY_WAS + query).build();
96              }
97          }
```

Sink: SqlOnlyInputValidationOnKeywords.java:54 Return()

```
52              }
53              AttackResult attackResult = lesson6a.injectableQuery(userId);
54              return new AttackResult(attackResult.isLessonCompleted(),
attackResult.getFeedback(), attackResult.getOutput(), getClass().getSimpleName(),
true);
55          }
56      }
```

## SqlInjectionLesson5b.java, line 52 (System Information Leak: External)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | . | | |

Source: SqlInjectionLesson5b.java:92 java.lang.Throwable.getMessage()

```
90                  } catch (SQLException sqle) {
91
92                      return failed(this).output(sqle.getMessage() + "<br> Your query was: "
+ queryString.replace("?", login_count)).build();
93                  }
94              } catch (Exception e) {
```

Sink: SqlInjectionLesson5b.java:52 Return()

```
50                      @ResponseBody
51                      public AttackResult completed(@RequestParam String userid, @RequestParam String
login_count, HttpServletRequest request) throws IOException {
52                          return injectableQuery(login_count, userid);
53                      }
```

## JWTRefreshEndpoint.java, line 112 (System Information Leak: External)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | . | | |

Source: JWTRefreshEndpoint.java:112 java.lang.Throwable.getMessage()

```
110                         return ok(failed(this).feedback("jwt-refresh-not-
tom").feedbackArgs(user).build());
111                     } catch (ExpiredJwtException e) {
112                         return ok(failed(this).output(e.getMessage()).build());
113                     } catch (JwtException e) {
114                         return ok(failed(this).feedback("jwt-invalid-token").build());
```

Sink: JWTRefreshEndpoint.java:112 Return()

```
110                         return ok(failed(this).feedback("jwt-refresh-not-
tom").feedbackArgs(user).build());
111                     } catch (ExpiredJwtException e) {
112                         return ok(failed(this).output(e.getMessage()).build());
113                     } catch (JwtException e) {
114                         return ok(failed(this).feedback("jwt-invalid-token").build());
```

## SqlInjectionLesson2.java, line 56 (System Information Leak: External)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | . | | |

Source: SqlInjectionLesson2.java:75 java.lang.Throwable.getMessage()

```
73                      }
74                  } catch (SQLException sqle) {
75                      return failed(this).feedback("sql-
injection.2.failed").output(sqle.getMessage()).build();
76                  }
77              }
```

Sink: SqlInjectionLesson2.java:56 Return()

```
54                  @ResponseBody
55                  public AttackResult completed(@RequestParam String query) {
56                      return injectableQuery(query);
57                  }
```

## SqlInjectionLesson6a.java, line 52 (System Information Leak: External)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | . | | |

Source: SqlInjectionLesson6a.java:92 java.lang.Throwable.getMessage()

```
90                      }
91                  } catch (SQLException sqle) {
92                      return failed(this).output(sqle.getMessage() + YOUR_QUERY_WAS +
query).build();
93                  }
94              } catch (Exception e) {
```

Sink: SqlInjectionLesson6a.java:52 Return()

```
50                  @ResponseBody
51                  public AttackResult completed(@RequestParam(value="userid_6a")  String userId) {
52                      return injectableQuery(userId);
53                      // The answer: Smith' union select userid,user_name, password,cookie,cookie,
cookie,userid from user_system_data --
54                  }
```

ot Copyright 2024 Open Text.

## ProfileZipSlip.java, line 40 (System Information Leak: External)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | . | | |

Source: ProfileZipSlip.java:67 java.lang.Throwable.getMessage()

```
65                  return isSolved(currentImage, getProfilePictureAsBase64());
66              } catch (IOException e) {
67                  return failed(this).output(e.getMessage()).build();
68              }
69          }
```

Sink: ProfileZipSlip.java:40 Return()

```
38                  return failed(this).feedback("path-traversal-zip-slip.no-zip").build();
39              } else {
40                  return processZipUpload(file);
41              }
42          }
```

## SqlInjectionLesson8.java, line 55 (System Information Leak: External)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | . | | |

Source: SqlInjectionLesson8.java:93 java.lang.Throwable.getMessage()

```
91
92              } catch (Exception e) {
93                  return failed(this).output("<br><span class='feedback-negative'>" +
e.getMessage() + "</span>").build();
94              }
95          }
```

Sink: SqlInjectionLesson8.java:55 Return()

```
53      @ResponseBody
54      public AttackResult completed(@RequestParam String name, @RequestParam String
auth_tan) {
55          return injectableQueryConfidentiality(name, auth_tan);
56      }
```

## SqlInjectionLesson4.java, line 57 (System Information Leak: External)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | . | | |

Source: SqlInjectionLesson4.java:75 java.lang.Throwable.getMessage()

```
73                  }
74              } catch (SQLException sqle) {
75                  return failed(this).output(sqle.getMessage()).build();
76              }
77          } catch (Exception e) {
```

Sink: SqlInjectionLesson4.java:57 Return()

```
55      @ResponseBody
56      public AttackResult completed(@RequestParam String query) {
57          return injectableQuery(query);
58      }
```

## SqlInjectionLesson10.java, line 53 (System Information Leak: External)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | . | | |

Source: SqlInjectionLesson10.java:78 java.lang.Throwable.getMessage()

```
76              } catch (SQLException e) {
77                  if (tableExists(connection)) {
```

ot Copyright 2024 Open Text.

```
78                                  return failed(this).output("<span class='feedback-negative'>" +
e.getMessage() + "</span><br>" + output.toString()).build();
79                              } else {
80                                  return success(this).feedback("sql-injection.10.success").build();
```

Sink: SqlInjectionLesson10.java:53 Return()

```
51      @ResponseBody
52      public AttackResult completed(@RequestParam String action_string) {
53          return injectableQueryAvailability(action_string);
54      }
```

## SqlInjectionLesson3.java, line 57 (System Information Leak: External)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | . | | |

Source: SqlInjectionLesson3.java:79 java.lang.Throwable.getMessage()

```
77
78                  } catch (SQLException sqle) {
79                      return failed(this).output(sqle.getMessage()).build();
80                  }
81              } catch (Exception e) {
```

Sink: SqlInjectionLesson3.java:57 Return()

```
55      @ResponseBody
56      public AttackResult completed(@RequestParam String query) {
57          return injectableQuery(query);
58      }
```

## application-webwolf.properties, line 1 (System Information Leak: External)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | . | | |

Sink: application-webwolf.properties:1 server.error.include-stacktrace()

```
-1  server.error.include-stacktrace=always
0   server.error.path=/error.html
1   server.port=${webwolf.port:9090}
```

## SimpleXXE.java, line 79 (System Information Leak: External)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | . | | |

Source: SimpleXXE.java:77 org.apache.commons.lang3.exception.ExceptionUtils.getStackTrace()

```
75                  }
76              } catch (Exception e) {
77                  error = ExceptionUtils.getStackTrace(e);
78              }
79              return failed(this).output(error).build();
```

Sink: SimpleXXE.java:79 Return()

```
77                  error = ExceptionUtils.getStackTrace(e);
78              }
79              return failed(this).output(error).build();
80      }
```

## SimpleXXE.java, line 79 (System Information Leak: External)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | . | | |

Source: SimpleXXE.java:77 Read e()

```
75                  }
```

ot Copyright 2024 Open Text.

```
76                      } catch (Exception e) {
77                          error = ExceptionUtils.getStackTrace(e);
78                      }
79                      return failed(this).output(error).build();
```

Sink: SimpleXXE.java:79 Return()

```
77                          error = ExceptionUtils.getStackTrace(e);
78                      }
79                      return failed(this).output(error).build();
80                  }
```

## JWTFinalEndpoint.java, line 118 (System Information Leak: External)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | . | | |

Source: JWTFinalEndpoint.java:118 Read e()

```
116                     }
117                 } catch (JwtException e) {
118                     return failed(this).feedback("jwt-invalid-
token").output(e.toString()).build();
119                 }
120             }
```

Sink: JWTFinalEndpoint.java:118 Return()

```
116                     }
117                 } catch (JwtException e) {
118                     return failed(this).feedback("jwt-invalid-
token").output(e.toString()).build();
119                 }
120             }
```

## SqlOnlyInputValidationOnKeywords.java, line 54 (System Information Leak: External)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | . | | |

Source: SqlInjectionLesson6a.java:92 java.lang.Throwable.getMessage()

```
90                      }
91                  } catch (SQLException sqle) {
92                      return failed(this).output(sqle.getMessage() + YOUR_QUERY_WAS +
query).build();
93                  }
94              } catch (Exception e) {
```

Sink: SqlOnlyInputValidationOnKeywords.java:54 Return()

```
52              }
53              AttackResult attackResult = lesson6a.injectableQuery(userId);
54              return new AttackResult(attackResult.isLessonCompleted(),
attackResult.getFeedback(), attackResult.getOutput(), getClass().getSimpleName(),
true);
55          }
56      }
```

## ProfileUploadRetrieval.java, line 96 (System Information Leak: External)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | ProfileUploadRetrieval.java getProfilePicture() 96 location()() . location() . | | |

Source: ProfileUploadRetrieval.java:46 ProfileUploadRetrieval(0)

```
44      private final File catPicturesDirectory;
45
46      public ProfileUploadRetrieval(@Value("${webgoat.server.directory}") String
webGoatHomeDirectory) {
47          this.catPicturesDirectory = new File(webGoatHomeDirectory, "/PathTraversal/" +
"/cats");
```

```
48                      this.catPicturesDirectory.mkdirs();
```

Sink: ProfileUploadRetrieval.java:96 org.springframework.http.ResponseEntity.HeadersBuilder.location()

```
94                      return ResponseEntity.ok()
95                              .contentType(MediaType.parseMediaType(MediaType.IMAGE_JPEG_VALUE))
96                              .location(new URI("/PathTraversal/random-picture?id=" + catPicture.getName()))
97                              .body(Base64.getEncoder().encode(FileCopyUtils.copyToByteArray(catPicture)));
98                  }
```

## SqlInjectionLesson9.java, line 56 (System Information Leak: External)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | . | | |

Source: SqlInjectionLesson9.java:78 java.lang.Throwable.getMessage()

```
76                  } catch (SQLException e) {
77                      System.err.println(e.getMessage());
78                      return failed(this).output("<br><span class='feedback-negative'>" + e.getMessage() + "</span>").build();
79                  }
```

Sink: SqlInjectionLesson9.java:56 Return()

```
54      @ResponseBody
55      public AttackResult completed(@RequestParam String name, @RequestParam String auth_tan) {
56          return injectableQueryIntegrity(name, auth_tan);
57      }
```

## SqlInjectionLesson6a.java, line 52 (System Information Leak: External)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | . | | |

Source: SqlInjectionLesson6a.java:95 java.lang.Throwable.getMessage()

```
93                  }
94              } catch (Exception e) {
95                  return failed(this).output(this.getClass().getName() + " : " + e.getMessage() + YOUR_QUERY_WAS + query).build();
96              }
97      }
```

Sink: SqlInjectionLesson6a.java:52 Return()

```
50      @ResponseBody
51      public AttackResult completed(@RequestParam(value="userid_6a")  String userId) {
52          return injectableQuery(userId);
53          // The answer: Smith' union select userid,user_name, password,cookie,cookie, cookie,userid from user_system_data --
54      }
```

## SqlInjectionLesson4.java, line 57 (System Information Leak: External)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | . | | |

Source: SqlInjectionLesson4.java:78 java.lang.Throwable.getMessage()

```
76                  }
77              } catch (Exception e) {
78                  return failed(this).output(this.getClass().getName() + " : " + e.getMessage()).build();
79              }
80      }
```

Sink: SqlInjectionLesson4.java:57 Return()

```
55      @ResponseBody
```

Copyright 2024 Open Text.

```
56                      public AttackResult completed(@RequestParam String query) {
57                          return injectableQuery(query);
58                      }
```

## MailAssignment.java, line 65 (System Information Leak: External)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | . | | |

Source: MailAssignment.java:65 org.springframework.core.NestedRuntimeException.getMessage()

```
63                          restTemplate.postForEntity(webWolfURL, mailEvent, Object.class);
64                      } catch (RestClientException e ) {
65                          return informationMessage(this).feedback("webwolf.email_failed").output(e.getMessage()).build();
66                      }
67                      return informationMessage(this).feedback("webwolf.email_send").feedbackArgs(email).build();
```

Sink: MailAssignment.java:65 Return()

```
63                          restTemplate.postForEntity(webWolfURL, mailEvent, Object.class);
64                      } catch (RestClientException e ) {
65                          return informationMessage(this).feedback("webwolf.email_failed").output(e.getMessage()).build();
66                      }
67                      return informationMessage(this).feedback("webwolf.email_send").feedbackArgs(email).build();
```

## SqlInjectionLesson5b.java, line 52 (System Information Leak: External)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | . | | |

Source: SqlInjectionLesson5b.java:95 java.lang.Throwable.getMessage()

```
93                      }
94                  } catch (Exception e) {
95                      return failed(this).output(this.getClass().getName() + " : " + e.getMessage() + "<br> Your query was: " + queryString.replace("?", login_count)).build();
96                  }
97              }
```

Sink: SqlInjectionLesson5b.java:52 Return()

```
50              @ResponseBody
51              public AttackResult completed(@RequestParam String userid, @RequestParam String login_count, HttpServletRequest request) throws IOException {
52                  return injectableQuery(login_count, userid);
53              }
```

## SqlOnlyInputValidation.java, line 53 (System Information Leak: External)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | . | | |

Source: SqlInjectionLesson6a.java:95 java.lang.Throwable.getMessage()

```
93                      }
94                  } catch (Exception e) {
95                      return failed(this).output(this.getClass().getName() + " : " + e.getMessage() + YOUR_QUERY_WAS + query).build();
96                  }
97              }
```

Sink: SqlOnlyInputValidation.java:53 Return()

```
51                  }
52                  AttackResult attackResult = lesson6a.injectableQuery(userId);
```

ot Copyright 2024 Open Text.

```
53                  return new AttackResult(attackResult.isLessonCompleted(),
attackResult.getFeedback(), attackResult.getOutput(), getClass().getSimpleName(),
true);
54              }
55          }
```

## application-webgoat.properties, line 1 (System Information Leak: External)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | . | | |

Sink: application-webgoat.properties:1 server.error.include-stacktrace()

```
-1  server.error.include-stacktrace=always
0   server.error.path=/error.html
1   server.servlet.context-path=/WebGoat
```

## FileServer.java, line 68 (System Information Leak: External)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | . | | |

Source: FileServer.java:68 Read this.fileLocation()

```
66      @ResponseBody
67      public String getFileLocation() {
68          return fileLocation;
69      }
```

Sink: FileServer.java:68 Return this.fileLocation()

```
66      @ResponseBody
67      public String getFileLocation() {
68          return fileLocation;
69      }
```

## CSRFFeedback.java, line 71 (System Information Leak: External)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | . | | |

Source: CSRFFeedback.java:71 org.apache.commons.lang3.exception.ExceptionUtils.getStackTrace()

```
69              objectMapper.readValue(feedback.getBytes(), Map.class);
70          } catch (IOException e) {
71              return failed(this).feedback(ExceptionUtils.getStackTrace(e)).build();
72          }
73          boolean correctCSRF = requestContainsWebGoatCookie(request.getCookies()) &&
request.getContentType().contains(MediaType.TEXT_PLAIN_VALUE);
```

Sink: CSRFFeedback.java:71 Return()

```
69              objectMapper.readValue(feedback.getBytes(), Map.class);
70          } catch (IOException e) {
71              return failed(this).feedback(ExceptionUtils.getStackTrace(e)).build();
72          }
73          boolean correctCSRF = requestContainsWebGoatCookie(request.getCookies()) &&
request.getContentType().contains(MediaType.TEXT_PLAIN_VALUE);
```

## SpoofCookieAssignment.java, line 73 (System Information Leak: External)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | . | | |

Source: SpoofCookieAssignment.java:109 java.lang.Throwable.getMessage()

```
107         } catch (Exception e) {
108             // for providing some instructive guidance, we won't return 4xx error here
109             return failed(this).output(e.getMessage()).build();
110         }
```

ot Copyright 2024 Open Text.

```
111                 if (users.containsKey(cookieUsername)) {
```

Sink: SpoofCookieAssignment.java:73 Return()

```
71                      return credentialsLoginFlow(username, password, response);
72                  } else {
73                      return cookieLoginFlow(cookieValue);
74                  }
75              }
```

## ContentTypeAssignment.java, line 82 (System Information Leak: External)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | . | | |

Source: ContentTypeAssignment.java:77 org.apache.commons.lang3.exception.ExceptionUtils.getStackTrace()

```
75                      }
76                  } catch (Exception e) {
77                      error = ExceptionUtils.getStackTrace(e);
78                      attackResult =
failed(this).feedback("xxe.content.type.feedback.xml").output(error).build();
79                  }
```

Sink: ContentTypeAssignment.java:82 Return attackResult()

```
80              }
81
82              return attackResult;
83          }
```

## ProfileUploadRemoveUserInput.java, line 27 (System Information Leak: External)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | . | | |

Source: ProfileUploadBase.java:53 java.lang.Throwable.getMessage()

```
51
52              } catch (IOException e) {
53                  return failed(this).output(e.getMessage()).build();
54              }
55          }
```

Sink: ProfileUploadRemoveUserInput.java:27 Return()

```
25      @ResponseBody
26      public AttackResult uploadFileHandler(@RequestParam("uploadedFileRemoveUserInput")
MultipartFile file) {
27          return super.execute(file, file.getOriginalFilename());
28      }
29  }
```

## SqlInjectionLesson8.java, line 55 (System Information Leak: External)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | . | | |

Source: SqlInjectionLesson8.java:89 java.lang.Throwable.getMessage()

```
87                      }
88                  } catch (SQLException e) {
89                      return failed(this).output("<br><span class='feedback-negative'>" +
e.getMessage() + "</span>").build();
90                  }
```

Sink: SqlInjectionLesson8.java:55 Return()

```
53      @ResponseBody
54      public AttackResult completed(@RequestParam String name, @RequestParam String
auth_tan) {
55          return injectableQueryConfidentiality(name, auth_tan);
56      }
```

Copyright 2024 Open Text.

## SqlInjectionLesson9.java, line 56 (System Information Leak: External)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | . | | |

Source: SqlInjectionLesson9.java:85 java.lang.Throwable.getMessage()

```
83          } catch (Exception e) {
84              System.err.println(e.getMessage());
85              return failed(this).output("<br><span class='feedback-negative'>" +
e.getMessage() + "</span>").build();
86          }
87      }
```

Sink: SqlInjectionLesson9.java:56 Return()

```
54      @ResponseBody
55      public AttackResult completed(@RequestParam String name, @RequestParam String
auth_tan) {
56          return injectableQueryIntegrity(name, auth_tan);
57      }
```

## ContentTypeAssignment.java, line 82 (System Information Leak: External)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | . | | |

Source: ContentTypeAssignment.java:77 Read e()

```
75                  }
76              } catch (Exception e) {
77                  error = ExceptionUtils.getStackTrace(e);
78                  attackResult =
failed(this).feedback("xxe.content.type.feedback.xml").output(error).build();
79              }
```

Sink: ContentTypeAssignment.java:82 Return attackResult()

```
80          }
81
82          return attackResult;
83      }
```

## SqlInjectionLesson5.java, line 67 (System Information Leak: External)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | . | | |

Source: SqlInjectionLesson5.java:80 java.lang.Throwable.getMessage()

```
78              }
79          } catch (Exception e) {
80              return failed(this).output(this.getClass().getName() + " : " +
e.getMessage() + "<br> Your query was: " + query).build();
81          }
82      }
```

Sink: SqlInjectionLesson5.java:67 Return()

```
65      public AttackResult completed(String query) {
66          createUser();
67          return injectableQuery(query);
68      }
```

## SSRFTask2.java, line 47 (System Information Leak: External)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | . | | |

Source: SSRFTask2.java:57 java.lang.Throwable.getMessage()

```
55                              .replaceAll("\n","<br>"); // Otherwise the \n gets escaped in
the response
```

```
56                      } catch (MalformedURLException e) {
57                          return getFailedResult(e.getMessage());
58                      } catch (IOException e) {
59                          //in case the external site is down, the test and lesson should still
be ok
```

Sink: SSRFTask2.java:47 Return()

```
45      @ResponseBody
46      public AttackResult completed(@RequestParam String url) {
47          return furBall(url);
48      }
```

## JWTRefreshEndpoint.java, line 112 (System Information Leak: External)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | JWTRefreshEndpoint.java checkout() 112 ok()() . ok() . | | |

Source: JWTRefreshEndpoint.java:112 java.lang.Throwable.getMessage()

```
110                     return ok(failed(this).feedback("jwt-refresh-not-
tom").feedbackArgs(user).build());
111             } catch (ExpiredJwtException e) {
112                 return ok(failed(this).output(e.getMessage()).build());
113             } catch (JwtException e) {
114                 return ok(failed(this).feedback("jwt-invalid-token").build());
```

Sink: JWTRefreshEndpoint.java:112 org.springframework.http.ResponseEntity.ok()

```
110                     return ok(failed(this).feedback("jwt-refresh-not-
tom").feedbackArgs(user).build());
111             } catch (ExpiredJwtException e) {
112                 return ok(failed(this).output(e.getMessage()).build());
113             } catch (JwtException e) {
114                 return ok(failed(this).feedback("jwt-invalid-token").build());
```

## SqlInjectionLesson10.java, line 53 (System Information Leak: External)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | . | | |

Source: SqlInjectionLesson10.java:85 java.lang.Throwable.getMessage()

```
83
84              } catch (Exception e) {
85                  return failed(this).output("<span class='feedback-negative'>" +
e.getMessage() + "</span>").build();
86              }
87      }
```

Sink: SqlInjectionLesson10.java:53 Return()

```
51      @ResponseBody
52      public AttackResult completed(@RequestParam String action_string) {
53          return injectableQueryAvailability(action_string);
54      }
```

## SimpleMailAssignment.java, line 73 (System Information Leak: External)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | . | | |

Source: SimpleMailAssignment.java:93 org.springframework.core.NestedRuntimeException.getMessage()

```
91                      restTemplate.postForEntity(webWolfURL, mailEvent, Object.class);
92                  } catch (RestClientException e) {
93                      return informationMessage(this).feedback("password-reset-
simple.email_failed").output(e.getMessage()).build();
94                  }
95                  return informationMessage(this).feedback("password-reset-
simple.email_send").feedbackArgs(email).build();
```

Sink: SimpleMailAssignment.java:73 Return()

```
71                      public AttackResult resetPassword(@RequestParam String emailReset) {
72                          String email = ofNullable(emailReset).orElse("unknown@webgoat.org");
73                          return sendEmail(extractUsername(email), email);
74                      }
```

## ProfileUpload.java, line 29 (System Information Leak: External)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | . | | |

**Source:** ProfileUploadBase.java:53 java.lang.Throwable.getMessage()

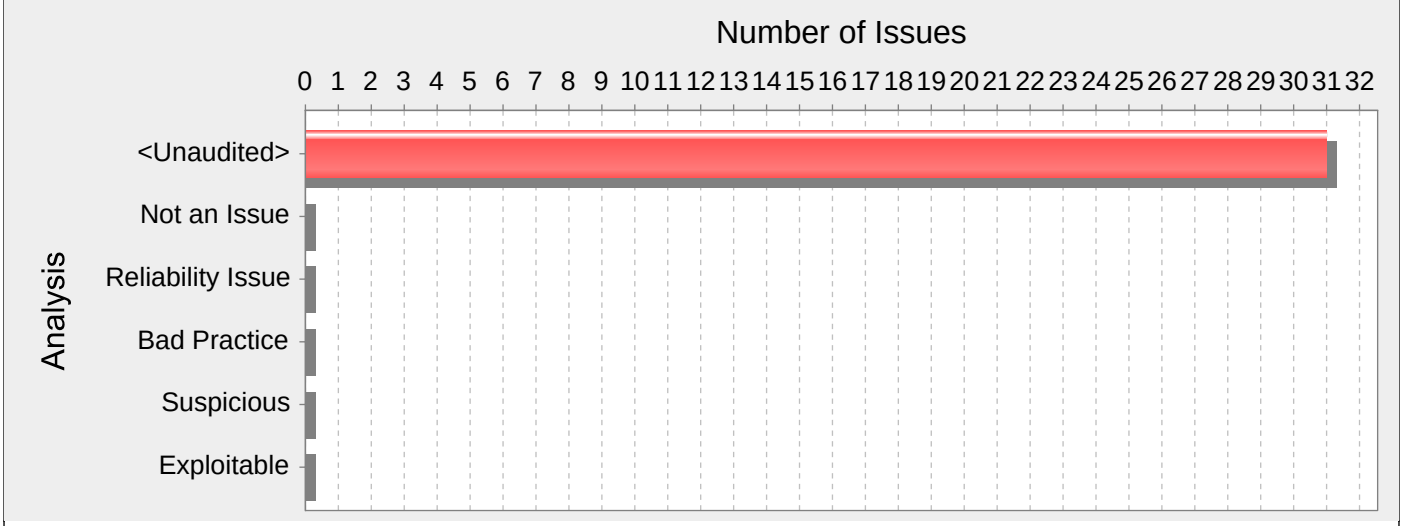
```
51
52                          } catch (IOException e) {
53                              return failed(this).output(e.getMessage()).build();
54                          }
55                      }
```

**Sink:** ProfileUpload.java:29 Return()

```
27                      @ResponseBody
28                      public AttackResult uploadFileHandler(@RequestParam("uploadedFile") MultipartFile
file, @RequestParam(value = "fullName", required = false) String fullName) {
29                          return super.execute(file, fullName);
30                      }
```

ot Copyright 2024 Open Text.

## 49: 04.01. (31 Issues)




**Abstract:**

MavenWrapperDownloader.java main() 92 printStackTrace()() . printStackTrace() .

**Explanation:**

.

1: .

```
try {
...
} catch (Exception e) {
e.printStackTrace();
}
```

. , " " " " . "syslog" . .

. , SQL injection . . Example 1 , .

. NFC( ) . NFC . NFC NFC , .

2: Android NFC . .

```
...
public static final String TAG = "NfcActivity";
private static final String DATA_SPLITTER = "__:DATA:__";
private static final String MIME_TYPE = "application/my.applications.mimetype";
...
public NdefMessage createNdefMessage(NfcEvent event) {
TelephonyManager tm = (TelephonyManager)Context.getSystemService(Context.TELEPHONY_SERVICE);
String VERSION = tm.getDeviceSoftwareVersion();
String text = TAG + DATA_SPLITTER + VERSION;
NdefRecord record = new NdefRecord(NdefRecord.TNF_MIME_MEDIA,
MIME_TYPE.getBytes(), new byte[0], text.getBytes());
NdefRecord[] records = { record };
NdefMessage msg = new NdefMessage(records);
return msg;
}
...
```

NDEF(NFC ) , URI . , MIME . Example 2 Fortify Static Code Analyzer return system information leak .

**Recommendations:**

. . . (: HTML ).

. , " " .

ot Copyright 2024 Open Text.

Android NFC . , , .

**Tips:**

1. System information leak , IT . .

2. . , , Audit Guide .

## SqlInjectionLesson10b.java, line 112 (System Information Leak)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | SqlInjectionLesson10b.java getJavaFileContentsAsString() 112 printStackTrace()() . printStackTrace() . | | |

**Sink:** SqlInjectionLesson10b.java:112 printStackTrace()

```
110                 javaFileObject = new JavaObjectFromString("TestClass.java",
javaFileContents.toString());
111             } catch (Exception exception) {
112                 exception.printStackTrace();
113             }
114             return javaFileObject;
```

## ProfileUploadRetrieval.java, line 48 (System Information Leak: Internal)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | ProfileUploadRetrieval.java ProfileUploadRetrieval() 48 mkdirs()() . mkdirs() . | | |

**Source:** ProfileUploadRetrieval.java:46 ProfileUploadRetrieval(0)

```
44      private final File catPicturesDirectory;
45
46      public ProfileUploadRetrieval(@Value("${webgoat.server.directory}") String
webGoatHomeDirectory) {
47          this.catPicturesDirectory = new File(webGoatHomeDirectory, "/PathTraversal/" +
"/cats");
48          this.catPicturesDirectory.mkdirs();
```

**Sink:** ProfileUploadRetrieval.java:48 java.io.File.mkdirs()

```
46      public ProfileUploadRetrieval(@Value("${webgoat.server.directory}") String
webGoatHomeDirectory) {
47          this.catPicturesDirectory = new File(webGoatHomeDirectory, "/PathTraversal/" +
"/cats");
48          this.catPicturesDirectory.mkdirs();
49      }
```

## FileServer.java, line 75 (System Information Leak: Internal)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | FileServer.java importFile() 75 mkdirs()() . mkdirs() . | | |

**Source:** FileServer.java:74 Read this.fileLocation()

```
72      public ModelAndView importFile(@RequestParam("file") MultipartFile myFile) throws
IOException {
73          var user = (WebGoatUser)
SecurityContextHolder.getContext().getAuthentication().getPrincipal();
74          var destinationDir = new File(fileLocation, user.getUsername());
75          destinationDir.mkdirs();
76          myFile.transferTo(new File(destinationDir, myFile.getOriginalFilename()));
```

**Sink:** FileServer.java:75 java.io.File.mkdirs()

```
73          var user = (WebGoatUser)
SecurityContextHolder.getContext().getAuthentication().getPrincipal();
74          var destinationDir = new File(fileLocation, user.getUsername());
75          destinationDir.mkdirs();
76          myFile.transferTo(new File(destinationDir, myFile.getOriginalFilename()));
77          log.debug("File saved to {}", new File(destinationDir,
myFile.getOriginalFilename()));
```

## Salaries.java, line 63 (System Information Leak: Internal)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | Salaries.java copyFiles() 63 mkdir()() . mkdir() . | | |

Source: Salaries.java:61 Read this.webGoatHomeDirectory()

```
59          public void copyFiles() {
60              ClassPathResource classPathResource = new
ClassPathResource("lessons/employees.xml");
61              File targetDirectory = new File(webGoatHomeDirectory, "/ClientSideFiltering");
62              if (!targetDirectory.exists()) {
63                  targetDirectory.mkdir();
```

Sink: Salaries.java:63 java.io.File.mkdir()

```
61              File targetDirectory = new File(webGoatHomeDirectory, "/ClientSideFiltering");
62              if (!targetDirectory.exists()) {
63                  targetDirectory.mkdir();
64              }
65              try {
```

## BlindSendFileAssignment.java, line 69 (System Information Leak: Internal)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | BlindSendFileAssignment.java createSecretFileWithRandomContents() 69 mkdirs()() . mkdirs() . | | |

Source: BlindSendFileAssignment.java:59 BlindSendFileAssignment(0)

```
57          private final Map<WebGoatUser, String> userToFileContents = new HashMap<>();
58
59          public BlindSendFileAssignment(@Value("${webgoat.user.directory}") String
webGoatHomeDirectory, CommentsCache comments) {
60              this.webGoatHomeDirectory = webGoatHomeDirectory;
61              this.comments = comments;
```

Sink: BlindSendFileAssignment.java:69 java.io.File.mkdirs()

```
67              File targetDirectory = new File(webGoatHomeDirectory, "/XXE/" +
user.getUsername());
68              if (!targetDirectory.exists()) {
69                  targetDirectory.mkdirs();
70              }
71              try {
```

## MavenWrapperDownloader.java, line 92 (System Information Leak)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | MavenWrapperDownloader.java main() 92 printStackTrace()() . printStackTrace() . | | |

Sink: MavenWrapperDownloader.java:92 printStackTrace()

```
90              } catch (Throwable e) {
91                  System.out.println("- Error downloading");
92                  e.printStackTrace();
93                  System.exit(1);
94              }
```

## ProfileUploadBase.java, line 42 (System Information Leak: Internal)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | ProfileUploadBase.java execute() 42 mkdirs()() . mkdirs() . | | |

Source: ProfileUploadFix.java:22 ProfileUploadFix(0)

```
20  public class ProfileUploadFix extends ProfileUploadBase {
21
22      public ProfileUploadFix(@Value("${webgoat.server.directory}") String
webGoatHomeDirectory, WebSession webSession) {
23          super(webGoatHomeDirectory, webSession);
```

Copyright 2024 Open Text.

```
24                      }
```

Sink: ProfileUploadBase.java:42 java.io.File.mkdirs()

```
40
41                  try {
42                      uploadDirectory.mkdirs();
43                      var uploadedFile = new File(uploadDirectory, fullName);
44                      uploadedFile.createNewFile();
```

## ProfileUploadRetrieval.java, line 103 (System Information Leak: Internal)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | ProfileUploadRetrieval.java getProfilePicture() 103 error()() . error() . | | |

Source: ProfileUploadRetrieval.java:103 Read e()

```
101
.body(StringUtils.arrayToCommaDelimitedString(catPicture.getParentFile().listFiles()).
getBytes());
102             } catch (IOException | URISyntaxException e) {
103                 log.error("Image not found", e);
104             }
```

Sink: ProfileUploadRetrieval.java:103 org.slf4j.Logger.error()

```
101
.body(StringUtils.arrayToCommaDelimitedString(catPicture.getParentFile().listFiles()).
getBytes());
102             } catch (IOException | URISyntaxException e) {
103                 log.error("Image not found", e);
104             }
```

## CryptoUtil.java, line 94 (System Information Leak: Internal)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | CryptoUtil.java verifyMessage() 94 error()() . error() . | | |

Source: CryptoUtil.java:94 Read e()

```
92            log.info("Verified the signature with result: {}", result);
93          } catch (Exception e) {
94            log.error("Signature verification failed", e);
95          }
```

Sink: CryptoUtil.java:94 org.slf4j.Logger.error()

```
92            log.info("Verified the signature with result: {}", result);
93          } catch (Exception e) {
94            log.error("Signature verification failed", e);
95          }
```

## SqlInjectionLesson9.java, line 77 (System Information Leak: Internal)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | SqlInjectionLesson9.java injectableQueryIntegrity() 77 println()() . println() . | | |

Source: SqlInjectionLesson9.java:77 java.lang.Throwable.getMessage()

```
75                      }
76                  } catch (SQLException e) {
77                      System.err.println(e.getMessage());
78                      return failed(this).output("<br><span class='feedback-negative'>" +
e.getMessage() + "</span>").build();
79                  }
```

Sink: SqlInjectionLesson9.java:77 java.io.PrintStream.println()

```
75                      }
76                  } catch (SQLException e) {
77                      System.err.println(e.getMessage());
78                      return failed(this).output("<br><span class='feedback-negative'>" +
e.getMessage() + "</span>").build();
79                  }
```

ot Copyright 2024 Open Text.

## ProfileUploadBase.java, line 42 (System Information Leak: Internal)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |

**Abstract:** ProfileUploadBase.java execute() 42 mkdirs()() . mkdirs() .

**Source:** ProfileUpload.java:22 ProfileUpload(0)

```
20  public class ProfileUpload extends ProfileUploadBase {
21
22      public ProfileUpload(@Value("${webgoat.server.directory}") String
    webGoatHomeDirectory, WebSession webSession) {
23          super(webGoatHomeDirectory, webSession);
24      }
```

**Sink:** ProfileUploadBase.java:42 java.io.File.mkdirs()

```
40
41          try {
42              uploadDirectory.mkdirs();
43              var uploadedFile = new File(uploadDirectory, fullName);
44              uploadedFile.createNewFile();
```

## CryptoIntegrationTest.java, line 35 (System Information Leak)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |

**Abstract:** CryptoIntegrationTest.java runTests() 35 printStackTrace()() . printStackTrace() .

**Sink:** CryptoIntegrationTest.java:35 printStackTrace()

```
33      checkAssignment4();
34    } catch (NoSuchAlgorithmException e) {
35      e.printStackTrace();
36      fail();
37    }
```

## SSRFTask1.java, line 68 (System Information Leak)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |

**Abstract:** SSRFTask1.java stealTheCheese() 68 printStackTrace()() . printStackTrace() .

**Sink:** SSRFTask1.java:68 printStackTrace()

```
66              }
67          } catch (Exception e) {
68              e.printStackTrace();
69              return failed(this)
70                      .output(e.getMessage())
```

## SqlInjectionLesson6b.java, line 77 (System Information Leak)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |

**Abstract:** SqlInjectionLesson6b.java getPassword() 77 printStackTrace()() . printStackTrace() .

**Sink:** SqlInjectionLesson6b.java:77 printStackTrace()

```
75              }
76          } catch (Exception e) {
77              e.printStackTrace();
78              // do nothing
79          }
```

## SqlInjectionLesson9.java, line 84 (System Information Leak: Internal)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |

ot Copyright 2024 Open Text.

| | |
|---|---|
| **Abstract:** | SqlInjectionLesson9.java injectableQueryIntegrity() 84 println()() . println() . |
| **Source:** | SqlInjectionLesson9.java:84 java.lang.Throwable.getMessage() |

```
82
83              } catch (Exception e) {
84                  System.err.println(e.getMessage());
85                  return failed(this).output("<br><span class='feedback-negative'>" +
e.getMessage() + "</span>").build();
86              }
```

| | |
|---|---|
| **Sink:** | SqlInjectionLesson9.java:84 java.io.PrintStream.println() |

```
82
83              } catch (Exception e) {
84                  System.err.println(e.getMessage());
85                  return failed(this).output("<br><span class='feedback-negative'>" +
e.getMessage() + "</span>").build();
86              }
```

## Requests.java, line 102 (System Information Leak: Internal)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | Requests.java toJsonString() 102 error()() . error() . | | |
| **Source:** | Requests.java:102 Read e() | | |

```
100                 return objectMapper.writeValueAsString(t);
101             } catch (JsonProcessingException e) {
102                 log.error("Unable to create json", e);
103             }
104             return "No request(s) found";
```

| | |
|---|---|
| **Sink:** | Requests.java:102 org.slf4j.Logger.error() |

```
100                 return objectMapper.writeValueAsString(t);
101             } catch (JsonProcessingException e) {
102                 log.error("Unable to create json", e);
103             }
104             return "No request(s) found";
```

## StartupMessage.java, line 22 (System Information Leak: Internal)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | StartupMessage.java onStartup() 22 info()() . info() . | | |
| **Source:** | StartupMessage.java:26 org.springframework.core.env.PropertyResolver.getProperty() | | |

```
24              if (event.getApplicationContext().getApplicationName().contains("WebGoat")) {
25                  port =
event.getApplicationContext().getEnvironment().getProperty("server.port");
26                  address =
event.getApplicationContext().getEnvironment().getProperty("server.address");
27              }
28          }
```

| | |
|---|---|
| **Sink:** | StartupMessage.java:22 org.slf4j.Logger.info() |

```
20          void onStartup(ApplicationReadyEvent event) {
21              if (StringUtils.hasText(port) &&
!StringUtils.hasText(System.getProperty("running.in.docker"))) {
22                  log.info("Please browse to http://{}:{}/WebGoat to get started...",
address, port);
23              }
24              if (event.getApplicationContext().getApplicationName().contains("WebGoat")) {
```

## ProfileUploadBase.java, line 42 (System Information Leak: Internal)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | ProfileUploadBase.java execute() 42 mkdirs()() . mkdirs() . | | |

Source: ProfileUploadRemoveUserInput.java:20 ProfileUploadRemoveUserInput(0)

```
18  public class ProfileUploadRemoveUserInput extends ProfileUploadBase {
19
20      public ProfileUploadRemoveUserInput(@Value("${webgoat.server.directory}") String
    webGoatHomeDirectory, WebSession webSession) {
21          super(webGoatHomeDirectory, webSession);
22      }
```

Sink: ProfileUploadBase.java:42 java.io.File.mkdirs()

```
40
41          try {
42              uploadDirectory.mkdirs();
43              var uploadedFile = new File(uploadDirectory, fullName);
44              uploadedFile.createNewFile();
```

## SqlInjectionLesson6b.java, line 73 (System Information Leak)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |

Abstract: SqlInjectionLesson6b.java getPassword() 73 printStackTrace()() . printStackTrace() .

Sink: SqlInjectionLesson6b.java:73 printStackTrace()

```
71                  }
72              } catch (SQLException sqle) {
73                  sqle.printStackTrace();
74                  // do nothing
75              }
```

## SqlInjectionLesson13.java, line 64 (System Information Leak: Internal)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |

Abstract: SqlInjectionLesson13.java completed() 64 error()() . error() .

Source: SqlInjectionLesson13.java:64 Read e()

```
62              return failed(this).build();
63          } catch (SQLException e) {
64              log.error("Failed", e);
65              return (failed(this).build());
66          }
```

Sink: SqlInjectionLesson13.java:64 org.slf4j.Logger.error()

```
62              return failed(this).build();
63          } catch (SQLException e) {
64              log.error("Failed", e);
65              return (failed(this).build());
66          }
```

## CryptoIntegrationTest.java, line 42 (System Information Leak)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |

Abstract: CryptoIntegrationTest.java runTests() 42 printStackTrace()() . printStackTrace() .

Sink: CryptoIntegrationTest.java:42 printStackTrace()

```
40    checkAssignmentSigning();
41   } catch (Exception e) {
42    e.printStackTrace();
43    fail();
44   }
```

## Salaries.java, line 107 (System Information Leak: Internal)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |

Copyright 2024 Open Text.

| | |
|---|---|
| **Abstract:** | Salaries.java invoke() 107 error()() . error() . |
| **Source:** | Salaries.java:76 Read this.webGoatHomeDirectory() |

```
74                  public List<Map<String, Object>> invoke() {
75                      NodeList nodes = null;
76                      File d = new File(webGoatHomeDirectory, "ClientSideFiltering/employees.xml");
77                      XPathFactory factory = XPathFactory.newInstance();
78                      XPath path = factory.newXPath();
```

| | |
|---|---|
| **Sink:** | Salaries.java:107 org.slf4j.Logger.error() |

```
105                         log.error("Unable to parse xml", e);
106                     } catch (IOException e) {
107                         log.error("Unable to read employees.xml at location: '{}'", d);
108                     }
109                     return json;
```

## CryptoUtilTest.java, line 28 (System Information Leak: Internal)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | CryptoUtilTest.java testSigningAssignment() 28 error()() . error() . | | |
| **Source:** | CryptoUtilTest.java:28 Read e() | | |

```
26                    assertTrue(CryptoUtil.verifyAssignment(modulus, signature, keyPair.getPublic()));
27                  } catch (Exception e) {
28                    log.error("signing failed", e);;
29                    fail();
30                  }
```

| | |
|---|---|
| **Sink:** | CryptoUtilTest.java:28 org.slf4j.Logger.error() |

```
26                    assertTrue(CryptoUtil.verifyAssignment(modulus, signature, keyPair.getPublic()));
27                  } catch (Exception e) {
28                    log.error("signing failed", e);;
29                    fail();
30                  }
```

## MvcConfiguration.java, line 63 (System Information Leak: Internal)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | MvcConfiguration.java createDirectory() 63 mkdirs()() . mkdirs() . | | |
| **Source:** | MvcConfiguration.java:61 Read this.fileLocation() | | |

```
59                  @PostConstruct
60                  public void createDirectory() {
61                      File file = new File(fileLocation);
62                      if (!file.exists()) {
63                          file.mkdirs();
```

| | |
|---|---|
| **Sink:** | MvcConfiguration.java:63 java.io.File.mkdirs() |

```
61                      File file = new File(fileLocation);
62                      if (!file.exists()) {
63                          file.mkdirs();
64                      }
65                  }
```

## ProfileUploadRetrieval.java, line 57 (System Information Leak: Internal)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | ProfileUploadRetrieval.java initAssignment() 57 error()() . error() . | | |
| **Source:** | ProfileUploadRetrieval.java:57 java.lang.Throwable.getMessage() | | |

```
55                          FileCopyUtils.copy(is, new FileOutputStream(new
File(catPicturesDirectory, i + ".jpg")));
56                      } catch (Exception e) {
57                          log.error("Unable to copy pictures" + e.getMessage());
58                      }
59                  }
```

**Sink:** ProfileUploadRetrieval.java:57 org.slf4j.Logger.error()

```
55                          FileCopyUtils.copy(is, new FileOutputStream(new
File(catPicturesDirectory, i + ".jpg")));
56                      } catch (Exception e) {
57                          log.error("Unable to copy pictures" + e.getMessage());
58                      }
59                  }
```

## SqlInjectionLesson10.java, line 100 (System Information Leak: Internal)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |

**Abstract:** SqlInjectionLesson10.java tableExists() 100 println()() . println() .

**Source:** SqlInjectionLesson10.java:100 java.lang.Throwable.getMessage()

```
98                          return false;
99                      } else {
100                         System.err.println(e.getMessage());
101                         return false;
102                     }
```

**Sink:** SqlInjectionLesson10.java:100 java.io.PrintStream.println()

```
98                          return false;
99                      } else {
100                         System.err.println(e.getMessage());
101                         return false;
102                     }
```

## PluginMessages.java, line 68 (System Information Leak: Internal)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |

**Abstract:** PluginMessages.java refreshProperties() 68 error()() . error() .

**Source:** PluginMessages.java:68 Read e()

```
66                      }
67                  } catch (IOException e) {
68                      logger.error("Unable to read plugin message", e);
69                  }
```

**Sink:** PluginMessages.java:68 org.apache.commons.logging.Log.error()

```
66                      }
67                  } catch (IOException e) {
68                      logger.error("Unable to read plugin message", e);
69                  }
```

## MD5.java, line 56 (System Information Leak: Internal)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |

**Abstract:** MD5.java main() 56 println()() . println() .

**Source:** MD5.java:56 java.lang.Throwable.getMessage()

```
54                          System.out.println(MD5.getHashString(new File(element)) + " " +
element);
55                      } catch (IOException x) {
56                          System.err.println(x.getMessage());
57                      }
58                  }
```

**Sink:** MD5.java:56 java.io.PrintStream.println()

```
54                          System.out.println(MD5.getHashString(new File(element)) + " " +
element);
55                      } catch (IOException x) {
56                          System.err.println(x.getMessage());
57                      }
58                  }
```

## SqlInjectionLesson8.java, line 140 (System Information Leak: Internal)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | SqlInjectionLesson8.java log() 140 println()() . println() . | | |

Source: SqlInjectionLesson8.java:140 java.lang.Throwable.getMessage()

```
138                 statement.executeUpdate(logQuery);
139             } catch (SQLException e) {
140                 System.err.println(e.getMessage());
141             }
142         }
```

Sink: SqlInjectionLesson8.java:140 java.io.PrintStream.println()

```
138                 statement.executeUpdate(logQuery);
139             } catch (SQLException e) {
140                 System.err.println(e.getMessage());
141             }
142         }
```

## LabelAndHintIntegrationTest.java, line 119 (System Information Leak)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | LabelAndHintIntegrationTest.java getProperties() 119 printStackTrace()() . printStackTrace() . | | |

Sink: LabelAndHintIntegrationTest.java:119 printStackTrace()

```
117                 prop.load(input);
118             } catch (Exception e) {
119                 e.printStackTrace();
120             }
121             return prop;
```

## CryptoUtil.java, line 64 (System Information Leak: Internal)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | CryptoUtil.java signMessage() 64 error()() . error() . | | |

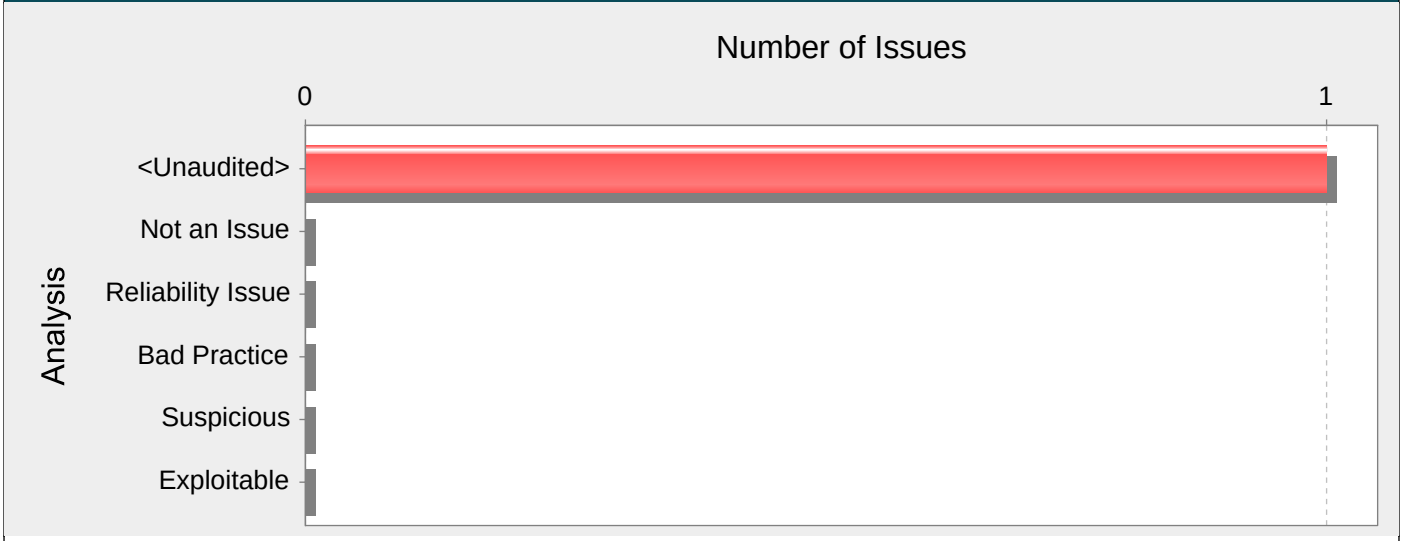
Source: CryptoUtil.java:64 Read e()

```
62             log.info("signe the signature with result: {}", signature);
63           } catch (Exception e) {
64             log.error("Signature signing failed", e);
65           }
```

Sink: CryptoUtil.java:64 org.slf4j.Logger.error()

```
62             log.info("signe the signature with result: {}", signature);
63           } catch (Exception e) {
64             log.error("Signature signing failed", e);
65           }
```

## 49: 02.15. (1 Issues)




**Abstract:**
WebWolfRedirect.java 19 openWebWolf() . .

**Explanation:**
.
1. .

2. .

1: .

...
String returnURL = request.getParameter("returnURL");
return new ModelAndView(returnURL);
...

URL , . "http://www.yourcorp.com/webApp/logic?returnURL=WEB-INF/applicationContext.xml" applicationContext.xml .
applicationContext.xml applicationContext.xml jar . .

**Recommendations:**
. .
.
< a href="http://www.yourcorp.com/webApp/logic?nextPage=WEB-INF/signup.jsp">New Customer</a>
.
< a href="http://www.yourcorp.com/webApp/logic?nextPage=newCustomer">New Customer</a>
"newCustomer" "/WEB-INF/signup.jsp" .

### WebWolfRedirect.java, line 19 (File Disclosure: Spring)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | API Abuse | | |
| **Abstract:** | WebWolfRedirect.java 19 openWebWolf() . . | | |

**Source:** WebWolfRedirect.java:17 org.springframework.core.env.PropertyResolver.getProperty()

```
15      @GetMapping("/WebWolf")
16      public ModelAndView openWebWolf() {
17          var url = applicationContext.getEnvironment().getProperty("webwolf.url");
18
19          return new ModelAndView("redirect:" + url + "/home");
```

**Sink:** WebWolfRedirect.java:19 org.springframework.web.servlet.ModelAndView.ModelAndView()

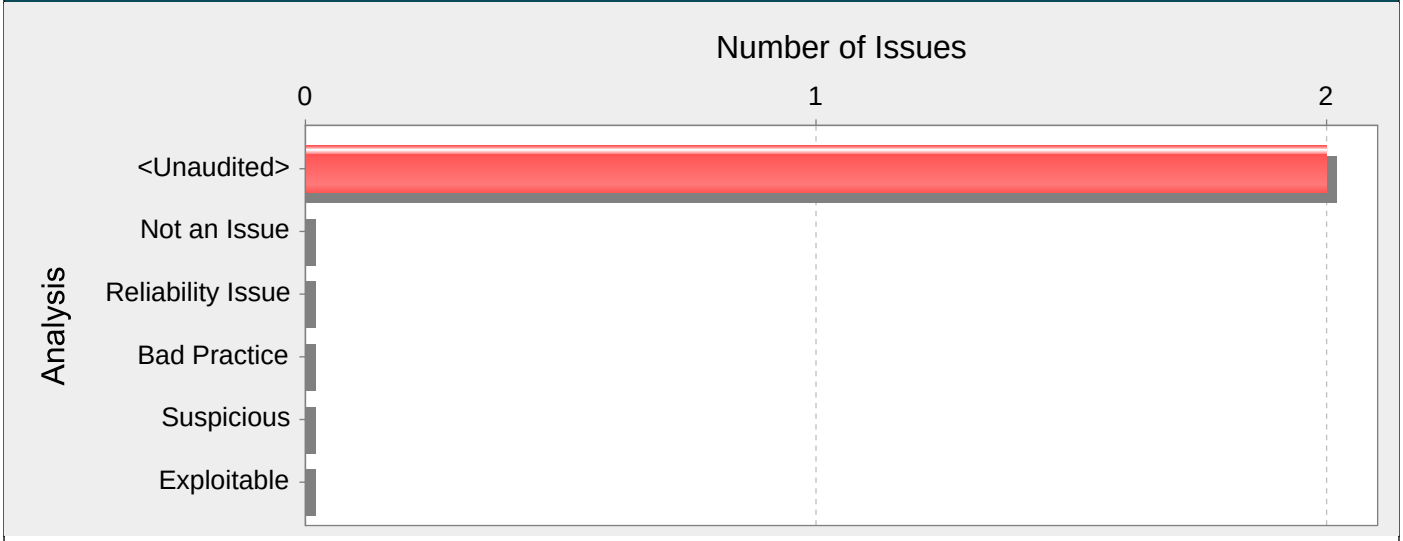
```
17          var url = applicationContext.getEnvironment().getProperty("webwolf.url");
```

ot Copyright 2024 Open Text.

```
18
19                          return new ModelAndView("redirect:" + url + "/home");
20                      }
21                  }
```

Copyright 2024 Open Text.

## 49: 02.11. (2 Issues)




**Abstract:**
ClientSideFiltering.html 96 .

**Explanation:**
. (: HTTP) (: HTTPS) MiTM(Man-in-The-Middle) . .

: .

<a href="http://www.example.com/index.html"/>

www.example.com .

.

**Recommendations:**
. . .

### ClientSideFiltering.html, line 96 (Insecure Transport: External Link)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Medium | **Folder** | Medium |
| **Kingdom:** | Security Features | | |
| **Abstract:** | ClientSideFiltering.html 96 . | | |

**Sink:** ClientSideFiltering.html:96

```
94                          <div class="col-xs-5" style="border:0px solid gray">
95                              <h3>Samsung Galaxy S8</h3>
96                              <h5 style="color:#337ab7"><a
href="http://www.samsung.com">Samsung</a> ·
97                                  <small style="color:#337ab7">(124421 reviews)</small>
98                              </h5>
```

### ClientSideFiltering.html, line 96 (Insecure Transport: External Link)

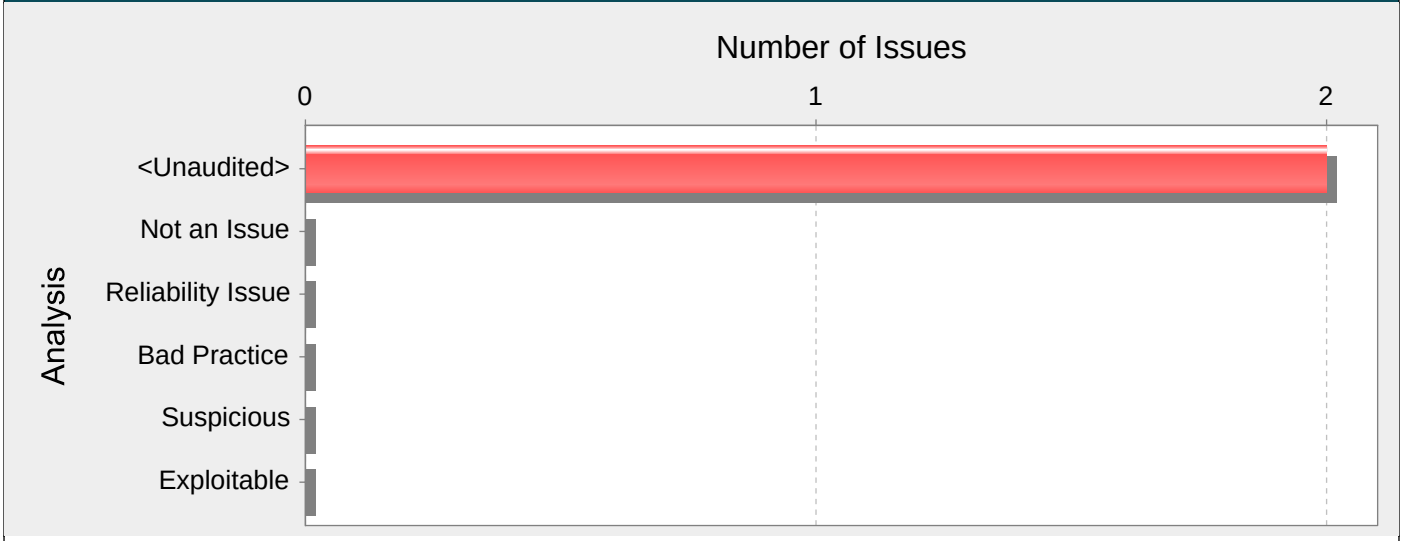
| | | | |
|---|---|---|---|
| **Fortify Priority:** | Medium | **Folder** | Medium |
| **Kingdom:** | Security Features | | |
| **Abstract:** | ClientSideFiltering.html 96 . | | |

**Sink:** ClientSideFiltering.html:96

```
94                          <div class="col-xs-5" style="border:0px solid gray">
95                              <h3>Samsung Galaxy S8</h3>
96                              <h5 style="color:#337ab7"><a
href="http://www.samsung.com">Samsung</a> ·
97                                  <small style="color:#337ab7">(124421 reviews)</small>
98                              </h5>
```

Copyright 2024 Open Text.

## 49: 02.09. (2 Issues)




**Abstract:**

getInstance()() .

**Explanation:**

. base64 .

1: .

```
...
Properties prop = new Properties();
prop.load(new FileInputStream("config.properties"));
String password = Base64.decode(prop.getProperty("password"));
DriverManager.getConnection(url, usr, password);
...
```

config.properties password base64 . .

.

2: Android WebView .

```
...
webview.setWebViewClient(new WebViewClient() {
public void onReceivedHttpAuthRequest(WebView view,
HttpAuthHandler handler, String host, String realm) {
String[] credentials = view.getHttpAuthUsernamePassword(host, realm);
String username = new String(Base64.decode(credentials[0], DEFAULT));
String password = new String(Base64.decode(credentials[1], DEFAULT));
handler.proceed(username, password);
}
});
...
```

WebView . .

3: MD5 Spring Security .

```
@Bean
public PasswordEncoder passwordEncoder() {
return new MD5PasswordEncoder();
}
```

**Recommendations:**

. . . .

. , WebSphere Application Server 4.x XOR . WebSphere .

Android SQLite SQLCipher . SQLCipher SQLite , 256 AES . .

4: SQLCipher Android .

```
import net.sqlcipher.database.SQLiteDatabase;
...
SQLiteDatabase.loadLibs(this);
File dbFile = getDatabasePath("credentials.db");
dbFile.mkdirs();
dbFile.delete();
SQLiteDatabase db = SQLiteDatabase.openOrCreateDatabase(dbFile, "credentials", null);
db.execSQL("create table credentials(u, p)");
db.execSQL("insert into credentials(u, p) values(?, ?)", new Object[]{username, password});
...
```

android.database.sqlite.SQLiteDatabase net.sqlcipher.database.SQLiteDatabase .

WebView sqlcipher.so WebKit .

Spring Security . BCryptPasswordEncoder, Pbkdf2PasswordEncoder SCryptPasswordEncoder . DelegatingPasswordEncoder . . .

5: Spring Security .

```
@Bean
public PasswordEncoder passwordEncoder() {
PasswordEncoder encoder = PasswordEncoderFactories.createDelegatingPasswordEncoder();
return encoder;
}
```

**Tips:**

1. Fortify Secure Coding Rulepacks . Fortify Static Code Analyzer .

. . . .

(pass-through rule) . .

2. (Struts Struts 2 ). , Fortify Fortify Static Code Analyzer . Context-Sensitive Ranking( ) . Fortify , Fortify Software Security Research Group .

## WebSecurityConfig.java, line 89 (Password Management: Weak Cryptography)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Security Features | | |
| **Abstract:** | getInstance()() . | | |
| **Sink:** | WebSecurityConfig.java:89 FunctionCall: getInstance() | | |

```
87      @Bean
88      public NoOpPasswordEncoder passwordEncoder() {
89          return (NoOpPasswordEncoder) NoOpPasswordEncoder.getInstance();
90      }
91  }
```

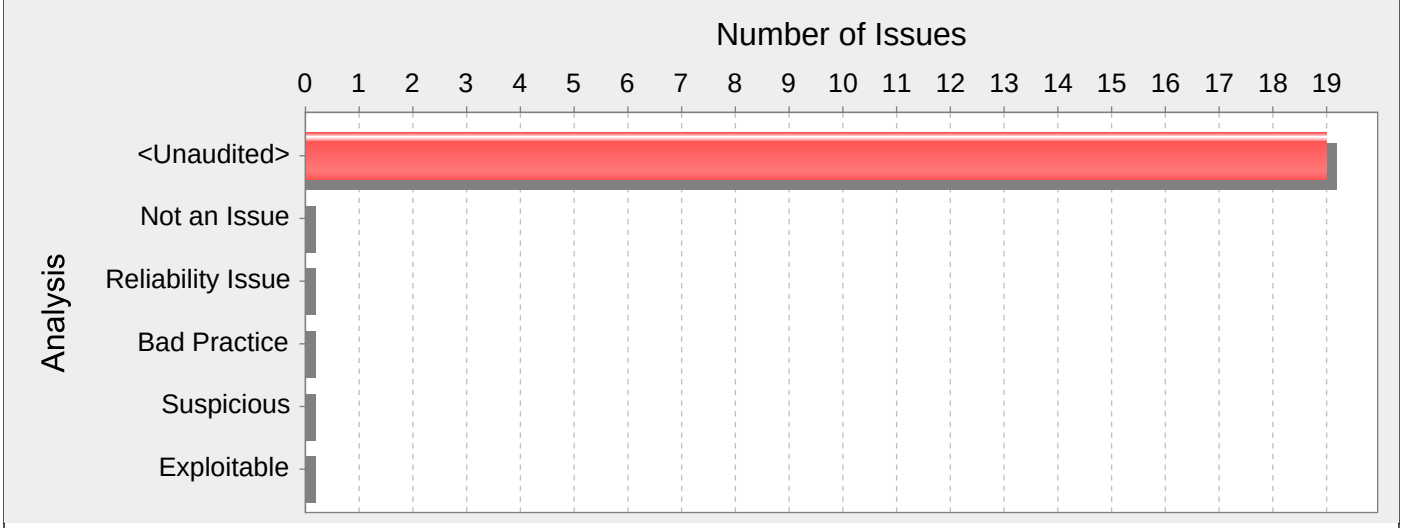
## WebSecurityConfig.java, line 100 (Password Management: Weak Cryptography)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Security Features | | |
| **Abstract:** | getInstance()() . | | |
| **Sink:** | WebSecurityConfig.java:100 FunctionCall: getInstance() | | |

```
98      @Bean
99      public NoOpPasswordEncoder passwordEncoder() {
100         return (NoOpPasswordEncoder) NoOpPasswordEncoder.getInstance();
101     }
102 }
```

ot Copyright 2024 Open Text.

## 49: 02.08. (19 Issues)




**Abstract:**

PasswordResetLink.java createPasswordReset() . .

**Explanation:**

Random.setSeed() . (PRNG) Random.nextInt(), Random.nextShort(), Random.nextLong() , Random.nextBoolean() , Random.nextBytes(byte[]) ( ) .

**Recommendations:**

, , PRNG(: java.security.SecureRandom) .

### HijackSessionAuthenticationProvider.java, line 51 (Insecure Randomness)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Security Features | | |
| **Abstract:** | nextLong() . | | |
| **Sink:** | HijackSessionAuthenticationProvider.java:51 nextLong() | | |

```
49
50                  private Queue<String> sessions = new LinkedList<>();
51                  private static long id = new Random().nextLong() & Long.MAX_VALUE;
52                  protected static final int MAX_SESSIONS = 50;
```

### CSRFGetFlag.java, line 63 (Insecure Randomness)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Security Features | | |
| **Abstract:** | nextInt() . | | |
| **Sink:** | CSRFGetFlag.java:63 nextInt() | | |

```
61                  if ("true".equals(req.getParameter("csrf"))) {
62                      Random random = new Random();
63                      userSessionData.setValue("csrf-get-success", random.nextInt(65536));
64                      response.put("success", true);
65                      response.put("message", pluginMessages.getMessage("csrf-get-null-
referer.success"));
```

### CSRFGetFlag.java, line 69 (Insecure Randomness)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Security Features | | |
| **Abstract:** | nextInt() . | | |
| **Sink:** | CSRFGetFlag.java:69 nextInt() | | |

```
67                  } else {
68                      Random random = new Random();
69                      userSessionData.setValue("csrf-get-success", random.nextInt(65536));
70                      response.put("success", true);
```

ot Copyright 2024 Open Text.

```
71                          response.put("message", pluginMessages.getMessage("csrf-get-other-
referer.success"));
```

## JWTRefreshEndpoint.java, line 90 (Insecure Randomness)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Security Features | | |
| **Abstract:** | randomAlphabetic() . | | |
| **Sink:** | JWTRefreshEndpoint.java:90 randomAlphabetic() | | |

```
88                          .compact();
89                  Map<String, Object> tokenJson = new HashMap<>();
90                  String refreshToken = RandomStringUtils.randomAlphabetic(20);
91                  validRefreshTokens.add(refreshToken);
92                  tokenJson.put("access_token", token);
```

## BlindSendFileAssignment.java, line 65 (Insecure Randomness)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Security Features | | |
| **Abstract:** | randomAlphabetic() . | | |
| **Sink:** | BlindSendFileAssignment.java:65 randomAlphabetic() | | |

```
63
64              private void createSecretFileWithRandomContents(WebGoatUser user) {
65                  var fileContents = "WebGoat 8.0 rocks... (" + randomAlphabetic(10) + ")";
66                  userToFileContents.put(user, fileContents);
67                  File targetDirectory = new File(webGoatHomeDirectory, "/XXE/" +
user.getUsername());
```

## ChromeDevTools.html, line 53 (Insecure Randomness)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Security Features | | |
| **Abstract:** | random() . | | |
| **Sink:** | ChromeDevTools.html:53 FunctionPointerCall: random() | | |

```
51                          // a namespace has been assigned for it, but you can roll your own if
you prefer
52                          document.getElementById("btn").addEventListener("click", function() {
53                              document.getElementById("networkNum").value = Math.random() * 100;
54                              document.getElementById("networkNumCopy").value =
document.getElementById("networkNum").value;
55                          });
```

## HashingAssignment.java, line 54 (Insecure Randomness)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Security Features | | |
| **Abstract:** | nextInt() . | | |
| **Sink:** | HashingAssignment.java:54 nextInt() | | |

```
52            if (md5Hash == null) {
53
54              String secret = SECRETS[new Random().nextInt(SECRETS.length)];
55
56                 MessageDigest md = MessageDigest.getInstance("MD5");
```

## CSRFGetFlag.java, line 80 (Insecure Randomness)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Security Features | | |
| **Abstract:** | nextInt() . | | |
| **Sink:** | CSRFGetFlag.java:80 nextInt() | | |

```
78                  } else {
79                      Random random = new Random();
80                      userSessionData.setValue("csrf-get-success", random.nextInt(65536));
81                      response.put("success", true);
```

ot Copyright 2024 Open Text.

```
82                      response.put("message", pluginMessages.getMessage("csrf-get-other-
referer.success"));
```

## PasswordResetLink.java, line 25 (Insecure Randomness)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Security Features | | |
| Abstract: | nextInt() . | | |

Sink: PasswordResetLink.java:25 nextInt()

```
23          char[] a = inputString.toCharArray();
24          for (int i = 0; i < a.length; i++) {
25              int j = random.nextInt(a.length);
26              char temp = a[i];
27              a[i] = a[j];
```

## EncDec.java, line 41 (Insecure Randomness)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Security Features | | |
| Abstract: | randomAlphabetic() . | | |

Sink: EncDec.java:41 randomAlphabetic()

```
39      // PoC: weak encoding method
40
41      private static final String SALT = RandomStringUtils.randomAlphabetic(10);
42
43      private EncDec() {
```

## HashingAssignment.java, line 73 (Insecure Randomness)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Security Features | | |
| Abstract: | nextInt() . | | |

Sink: HashingAssignment.java:73 nextInt()

```
71    String sha256 = (String) request.getSession().getAttribute("sha256");
72    if (sha256 == null) {
73     String secret = SECRETS[new Random().nextInt(SECRETS.length)];
74        sha256 = getHash(secret, "SHA-256");
75     request.getSession().setAttribute("sha256Hash", sha256);
```

## HijackSessionAuthenticationProvider.java, line 82 (Insecure Randomness)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Security Features | | |
| Abstract: | nextDouble() . | | |

Sink: HijackSessionAuthenticationProvider.java:82 nextDouble()

```
80
81      protected void authorizedUserAutoLogin() {
82          if
(!PROBABILITY_DOUBLE_PREDICATE.test(ThreadLocalRandom.current().nextDouble())) {
83              Authentication authentication = AUTHENTICATION_SUPPLIER.get();
84              authentication.setAuthenticated(true);
```

## HttpBasics.html, line 59 (Insecure Randomness)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Security Features | | |
| Abstract: | random() . | | |

Sink: HttpBasics.html:59 FunctionPointerCall: random()

```
57          // a namespace has been assigned for it, but you can roll your own if you prefer
58          webgoat.customjs.assignRandomVal = function () {
59            var x = Math.floor((Math.random() * 100) + 1);
60            document.getElementById("magic_num").value = x;
61          };
```

## EncodingAssignment.java, line 52 (Insecure Randomness)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Security Features | | |
| Abstract: | nextInt() . | | |
| Sink: | EncodingAssignment.java:52 nextInt() | | |

```
50                  String username = request.getUserPrincipal().getName();
51                  if (basicAuth == null) {
52                   String password =****** Random().nextInt(HashingAssignment.SECRETS.length)];
53                   basicAuth = getBasicAuth(username, password);
54                   request.getSession().setAttribute("basicAuth", basicAuth);
```

## ChromeDevTools.html, line 53 (Insecure Randomness)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Security Features | | |
| Abstract: | random() . | | |
| Sink: | ChromeDevTools.html:53 FunctionPointerCall: random() | | |

```
51                      // a namespace has been assigned for it, but you can roll your own if
you prefer
52                      document.getElementById("btn").addEventListener("click", function() {
53                          document.getElementById("networkNum").value = Math.random() * 100;
54                          document.getElementById("networkNumCopy").value =
document.getElementById("networkNum").value;
55                      });
```

## HttpBasics.html, line 59 (Insecure Randomness)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Security Features | | |
| Abstract: | random() . | | |
| Sink: | HttpBasics.html:59 FunctionPointerCall: random() | | |

```
57                  // a namespace has been assigned for it, but you can roll your own if you prefer
58                  webgoat.customjs.assignRandomVal = function () {
59                var x = Math.floor((Math.random() * 100) + 1);
60                document.getElementById("magic_num").value = x;
61              };
```

## ProfileUploadRetrieval.java, line 86 (Insecure Randomness)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Security Features | | |
| Abstract: | nextInt() . | | |
| Sink: | ProfileUploadRetrieval.java:86 nextInt() | | |

```
84                  try {
85                      var id = request.getParameter("id");
86                      var catPicture = new File(catPicturesDirectory, (id == null ?
RandomUtils.nextInt(1, 11) : id) + ".jpg");
87
88                      if (catPicture.getName().toLowerCase().contains("path-traversal-
secret.jpg")) {
```

## PasswordResetLink.java, line 17 (Insecure Randomness: User-Controlled Seed)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Security Features | | |
| Abstract: | PasswordResetLink.java createPasswordReset() . . | | |
| Source: | PasswordResetLink.java:33 main(0) | | |

```
31      }
32
33      public static void main(String[] args) {
34          if (args == null || args.length != 2) {
35              System.out.println("Need a username and key");
```

ot Copyright 2024 Open Text.

| Sink: | PasswordResetLink.java:17 java.util.Random.setSeed() |
|---|---|

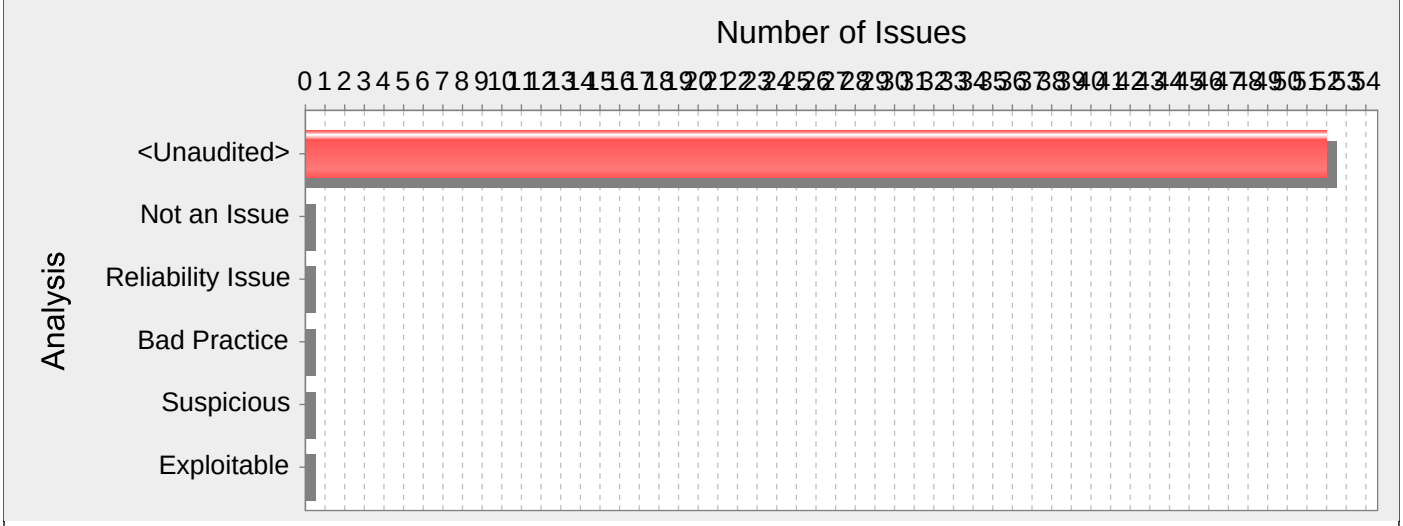
```
15                  if (username.equalsIgnoreCase("admin")) {
16                      //Admin has a fix reset link
17                      random.setSeed(key.length());
18                  }
19                  return scramble(random, scramble(random, scramble(random,
MD5.getHashString(username))));
```

| JWTSecretKeyEndpoint.java, line 55 (Insecure Randomness) | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Security Features | | |
| Abstract: | nextInt() . | | |
| Sink: | JWTSecretKeyEndpoint.java:55 nextInt() | | |

```
53
54      public static final String[] SECRETS = {"victory", "business", "available",
"shipping", "washington"};
55      public static final String JWT_SECRET = TextCodec.BASE64.encode(SECRETS[new
Random().nextInt(SECRETS.length)]);
56      private static final String WEBGOAT_USER = "WebGoat";
57      private static final List<String> expectedClaims = List.of("iss", "iat", "exp",
"aud", "sub", "username", "Email", "Role");
```

Copyright 2024 Open Text.

## 49: 02.06. (52 Issues)




**Abstract:**

Hardcoded Password .

**Explanation:**

. . . .

1: .

```
...
DriverManager.getConnection(url, "scott", "tiger");
...
```

. “scott” “tiger” . . javap -c . Example 1 .

```
javap -c ConnMngr.class
22: ldc #36; //String jdbc:mysql://ixne.com/rxsql
24: ldc #38; //String scott
26: ldc #17; //String tiger
```

.

2: Android WebView .

```
...
webview.setWebViewClient(new WebViewClient() {
public void onReceivedHttpAuthRequest(WebView view,
HttpAuthHandler handler, String host, String realm) {
handler.proceed("guest", "allow");
}
});
...
```

Example 1 .

**Recommendations:**

. . .

. , WebSphere Application Server 4.x XOR . WebSphere . .

Android SQLite SQLCipher . SQLCipher SQLite , 256 AES . .

3: SQLCipher Android .

```
import net.sqlcipher.database.SQLiteDatabase;
...
SQLiteDatabase.loadLibs(this);
File dbFile = getDatabasePath("credentials.db");
```

Copyright 2024 Open Text.

dbFile.mkdirs();
dbFile.delete();
SQLiteDatabase db = SQLiteDatabase.openOrCreateDatabase(dbFile, "credentials", null);
db.execSQL("create table credentials(u, p)");
db.execSQL("insert into credentials(u, p) values(?, ?)", new Object[]{username, password});
...

android.database.sqlite.SQLiteDatabase net.sqlcipher.database.SQLiteDatabase .

WebView sqlcipher.so WebKit .

**Tips:**

1. Fortify Java Annotations FortifyPassword FortifyNotPassword .

2. null , Hardcoded Password password . Fortify Custom Rules Editor Password Management .

## JWT_libraries.adoc, line 28 (Credential Management: Hardcoded API Credentials)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Security Features | | |
| **Abstract:** | API . | | |

**Sink:** JWT_libraries.adoc:28

```
26  [source]
27  ----
28  var token =
    "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9l
    IiwiaWF0IjoxNTE2MjM5MDIyfQ.NFvYpuwbF6YWbPyaNAGEPw9wbhiQSovvSrD89B8K7Ng";
29
30  Jwts.parser().setSigningKey("test").parseClaimsJws(token);
```

## JWT_libraries.adoc, line 40 (Credential Management: Hardcoded API Credentials)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Security Features | | |
| **Abstract:** | API . | | |

**Sink:** JWT_libraries.adoc:40

```
38  [source]
39  ----
40  var token = "
    eyJhbGciOiJub25lIiwidHlwIjoiSldUIn0.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIi
    wiaWF0IjoxNTE2MjM5MDIyfQ.NFvYpuwbF6YWbPyaNAGEPw9wbhiQSovvSrD89B8K7Ng";
41
42  Jwts.parser().setSigningKey("test").parseClaimsJws(token);
```

## JWTRefreshEndpointTest.java, line 179 (Password Management: Hardcoded Password)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Security Features | | |
| **Abstract:** | Hardcoded Password . | | |

**Sink:** JWTRefreshEndpointTest.java:179 FunctionCall: put()

```
177         Map<String, Object> loginJson = new HashMap<>();
178         loginJson.put("user", "Jerry");
179         loginJson.put("password", PASSWORD);
180         MvcResult result =
    mockMvc.perform(MockMvcRequestBuilders.post("/JWT/refresh/login")
181                 .contentType(MediaType.APPLICATION_JSON)
```

## CryptoUtil.java, line 123 (Key Management: Hardcoded Encryption Key)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Security Features | | |
| **Abstract:** | . | | |

**Sink:** CryptoUtil.java:123

```
121
```

ot Copyright 2024 Open Text.

```
122                  public static PrivateKey getPrivateKeyFromPEM(String privateKeyPem) throws
                    NoSuchAlgorithmException, InvalidKeySpecException {
123                    privateKeyPem = privateKeyPem.replace("-----BEGIN PRIVATE KEY-----", "");
124                    privateKeyPem = privateKeyPem.replace("-----END PRIVATE KEY-----", "");
125                    privateKeyPem = privateKeyPem.replace("\n", "").replace("\r", "");
```

## jwt-refresh.js, line 10 (Password Management: Hardcoded Password)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Security Features | | |
| **Abstract:** | Hardcoded Password . | | |

**Sink:** jwt-refresh.js:10 FieldAccess: password()

```
8                          url: 'JWT/refresh/login',
9                          contentType: "application/json",
10                         data: JSON.stringify({user: user, password: "bm5nhSkxCXZkKRy4"})
11                     }).success(
12                         function (response) {
```

## JWTRefreshEndpointTest.java, line 94 (Credential Management: Hardcoded API Credentials)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Security Features | | |
| **Abstract:** | API . | | |

**Sink:** JWTRefreshEndpointTest.java:94

```
92                     @Test
93                     public void checkoutWithTomsTokenFromAccessLogShouldFail() throws Exception {
94                         String accessTokenTom =
                    "eyJhbGciOiJIUzUxMiJ9.eyJpYXQiOjE1MjYxMzE0MTEsImV4cCI6MTUyNjIxNzgxMSwiYWRtaW4iOiJmYWxz
                    ZSIsInVzZXIiOiJUb20ifQ.DCoaq9zQkyDH25EcVWKcdbyVfUL4c9D4jRvsqOqvi9iAd4QuqmKcchfbU8FNzeB
                    NF9tLeFXHZLU4yRkq-bjm7Q";
95                         mockMvc.perform(MockMvcRequestBuilders.post("/JWT/refresh/checkout")
96                                 .header("Authorization", "Bearer " + accessTokenTom))
```

## JWT_decode.adoc, line 8 (Credential Management: Hardcoded API Credentials)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Security Features | | |
| **Abstract:** | API . | | |

**Sink:** JWT_decode.adoc:8

```
6                  [source]
7                  ----
8                  eyJhbGciOiJIUzI1NiJ9.ew0KICAiYXV0aG9yaXRpZXMiIDogWyAiUk9MRV9BRE1JTiIsICJST0xFX1VTRVIiI
                   F0sDQogICJjbGllbnRfaWQiIDogIm15LWNsaWVudC13aXRoLXNlY3JldCIsDQogICJleHAiIDogMTYwNzA5OTY
                   wOCwNCiAgImp0aSIgOiAiOWJjOTJhNDQtMGIxYS00YzVlLWJlNzAtZGE1MjA3NWI5YTg0IiwNCiAgInNjb3BlI
                   iA6IFsgInJlYWQiLCAid3JpdGUiF0sDQogICJ1c2VyX25hbWUiIDogInVzZXIiDQp9.9lYaULTuoIDJ86-
                   zKDSntJQyHPpJ2mZAbnWRfel99iI
9                  ----
```

## SqlInjectionLesson6b.java, line 61 (Password Management: Hardcoded Password)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Security Features | | |
| **Abstract:** | Hardcoded Password . | | |

**Sink:** SqlInjectionLesson6b.java:61 VariableAccess: password()

```
59
60                     protected String getPassword() {
61                         String password =******
62                         try (Connection connection = dataSource.getConnection()) {
63                             String query = "SELECT password FROM user_system_data WHERE user_name =
                    'dave'";
```

## IDORIntegrationTest.java, line 48 (Password Management: Hardcoded Password)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Security Features | | |
| **Abstract:** | Hardcoded Password . | | |

ot Copyright 2024 Open Text.

| Sink: | IDORIntegrationTest.java:48 FunctionCall: put() |
|---|---|
| 46 | params.clear(); |
| 47 | params.put("username", "tom"); |
| 48 | params.put("password", "cat"); |
| 49 | |
| 50 | |

## JWT_libraries.adoc, line 28 (Credential Management: Hardcoded API Credentials)

| Fortify Priority: | Critical | Folder | Critical |
|---|---|---|---|
| Kingdom: | Security Features | | |
| Abstract: | API . | | |

| Sink: | JWT_libraries.adoc:28 |
|---|---|
| 26 | [source] |
| 27 | ---- |
| 28 | var token = "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.NFvYpuwbF6YWbPyaNAGEPw9wbhiQSovvSrD89B8K7Ng"; |
| 29 | |
| 30 | Jwts.parser().setSigningKey("test").parseClaimsJws(token); |

## PathTraversal.html, line 258 (Password Management: Password in HTML Form)

| Fortify Priority: | Critical | Folder | Critical |
|---|---|---|---|
| Kingdom: | Security Features | | |
| Abstract: | HTML . | | |

| Sink: | PathTraversal.html:258 |
|---|---|
| 256 |  |
| 257 | <label>Password:</label> |
| 258 | <input class="form-control" type="password" id="passwordZipSlip" name="password" required |
| 259 | placeholder="Enter Password" value="test"/> |
| 260 |  |

## JWT_signing_solution.adoc, line 35 (Credential Management: Hardcoded API Credentials)

| Fortify Priority: | Critical | Folder | Critical |
|---|---|---|---|
| Kingdom: | Security Features | | |
| Abstract: | API . | | |

| Sink: | JWT_signing_solution.adoc:35 |
|---|---|
| 33 | GET http://localhost:8080/WebGoat/JWT/votings/login?user=Tom HTTP/1.1 |
| 34 | |
| 35 | access_token=eyJhbGciOiJIUzUxMiJ9.eyJpYXQiOjE2MDgxMjg1NjYsImFkbWluIjoiZmFsc2UiLCJ1c2VyIjoiVG9tIn0.rTSX6PSXqUoGUvQQDBiqX0re2BSt7s2-X6FPf34Qly9SMpqIUSP8jykedJbjOBNlM3_CTjgk1SvUv48Pz8zIzA |
| 36 | ---- |

## logs.txt, line 2 (Credential Management: Hardcoded API Credentials)

| Fortify Priority: | Critical | Folder | Critical |
|---|---|---|---|
| Kingdom: | Security Features | | |
| Abstract: | API . | | |

| Sink: | logs.txt:2 |
|---|---|
| 0 | |
| 1 | 194.201.170.15 - - [28/Jan/2016:21:28:01 +0100] "GET /JWT/refresh/checkout?token=eyJhbGciOiJIUzUxMiJ9.eyJpYXQiOjE1MjYxMzE0MTEsImV4cCI6MTUyNjIxNzgxMSwiYWRtaW4iOiJmYWxzZSIsInVzZXIiOiJUb20ifQ.DCoaq9zQkyDH25EcVWKcdbyVfUL4c9D4jRvsqOqvi9iAd4QuqmKcchfbU8FNzeBNF9tLeFXHZLU4yRkq-bjm7Q HTTP/1.1" 401 242 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0" "-" |
| 2 | 194.201.170.15 - - [28/Jan/2016:21:28:01 +0100] "POST /JWT/refresh/moveToCheckout HTTP/1.1" 200 12783 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0" "-" |
| 3 | 194.201.170.15 - - [28/Jan/2016:21:28:01 +0100] "POST /JWT/refresh/login HTTP/1.1" 200 212 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0" "-" |

## JWTTokenTest.java, line 40 (Credential Management: Hardcoded API Credentials)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Security Features | | |
| Abstract: | API . | | |

Sink: JWTTokenTest.java:40

```
38      @Test
39      void decodeValidSignedToken() {
40          var token = JWTToken.decode("eyJhbGciOiJIUzI1NiJ9.eyJ0ZXN0IjoidGVzdCJ9.KOobRHDYyaesV_doOk11XXGKSONwzllraAaqqM4VFE4", "test");
41
42          assertThat(token.getHeader()).contains("\"alg\" : \"HS256\"");
```

## WebGoatLabels.properties, line 22 (Credential Management: Hardcoded API Credentials)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Security Features | | |
| Abstract: | API . | | |

Sink: WebGoatLabels.properties:22

```
20  jwt-refresh-hint2=The token from the access log is no longer valid, can you find a way to refresh it?
21  jwt-refresh-hint3=The endpoint for refreshing a token is 'JWT/refresh/newToken'
22  jwt-refresh-hint4=Use the found access token in the Authorization: Bearer header and use your own refresh token
23  jwt-refresh-not-tom=User is not Tom but {0}, please try again
```

## PathTraversal.html, line 49 (Password Management: Password in HTML Form)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Security Features | | |
| Abstract: | HTML . | | |

Sink: PathTraversal.html:49

```
47                  <div class="form-group">
48                      <label>Password:</label>
49                      <input class="form-control" type="password" id="password" name="password" required
50                             placeholder="Enter Password" value="test"/>
51                      <span class="Error"></span>
```

## PathTraversal.html, line 160 (Password Management: Password in HTML Form)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Security Features | | |
| Abstract: | HTML . | | |

Sink: PathTraversal.html:160

```
158                 <div class="form-group">
159                     <label>Password:</label>
160                     <input class="form-control" type="password" id="passwordRemoveUserInput" name="password" required
161                            placeholder="Enter Password" value="test"/>
162                     <span class="Error"></span>
```

## UserServiceTest.java, line 51 (Password Management: Hardcoded Password)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Security Features | | |
| Abstract: | Hardcoded Password . | | |

Sink: UserServiceTest.java:51 VariableAccess: password()

```
49      public void testLoadUserByUsername(){
50          var username = "guest";
51          var password =******
52          WebGoatUser user = new WebGoatUser(username, password);
53          when(mockUserRepository.findByUsername(username)).thenReturn(user);
```

ot Copyright 2024 Open Text.

## JWTTokenTest.java, line 28 (Credential Management: Hardcoded API Credentials)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Security Features | | |
| Abstract: | API . | | |

Sink: JWTTokenTest.java:28

```
26          var token = JWTToken.encode(toString(headers), toString(payload), "webgoat");
27
28
            assertThat(token.getEncoded()).isEqualTo("eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0ZXN
            0IjoidGVzdCJ9.axNp9BkswwK_YRF2URJ5P1UejQNYZbK4qYcMnkusg6I");
29      }
```

## InsecureLoginTask.java, line 36 (Password Management: Hardcoded Password)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Security Features | | |
| Abstract: | Hardcoded Password . | | |

Sink: InsecureLoginTask.java:36 FunctionCall: equals()

```
34      @ResponseBody
35      public AttackResult completed(@RequestParam String username, @RequestParam String
        password) {
36       if ("CaptainJack".equals(username) && "BlackPearl".equals(password)) {
37        return success(this).build();
38       }
```

## goatkeystore.pkcs12, line 0 (Key Management: Hardcoded Encryption Key)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Security Features | | |
| Abstract: | . | | |

Sink: goatkeystore.pkcs12:0

```
-2      0 ^B^A^C0 ^F *H
-1      ^A^G^A ^D 0 0^Ee^F *H
```

## SolutionConstants.java, line 34 (Password Management: Hardcoded Password)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Security Features | | |
| Abstract: | Hardcoded Password . | | |

Sink: SolutionConstants.java:34 FieldAccess: PASSWORD()

```
32
33      //TODO should be random generated when starting the server
34      String PASSWORD =******
35      String PASSWORD_TOM = "thisisasecretfortomonly";
36      String ADMIN_PASSWORD_LINK = "375afe1104f4a487a73823c50a9292a2";
```

## JWT_libraries.adoc, line 40 (Credential Management: Hardcoded API Credentials)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Security Features | | |
| Abstract: | API . | | |

Sink: JWT_libraries.adoc:40

```
38      [source]
39      ----
40      var token = "
        eyJhbGciOiJub25lIiwidHlwIjoiSldUIn0.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIi
        wiaWF0IjoxNTE2MjM5MDIyfQ.NFvYpuwbF6YWbPyaNAGEPw9wbhiQSovvSrD89B8K7Ng";
41
42      Jwts.parser().setSigningKey("test").parseClaimsJws(token);
```

## JWT_signing_solution.adoc, line 35 (Credential Management: Hardcoded API Credentials)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |

| | |
|---|---|
| **Kingdom:** | Security Features |
| **Abstract:** | API . |
| **Sink:** | JWT_signing_solution.adoc:35 |

```
33  GET http://localhost:8080/WebGoat/JWT/votings/login?user=Tom HTTP/1.1
34
35  access_token=eyJhbGciOiJIUzUxMiJ9.eyJpYXQiOjE2MDgxMjg1NjYsImFkbWluIjoiZmFsc2UiLCJ1c2VyIjoiVG9tIn0.rTSX6PSXqUoGUvQQDBiqX0re2BSt7s2-X6FPf34Qly9SMpqIUSP8jykedJbjOBNlM3_CTjgk1SvUv48Pz8zIzA
36  ----
```

## JWT.html, line 310 (Credential Management: Hardcoded API Credentials)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Security Features | | |
| **Abstract:** | API . | | |
| **Sink:** | JWT.html:310 | | |

```
308             <form class="attack-form" accept-charset="UNKNOWN"
309                   method="POST"
310
action="/WebGoat/JWT/final/delete?token=eyJ0eXAiOiJKV1QiLCJraWQiOiJ3ZWJnb2F0X2tleSIsImFsZyI6IkhTMjU2In0.eyJpc3MiOiJXZWJHb2F0IFRva2VuIEJ1aWxkZXIiLCJpYXQiOjE1MjQyMTA5MDQsImV4cCI6MTYxODkwNTMwNCwiYXVkIjoid2ViZ29hdC5vcmciLCJzdWIiOiJqZXJyeUB3ZWJnb2F0LmNvbSIsInVzZXJuYW1lIjoiSmVycnkiLCJFbWFpbCI6ImplcnJ5QHdlYmdvYXQuY29tIiwiUm9sZSI6WyJDYXQiXX0.CgZ27DzgVW8gzc0n6izOU638uUCi6UhiOJKYzoEZGE8">
311                 <div class="container-fluid">
312                     <div id="toast"></div>
```

## PathTraversal.html, line 258 (Password Management: Password in HTML Form)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Security Features | | |
| **Abstract:** | HTML . | | |
| **Sink:** | PathTraversal.html:258 | | |

```
256                     <div class="form-group">
257                         <label>Password:</label>
258                         <input class="form-control" type="password" id="passwordZipSlip" name="password" required
259                                placeholder="Enter Password" value="test"/>
260                         <span class="Error"></span>
```

## JWTRefreshEndpoint.java, line 61 (Password Management: Hardcoded Password)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Security Features | | |
| **Abstract:** | Hardcoded Password . | | |
| **Sink:** | JWTRefreshEndpoint.java:61 FieldAccess: PASSWORD() | | |

```
59  public class JWTRefreshEndpoint extends AssignmentEndpoint {
60
61      public static final String PASSWORD =******
62      private static final String JWT_PASSWORD =******
63      private static final List<String> validRefreshTokens = new ArrayList<>();
```

## UserServiceTest.java, line 73 (Password Management: Hardcoded Password)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Security Features | | |
| **Abstract:** | Hardcoded Password . | | |
| **Sink:** | UserServiceTest.java:73 VariableAccess: password() | | |

```
71      public void testAddUser(){
72          var username = "guest";
73          var password =******
74
75          sut.addUser(username, password);
```

ot Copyright 2024 Open Text.

## JWTRefreshEndpointTest.java, line 103 (Credential Management: Hardcoded API Credentials)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Security Features | | |
| Abstract: | API . | | |

Sink: JWTRefreshEndpointTest.java:103

```
101     @Test
102     public void checkoutWitRandomTokenShouldFail() throws Exception {
103         String accessTokenTom =
"eyJhbGciOiJIUzUxMiJ9.eyJpLXQiOjE1MjYxMzE0MTEsImV4cCI6MTUyNjIxNzgxMSwiYWRtaW4iOiJmYWxzZSIsInVzZXIiOiJUb20ifQ.DCoaq9zQkyDH25EcVWKcdbyVfUL4c9D4jRvsqOqvi9iAd4QuqmKcchfbU8FNzeBNF9tLeFXHZLU4yRkq-bjm7Q";
104         mockMvc.perform(MockMvcRequestBuilders.post("/JWT/refresh/checkout")
105                 .header("Authorization", "Bearer " + accessTokenTom))
```

## IDORLogin.java, line 46 (Password Management: Hardcoded Password)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Security Features | | |
| Abstract: | Hardcoded Password . | | |

Sink: IDORLogin.java:46 FunctionCall: put()

```
44
45          idorUserInfo.put("tom", new HashMap<String, String>());
46          idorUserInfo.get("tom").put("password", "cat");
47          idorUserInfo.get("tom").put("id", "2342384");
48          idorUserInfo.get("tom").put("color", "yellow");
```

## PathTraversal.html, line 160 (Password Management: Password in HTML Form)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Security Features | | |
| Abstract: | HTML . | | |

Sink: PathTraversal.html:160

```
158                 <div class="form-group">
159                     <label>Password:</label>
160                     <input class="form-control" type="password"
id="passwordRemoveUserInput" name="password" required
161                            placeholder="Enter Password" value="test"/>
162                     <span class="Error"></span>
```

## JWTRefreshEndpointTest.java, line 156 (Password Management: Hardcoded Password)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Security Features | | |
| Abstract: | Hardcoded Password . | | |

Sink: JWTRefreshEndpointTest.java:156 FunctionCall: put()

```
154         Map<String, Object> loginJson = new HashMap<>();
155         loginJson.put("user", "Jerry");
156         loginJson.put("password", PASSWORD);
157         MvcResult result =
mockMvc.perform(MockMvcRequestBuilders.post("/JWT/refresh/login")
158                 .contentType(MediaType.APPLICATION_JSON)
```

## JWTRefreshEndpoint.java, line 74 (Password Management: Hardcoded Password)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Security Features | | |
| Abstract: | Hardcoded Password . | | |

Sink: JWTRefreshEndpoint.java:74 FunctionCall: equals()

```
72          String password =****** json.get("password");
73
74          if ("Jerry".equalsIgnoreCase(user) && PASSWORD.equals(password)) {
75              return ok(createNewTokens(user));
```

```
76                      }
```

## PathTraversal.html, line 103 (Password Management: Password in HTML Form)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Security Features | | |
| **Abstract:** | HTML . | | |

**Sink:** PathTraversal.html:103

```
101                     <div class="form-group">
102                         <label>Password:</label>
103                         <input class="form-control" type="password" id="passwordFix"
name="password" required
104                                placeholder="Enter Password" value="test"/>
105                         <span class="Error"></span>
```

## CSRFIntegrationTest.java, line 187 (Password Management: Hardcoded Password)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Security Features | | |
| **Abstract:** | Hardcoded Password . | | |

**Sink:** CSRFIntegrationTest.java:187 FunctionCall: put()

```
185             params.clear();
186             params.put("username", "csrf-" + this.getUser());
187             params.put("password", "password");
188
189             //login and get the new cookie
```

## GeneralLessonIntegrationTest.java, line 92 (Password Management: Hardcoded Password)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Security Features | | |
| **Abstract:** | Hardcoded Password . | | |

**Sink:** GeneralLessonIntegrationTest.java:92 FunctionCall: put()

```
90              params.clear();
91              params.put("username", "CaptainJack");
92              params.put("password", "BlackPearl");
93              checkAssignment(url("/WebGoat/InsecureLogin/task"), params, true);
94              checkResults("/InsecureLogin/");
```

## PathTraversal.html, line 49 (Password Management: Password in HTML Form)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Security Features | | |
| **Abstract:** | HTML . | | |

**Sink:** PathTraversal.html:49

```
47                      <div class="form-group">
48                          <label>Password:</label>
49                          <input class="form-control" type="password" id="password"
name="password" required
50                                 placeholder="Enter Password" value="test"/>
51                          <span class="Error"></span>
```

## JWTRefreshEndpointTest.java, line 73 (Credential Management: Hardcoded API Credentials)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Security Features | | |
| **Abstract:** | API . | | |

**Sink:** JWTRefreshEndpointTest.java:73

```
71
72              //Now create a new refresh token for Tom based on Toms old access token and
send the refresh token of Jerry
```

Copyright 2024 Open Text.

```
73                  String accessTokenTom =
"eyJhbGciOiJIUzUxMiJ9.eyJpYXQiOjE1MjYxMzE0MTEsImV4cCI6MTUyNjIxNzgxMSwiYWRtaW4iOiJmYWxz
ZSIsInVzZXIiOiJUb20ifQ.DCoaq9zQkyDH25EcVWKcdbyVfUL4c9D4jRvsqOqvi9iAd4QuqmKcchfbU8FNzeB
NF9tLeFXHZLU4yRkq-bjm7Q";
74                  Map<String, Object> refreshJson = new HashMap<>();
75                  refreshJson.put("refresh_token", refreshToken);
```

## CryptoUtil.java, line 42 (Key Management: Hardcoded Encryption Key)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Security Features | | |
| **Abstract:** | . | | |

**Sink:** CryptoUtil.java:42

```
40
41              public static String getPrivateKeyInPEM(KeyPair keyPair) {
42                  String encodedString = "-----BEGIN PRIVATE KEY-----\n";
43                  encodedString = encodedString+new
String(Base64.getEncoder().encode(keyPair.getPrivate().getEncoded()),Charset.forName("
UTF-8"))+"\n";
44                  encodedString = encodedString+"-----END PRIVATE KEY-----\n";
```

## GeneralLessonIntegrationTest.java, line 102 (Password Management: Hardcoded Password)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Security Features | | |
| **Abstract:** | Hardcoded Password . | | |

**Sink:** GeneralLessonIntegrationTest.java:102 FunctionCall: put()

```
100                 Map<String, Object> params = new HashMap<>();
101                 params.clear();
102                 params.put("password", "ajnaeliclm^&&@kjn.");
103                 checkAssignment(url("/WebGoat/SecurePasswords/assignment"), params, true);
104                 checkResults("SecurePasswords/");
```

## JWT.html, line 310 (Credential Management: Hardcoded API Credentials)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Security Features | | |
| **Abstract:** | API . | | |

**Sink:** JWT.html:310

```
308                 <form class="attack-form" accept-charset="UNKNOWN"
309                       method="POST"
310
action="/WebGoat/JWT/final/delete?token=eyJ0eXAiOiJKV1QiLCJraWQiOiJ3ZWJnb2F0X2tleSIsIm
FsZyI6IkhTMjU2In0.eyJpc3MiOiJXZWJHb2F0IFRva2VuIEJ1aWxkZXIiLCJpYXQiOjE1MjQyMTA5MDQsImV4
cCI6MTYxODkwNTMwNCwiYXVkIjoid2ViZ29hdC5vcmciLCJzdWIiOiJqZXJyeUB3ZWJnb2F0LmNvbSIsInVzZX
JuYW1lIjoiSmVycnkiLCJFbWFpbCI6ImplcnJ5QHdlYmdvYXQuY29tIiwiUm9sZSI6WyJDYXQiXX0.CgZ27Dzg
VW8gzc0n6izOU638uUCi6UhiOJKYzoEZGE8">
311                     <div class="container-fluid">
312                         <div id="toast"></div>
```

## IDORLogin.java, line 52 (Password Management: Hardcoded Password)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Security Features | | |
| **Abstract:** | Hardcoded Password . | | |

**Sink:** IDORLogin.java:52 FunctionCall: put()

```
50
51                  idorUserInfo.put("bill", new HashMap<String, String>());
52                  idorUserInfo.get("bill").put("password", "buffalo");
53                  idorUserInfo.get("bill").put("id", "2342388");
54                  idorUserInfo.get("bill").put("color", "brown");
```

## WebGoatLabels.properties, line 22 (Credential Management: Hardcoded API Credentials)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Security Features | | |

| | |
|---|---|
| **Abstract:** | API . |
| **Sink:** | WebGoatLabels.properties:22 |

```
20  jwt-refresh-hint2=The token from the access log is no longer valid, can you find a way to refresh it?
21  jwt-refresh-hint3=The endpoint for refreshing a token is 'JWT/refresh/newToken'
22  jwt-refresh-hint4=Use the found access token in the Authorization: Bearer header and use your own refresh token
23  jwt-refresh-not-tom=User is not Tom but {0}, please try again
```

## JWTFinalEndpointTest.java, line 28 (Credential Management: Hardcoded API Credentials)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Security Features | | |
| **Abstract:** | API . | | |
| **Sink:** | JWTFinalEndpointTest.java:28 | | |

```
26  public class JWTFinalEndpointTest extends LessonTest {
27
28      private static final String TOKEN_JERRY = "eyJraWQiOiJ3ZWJnb2F0X2tleSIsImFsZyI6IkhTNTEyIn0.eyJhdWQiOiJ3ZWJnb2F0Lm9yZyIsImVtYWlsIjoiamVycnlAd2ViZ29hdC5jb20iLCJ1c2VybmFtZSI6IkplcnJ5In0.xBc5FFwaOcuxjdr_VJ16n8Jb7vScuaZulNTl66F2MWF1aBe47QsUosvbjWGORNcMPiPNwnMu1Yb0WZVNrp2ZXA";
29
30      @BeforeEach
```

## JWTTokenTest.java, line 48 (Credential Management: Hardcoded API Credentials)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Security Features | | |
| **Abstract:** | API . | | |
| **Sink:** | JWTTokenTest.java:48 | | |

```
46      @Test
47      void decodeInvalidSignedToken() {
48          var token = JWTToken.decode("eyJhbGciOiJIUzI1NiJ9.eyJ0ZXsdfdfsaasfddfasN0IjoidGVzdCJ9.KOobRHDYyaesV_doOk11XXGKSONwzllraAaqqM4VFE4", "");
49
50          assertThat(token.getHeader()).contains("\"alg\" : \"HS256\"");
```

## JWT_decode.adoc, line 8 (Credential Management: Hardcoded API Credentials)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Security Features | | |
| **Abstract:** | API . | | |
| **Sink:** | JWT_decode.adoc:8 | | |

```
6  [source]
7  ----
8  eyJhbGciOiJIUzI1NiJ9.ew0KICAiYXV0aG9yaXRpZXMiIDogWyAiUk9MRV9BRE1JTiIsICJST0xFX1VTRVIiIF0sDQogICJjbGllbnRfaWQiIDogIm15LWNsaWVudC13aXRoLXNlY3JldCIsDQogICJleHAiIDogMTYwNzA5OTYwOCwNCiAgImp0aSIgOiAiOWJjOTJhNDQtMGIxYS00YzVlLWJlNzAtZGE1MjA3NWI5YTg0IiwNCiAgInNjb3BlIiA6IFsgInJlYWQiLCAid3JpdGUiIF0sDQogICJ1c2VyX25hbWUiIDogInVzZXIiDQp9.9lYaULTuoIDJ86-zKDSntJQyHPpJ2mZAbnWRfel99iI
9  ----
```

## jwt-refresh.js, line 10 (Password Management: Hardcoded Password)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Security Features | | |
| **Abstract:** | Hardcoded Password . | | |
| **Sink:** | jwt-refresh.js:10 FieldAccess: password() | | |

```
8          url: 'JWT/refresh/login',
9          contentType: "application/json",
10         data: JSON.stringify({user: user, password: "bm5nhSkxCXZkKRy4"})
11     }).success(
12         function (response) {
```

## SpoofCookieAssignmentTest.java, line 92 (Password Management: Hardcoded Password)

| Fortify Priority: | High | Folder | High |
|---|---|---|---|
| Kingdom: | Security Features | | |
| Abstract: | Hardcoded Password . | | |
| Sink: | SpoofCookieAssignmentTest.java:92 VariableAccess: password() | | |

```
90      void validLoginWithoutCookieTest() throws Exception {
91          String username = "webgoat";
92          String password =******
93
94          ResultActions result = mockMvc.perform(MockMvcRequestBuilders
```

## User.java, line 31 (Password Management: Empty Password)

| Fortify Priority: | High | Folder | High |
|---|---|---|---|
| Kingdom: | Security Features | | |
| Abstract: | . | | |
| Sink: | User.java:31 FieldAccess: password() | | |

```
29
30      private String username = "";
31      private String password =******
32
33      public String getPassword() {
```

## logs.txt, line 2 (Credential Management: Hardcoded API Credentials)

| Fortify Priority: | Critical | Folder | Critical |
|---|---|---|---|
| Kingdom: | Security Features | | |
| Abstract: | API . | | |
| Sink: | logs.txt:2 | | |

```
0
1   194.201.170.15 - - [28/Jan/2016:21:28:01 +0100] "GET
    /JWT/refresh/checkout?token=eyJhbGciOiJIUzUxMiJ9.eyJpYXQiOjE1MjYxMzE0MTEsImV4cCI6MTUyN
    jIxNzgxMSwiYWRtaW4iOiJmYWxzZSIsInVzZXIiOiJUb20ifQ.DCoaq9zQkyDH25EcVWKcdbyVfUL4c9D4jRvs
    qOqvi9iAd4QuqmKcchfbU8FNzeBNF9tLeFXHZLU4yRkq-bjm7Q HTTP/1.1" 401 242 "-" "Mozilla/5.0
    (X11; Ubuntu; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0" "-"
2   194.201.170.15 - - [28/Jan/2016:21:28:01 +0100] "POST /JWT/refresh/moveToCheckout
    HTTP/1.1" 200 12783 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:60.0)
    Gecko/20100101 Firefox/60.0" "-"
3   194.201.170.15 - - [28/Jan/2016:21:28:01 +0100] "POST /JWT/refresh/login HTTP/1.1" 200
    212 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:60.0) Gecko/20100101 Firefox/60.0"
    "-"
```

## PathTraversal.html, line 103 (Password Management: Password in HTML Form)

| Fortify Priority: | Critical | Folder | Critical |
|---|---|---|---|
| Kingdom: | Security Features | | |
| Abstract: | HTML . | | |
| Sink: | PathTraversal.html:103 | | |

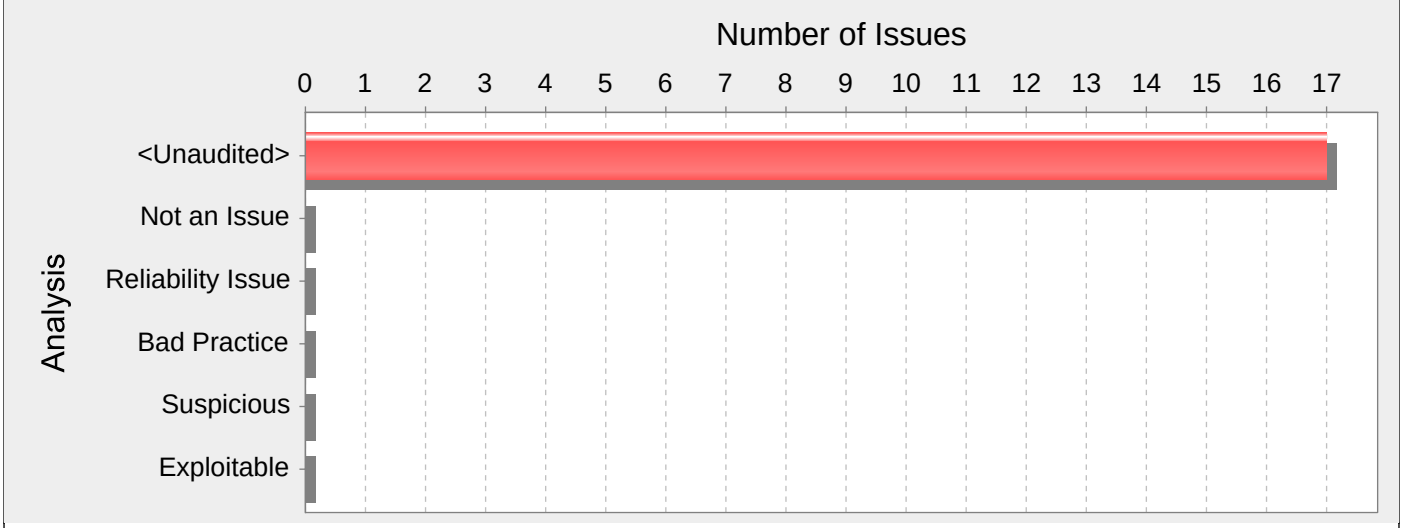
```
101                 <div class="form-group">
102                     <label>Password:</label>
103                     <input class="form-control" type="password" id="passwordFix"
    name="password" required
104                            placeholder="Enter Password" value="test"/>
105                     <span class="Error"></span>
```

## goatkeystore.pkcs12, line 0 (Key Management: Hardcoded Encryption Key)

| Fortify Priority: | High | Folder | High |
|---|---|---|---|
| Kingdom: | Security Features | | |
| Abstract: | . | | |
| Sink: | goatkeystore.pkcs12:0 | | |

```
-2  0 ^B^A^C0 ^F *H
-1  ^A^G^A ^D 0 0^Ee^F *H
```

## 49: 02.05. (17 Issues)




**Abstract:**
true Secure .

**Explanation:**
Secure . HTTPS . . ID .

1: Secure .

```
Cookie cookie = new Cookie("emailCookie", email);
response.addCookie(cookie);
```

HTTPS HTTP Secure , HTTPS HTTP . HTTP ( ID ) .

**Recommendations:**
Secure . setSecure(true) .

2:

```
Cookie cookie = new Cookie("emailCookie", email);
cookie.setSecure(true);
response.addCookie(cookie);
```

### JWTVotesEndpointTest.java, line 73 (Cookie Security: Cookie not Sent Over SSL)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Security Features | | |
| **Abstract:** | true Secure . | | |
| **Sink:** | JWTVotesEndpointTest.java:73 new Cookie(...) | | |

```
71          mockMvc.perform(MockMvcRequestBuilders.post("/JWT/votings")
72                  .contentType(MediaType.APPLICATION_JSON)
73                  .cookie(new Cookie("access_token", token)))
74                  .andExpect(status().isOk())
75                  .andExpect(jsonPath("$.lessonCompleted", is(true)));
```

### messages.properties, line 43 (Password Management: Password in Configuration File)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Environment | | |
| **Abstract:** | . | | |
| **Sink:** | messages.properties:43 accounts.table.password() | | |

```
41  accounts.table.account=Account
42  accounts.table.user=User
43  accounts.table.password=******
44  logout=Logout
45  version=Version
```

### JWTVotesEndpointTest.java, line 105 (Cookie Security: Cookie not Sent Over SSL)

ot Copyright 2024 Open Text.

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Security Features | | |
| Abstract: | true Secure . | | |

Sink: JWTVotesEndpointTest.java:105 new Cookie(...)

```
103     public void guestShouldNotSeeNumberOfVotes() throws Exception {
104         mockMvc.perform(MockMvcRequestBuilders.get("/JWT/votings")
105                 .cookie(new Cookie("access_token", "")))
106                 .andExpect(status().isOk())
107                 .andExpect(jsonPath("$[0].numberOfVotes").doesNotExist())
```

## JWTVotesEndpointTest.java, line 130 (Cookie Security: Cookie not Sent Over SSL)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Security Features | | |
| Abstract: | true Secure . | | |

Sink: JWTVotesEndpointTest.java:130 new Cookie(...)

```
128     public void invalidTokenShouldSeeGuestView() throws Exception {
129         mockMvc.perform(MockMvcRequestBuilders.get("/JWT/votings")
130                 .cookie(new Cookie("access_token", "abcd.efgh.ijkl")))
131                 .andExpect(status().isOk())
132                 .andExpect(jsonPath("$[0].numberOfVotes").doesNotExist())
```

## JWTVotesEndpoint.java, line 111 (Cookie Security: Cookie not Sent Over SSL)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Security Features | | |
| Abstract: | true Secure . | | |

Sink: JWTVotesEndpoint.java:111 addCookie(cookie)

```
109             } else {
110                 Cookie cookie = new Cookie("access_token", "");
111                 response.addCookie(cookie);
112                 response.setStatus(HttpStatus.UNAUTHORIZED.value());
113                 response.setContentType(MediaType.APPLICATION_JSON_VALUE);
```

## messages.properties, line 34 (Password Management: Password in Configuration File)

| Fortify Priority: | High | Folder | High |
|---|---|---|---|
| Kingdom: | Environment | | |
| Abstract: | . | | |

Sink: messages.properties:34 password()

```
32  InvalidData=Invalid Data
33  Go!=Go!
34  password=******
35  password.confirm=Confirm password
36  username=Username
```

## JWTVotesEndpointTest.java, line 199 (Cookie Security: Cookie not Sent Over SSL)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Security Features | | |
| Abstract: | true Secure . | | |

Sink: JWTVotesEndpointTest.java:199 new Cookie(...)

```
197
198             mockMvc.perform(MockMvcRequestBuilders.get("/JWT/votings/")
199                     .cookie(new Cookie("access_token", token)))
200                     .andExpect(status().isOk())
201                     .andExpect(jsonPath("$[0].numberOfVotes").doesNotExist())
```

## JWTVotesEndpoint.java, line 106 (Cookie Security: Cookie not Sent Over SSL)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Security Features | | |

| | |
|---|---|
| **Abstract:** | true Secure . |
| **Sink:** | JWTVotesEndpoint.java:106 addCookie(cookie) |

```
104                             .compact();
105                     Cookie cookie = new Cookie("access_token", token);
106                     response.addCookie(cookie);
107                     response.setStatus(HttpStatus.OK.value());
108                     response.setContentType(MediaType.APPLICATION_JSON_VALUE);
```

**HijackSessionAssignmentTest.java, line 84 (Cookie Security: Cookie not Sent Over SSL)**

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Security Features | | |
| **Abstract:** | true Secure . | | |
| **Sink:** | HijackSessionAssignmentTest.java:84 cookie = new Cookie(...) | | |

```
82              ReflectionTestUtils.setField(assignment, "provider", providerMock);
83
84              Cookie cookie = new Cookie(COOKIE_NAME, "value");
85
86              ResultActions result = mockMvc.perform(MockMvcRequestBuilders
```

**messages.properties, line 43 (Password Management: Password in Configuration File)**

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Environment | | |
| **Abstract:** | . | | |
| **Sink:** | messages.properties:43 accounts.table.password() | | |

```
41  accounts.table.account=Account
42  accounts.table.user=User
43  accounts.table.password=******
44  logout=Logout
45  version=Version
```

**JWTVotesEndpointTest.java, line 187 (Cookie Security: Cookie not Sent Over SSL)**

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Security Features | | |
| **Abstract:** | true Secure . | | |
| **Sink:** | JWTVotesEndpointTest.java:187 new Cookie(...) | | |

```
185
186             mockMvc.perform(MockMvcRequestBuilders.post("/JWT/votings/Admin lost
password")
187                     .cookie(new Cookie("access_token", token)))
188                     .andExpect(status().isUnauthorized());
189         }
```

**SpoofCookieAssignment.java, line 81 (Cookie Security: Cookie not Sent Over SSL)**

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Security Features | | |
| **Abstract:** | true Secure . | | |
| **Sink:** | SpoofCookieAssignment.java:81 addCookie(cookie) | | |

```
79              Cookie cookie = new Cookie(COOKIE_NAME, "");
80              cookie.setMaxAge(0);
81              response.addCookie(cookie);
82          }
```

**messages.properties, line 34 (Password Management: Password in Configuration File)**

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Environment | | |
| **Abstract:** | . | | |
| **Sink:** | messages.properties:34 password() | | |

ot Copyright 2024 Open Text.

```
32                  InvalidData=Invalid Data
33                  Go!=Go!
34                  password=******
35                  password.confirm=Confirm password
36                  username=Username
```

## SpoofCookieAssignmentTest.java, line 75 (Cookie Security: Cookie not Sent Over SSL)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Security Features | | |
| **Abstract:** | true Secure . | | |
| **Sink:** | SpoofCookieAssignmentTest.java:75 cookie = new Cookie(...) | | |

```
73                      @DisplayName("Lesson completed")
74                      void success() throws Exception {
75                          Cookie cookie = new Cookie(COOKIE_NAME,
"NjI2MTcwNGI3YTQxNGE1OTU2NzQ2ZDZmNzQ=");
76
77                          ResultActions result = mockMvc.perform(MockMvcRequestBuilders
```

## SpoofCookieAssignmentTest.java, line 113 (Cookie Security: Cookie not Sent Over SSL)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Security Features | | |
| **Abstract:** | true Secure . | | |
| **Sink:** | SpoofCookieAssignmentTest.java:113 cookie = new Cookie(...) | | |

```
111                             + "3.- Valid cookie with not known username sent ")
112                     void cookieLoginNotSolvedFlow(String cookieValue) throws Exception {
113                         Cookie cookie = new Cookie(COOKIE_NAME, cookieValue);
114                         mockMvc.perform(MockMvcRequestBuilders
115                             .post(LOGIN_CONTEXT_PATH)
```

## JWTVotesEndpointTest.java, line 175 (Cookie Security: Cookie not Sent Over SSL)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Security Features | | |
| **Abstract:** | true Secure . | | |
| **Sink:** | JWTVotesEndpointTest.java:175 new Cookie(...) | | |

```
173                     public void guestShouldNotBeAbleToVote() throws Exception {
174                         mockMvc.perform(MockMvcRequestBuilders.post("/JWT/votings/Admin lost
password")
175                                 .cookie(new Cookie("access_token", "")))
176                                 .andExpect(status().isUnauthorized());
177                     }
```

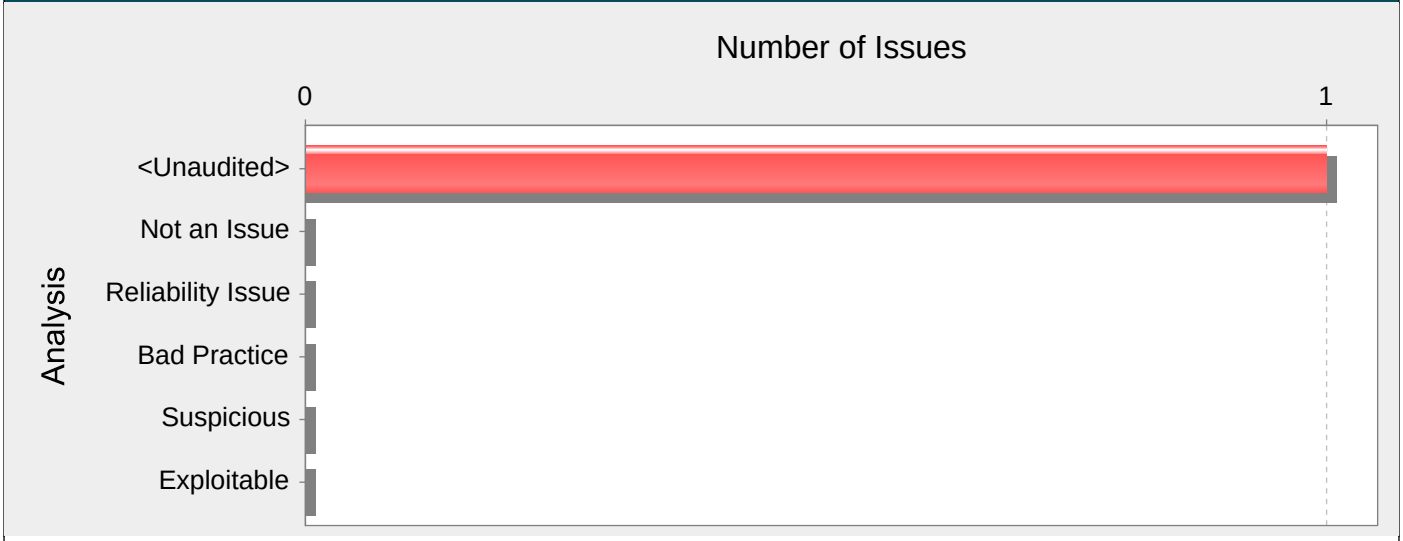
## CSRFFeedbackTest.java, line 69 (Cookie Security: Cookie not Sent Over SSL)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Security Features | | |
| **Abstract:** | true Secure . | | |
| **Sink:** | CSRFFeedbackTest.java:69 new Cookie(...) | | |

```
67                      mockMvc.perform(post("/csrf/feedback/message")
68                              .contentType(MediaType.TEXT_PLAIN)
69                              .cookie(new Cookie("JSESSIONID", "test"))
70                              .header("host", "localhost:8080")
71                              .header("referer", "webgoat.org")
```

ot Copyright 2024 Open Text.

## 49: 02.04. (1 Issues)




**Abstract:**

.

**Explanation:**

MD2, MD4, MD5, RIPEMD-160 SHA-1 . .

MD RIPEMD . SHA-1 , . .

**Recommendations:**

MD2, MD4, MD5, RIPEMD-160 SHA-1 . SHA-224, SHA-256, SHA-384, SHA-512 SHA-3 . SHA-1 .

### HashingAssignment.java, line 56 (Weak Cryptographic Hash)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Security Features | | |

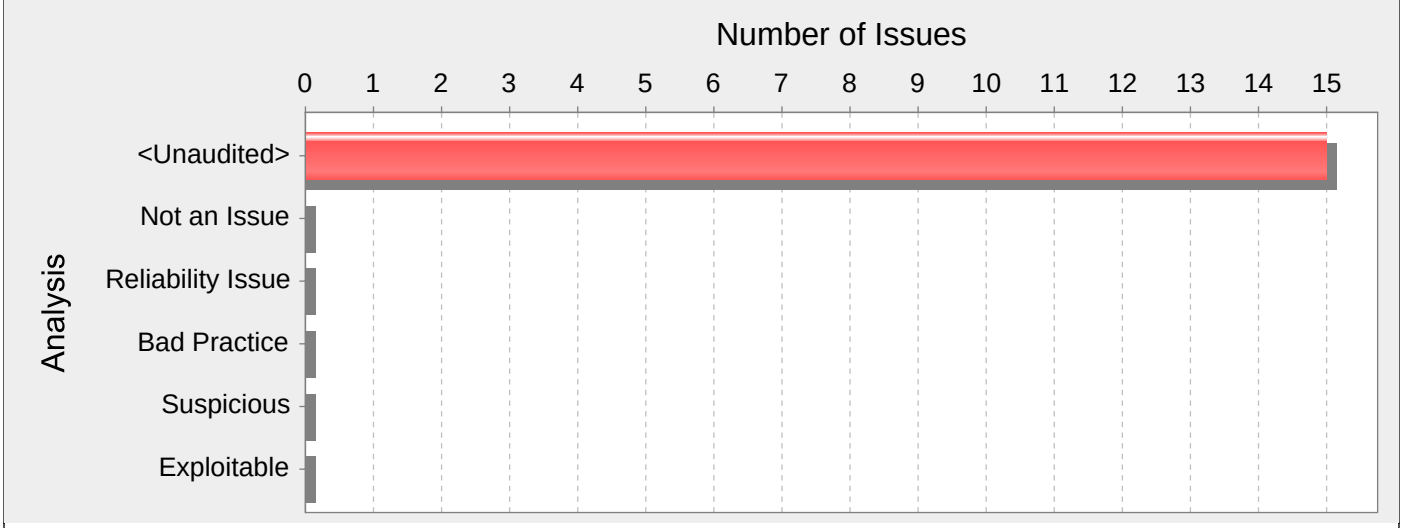
**Abstract:** .

**Sink:** HashingAssignment.java:56 getInstance()

```
54                String secret = SECRETS[new Random().nextInt(SECRETS.length)];
55
56                MessageDigest md = MessageDigest.getInstance("MD5");
57                md.update(secret.getBytes());
58                byte[] digest = md.digest();
```

Copyright 2024 Open Text.

## 49: 02.03. (15 Issues)




**Abstract:**
HijackSessionAssignment.java 86 HttpOnly true .

**Explanation:**
HttpOnly . Cross-Site Scripting ID . HttpOnly .

1: HttpOnly .

```
javax.servlet.http.Cookie cookie = new javax.servlet.http.Cookie("emailCookie", email);
// Missing a call to: cookie.setHttpOnly(true);
```

**Recommendations:**
HttpOnly . javax.servlet.http.Cookie true setHttpOnly(boolean) .

2: Example 1 HttpOnly true .

```
javax.servlet.http.Cookie cookie = new javax.servlet.http.Cookie("emailCookie", email);
cookie.setHttpOnly(true);
```

HttpOnly true .

### SpoofCookieAssignment.java, line 93 (Cookie Security: HTTPOnly not Set)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Security Features | | |
| **Abstract:** | SpoofCookieAssignment.java 93 HttpOnly true . | | |
| **Sink:** | SpoofCookieAssignment.java:93 newCookie = new Cookie(...) | | |

```
91          if (!authPassword.isBlank() && authPassword.equals(password)) {
92              String newCookieValue = EncDec.encode(lowerCasedUsername);
93              Cookie newCookie = new Cookie(COOKIE_NAME, newCookieValue);
94              newCookie.setPath("/WebGoat");
95              newCookie.setSecure(true);
```

### HijackSessionAssignmentTest.java, line 84 (Cookie Security: HTTPOnly not Set)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Security Features | | |
| **Abstract:** | HijackSessionAssignmentTest.java 84 HttpOnly true . | | |
| **Sink:** | HijackSessionAssignmentTest.java:84 cookie = new Cookie(...) | | |

```
82          ReflectionTestUtils.setField(assignment, "provider", providerMock);
83
84          Cookie cookie = new Cookie(COOKIE_NAME, "value");
85
86          ResultActions result = mockMvc.perform(MockMvcRequestBuilders
```

### JWTVotesEndpointTest.java, line 199 (Cookie Security: HTTPOnly not Set)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |

ot Copyright 2024 Open Text.

| | |
|---|---|
| Kingdom: | Security Features |
| Abstract: | JWTVotesEndpointTest.java 199 HttpOnly true . |
| Sink: | JWTVotesEndpointTest.java:199 new Cookie(...) |

```
197
198                     mockMvc.perform(MockMvcRequestBuilders.get("/JWT/votings/")
199                             .cookie(new Cookie("access_token", token)))
200                             .andExpect(status().isOk())
201                             .andExpect(jsonPath("$[0].numberOfVotes").doesNotExist())
```

## SpoofCookieAssignmentTest.java, line 113 (Cookie Security: HTTPOnly not Set)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Security Features | | |
| Abstract: | SpoofCookieAssignmentTest.java 113 HttpOnly true . | | |
| Sink: | SpoofCookieAssignmentTest.java:113 cookie = new Cookie(...) | | |

```
111                     + "3.- Valid cookie with not known username sent ")
112                 void cookieLoginNotSolvedFlow(String cookieValue) throws Exception {
113                     Cookie cookie = new Cookie(COOKIE_NAME, cookieValue);
114                     mockMvc.perform(MockMvcRequestBuilders
115                         .post(LOGIN_CONTEXT_PATH)
```

## JWTVotesEndpoint.java, line 110 (Cookie Security: HTTPOnly not Set)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Security Features | | |
| Abstract: | JWTVotesEndpoint.java 110 HttpOnly true . | | |
| Sink: | JWTVotesEndpoint.java:110 cookie = new Cookie(...) | | |

```
108                     response.setContentType(MediaType.APPLICATION_JSON_VALUE);
109                 } else {
110                     Cookie cookie = new Cookie("access_token", "");
111                     response.addCookie(cookie);
112                     response.setStatus(HttpStatus.UNAUTHORIZED.value());
```

## JWTVotesEndpointTest.java, line 130 (Cookie Security: HTTPOnly not Set)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Security Features | | |
| Abstract: | JWTVotesEndpointTest.java 130 HttpOnly true . | | |
| Sink: | JWTVotesEndpointTest.java:130 new Cookie(...) | | |

```
128                 public void invalidTokenShouldSeeGuestView() throws Exception {
129                     mockMvc.perform(MockMvcRequestBuilders.get("/JWT/votings")
130                             .cookie(new Cookie("access_token", "abcd.efgh.ijkl")))
131                             .andExpect(status().isOk())
132                             .andExpect(jsonPath("$[0].numberOfVotes").doesNotExist())
```

## JWTVotesEndpointTest.java, line 175 (Cookie Security: HTTPOnly not Set)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Security Features | | |
| Abstract: | JWTVotesEndpointTest.java 175 HttpOnly true . | | |
| Sink: | JWTVotesEndpointTest.java:175 new Cookie(...) | | |

```
173                 public void guestShouldNotBeAbleToVote() throws Exception {
174                     mockMvc.perform(MockMvcRequestBuilders.post("/JWT/votings/Admin lost
password")
175                             .cookie(new Cookie("access_token", "")))
176                             .andExpect(status().isUnauthorized());
177                 }
```

## JWTVotesEndpointTest.java, line 73 (Cookie Security: HTTPOnly not Set)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Security Features | | |

ot Copyright 2024 Open Text.

| | |
|---|---|
| **Abstract:** | JWTVotesEndpointTest.java 73 HttpOnly true . |
| **Sink:** | JWTVotesEndpointTest.java:73 new Cookie(...) |

```
71                  mockMvc.perform(MockMvcRequestBuilders.post("/JWT/votings")
72                          .contentType(MediaType.APPLICATION_JSON)
73                          .cookie(new Cookie("access_token", token)))
74                          .andExpect(status().isOk())
75                          .andExpect(jsonPath("$.lessonCompleted", is(true)));
```

**JWTVotesEndpoint.java, line 105 (Cookie Security: HTTPOnly not Set)**

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Security Features | | |
| **Abstract:** | JWTVotesEndpoint.java 105 HttpOnly true . | | |
| **Sink:** | JWTVotesEndpoint.java:105 cookie = new Cookie(...) | | |

```
103                         .signWith(io.jsonwebtoken.SignatureAlgorithm.HS512, JWT_PASSWORD)
104                         .compact();
105                 Cookie cookie = new Cookie("access_token", token);
106                 response.addCookie(cookie);
107                 response.setStatus(HttpStatus.OK.value());
```

**SpoofCookieAssignmentTest.java, line 75 (Cookie Security: HTTPOnly not Set)**

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Security Features | | |
| **Abstract:** | SpoofCookieAssignmentTest.java 75 HttpOnly true . | | |
| **Sink:** | SpoofCookieAssignmentTest.java:75 cookie = new Cookie(...) | | |

```
73      @DisplayName("Lesson completed")
74      void success() throws Exception {
75          Cookie cookie = new Cookie(COOKIE_NAME,
"NjI2MTcwNGI3YTQxNGE1OTU2NzQ2ZDZmNzQ=");
76
77          ResultActions result = mockMvc.perform(MockMvcRequestBuilders
```

**HijackSessionAssignment.java, line 86 (Cookie Security: HTTPOnly not Set)**

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Security Features | | |
| **Abstract:** | HijackSessionAssignment.java 86 HttpOnly true . | | |
| **Sink:** | HijackSessionAssignment.java:86 cookie = new Cookie(...) | | |

```
84
85      private void setCookie(HttpServletResponse response, String cookieValue) {
86          Cookie cookie = new Cookie(COOKIE_NAME, cookieValue);
87          cookie.setPath("/WebGoat");
88          cookie.setSecure(true);
```

**SpoofCookieAssignment.java, line 79 (Cookie Security: HTTPOnly not Set)**

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Security Features | | |
| **Abstract:** | SpoofCookieAssignment.java 79 HttpOnly true . | | |
| **Sink:** | SpoofCookieAssignment.java:79 cookie = new Cookie(...) | | |

```
77      @GetMapping(path = "/SpoofCookie/cleanup")
78      public void cleanup(HttpServletResponse response) {
79          Cookie cookie = new Cookie(COOKIE_NAME, "");
80          cookie.setMaxAge(0);
81          response.addCookie(cookie);
```

**CSRFFeedbackTest.java, line 69 (Cookie Security: HTTPOnly not Set)**

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Security Features | | |
| **Abstract:** | CSRFFeedbackTest.java 69 HttpOnly true . | | |

ot Copyright 2024 Open Text.

| Sink: | CSRFFeedbackTest.java:69 new Cookie(...) |
|---|---|

```
67              mockMvc.perform(post("/csrf/feedback/message")
68                      .contentType(MediaType.TEXT_PLAIN)
69                      .cookie(new Cookie("JSESSIONID", "test"))
70                      .header("host", "localhost:8080")
71                      .header("referer", "webgoat.org")
```

**JWTVotesEndpointTest.java, line 187 (Cookie Security: HTTPOnly not Set)**

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Security Features | | |
| Abstract: | JWTVotesEndpointTest.java 187 HttpOnly true . | | |
| Sink: | JWTVotesEndpointTest.java:187 new Cookie(...) | | |

```
185
186             mockMvc.perform(MockMvcRequestBuilders.post("/JWT/votings/Admin lost
password")
187                     .cookie(new Cookie("access_token", token)))
188                     .andExpect(status().isUnauthorized());
189         }
```

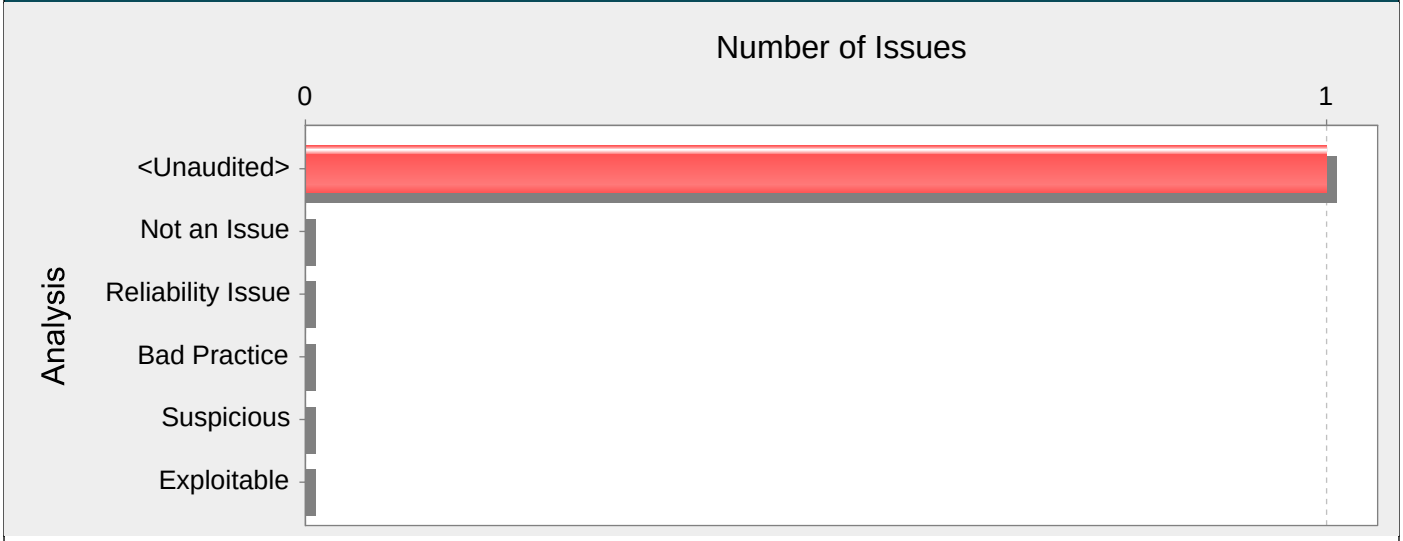
**JWTVotesEndpointTest.java, line 105 (Cookie Security: HTTPOnly not Set)**

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Security Features | | |
| Abstract: | JWTVotesEndpointTest.java 105 HttpOnly true . | | |
| Sink: | JWTVotesEndpointTest.java:105 new Cookie(...) | | |

```
103         public void guestShouldNotSeeNumberOfVotes() throws Exception {
104             mockMvc.perform(MockMvcRequestBuilders.get("/JWT/votings")
105                     .cookie(new Cookie("access_token", "")))
106                     .andExpect(status().isOk())
107                     .andExpect(jsonPath("$[0].numberOfVotes").doesNotExist())
```

ot Copyright 2024 Open Text.

## 49: 01.15. , 07.01. DNS lookup (1 Issues)




**Abstract:**
getLocalHost() . DNS . DNS .

**Explanation:**
DNS DNS . DNS (DNS (cache-poisoning) ) IP . DNS .

: DNS . DNS .

```
String ip = request.getRemoteAddr();
InetAddress addr = InetAddress.getByName(ip);
if (addr.getCanonicalHostName().endsWith("trustme.com")) {
trusted = true;
}
```

IP DNS . IP IP . IP . . . , IP authentication .

**Recommendations:**
DNS . DNS . . . DNS authentication .

authentication authentication . , password management . SSL . , authentication .

**Tips:**
1. DNS . authentication DNS . , , ?

### Assignment1Test.java, line 62 (Often Misused: Authentication)

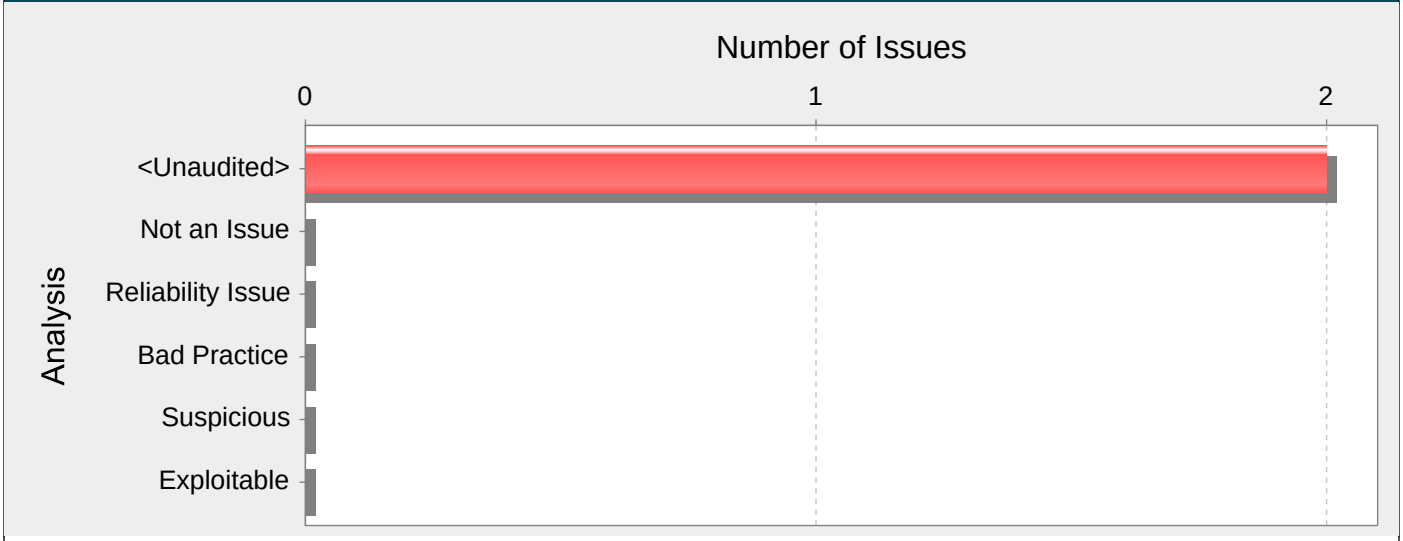
| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | API Abuse | | |
| **Abstract:** | getLocalHost() . DNS . DNS . | | |

**Sink:** Assignment1Test.java:62 getLocalHost()

```
60      @Test
61      void success() throws Exception {
62          InetAddress addr = InetAddress.getLocalHost();
63          String host = addr.getHostAddress();
64          mockMvc.perform(MockMvcRequestBuilders.post("/challenge/1")
```

ot Copyright 2024 Open Text.

## 49: 01.12. (2 Issues)




**Abstract:**
99 exchange() URI . . IP .

**Explanation:**
SSRF(Server-Side Request Forgery) . IP , .

: URL .

```
String url = request.getParameter("url");
CloseableHttpClient httpclient = HttpClients.createDefault();
HttpGet httpGet = new HttpGet(url);
CloseableHttpResponse response1 = httpclient.execute(httpGet);
```

URI . , URI http https .

- up://
- ldap://
- jar://
- gopher://
- mailto://
- ssh2://
- telnet://
- expect://

.

-
-
-
- Injection CSRF /
- file://
- Windows file:// UNC
- DNS

**Recommendations:**
. . URI . , . .

. . . . .

. URI , , . .

### ResetLinkAssignmentForgotPassword.java, line 99 (Server-Side Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | 99 exchange() URI . . IP . | | |

| | |
|---|---|
| Source: | ResetLinkAssignmentForgotPassword.java:69 javax.servlet.http.HttpServletRequest.getHeader() |

```
67                  String resetLink = UUID.randomUUID().toString();
68                  ResetLinkAssignment.resetLinks.add(resetLink);
69                  String host = request.getHeader("host");
70                  if (ResetLinkAssignment.TOM_EMAIL.equals(email) && (host.contains(webWolfPort)
|| host.contains(webWolfHost))) { //User indeed changed the host header.
71                      ResetLinkAssignment.userToTomResetLink.put(getWebSession().getUserName(),
resetLink);
```

| | |
|---|---|
| Sink: | ResetLinkAssignmentForgotPassword.java:99 org.springframework.web.client.RestTemplate.exchange() |

```
97                      HttpHeaders httpHeaders = new HttpHeaders();
98                      HttpEntity httpEntity = new HttpEntity(httpHeaders);
99                      new
RestTemplate().exchange(String.format("http://%s/PasswordReset/reset/reset-
password/%s", host, resetLink), HttpMethod.GET, httpEntity, Void.class);
100                 } catch (Exception e) {
101                     //don't care
```

## SSRFTask2.java, line 53 (Server-Side Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | 53 openStream() URI . . IP . | | |

| | |
|---|---|
| Source: | SSRFTask2.java:46 completed(0) |

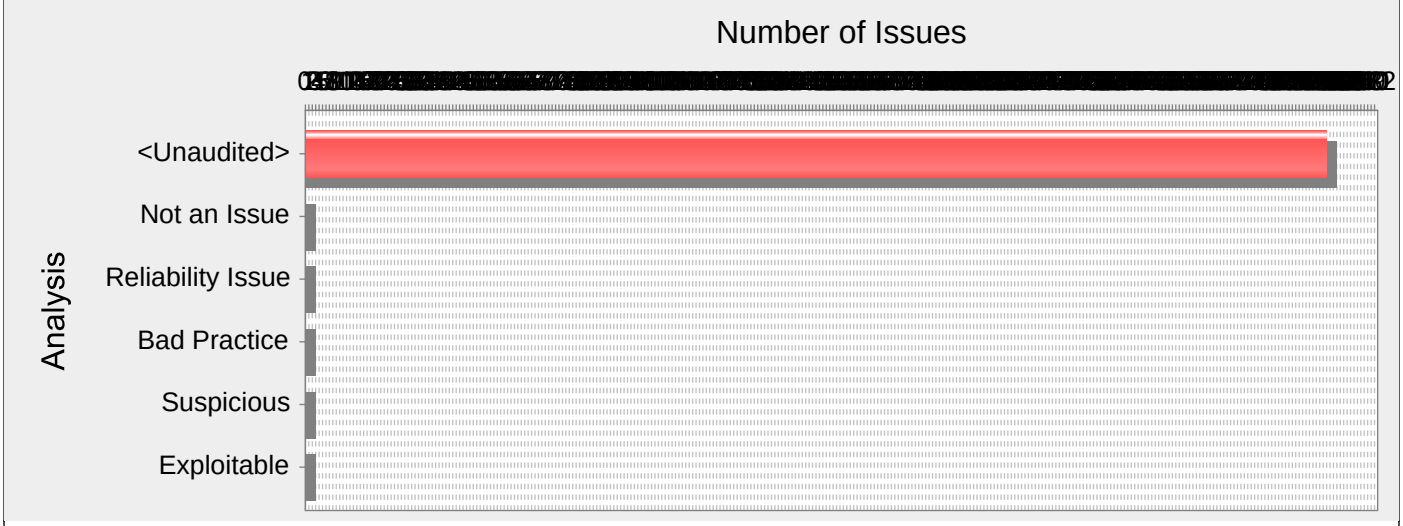
```
44      @PostMapping("/SSRF/task2")
45      @ResponseBody
46      public AttackResult completed(@RequestParam String url) {
47          return furBall(url);
48      }
```

| | |
|---|---|
| Sink: | SSRFTask2.java:53 java.net.URL.openStream() |

```
51              if (url.matches("http://ifconfig.pro")) {
52                  String html;
53                  try (InputStream in = new URL(url).openStream()) {
54                      html = new String(in.readAllBytes(), StandardCharsets.UTF_8)
55                              .replaceAll("\n","<br>"); // Otherwise the \n gets escaped in
the response
```

Copyright 2024 Open Text.

## 49: 01.11. (298 Issues)




### Abstract:

WebSecurityConfig.java 74 <a href="location://src/main/java/org/owasp/webgoat/container/WebSecurityConfig.java###74###31###0">disable()</a> HTTP .

### Explanation:

CSRF(cross-site request forgery) .

1. .

2. HTTP .

Nonce . , CSRF ( ). . .

```
RequestBuilder rb = new RequestBuilder(RequestBuilder.POST, "/new_user");
body = addToPost(body, new_username);
body = addToPost(body, new_passwd);
rb.sendRequest(body, new NewAccountCallback(callback));
```

.

```
RequestBuilder rb = new RequestBuilder(RequestBuilder.POST, "http://www.example.com/new_user");
body = addToPost(body, "attacker";
body = addToPost(body, "haha");
rb.sendRequest(body, new NewAccountCallback(callback));
```

example.com . CSRF . . . .

ID URL ID CSRF .

CSRF 2007 OWASP Top 10 5.

### Recommendations:

. ID Nonce .

```
RequestBuilder rb = new RequestBuilder(RequestBuilder.POST, "/new_user");
body = addToPost(body, new_username);
body = addToPost(body, new_passwd);
body = addToPost(body, request_id);
rb.sendRequest(body, new NewAccountCallback(callback));
```

ID . ID . ID ID CSRF . (: SSLv3 ) .

.

: CSRF CSRF .

- : CSRF . CAPTCHA, , .

HTTP Referer/Origin : CSRF . CSRF .

: ID ID CSRF . . ID .
: CSRF ID . .

XSS . CSRF XSS .

**Tips:**

1. Fortify Static Code Analyzer POST HTML XMLHttpRequest . CSRF .

## SqlInjectionAdvanced.html, line 80 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | SqlInjectionAdvanced.html 80 . | | |

**Sink:** SqlInjectionAdvanced.html:80

```
78                                  <div class="row">
79                                      <div class="col-lg-12">
80                                          <form id="login-form" class="attack-form" accept-charset="UNKNOWN"
81                                                  method="POST" name="form"
82                  action="/WebGoat/SqlInjectionAdvanced/challenge_Login"
```

## jquery.form.js, line 245 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | jquery.form.js 245 HTTP . | | |

**Sink:** jquery.form.js:245 FunctionPointerCall: get()

```
243             // see:  http://groups.google.com/group/jquery-dev/browse_thread/thread/36395b7ab510dd5d
244             if (options.closeKeepAlive) {
245                 $.get(options.closeKeepAlive, function() {
246                     jqxhr = fileUploadIframe(a);
247                 });
```

## HttpBasics.html, line 22 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | HttpBasics.html 22 . | | |

**Sink:** HttpBasics.html:22

```
20                  <!-- you can write your own custom forms, but standard form submission will take you to your endpoint and outside of the WebGoat framework -->
21                  <!-- of course, you can write your own ajax submission /handling in your own javascript if you like -->
22                <form class="attack-form" accept-charset="UNKNOWN"
23                 method="POST" name="form"
24                 action="/WebGoat/HttpBasics/attack1">
```

## Cryptography.html, line 31 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | Cryptography.html 31 . | | |

**Sink:** Cryptography.html:31

```
29                Now suppose you have intercepted the following header:<br/>
30                <div id="basicauthtoken" ></div><br/>
31                <form class="attack-form" method="POST" name="form" action="/WebGoat/crypto/encoding/basic-auth">
32                Then what was the username
33                <input name="answer_user" value="" type="TEXT"/>
```

## JWT.html, line 308 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |

ot Copyright 2024 Open Text.

| | |
|---|---|
| **Abstract:** | JWT.html 308 . |

**Sink:** JWT.html:308

```
306                 <div class="attack-container">
307                     <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
308                     <form class="attack-form" accept-charset="UNKNOWN"
309                           method="POST"
310
action="/WebGoat/JWT/final/delete?token=eyJ0eXAiOiJKV1QiLCJraWQiOiJ3ZWJnb2F0X2tleSIsIm
FsZyI6IkhTMjU2In0.eyJpc3MiOiJXZWJHb2F0IFRva2VuIEJ1aWxkZXIiLCJpYXQiOjE1MjQyMTA5MDQsImV4
cCI6MTYxODkwNTMwNCwiYXVkIjoid2ViZ29hdC5vcmciLCJzdWIiOiJqZXJyeUB3ZWJnb2F0LmNvbSIsInVzZX
JuYW1lIjoiSmVycnkiLCJFbWFpbCI6ImplcnJ5QHdlYmdvYXQuY29tIiwiUm9sZSI6WyJDYXQiXX0.CgZ27Dzg
VW8gzc0n6izOU638uUCi6UhiOJKYzoEZGE8">
```

## InsecureLogin.html, line 26 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | InsecureLogin.html 26 . | | |

**Sink:** InsecureLogin.html:26

```
24                  </form>
25                  <br></br>
26                  <form class="attack-form" accept-charset="UNKNOWN" name="task"
27                        method="POST"
28                        action="/WebGoat/InsecureLogin/task">
```

## PathTraversal.html, line 192 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | PathTraversal.html 192 . | | |

**Sink:** PathTraversal.html:192

```
190
191                 <br/>
192                 <form class="attack-form" method="POST" name="form"
action="/WebGoat/PathTraversal/random">
193                     <div class="assignment-success"><i class="fa fa-2 fa-check hidden"
aria-hidden="true"></i></div>
194                     <div class="form-group">
```

## SqlInjection.html, line 245 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | SqlInjection.html 245 . | | |

**Sink:** SqlInjection.html:245

```
243     <div class="attack-container">
244         <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
245         <form class="attack-form" accept-charset="UNKNOWN"
246               method="POST" name="form"
247               action="/WebGoat/SqlInjection/attack9"
```

## SqlInjectionMitigations.html, line 26 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | SqlInjectionMitigations.html 26 . | | |

**Sink:** SqlInjectionMitigations.html:26

```
24      <div class="attack-container">
25          <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
26          <form class="attack-form" accept-charset="UNKNOWN" method="POST" name="form"
action="/WebGoat/SqlInjectionMitigations/attack10a">
27              <div>
```

```
28                          <p>Connection conn = DriverManager.<input type="text" name="field1"
id="field1" />(DBURL, DBUSER, DBPW);</p>
```

## ChromeDevTools.html, line 25 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | ChromeDevTools.html 25 . | | |

**Sink:** ChromeDevTools.html:25

```
23      <div class="attack-container">
24          <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
25          <form class="attack-form" accept-charset="UNKNOWN"
26                method="POST" name="DOMFollowUp"
27                action="/WebGoat/ChromeDevTools/dummy">
```

## stored-xss.js, line 35 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | stored-xss.js 35 HTTP . | | |

**Sink:** stored-xss.js:35 FunctionPointerCall: get()

```
33      function getChallenges() {
34          $("#list").empty();
35          $.get('CrossSiteScripting/stored-xss', function (result, status) {
36              for (var i = 0; i < result.length; i++) {
37                  var comment = html.replace('USER', result[i].user);
```

## Challenge6.html, line 102 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | Challenge6.html 102 . | | |

**Sink:** Challenge6.html:102

```
100         </div>
101         <br/>
102         <form class="attack-form" method="POST" name="form"
action="/WebGoat/challenge/flag">
103             <div class="form-group">
104                 <div class="input-group">
```

## CSRF.html, line 213 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | CSRF.html 213 . | | |

**Sink:** CSRF.html:213

```
211             </i>
212         </div>
213         <form class="attack-form" accept-charset="UNKNOWN" id="confirm-flag-feedback"
214               method="POST" name="form2"
215               action="/WebGoat/csrf/feedback">
```

## WebWolfIntroduction.html, line 40 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | WebWolfIntroduction.html 40 . | | |

**Sink:** WebWolfIntroduction.html:40

```
38          <br/>
39          <!-- <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>-->
40          <form class="attack-form" accept-charset="UNKNOWN"
style="position:relative;top:-50px"
```

ot Copyright 2024 Open Text.

```
41                          method="POST" name="secondform"
42                          action="/WebGoat/WebWolf/mail/send">
```

## path_traversal.js, line 46 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | path_traversal.js 46 HTTP . | | |
| **Sink:** | path_traversal.js:46 FunctionPointerCall: get() | | |

```
44
45                  webgoat.customjs.profileUploadCallbackRemoveUserInput = function () {
46                      $.get("PathTraversal/profile-picture", function (result, status) {
47                          document.getElementById("previewRemoveUserInput").src =
"" + result;
48                      });
```

## HttpBasics.html, line 26 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | HttpBasics.html 26 . | | |
| **Sink:** | HttpBasics.html:26 | | |

```
24                      action="/WebGoat/HttpBasics/attack1">
25                      <div id="lessonContent">
26                       <form accept-charset="UNKNOWN" method="POST" name="form"
27                        action="#attack/307/100">
28                        Enter Your Name: <input name="person" value="" type="TEXT"/><input
```

## Challenge7.html, line 60 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | Challenge7.html 60 . | | |
| **Sink:** | Challenge7.html:60 | | |

```
58                  </div>
59                  <br/>
60                  <form class="attack-form" method="POST" name="form"
action="/WebGoat/challenge/flag">
61                      <div class="form-group">
62                          <div class="input-group">
```

## ClientSideFiltering.html, line 16 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | ClientSideFiltering.html 16 . | | |
| **Sink:** | ClientSideFiltering.html:16 | | |

```
14                  <input type="hidden" id="user_id" value="102"/>
15                  <!-- using attack-form class on your form, will allow your request to be
ajaxified and stay within the display framework for webgoat -->
16                  <form class="attack-form" accept-charset="UNKNOWN" method="POST" name="form"
17                        action="/WebGoat/clientSideFiltering/attack1">
18                      <link rel="stylesheet" type="text/css"
```

## LessonController.js, line 147 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | LessonController.js 147 HTTP . | | |
| **Sink:** | LessonController.js:147 AssignmentStatement() | | |

```
145                 $.ajax({
146                     url:'service/restartlesson.mvc',
147                     method:'GET'
```

ot Copyright 2024 Open Text.

```
148                                 }).done(function(lessonLink) {
149                                     self.loadLesson(self.name);
```

## webwolfPasswordReset.html, line 12 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | webwolfPasswordReset.html 12 . | | |

**Sink:** webwolfPasswordReset.html:12

```
10      <div class="row">
11          <div class="col-xs-12 col-sm-8 col-md-6 col-sm-offset-2 col-md-offset-3">
12              <form role="form" method="GET" th:action="${webwolfUrl}">
13                  <h2 class="sign_up_title">Reset your password</h2>
14                  <input type="hidden" name="uniqueCode" th:value="${uniqueCode}"/>
```

## MissingFunctionAC.html, line 99 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | MissingFunctionAC.html 99 . | | |

**Sink:** MissingFunctionAC.html:99

```
97      <div class="attack-container">
98          <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
99          <form class="attack-form" accept-charset="UNKNOWN"
100               method="POST" name="form"
101               action="/WebGoat/access-control/user-hash-fix">
```

## challenge8.js, line 46 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | challenge8.js 46 HTTP . | | |

**Sink:** challenge8.js:46 FunctionPointerCall: get()

```
44  function doVote(stars) {
45      $("#voteResultMsg").hide();
46      $.get("challenge/8/vote/" + stars, function (result) {
47          if (result["error"]) {
48              $("#voteResultMsg").addClass('alert-danger alert-dismissable');
```

## Challenge1.html, line 18 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | Challenge1.html 18 . | | |

**Sink:** Challenge1.html:18

```
16                  </div>
17                  <div class="panel-body">
18                      <form class="attack-form" accept-charset="UNKNOWN"
19                            method="POST" name="form"
20                            action="/WebGoat/challenge/1"
```

## VulnerableComponents.html, line 104 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | VulnerableComponents.html 104 . | | |

**Sink:** VulnerableComponents.html:104

```
102     action="/WebGoat/VulnerableComponents/attack1">
103     <div id="lessonContent">
104      <form accept-charset="UNKNOWN" method="POST" name="form"
105       action="#attack/307/100">
106       <table>
```

ot Copyright 2024 Open Text.

| challenge8.js, line 26 (Cross-Site Request Forgery) | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | challenge8.js 26 HTTP . | | |
| Sink: | challenge8.js:26 FunctionPointerCall: get() | | |

```
24
25                  function average() {
26                      $.get("challenge/8/votes/average", function (average) {
27                              for (var i = 1; i <= 5; i++) {
28                                  var number = average["average"];
```

| MissingFunctionAC.html, line 99 (Cross-Site Request Forgery) | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | MissingFunctionAC.html 99 . | | |
| Sink: | MissingFunctionAC.html:99 | | |

```
97                      <div class="attack-container">
98                          <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
99                          <form class="attack-form" accept-charset="UNKNOWN"
100                               method="POST" name="form"
101                               action="/WebGoat/access-control/user-hash-fix">
```

| xxe.js, line 72 (Cross-Site Request Forgery) | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | xxe.js 72 HTTP . | | |
| Sink: | xxe.js:72 FunctionPointerCall: get() | | |

```
70
71                  function getComments(field) {
72                      $.get("xxe/comments", function (result, status) {
73                          $(field).empty();
74                          for (var i = 0; i < result.length; i++) {
```

| HtmlTampering.html, line 14 (Cross-Site Request Forgery) | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | HtmlTampering.html 14 . | | |
| Sink: | HtmlTampering.html:14 | | |

```
12                      <div class="attack-container">
13                          <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
14                          <form class="attack-form" accept-charset="UNKNOWN" id="task" name="task"
15                                method="POST"
16                                action="/WebGoat/HtmlTampering/task">
```

| SSRF.html, line 13 (Cross-Site Request Forgery) | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | SSRF.html 13 . | | |
| Sink: | SSRF.html:13 | | |

```
11                          <div class="attack-container">
12                              <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
13                              <form class="attack-form" accept-charset="UNKNOWN"
14                                    method="POST" name="form"
15                                    action="/WebGoat/SSRF/task1">
```

ot Copyright 2024 Open Text.

## path_traversal.js, line 53 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | path_traversal.js 53 HTTP . | | |
| Sink: | path_traversal.js:53 FunctionPointerCall: get() | | |

```
51
52  webgoat.customjs.profileUploadCallbackRetrieval = function () {
53      $.get("PathTraversal/profile-picture", function (result, status) {
54          document.getElementById("previewRetrieval").src = "" +
    result;
55      });
```

## CSRF.html, line 93 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | CSRF.html 93 . | | |
| Sink: | CSRF.html:93 | | |

```
91                          <div class="post-footer">
92                              <div class="input-group">
93                                  <form class="attack-form" accept-charset="UNKNOWN"
    id="csrf-review"
94                                    method="POST" name="review-form"
95                                    successCallback=""
```

## jwt.html, line 32 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | jwt.html 32 . | | |
| Sink: | jwt.html:32 | | |

```
30              </form>
31          </div>
32          <form id="encodeForm">
33              <div class="form-group">
34                  <label>Decoded</label>
```

## XXE.html, line 164 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | XXE.html 164 . | | |
| Sink: | XXE.html:164 | | |

```
162         <img th:src="@{/images/wolf-enabled.png}" class="webwolf-enabled"/>
163         <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
    hidden="true"></i></div>
164         <form class="attack-form" accept-charset="UNKNOWN"
165               method="POST" name="form"
166               prepareData="blindXXE"
```

## Challenge5.html, line 26 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | Challenge5.html 26 . | | |
| Sink: | Challenge5.html:26 | | |

```
24                              <div class="row">
25                                  <div class="col-lg-12">
26                                      <form id="login-form" class="attack-form" accept-
    charset="UNKNOWN"
27                                            method="POST" name="form"
28                                            action="/WebGoat/challenge/5" role="form">
```

| IDOR.html, line 23 (Cross-Site Request Forgery) | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | IDOR.html 23 . | | |
| Sink: | IDOR.html:23 | | |

```
21
22                  <!-- modify the action to point to the intended endpoint -->
23                  <form class="attack-form" accept-charset="UNKNOWN"
24                        method="POST" name="form"
25                        action="/WebGoat/IDOR/login">
```

| text.js, line 270 (Cross-Site Request Forgery) | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | text.js 270 HTTP . | | |
| Sink: | text.js:270 FunctionPointerCall: open() | | |

```
268         text.get = function (url, callback, errback, headers) {
269             var xhr = text.createXhr(), header;
270             xhr.open('GET', url, true);
271
272             //Allow plugins direct access to xhr headers
```

| jwt.html, line 26 (Cross-Site Request Forgery) | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | jwt.html 26 . | | |
| Sink: | jwt.html:26 | | |

```
24          <div class="form-group">
25              <label for="token">Encoded</label>
26              <form id="decodeForm">
27              <textarea class="form-control" style="font-size: 14pt; font-
family:monospace;" id="token" name="token"
28                        rows="4"
```

| IDOR.html, line 108 (Cross-Site Request Forgery) | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | IDOR.html 108 . | | |
| Sink: | IDOR.html:108 | | |

```
106
107                 <!-- modify the action to point to the intended endpoint -->
108                 <form class="attack-form" accept-charset="UNKNOWN"
109                       method="POST" name="form"
110                       action="/WebGoat/IDOR/profile/alt-path">
```

| idor.js, line 14 (Cross-Site Request Forgery) | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | idor.js 14 HTTP . | | |
| Sink: | idor.js:14 AssignmentStatement() | | |

```
12      console.warn("on view profile activated")
13      webgoat.customjs.jquery.ajax({
14          method: "GET",
15          url: "/WebGoat/IDOR/profile",
16          contentType: 'application/json; charset=UTF-8'
```

| clientSideFiltering.js, line 17 (Cross-Site Request Forgery) |
|---|

Copyright 2024 Open Text.

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | clientSideFiltering.js 17 HTTP . | | |
| Sink: | clientSideFiltering.js:17 FunctionPointerCall: get() | | |

```
15
16  function ajaxFunction(userId) {
17      $.get("clientSideFiltering/salaries?userId=" + userId, function (result, status) {
18          var html = "<table border = '1' width = '90%' align = 'center'";
19          html = html + '<tr>';
```

## SqlInjection.html, line 16 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | SqlInjection.html 16 . | | |
| Sink: | SqlInjection.html:16 | | |

```
14      <div class="attack-container">
15          <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
16          <form class="attack-form" accept-charset="UNKNOWN"
17                method="POST" name="form"
18                action="/WebGoat/SqlInjection/attack2"
```

## SqlInjectionMitigations.html, line 176 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | SqlInjectionMitigations.html 176 . | | |
| Sink: | SqlInjectionMitigations.html:176 | | |

```
174             </div>
175         </form>
176         <form class="attack-form" method="POST" name="form"
action="/WebGoat/SqlInjectionMitigations/attack12a">
177             <div class="form-group">
178                 <div class="input-group">
```

## WebWolfIntroduction.html, line 40 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | WebWolfIntroduction.html 40 . | | |
| Sink: | WebWolfIntroduction.html:40 | | |

```
38          <br/>
39          <!-- <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>-->
40          <form class="attack-form" accept-charset="UNKNOWN"
style="position:relative;top:-50px"
41                method="POST" name="secondform"
42                action="/WebGoat/WebWolf/mail/send">
```

## path_traversal.js, line 29 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | path_traversal.js 29 HTTP . | | |
| Sink: | path_traversal.js:29 FunctionPointerCall: get() | | |

```
27
28  webgoat.customjs.profileUploadCallbackFix = function () {
29      $.get("PathTraversal/profile-picture", function (result, status) {
30          document.getElementById("previewFix").src = "" + result;
31      });
```

## WebSecurityConfig.java, line 58 (Cross-Site Request Forgery)

ot Copyright 2024 Open Text.

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | WebSecurityConfig.java 58 <a href="location://src/main/java/org/owasp/webgoat/webwolf/WebSecurityConfig.java###58###31###0">disable()</a> HTTP . | | |
| **Sink:** | WebSecurityConfig.java:58 FunctionCall: disable() | | |

```
56                      .antMatchers("/files").authenticated()
57                      .anyRequest().permitAll();
58              security.and().csrf().disable().formLogin()
59                      .loginPage("/login").failureUrl("/login?error=true");
60              security.and()
```

## Cryptography.html, line 90 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | Cryptography.html 90 . | | |
| **Sink:** | Cryptography.html:90 | | |

```
88      Now suppose you have the following private key:<br/>
89      <pre><div id="privatekey" ></div></pre><br/>
90      <form class="attack-form" method="POST" name="form"
action="/WebGoat/crypto/signing/verify">
91      Then what was the modulus of the public key
92      <input name="modulus" value="" type="TEXT"/>
```

## IDOR.html, line 108 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | IDOR.html 108 . | | |
| **Sink:** | IDOR.html:108 | | |

```
106
107             <!-- modify the action to point to the intended endpoint -->
108             <form class="attack-form" accept-charset="UNKNOWN"
109                   method="POST" name="form"
110                   action="/WebGoat/IDOR/profile/alt-path">
```

## LessonTemplate.html, line 48 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | LessonTemplate.html 48 . | | |
| **Sink:** | LessonTemplate.html:48 | | |

```
46
47              <!-- modify the action to point to the intended endpoint and set other
attributes as desired -->
48              <form class="attack-form" accept-charset="UNKNOWN"
49                    method="POST" name="form"
50                    action="/WebGoat/lesson-template/sample-attack">
```

## PasswordReset.html, line 48 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | PasswordReset.html 48 . | | |
| **Sink:** | PasswordReset.html:48 | | |

```
46                          </div>
47                      </form>
48                      <form class="attack-form" accept-charset="UNKNOWN"
novalidate="novalidate"
49                          method="POST"
50                          action="/WebGoat/PasswordReset/simple-mail">
```

| WebWolfIntroduction.html, line 77 (Cross-Site Request Forgery) | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | WebWolfIntroduction.html 77 . | | |
| Sink: | WebWolfIntroduction.html:77 | | |

```
75          <br/>
76          <br/>
77          <form class="attack-form" accept-charset="UNKNOWN"
78                method="POST" name="form"
79                action="/WebGoat/WebWolf/landing/">
```

| SqlInjection.html, line 64 (Cross-Site Request Forgery) | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | SqlInjection.html 64 . | | |
| Sink: | SqlInjection.html:64 | | |

```
62      <div class="attack-container">
63          <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
64          <form class="attack-form" accept-charset="UNKNOWN"
65                method="POST" name="form"
66                action="/WebGoat/SqlInjection/attack4"
```

| ClientSideFiltering.html, line 84 (Cross-Site Request Forgery) | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | ClientSideFiltering.html 84 . | | |
| Sink: | ClientSideFiltering.html:84 | | |

```
82
83          <div class="container-fluid">
84              <form class="attack-form" accept-charset="UNKNOWN"
85                    method="POST" name="form"
86                    action="/WebGoat/clientSideFiltering/getItForFree">
```

| PasswordReset.html, line 223 (Cross-Site Request Forgery) | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | PasswordReset.html 223 . | | |
| Sink: | PasswordReset.html:223 | | |

```
221                                     Forgot your password?
222                                 </h4>
223                                 <form class="attack-form" accept-charset="UNKNOWN"
224                                       method="POST" name="form"
225
action="/WebGoat/PasswordReset/ForgotPassword/create-password-reset-link"
```

| SqlInjectionMitigations.html, line 73 (Cross-Site Request Forgery) | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | SqlInjectionMitigations.html 73 . | | |
| Sink: | SqlInjectionMitigations.html:73 | | |

```
71      <div class="attack-container">
72          <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
73          <form class="attack-form" accept-charset="UNKNOWN"
74                method="POST" name="form"
75                action="/WebGoat/SqlOnlyInputValidation/attack"
```

ot Copyright 2024 Open Text.

## SqlInjection.html, line 245 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | SqlInjection.html 245 . | | |

Sink: SqlInjection.html:245

```
243     <div class="attack-container">
244         <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
245         <form class="attack-form" accept-charset="UNKNOWN"
246               method="POST" name="form"
247               action="/WebGoat/SqlInjection/attack9"
```

## jquery.form.js, line 245 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | jquery.form.js 245 HTTP . | | |

Sink: jquery.form.js:245 FunctionPointerCall: get()

```
243         // see:  http://groups.google.com/group/jquery-
dev/browse_thread/thread/36395b7ab510dd5d
244         if (options.closeKeepAlive) {
245             $.get(options.closeKeepAlive, function() {
246                 jqxhr = fileUploadIframe(a);
247             });
```

## SqlInjection.html, line 88 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | SqlInjection.html 88 . | | |

Sink: SqlInjection.html:88

```
86      <div class="attack-container">
87          <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
88          <form class="attack-form" accept-charset="UNKNOWN"
89                method="POST" name="form"
90                action="/WebGoat/SqlInjection/attack5"
```

## SqlInjectionMitigations.html, line 96 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | SqlInjectionMitigations.html 96 . | | |

Sink: SqlInjectionMitigations.html:96

```
94      <div class="attack-container">
95          <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
96          <form class="attack-form" accept-charset="UNKNOWN"
97                method="POST" name="form"
98                action="/WebGoat/SqlOnlyInputValidationOnKeywords/attack"
```

## VulnerableComponents.html, line 100 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | VulnerableComponents.html 100 . | | |

Sink: VulnerableComponents.html:100

```
98    <div class="attack-container">
99     <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
100    <form class="attack-form" accept-charset="UNKNOWN"
101     method="POST" name="form"
```

ot Copyright 2024 Open Text.

```
102                 action="/WebGoat/VulnerableComponents/attack1">
```

## HtmlTampering.html, line 14 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | HtmlTampering.html 14 . | | |

**Sink:** HtmlTampering.html:14

```
12                  <div class="attack-container">
13                      <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
14                      <form class="attack-form" accept-charset="UNKNOWN" id="task" name="task"
15                            method="POST"
16                            action="/WebGoat/HtmlTampering/task">
```

## credentials.js, line 3 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | credentials.js 3 HTTP . | | |

**Sink:** credentials.js:3 FunctionPointerCall: open()

```
1   function submit_secret_credentials() {
2       var xhttp = new XMLHttpRequest();
3       xhttp['open']('POST', '#attack/307/100', true);
4    //sending the request is obfuscated, to descourage js reading
5    var
_0xb7f9=["\x43\x61\x70\x74\x61\x69\x6E\x4A\x61\x63\x6B","\x42\x6C\x61\x63\x6B\x50\x65\
x61\x72\x6C","\x73\x74\x72\x69\x6E\x67\x69\x66\x79","\x73\x65\x6E\x64"];xhttp[_0xb7f9[
3]](JSON[_0xb7f9[2]]({username:_0xb7f9[0],password:_0xb7f9[1]}))
```

## idor.js, line 14 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | idor.js 14 HTTP . | | |

**Sink:** idor.js:14 AssignmentStatement()

```
12      console.warn("on view profile activated")
13      webgoat.customjs.jquery.ajax({
14          method: "GET",
15          url: "/WebGoat/IDOR/profile",
16          contentType: 'application/json; charset=UTF-8'
```

## SqlInjection.html, line 40 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | SqlInjection.html 40 . | | |

**Sink:** SqlInjection.html:40

```
38                  <div class="attack-container">
39                      <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
40                      <form class="attack-form" accept-charset="UNKNOWN"
41                            method="POST" name="form"
42                            action="/WebGoat/SqlInjection/attack3"
```

## path_traversal.js, line 46 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | path_traversal.js 46 HTTP . | | |

**Sink:** path_traversal.js:46 FunctionPointerCall: get()

```
44
45  webgoat.customjs.profileUploadCallbackRemoveUserInput = function () {
46      $.get("PathTraversal/profile-picture", function (result, status) {
```

ot Copyright 2024 Open Text.

```
47                      document.getElementById("previewRemoveUserInput").src =
"" + result;
48                  });
```

## CrossSiteScripting.html, line 149 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | CrossSiteScripting.html 149 . | | |

**Sink:** CrossSiteScripting.html:149

```
147                  <div class="attack-container">
148                    <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
149                    <form class="attack-form" accept-charset="UNKNOWN"
150                        method="POST" name="DOMFollowUp"
151                        action="/WebGoat/CrossSiteScripting/dom-follow-up">
```

## AuthBypass.html, line 43 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | AuthBypass.html 43 . | | |

**Sink:** AuthBypass.html:43

```
41                          </form>
42
43                          <form class="attack-form" accept-charset="UNKNOWN" id="change-password-
form"
44                                  method="POST" name="form"
45                                  successCallback="onBypassResponse"
```

## BypassRestrictions.html, line 19 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | BypassRestrictions.html 19 . | | |

**Sink:** BypassRestrictions.html:19

```
17                  <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
18                  <div class="container-fluid">
19                      <form class="attack-form" accept-charset="UNKNOWN"
name="fieldRestrictions"
20                              method="POST"
21                              action="/WebGoat/BypassRestrictions/FieldRestrictions">
```

## JWT.html, line 121 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | JWT.html 121 . | | |

**Sink:** JWT.html:121

```
119                 <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
120                 <div class="container-fluid">
121                     <form id="quiz-form" class="attack-form" accept-charset="UNKNOWN"
122                             method="POST" name="form"
123                             action="/WebGoat/JWT/quiz"
```

## IDOR.html, line 135 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | IDOR.html 135 . | | |

**Sink:** IDOR.html:135

```
133
```

ot Copyright 2024 Open Text.

```
134                     <!-- modify the action to point to the intended endpoint -->
135                     <form class="attack-form" accept-charset="UNKNOWN" id="view-other"
136                           method="GET" name="view-other-profile"
137                           action="/WebGoat/IDOR/profile/{userId}">
```

## MissingFunctionAC.html, line 77 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | MissingFunctionAC.html 77 . | | |

**Sink:** MissingFunctionAC.html:77

```
75          <div class="attack-container">
76              <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
77          <form class="attack-form" accept-charset="UNKNOWN"
78                method="POST" name="form"
79                action="/WebGoat/access-control/user-hash">
```

## PasswordReset.html, line 144 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | PasswordReset.html 144 . | | |

**Sink:** PasswordReset.html:144

```
142         <div class="attack-container">
143             <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
144         <form class="attack-form" accept-charset="UNKNOWN"
145               method="POST" name="form"
146               action="/WebGoat/PasswordReset/SecurityQuestions">
```

## SqlInjectionMitigations.html, line 45 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | SqlInjectionMitigations.html 45 . | | |

**Sink:** SqlInjectionMitigations.html:45

```
43          <div class="adoc-content"
th:replace="doc:lessons/sqlinjection/documentation/SqlInjection_jdbc_newcode.adoc"></d
iv>
44          <div class="attack-container" style="border: none !important; height: 100%; min-
height: 300px;">
45              <form id="codesubmit" style="height: 100%; min-height: 300px;" class="attack-
form" accept-charset="UNKNOWN" method="POST" name="form"
action="/WebGoat/SqlInjectionMitigations/attack10b">
46                  <div>
47                      <div id="editor" style="position: absolute; top: 0; right: 0; bottom:
0; left: 0; height: 300px;" name="editor"></div>
```

## quiz.js, line 15 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | quiz.js 15 HTTP . | | |

**Sink:** quiz.js:15 FunctionPointerCall: open()

```
13          var client = new XMLHttpRequest();
14          var quiz_id = document.getElementById("quiz_id").getAttribute("data-quiz_id");
15          client.open('GET', '/WebGoat/lesson_js/questions_' + quiz_id + '.json');
16          client.onreadystatechange = function() {
17              if (this.readyState == 4 && this.status == 200) {
```

## ClientSideFiltering.html, line 84 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |

ot Copyright 2024 Open Text.

| | |
|---|---|
| **Abstract:** | ClientSideFiltering.html 84 . |
| **Sink:** | ClientSideFiltering.html:84 |

```
82
83              <div class="container-fluid">
84                  <form class="attack-form" accept-charset="UNKNOWN"
85                        method="POST" name="form"
86                        action="/WebGoat/clientSideFiltering/getItForFree">
```

## SqlInjectionAdvanced.html, line 169 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | SqlInjectionAdvanced.html 169 . | | |
| **Sink:** | SqlInjectionAdvanced.html:169 | | |

```
167             <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
168             <div class="container-fluid">
169                 <form id="quiz-form" class="attack-form" accept-charset="UNKNOWN"
170                       method="POST" name="form"
171                       action="/WebGoat/SqlInjectionAdvanced/quiz"
```

## SqlInjection.html, line 217 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | SqlInjection.html 217 . | | |
| **Sink:** | SqlInjection.html:217 | | |

```
215         <div class="attack-container">
216             <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
217             <form class="attack-form" accept-charset="UNKNOWN"
218                   method="POST" name="form"
219                   action="/WebGoat/SqlInjection/attack8"
```

## SqlInjectionMitigations.html, line 45 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | SqlInjectionMitigations.html 45 . | | |
| **Sink:** | SqlInjectionMitigations.html:45 | | |

```
43          <div class="adoc-content"
th:replace="doc:lessons/sqlinjection/documentation/SqlInjection_jdbc_newcode.adoc"></d
iv>
44          <div class="attack-container" style="border: none !important; height: 100%; min-
height: 300px;">
45              <form id="codesubmit" style="height: 100%; min-height: 300px;" class="attack-
form" accept-charset="UNKNOWN" method="POST" name="form"
action="/WebGoat/SqlInjectionMitigations/attack10b">
46                  <div>
47                      <div id="editor" style="position: absolute; top: 0; right: 0; bottom:
0; left: 0; height: 300px;" name="editor"></div>
```

## PathTraversal.html, line 70 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | PathTraversal.html 70 . | | |
| **Sink:** | PathTraversal.html:70 | | |

```
68              <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
69              <div class="upload-container">
70                  <form class="attack-form" accept-charset="UNKNOWN"
71                        method="POST" name="form"
72                        onsubmit='return false'
```

ot Copyright 2024 Open Text.

| WebWolfIntroduction.html, line 77 (Cross-Site Request Forgery) | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | WebWolfIntroduction.html 77 . | | |
| Sink: | WebWolfIntroduction.html:77 | | |

```
75              <br/>
76              <br/>
77              <form class="attack-form" accept-charset="UNKNOWN"
78                    method="POST" name="form"
79                    action="/WebGoat/WebWolf/landing/">
```

| Challenge1.html, line 40 (Cross-Site Request Forgery) | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | Challenge1.html 40 . | | |
| Sink: | Challenge1.html:40 | | |

```
38              </div>
39
40              <form class="attack-form" method="POST" name="form"
action="/WebGoat/challenge/flag">
41                  <div class="form-group">
42                      <div class="input-group">
```

| VulnerableComponents.html, line 104 (Cross-Site Request Forgery) | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | VulnerableComponents.html 104 . | | |
| Sink: | VulnerableComponents.html:104 | | |

```
102         action="/WebGoat/VulnerableComponents/attack1">
103         <div id="lessonContent">
104          <form accept-charset="UNKNOWN" method="POST" name="form"
105           action="#attack/307/100">
106           <table>
```

| CSRF.html, line 35 (Cross-Site Request Forgery) | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | CSRF.html 35 . | | |
| Sink: | CSRF.html:35 | | |

```
33              </div>
34              <br/>
35              <form class="attack-form" accept-charset="UNKNOWN" id="confirm-flag-1"
36                    method="POST" name="form2"
37                    successCallback=""
```

| SqlInjectionAdvanced.html, line 169 (Cross-Site Request Forgery) | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | SqlInjectionAdvanced.html 169 . | | |
| Sink: | SqlInjectionAdvanced.html:169 | | |

```
167             <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
168             <div class="container-fluid">
169                 <form id="quiz-form" class="attack-form" accept-charset="UNKNOWN"
170                       method="POST" name="form"
171                       action="/WebGoat/SqlInjectionAdvanced/quiz"
```

| PasswordReset.html, line 104 (Cross-Site Request Forgery) |
|---|

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | PasswordReset.html 104 . | | |

Sink: PasswordReset.html:104

```
102     <div class="attack-container">
103         <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
104         <form class="attack-form" accept-charset="UNKNOWN"
105               method="POST"
106               action="/WebGoat/PasswordReset/questions">
```

## SqlInjectionMitigations.html, line 73 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | SqlInjectionMitigations.html 73 . | | |

Sink: SqlInjectionMitigations.html:73

```
71      <div class="attack-container">
72          <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
73          <form class="attack-form" accept-charset="UNKNOWN"
74                method="POST" name="form"
75                action="/WebGoat/SqlOnlyInputValidation/attack"
```

## CrossSiteScriptingMitigation.html, line 24 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | CrossSiteScriptingMitigation.html 24 . | | |

Sink: CrossSiteScriptingMitigation.html:24

```
22      <div class="adoc-content"
th:replace="doc:lessons/xss/documentation/CrossSiteScripting_content8b.adoc"></div>
23      <div class="attack-container" style="height: 100%; border: none !important;min-height:
450px;">
24        <form id="codesubmit" style="height: 100%; min-height: 350px;" class="attack-form"
accept-charset="UNKNOWN" method="POST" name="form"
action="/WebGoat/CrossSiteScripting/attack3">
25        <div>
26          <div id="editor" style="position: absolute; top: 0; right: 0; bottom: 0; left: 0;
height: 350px;" name="editor"></div>
```

## CIA.html, line 30 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | CIA.html 30 . | | |

Sink: CIA.html:30

```
28          <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
29          <div class="container-fluid">
30              <form id="quiz-form" class="attack-form" accept-charset="UNKNOWN"
31                    method="POST" name="form"
32                    action="cia/quiz" role="form">
```

## XXE.html, line 25 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | XXE.html 25 . | | |

Sink: XXE.html:25

```
23          <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
24
25          <form class="attack-form" accept-charset="UNKNOWN"
```

ot Copyright 2024 Open Text.

```
26                          method="POST" name="form"
27                          prepareData="simpleXXE"
```

## hijackform.html, line 3 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | hijackform.html 3 . | | |

**Sink:** hijackform.html:3

```
1  <div class="row">
2   <div class="col-md-4">
3    <form class="attack-form" accept-charset="UNKNOWN" method="POST"
4     action="/WebGoat/HijackSession/login">
5     <div style="padding: 20px;" id="password-login">
```

## PasswordReset.html, line 113 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | PasswordReset.html 113 . | | |

**Sink:** PasswordReset.html:113

```
111                             <a class="float-right btn btn-outline-primary">Login</a>
112                             <h4 class="card-title mb-4 mt-1">WebGoat Password
Recovery</h4>
113                             <form>
114                                 <div class="form-group">
115                                     <label>Your username</label>
```

## Cryptography.html, line 113 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | Cryptography.html 113 . | | |

**Sink:** Cryptography.html:113

```
111   <div class="attack-container">
112    <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
113    <form class="attack-form" method="POST" name="form"
action="/WebGoat/crypto/secure/defaults">
114    What is the unencrypted message<br/>
115    <input name="secretText" value="" type="TEXT"/><br/>
```

## goatApp.js, line 21 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | goatApp.js 21 HTTP . | | |

**Sink:** goatApp.js:21 FunctionPointerCall: getJSON()

```
19              initApp: function () {
20                  var locale = localStorage.getItem('locale') || 'en';
21                  $.getJSON('service/labels.mvc', function(data) {
22                      window.polyglot = new Polyglot({phrases: data});//i18n polyglot
labels
23                      asyncErrorHandler.init();
```

## Cryptography.html, line 65 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | Cryptography.html 65 . | | |

**Sink:** Cryptography.html:65

```
63   <div class="attack-container">
64    <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
```

ot Copyright 2024 Open Text.

```
65                    <form class="attack-form" method="POST" name="form" action="/WebGoat/crypto/hashing">
66                    Which password belongs to this hash: <div id="md5token" ></div>
67                    <input name="answer_pwd1" value="" type="TEXT"/><br/>
```

## csrf-review.js, line 35 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | csrf-review.js 35 HTTP . | | |

Sink: csrf-review.js:35 FunctionPointerCall: get()

```
33                    function getChallenges() {
34                        $("#list").empty();
35                        $.get('csrf/review', function (result, status) {
36                            for (var i = 0; i < result.length; i++) {
37                                var comment = html.replace('USER', result[i].user);
```

## PasswordReset.html, line 187 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | PasswordReset.html 187 . | | |

Sink: PasswordReset.html:187

```
185                                   </h4>
186                                   <div style="padding: 20px;" id="password-login">
187                                       <form id="login-form" class="attack-form" accept-
charset="UNKNOWN"
188                                         method="POST" name="form"
189                                         action="/WebGoat/PasswordReset/reset/login"
```

## IDOR.html, line 58 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | IDOR.html 58 . | | |

Sink: IDOR.html:58

```
56
57                        <!-- modify the action to point to the intended endpoint -->
58                        <form class="attack-form" accept-charset="UNKNOWN"
59                              method="GET" name="form"
60                              action="/WebGoat/IDOR/profile">
```

## credentials.js, line 3 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | credentials.js 3 HTTP . | | |

Sink: credentials.js:3 FunctionPointerCall: open()

```
1                     function submit_secret_credentials() {
2                         var xhttp = new XMLHttpRequest();
3                         xhttp['open']('POST', '#attack/307/100', true);
4                      //sending the request is obfuscated, to descourage js reading
5                      var
_0xb7f9=["\x43\x61\x70\x74\x61\x69\x6E\x4A\x61\x63\x6B","\x42\x6C\x61\x63\x6B\x50\x65\
x61\x72\x6C","\x73\x74\x72\x69\x6E\x67\x69\x66\x79","\x73\x65\x6E\x64"];xhttp[_0xb7f9[
3]](JSON[_0xb7f9[2]]({username:_0xb7f9[0],password:_0xb7f9[1]}))
```

## Cryptography.html, line 65 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | Cryptography.html 65 . | | |

Sink: Cryptography.html:65

```
63                    <div class="attack-container">
```

```
64                    <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
65                    <form class="attack-form" method="POST" name="form" action="/WebGoat/crypto/hashing">
66                    Which password belongs to this hash: <div id="md5token" ></div>
67                    <input name="answer_pwd1" value="" type="TEXT"/><br/>
```

## IDOR.html, line 159 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | IDOR.html 159 . | | |

**Sink:** IDOR.html:159

```
157
158                   <!-- modify the action to point to the intended endpoint -->
159                   <form class="attack-form" accept-charset="UNKNOWN" id="edit-other"
160                         method="GET" name="edit-other-profile"
161                         action="/WebGoat/IDOR/profile/{userId}">
```

## IDOR.html, line 159 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | IDOR.html 159 . | | |

**Sink:** IDOR.html:159

```
157
158                   <!-- modify the action to point to the intended endpoint -->
159                   <form class="attack-form" accept-charset="UNKNOWN" id="edit-other"
160                         method="GET" name="edit-other-profile"
161                         action="/WebGoat/IDOR/profile/{userId}">
```

## MissingFunctionAC.html, line 54 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | MissingFunctionAC.html 54 . | | |

**Sink:** MissingFunctionAC.html:54

```
52
53                    <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
54                    <form class="attack-form" accept-charset="UNKNOWN"
55                          method="POST" name="form"
56                          action="/WebGoat/access-control/hidden-menu">
```

## PathTraversal.html, line 125 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | PathTraversal.html 125 . | | |

**Sink:** PathTraversal.html:125

```
123                   <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
124                   <div class="upload-container">
125                       <form class="attack-form" accept-charset="UNKNOWN"
126                             method="POST" name="form"
127                             onsubmit='return false'
```

## CSRF.html, line 146 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | CSRF.html 146 . | | |

**Sink:** CSRF.html:146

```
144                             <div class="col-md-8">
```

ot Copyright 2024 Open Text.

```
145                                     <div class="well well-sm">
146                                         <form class="attack-form" accept-charset="UNKNOWN"
id="csrf-feedback"
147                                           method="POST"
148                                           prepareData="feedback"
```

## challenge8.js, line 7 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | challenge8.js 7 HTTP . | | |

**Sink:** challenge8.js:7 FunctionPointerCall: get()

```
5
6   function loadVotes() {
7       $.get("challenge/8/votes/", function (votes) {
8               var totalVotes = 0;
9               for (var i = 1; i <= 5; i++) {
```

## jwt-voting.js, line 43 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | jwt-voting.js 43 HTTP . | | |

**Sink:** jwt-voting.js:43 FunctionPointerCall: get()

```
41  function getVotings() {
42      $("#votesList").empty();
43      $.get("JWT/votings", function (result, status) {
44          for (var i = 0; i < result.length; i++) {
45              var voteTemplate = html.replace('IMAGE_SMALL', result[i].imageSmall);
```

## CrossSiteScripting.html, line 169 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | CrossSiteScripting.html 169 . | | |

**Sink:** CrossSiteScripting.html:169

```
167   <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
168   <div class="container-fluid">
169    <form id="quiz-form" class="attack-form" accept-charset="UNKNOWN"
170       method="POST" name="form"
171       action="/WebGoat/CrossSiteScripting/quiz" role="form">
```

## HttpBasics.html, line 52 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | HttpBasics.html 52 . | | |

**Sink:** HttpBasics.html:52

```
50                  <!-- you can write your own custom forms, but standard form submission
will take you to your endpoint and outside of the WebGoat framework -->
51                  <!-- of course, you can write your own ajax submission /handling in
your own javascript if you like -->
52      <form class="attack-form" accept-charset="UNKNOWN"
53       method="POST" name="form"
54       action="/WebGoat/HttpBasics/attack2">
```

## SqlInjectionAdvanced.html, line 116 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | SqlInjectionAdvanced.html 116 . | | |

**Sink:** SqlInjectionAdvanced.html:116

```
114                                     </div>
```

Copyright 2024 Open Text.

```
115                                                 </form>
116                                                 <form id="register-form" class="attack-form"
accept-charset="UNKNOWN"
117                                                       method="PUT" name="form"
118
action="/WebGoat/SqlInjectionAdvanced/challenge"
```

## registration.html, line 26 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | registration.html 26 . | | |
| **Sink:** | registration.html:26 | | |

```
24                  <fieldset>
25                      <legend th:text="#{register.title}">Please Sign Up</legend>
26                      <form class="form-horizontal" action="#" th:action="@{/register.mvc}"
th:object="${userForm}"
27                            method='POST'>
```

## path_traversal.js, line 13 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | path_traversal.js 13 HTTP . | | |
| **Sink:** | path_traversal.js:13 FunctionPointerCall: get() | | |

```
11
12  webgoat.customjs.profileUploadCallback = function () {
13      $.get("PathTraversal/profile-picture", function (result, status) {
14          document.getElementById("preview").src = "" + result;
15      });
```

## path_traversal.js, line 75 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | path_traversal.js 75 HTTP . | | |
| **Sink:** | path_traversal.js:75 FunctionPointerCall: get() | | |

```
73
74  webgoat.customjs.profileZipSlipRetrieval = function () {
75      $.get("PathTraversal/zip-slip", function (result, status) {
76          document.getElementById("previewZipSlip").src = "" +
result;
77      });
```

## JWT.html, line 308 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | JWT.html 308 . | | |
| **Sink:** | JWT.html:308 | | |

```
306     <div class="attack-container">
307         <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
308         <form class="attack-form" accept-charset="UNKNOWN"
309               method="POST"
310
action="/WebGoat/JWT/final/delete?token=eyJ0eXAiOiJKV1QiLCJraWQiOiJ3ZWJnb2F0X2tleSIsIm
FsZyI6IkhTMjU2In0.eyJpc3MiOiJXZWJHb2F0IFRva2VuIEJ1aWxkZXIiLCJpYXQiOjE1MjQyMTA5MDQsImV4
cCI6MTYxODkwNTMwNCwiYXVkIjoid2ViZ29hdC5vcmciLCJzdWIiOiJqZXJyeUB3ZWJnb2F0LmNvbSIsInVzZX
JuYW1lIjoiSmVycnkiLCJFbWFpbCI6ImplcnJ5QHdlYmdvYXQuY29tIiwiUm9sZSI6WyJDYXQiXX0.CgZ27Dzg
VW8gzc0n6izOU638uUCi6UhiOJKYzoEZGE8">
```

## HttpBasics.html, line 22 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |

| | |
|---|---|
| **Abstract:** | HttpBasics.html 22 . |
| **Sink:** | HttpBasics.html:22 |

```
20                  <!-- you can write your own custom forms, but standard form submission
will take you to your endpoint and outside of the WebGoat framework -->
21                  <!-- of course, you can write your own ajax submission /handling in your
own javascript if you like -->
22                <form class="attack-form" accept-charset="UNKNOWN"
23                 method="POST" name="form"
24                 action="/WebGoat/HttpBasics/attack1">
```

**stored-xss.js, line 35 (Cross-Site Request Forgery)**

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | stored-xss.js 35 HTTP . | | |
| **Sink:** | stored-xss.js:35 FunctionPointerCall: get() | | |

```
33              function getChallenges() {
34                  $("#list").empty();
35                  $.get('CrossSiteScripting/stored-xss', function (result, status) {
36                      for (var i = 0; i < result.length; i++) {
37                          var comment = html.replace('USER', result[i].user);
```

**PasswordReset.html, line 187 (Cross-Site Request Forgery)**

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | PasswordReset.html 187 . | | |
| **Sink:** | PasswordReset.html:187 | | |

```
185                                 </h4>
186                                 <div style="padding: 20px;" id="password-login">
187                                     <form id="login-form" class="attack-form" accept-
charset="UNKNOWN"
188                                       method="POST" name="form"
189                                       action="/WebGoat/PasswordReset/reset/login"
```

**assignment13.js, line 43 (Cross-Site Request Forgery)**

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | assignment13.js 43 HTTP . | | |
| **Sink:** | assignment13.js:43 FunctionPointerCall: get() | | |

```
41
42              function getServers(column) {
43                  $.get("SqlInjectionMitigations/servers?column=" + column, function (result,
status) {
44                      $("#servers").empty();
45                      for (var i = 0; i < result.length; i++) {
```

**SqlInjectionAdvanced.html, line 34 (Cross-Site Request Forgery)**

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | SqlInjectionAdvanced.html 34 . | | |
| **Sink:** | SqlInjectionAdvanced.html:34 | | |

```
32                  </table>
33                  </form>
34                  <form class="attack-form" accept-charset="UNKNOWN"
35                        method="POST" name="form"
36                        action="/WebGoat/SqlInjectionAdvanced/attack6b">
```

**PathTraversal.html, line 16 (Cross-Site Request Forgery)**

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |

ot Copyright 2024 Open Text.

| | |
|---|---|
| **Abstract:** | PathTraversal.html 16 . |
| **Sink:** | PathTraversal.html:16 |

```
14          <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
15          <div class="upload-container">
16              <form class="attack-form" accept-charset="UNKNOWN"
17                    method="POST" name="form"
18                    onsubmit='return false'
```

## registration.html, line 15 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | registration.html 15 . | | |
| **Sink:** | registration.html:15 | | |

```
13      <fieldset>
14          <legend th:text="#{register.title}">Please Sign Up</legend>
15          <form class="form-horizontal" action="#" th:action="@{/register.mvc}"
th:object="${userForm}"
16                method='POST'>
```

## JWT.html, line 154 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | JWT.html 154 . | | |
| **Sink:** | JWT.html:154 | | |

```
152     <div class="attack-container">
153         <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
154         <form class="attack-form" method="POST" name="form"
action="/WebGoat/JWT/secret">
155             <div class="form-group">
156                 <div class="input-group">
```

## SqlInjection.html, line 144 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | SqlInjection.html 144 . | | |
| **Sink:** | SqlInjection.html:144 | | |

```
142     <div class="attack-container">
143         <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
144         <form class="attack-form" accept-charset="UNKNOWN"
145               method="POST" name="form"
146               action="/WebGoat/SqlInjection/assignment5a">
```

## PasswordReset.html, line 113 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | PasswordReset.html 113 . | | |
| **Sink:** | PasswordReset.html:113 | | |

```
111                             <a class="float-right btn btn-outline-primary">Login</a>
112                             <h4 class="card-title mb-4 mt-1">WebGoat Password
Recovery</h4>
113                             <form>
114                                 <div class="form-group">
115                                     <label>Your username</label>
```

## jwt.html, line 26 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |

| Kingdom: | Encapsulation |
|---|---|
| Abstract: | jwt.html 26 . |
| Sink: | jwt.html:26 |

```
24              <div class="form-group">
25                  <label for="token">Encoded</label>
26                  <form id="decodeForm">
27                  <textarea class="form-control" style="font-size: 14pt; font-
family:monospace;" id="token" name="token"
28                            rows="4"
```

## SqlInjectionAdvanced.html, line 116 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | SqlInjectionAdvanced.html 116 . | | |
| Sink: | SqlInjectionAdvanced.html:116 | | |

```
114                                                 </div>
115                                             </form>
116                                             <form id="register-form" class="attack-form"
accept-charset="UNKNOWN"
117                                                 method="PUT" name="form"
118
action="/WebGoat/SqlInjectionAdvanced/challenge"
```

## WebSecurityConfig.java, line 74 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | WebSecurityConfig.java 74 <a href="location://src/main/java/org/owasp/webgoat/container/WebSecurityConfig.java###74###31###0">disable()</a> HTTP . | | |
| Sink: | WebSecurityConfig.java:74 FunctionCall: disable() | | |

```
72              security.and()
73                      .logout().deleteCookies("JSESSIONID").invalidateHttpSession(true);
74              security.and().csrf().disable();
75
76              http.headers().cacheControl().disable();
```

## InsecureLogin.html, line 18 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | InsecureLogin.html 18 . | | |
| Sink: | InsecureLogin.html:18 | | |

```
16              <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
17              <script th:src="@{/lesson_js/credentials.js}"></script>
18              <form class="attack-form" accept-charset="UNKNOWN" name="task"
19                    method="POST"
20                    action="/WebGoat/InsecureLogin/task">
```

## files.html, line 37 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | files.html 37 . | | |
| Sink: | files.html:37 | | |

```
35
36              <!-- Standard Form -->
37              <form th:action="@{/fileupload}" method="post" enctype="multipart/form-
data">
38                  <div class="form-inline">
39                      <div class="form-group">
```

ot Copyright 2024 Open Text.

## SqlInjection.html, line 64 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | SqlInjection.html 64 . | | |

Sink: SqlInjection.html:64

```
62      <div class="attack-container">
63          <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
64          <form class="attack-form" accept-charset="UNKNOWN"
65                method="POST" name="form"
66                action="/WebGoat/SqlInjection/attack4"
```

## login.html, line 31 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | login.html 31 . | | |

Sink: login.html:31

```
29              </div>
30              <br/><br/>
31              <form th:action="@{/login}" method='POST' style="width: 200px;">
32                  <div class="form-group">
33                      <label for="exampleInputEmail1"
th:text="#{username}">Username</label>
```

## CrossSiteScripting.html, line 47 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | CrossSiteScripting.html 47 . | | |

Sink: CrossSiteScripting.html:47

```
45   <div class="attack-container">
46    <div id="lessonContent">
47     <form class="attack-form" accept-charset="UNKNOWN"
48        method="GET" name="xss-5a"
49        action="/WebGoat/CrossSiteScripting/attack5a">
```

## SSRF.html, line 13 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | SSRF.html 13 . | | |

Sink: SSRF.html:13

```
11          <div class="attack-container">
12              <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
13              <form class="attack-form" accept-charset="UNKNOWN"
14                    method="POST" name="form"
15                    action="/WebGoat/SSRF/task1">
```

## Challenge7.html, line 60 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | Challenge7.html 60 . | | |

Sink: Challenge7.html:60

```
58          </div>
59          <br/>
60          <form class="attack-form" method="POST" name="form"
action="/WebGoat/challenge/flag">
61              <div class="form-group">
62                  <div class="input-group">
```

ot Copyright 2024 Open Text.

## CrossSiteScripting.html, line 149 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | CrossSiteScripting.html 149 . | | |

Sink: CrossSiteScripting.html:149

```
147  <div class="attack-container">
148  <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
     hidden="true"></i></div>
149  <form class="attack-form" accept-charset="UNKNOWN"
150  method="POST" name="DOMFollowUp"
151  action="/WebGoat/CrossSiteScripting/dom-follow-up">
```

## jwt-voting.js, line 43 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | jwt-voting.js 43 HTTP . | | |

Sink: jwt-voting.js:43 FunctionPointerCall: get()

```
41  function getVotings() {
42  $("#votesList").empty();
43  $.get("JWT/votings", function (result, status) {
44  for (var i = 0; i < result.length; i++) {
45  var voteTemplate = html.replace('IMAGE_SMALL', result[i].imageSmall);
```

## GoatUtils.js, line 56 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | GoatUtils.js 56 HTTP . | | |

Sink: GoatUtils.js:56 FunctionPointerCall: get()

```
54
55  showLessonCookiesAndParams: function() {
56  $.get(goatConstants.cookieService, {}, function(reply) {
57  $("#lesson_cookies").html(reply);
58  }, "html");
```

## jquery.form.js, line 245 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | jquery.form.js 245 HTTP . | | |

Sink: jquery.form.js:245 FunctionPointerCall: get()

```
243  // see: http://groups.google.com/group/jquery-
     dev/browse_thread/thread/36395b7ab510dd5d
244  if (options.closeKeepAlive) {
245  $.get(options.closeKeepAlive, function() {
246  jqxhr = fileUploadIframe(a);
247  });
```

## Challenge6.html, line 30 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | Challenge6.html 30 . | | |

Sink: Challenge6.html:30

```
28  <div class="row">
29  <div class="col-lg-12">
30  <form id="login-form" class="attack-form" accept-
     charset="UNKNOWN"
31  method="POST" name="form"
32  action="/WebGoat/challenge/6" role="form">
```

ot Copyright 2024 Open Text.

## challenge8.js, line 7 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | challenge8.js 7 HTTP . | | |

Sink: challenge8.js:7 FunctionPointerCall: get()

```
5
6   function loadVotes() {
7       $.get("challenge/8/votes/", function (votes) {
8           var totalVotes = 0;
9           for (var i = 1; i <= 5; i++) {
```

## Challenge5.html, line 69 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | Challenge5.html 69 . | | |

Sink: Challenge5.html:69

```
67          </div>
68          <br/>
69          <form class="attack-form" method="POST" name="form"
action="/WebGoat/challenge/flag">
70              <div class="form-group">
71                  <div class="input-group">
```

## AuthBypass.html, line 23 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | AuthBypass.html 23 . | | |

Sink: AuthBypass.html:23

```
21              <!-- modify the action to point to the intended endpoint and set other
attributes as desired -->
22              <script th:src="@{/lesson_js/bypass.js}" />
23              <form class="attack-form" accept-charset="UNKNOWN" id="verify-account-
form"
24                    method="POST" name="form"
25                    successCallback="onBypassResponse"
```

## JWT.html, line 121 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | JWT.html 121 . | | |

Sink: JWT.html:121

```
119         <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
120         <div class="container-fluid">
121             <form id="quiz-form" class="attack-form" accept-charset="UNKNOWN"
122                   method="POST" name="form"
123                   action="/WebGoat/JWT/quiz"
```

## CrossSiteScriptingMitigation.html, line 24 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | CrossSiteScriptingMitigation.html 24 . | | |

Sink: CrossSiteScriptingMitigation.html:24

```
22   <div class="adoc-content"
th:replace="doc:lessons/xss/documentation/CrossSiteScripting_content8b.adoc"></div>
23   <div class="attack-container" style="height: 100%; border: none !important;min-height:
450px;">
```

```
24                    <form id="codesubmit" style="height: 100%; min-height: 350px;" class="attack-form"
                    accept-charset="UNKNOWN" method="POST" name="form"
                    action="/WebGoat/CrossSiteScripting/attack3">
25                      <div>
26                       <div id="editor" style="position: absolute; top: 0; right: 0; bottom: 0; left: 0;
                    height: 350px;" name="editor"></div>
```

## ClientSideFiltering.html, line 16 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | ClientSideFiltering.html 16 . | | |

**Sink:** ClientSideFiltering.html:16

```
14                          <input type="hidden" id="user_id" value="102"/>
15                          <!-- using attack-form class on your form, will allow your request to be
                    ajaxified and stay within the display framework for webgoat -->
16                          <form class="attack-form" accept-charset="UNKNOWN" method="POST" name="form"
17                                action="/WebGoat/clientSideFiltering/attack1">
18                             <link rel="stylesheet" type="text/css"
```

## SqlInjectionAdvanced.html, line 34 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | SqlInjectionAdvanced.html 34 . | | |

**Sink:** SqlInjectionAdvanced.html:34

```
32                          </table>
33                      </form>
34                      <form class="attack-form" accept-charset="UNKNOWN"
35                            method="POST" name="form"
36                            action="/WebGoat/SqlInjectionAdvanced/attack6b">
```

## SqlInjection.html, line 274 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | SqlInjection.html 274 . | | |

**Sink:** SqlInjection.html:274

```
272                 <div class="attack-container">
273                     <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
                    hidden="true"></i></div>
274                     <form class="attack-form" accept-charset="UNKNOWN"
275                           method="POST" name="form"
276                           action="/WebGoat/SqlInjection/attack10"
```

## CrossSiteScripting.html, line 169 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | CrossSiteScripting.html 169 . | | |

**Sink:** CrossSiteScripting.html:169

```
167                   <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
                    hidden="true"></i></div>
168                   <div class="container-fluid">
169                    <form id="quiz-form" class="attack-form" accept-charset="UNKNOWN"
170                       method="POST" name="form"
171                       action="/WebGoat/CrossSiteScripting/quiz" role="form">
```

## PathTraversal.html, line 16 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | PathTraversal.html 16 . | | |

**Sink:** PathTraversal.html:16

ot Copyright 2024 Open Text.

```
14                      <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
15                      <div class="upload-container">
16                          <form class="attack-form" accept-charset="UNKNOWN"
17                                method="POST" name="form"
18                                onsubmit='return false'
```

## PasswordReset.html, line 176 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | PasswordReset.html 176 . | | |

Sink: PasswordReset.html:176

```
174                     <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
175
176                     <form class="attack-form" accept-charset="UNKNOWN"
177                           method="POST"
178                           action="/WebGoat/PasswordReset/reset/login">
```

## JWT.html, line 49 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | JWT.html 49 . | | |

Sink: JWT.html:49

```
47                      <div class="attack-output"></div>
48                      <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
49                      <form class="attack-form" accept-charset="UNKNOWN"
50                            method="POST"
51                            successCallback="jwtSigningCallback"
```

## spoofcookieform.html, line 3 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | spoofcookieform.html 3 . | | |

Sink: spoofcookieform.html:3

```
1  <div class="row">
2   <div class="col-md-4">
3    <form class="attack-form" accept-charset="UNKNOWN" method="POST"
4     action="/WebGoat/SpoofCookie/login">
5     <div style="padding: 20px;" id="password-login">
```

## MissingFunctionAC.html, line 77 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | MissingFunctionAC.html 77 . | | |

Sink: MissingFunctionAC.html:77

```
75                  <div class="attack-container">
76                      <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
77                      <form class="attack-form" accept-charset="UNKNOWN"
78                            method="POST" name="form"
79                            action="/WebGoat/access-control/user-hash">
```

## spoofcookieform.html, line 3 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | spoofcookieform.html 3 . | | |

Sink: spoofcookieform.html:3

```
1                   <div class="row">
2                    <div class="col-md-4">
3                     <form class="attack-form" accept-charset="UNKNOWN" method="POST"
4                      action="/WebGoat/SpoofCookie/login">
5                      <div style="padding: 20px;" id="password-login">
```

## SqlInjection.html, line 189 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | SqlInjection.html 189 . | | |

**Sink:** SqlInjection.html:189

```
187                 <div class="attack-container">
188                     <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
189                     <form class="attack-form" accept-charset="UNKNOWN"
190                           method="POST" name="form"
191                           action="/WebGoat/SqlInjection/assignment5b">
```

## files.html, line 37 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | files.html 37 . | | |

**Sink:** files.html:37

```
35
36                          <!-- Standard Form -->
37                          <form th:action="@{/fileupload}" method="post" enctype="multipart/form-
data">
38                              <div class="form-inline">
39                                  <div class="form-group">
```

## HttpProxies.html, line 25 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | HttpProxies.html 25 . | | |

**Sink:** HttpProxies.html:25

```
23                  <div class="attack-container">
24                      <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
25                      <form class="attack-form" accept-charset="UNKNOWN" name="intercept-
request"
26                            method="POST"
27                            action="/WebGoat/HttpProxies/intercept-request">
```

## SecurePasswords.html, line 21 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | SecurePasswords.html 21 . | | |

**Sink:** SecurePasswords.html:21

```
19                  <div class="attack-container">
20                      <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
21                      <form class="attack-form" accept-charset="UNKNOWN"
22                            method="POST" name="form"
23                            action="/WebGoat/SecurePasswords/assignment"
```

## JWT.html, line 154 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | JWT.html 154 . | | |

ot Copyright 2024 Open Text.

| | |
|---|---|
| **Sink:** | JWT.html:154 |

```
152                 <div class="attack-container">
153                     <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
154                     <form class="attack-form" method="POST" name="form"
action="/WebGoat/JWT/secret">
155                         <div class="form-group">
156                             <div class="input-group">
```

## clientSideFilteringFree.js, line 41 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | clientSideFilteringFree.js 41 HTTP . | | |
| **Sink:** | clientSideFilteringFree.js:41 FunctionPointerCall: get() | | |

```
39                  $(".checkoutCode").on("blur", function () {
40                      var checkoutCode = $(".checkoutCode").val();
41                      $.get("clientSideFiltering/challenge-store/coupons/" + checkoutCode, function
(result, status) {
42                          var discount = result.discount;
43                          if (discount > 0) {
```

## challenge8.js, line 26 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | challenge8.js 26 HTTP . | | |
| **Sink:** | challenge8.js:26 FunctionPointerCall: get() | | |

```
24
25              function average() {
26                  $.get("challenge/8/votes/average", function (average) {
27                      for (var i = 1; i <= 5; i++) {
28                          var number = average["average"];
```

## SecurePasswords.html, line 21 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | SecurePasswords.html 21 . | | |
| **Sink:** | SecurePasswords.html:21 | | |

```
19                  <div class="attack-container">
20                      <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
21                      <form class="attack-form" accept-charset="UNKNOWN"
22                            method="POST" name="form"
23                            action="/WebGoat/SecurePasswords/assignment"
```

## AuthBypass.html, line 43 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | AuthBypass.html 43 . | | |
| **Sink:** | AuthBypass.html:43 | | |

```
41                      </form>
42
43                      <form class="attack-form" accept-charset="UNKNOWN" id="change-password-
form"
44                            method="POST" name="form"
45                            successCallback="onBypassResponse"
```

## WebWolfIntroduction.html, line 19 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |

ot Copyright 2024 Open Text.

| | |
|---|---|
| **Abstract:** | WebWolfIntroduction.html 19 . |
| **Sink:** | WebWolfIntroduction.html:19 |

```
17
18
19                  <form class="attack-form" accept-charset="UNKNOWN"
style="position:relative;top:150px"
20                      method="POST" name="form"
21                      action="/WebGoat/WebWolf/mail/">
```

**InsecureDeserialization.html, line 26 (Cross-Site Request Forgery)**

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | InsecureDeserialization.html 26 . | | |
| **Sink:** | InsecureDeserialization.html:26 | | |

```
24              <div class="attack-container">
25                  <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
26                  <form class="attack-form" accept-charset="UNKNOWN" name="task"
27                        method="POST"
28                        action="/WebGoat/InsecureDeserialization/task">
```

**SqlInjection.html, line 274 (Cross-Site Request Forgery)**

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | SqlInjection.html 274 . | | |
| **Sink:** | SqlInjection.html:274 | | |

```
272         <div class="attack-container">
273             <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
274             <form class="attack-form" accept-charset="UNKNOWN"
275                   method="POST" name="form"
276                   action="/WebGoat/SqlInjection/attack10"
```

**SqlInjectionMitigations.html, line 125 (Cross-Site Request Forgery)**

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | SqlInjectionMitigations.html 125 . | | |
| **Sink:** | SqlInjectionMitigations.html:125 | | |

```
123         <div class="attack-container">
124             <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
125             <form class="attack-form" accept-charset="UNKNOWN"
126                   method="POST" name="form"
127                   action="/WebGoat/SqlInjectionMitigations/attack12a">
```

**CrossSiteScripting.html, line 13 (Cross-Site Request Forgery)**

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | CrossSiteScripting.html 13 . | | |
| **Sink:** | CrossSiteScripting.html:13 | | |

```
11  <div class="attack-container">
12    <div id="lessonContent">
13      <form class="attack-form" accept-charset="UNKNOWN"
14          method="POST" name="form"
15          action="/WebGoat/CrossSiteScripting/attack1">
```

**ChromeDevTools.html, line 25 (Cross-Site Request Forgery)**

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |

| | |
|---|---|
| **Abstract:** | ChromeDevTools.html 25 . |
| **Sink:** | ChromeDevTools.html:25 |

```
23                  <div class="attack-container">
24                      <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
25                      <form class="attack-form" accept-charset="UNKNOWN"
26                            method="POST" name="DOMFollowUp"
27                            action="/WebGoat/ChromeDevTools/dummy">
```

## path_traversal.js, line 59 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | path_traversal.js 59 HTTP . | | |
| **Sink:** | path_traversal.js:59 FunctionPointerCall: get() | | |

```
57
58              function newRandomPicture() {
59                  $.get("PathTraversal/random-picture", function (result, status) {
60                      document.getElementById("randomCatPicture").src = "" +
result;
61                  });
```

## AuthBypass.html, line 23 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | AuthBypass.html 23 . | | |
| **Sink:** | AuthBypass.html:23 | | |

```
21                      <!-- modify the action to point to the intended endpoint and set other
attributes as desired -->
22                      <script th:src="@{/lesson_js/bypass.js}" />
23                      <form class="attack-form" accept-charset="UNKNOWN" id="verify-account-
form"
24                            method="POST" name="form"
25                            successCallback="onBypassResponse"
```

## bypass.js, line 11 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | bypass.js 11 HTTP . | | |
| **Sink:** | bypass.js:11 AssignmentStatement() | | |

```
9               console.warn("on view profile activated")
10              webgoat.customjs.jquery.ajax({
11                  method: "GET",
12                  url: "/WebGoat/IDOR/profile",
13                  contentType: 'application/json; charset=UTF-8'
```

## Challenge1.html, line 18 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | Challenge1.html 18 . | | |
| **Sink:** | Challenge1.html:18 | | |

```
16                          </div>
17                          <div class="panel-body">
18                              <form class="attack-form" accept-charset="UNKNOWN"
19                                    method="POST" name="form"
20                                    action="/WebGoat/challenge/1"
```

## JWT.html, line 188 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |

ot Copyright 2024 Open Text.

| | |
|---|---|
| **Abstract:** | JWT.html 188 . |
| **Sink:** | JWT.html:188 |

```
186                 <div class="attack-container">
187                     <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
188                     <form class="attack-form" accept-charset="UNKNOWN"
189                           method="POST"
190                           additionalHeaders="addBearerToken"
```

## webwolfPasswordReset.html, line 12 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | webwolfPasswordReset.html 12 . | | |
| **Sink:** | webwolfPasswordReset.html:12 | | |

```
10                  <div class="row">
11                      <div class="col-xs-12 col-sm-8 col-md-6 col-sm-offset-2 col-md-offset-3">
12                          <form role="form" method="GET" th:action="${webwolfUrl}">
13                              <h2 class="sign_up_title">Reset your password</h2>
14                              <input type="hidden" name="uniqueCode" th:value="${uniqueCode}"/>
```

## CrossSiteScripting.html, line 47 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | CrossSiteScripting.html 47 . | | |
| **Sink:** | CrossSiteScripting.html:47 | | |

```
45                  <div class="attack-container">
46                   <div id="lessonContent">
47                    <form class="attack-form" accept-charset="UNKNOWN"
48                       method="GET" name="xss-5a"
49                       action="/WebGoat/CrossSiteScripting/attack5a">
```

## PathTraversal.html, line 125 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | PathTraversal.html 125 . | | |
| **Sink:** | PathTraversal.html:125 | | |

```
123                     <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
124                     <div class="upload-container">
125                         <form class="attack-form" accept-charset="UNKNOWN"
126                               method="POST" name="form"
127                               onsubmit='return false'
```

## SqlInjection.html, line 40 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | SqlInjection.html 40 . | | |
| **Sink:** | SqlInjection.html:40 | | |

```
38                  <div class="attack-container">
39                      <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
40                      <form class="attack-form" accept-charset="UNKNOWN"
41                            method="POST" name="form"
42                            action="/WebGoat/SqlInjection/attack3"
```

## clientSideFiltering.js, line 17 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |

ot Copyright 2024 Open Text.

| | |
|---|---|
| **Abstract:** | clientSideFiltering.js 17 HTTP . |
| **Sink:** | clientSideFiltering.js:17 FunctionPointerCall: get() |

```
15
16  function ajaxFunction(userId) {
17      $.get("clientSideFiltering/salaries?userId=" + userId, function (result, status) {
18          var html = "<table border = '1' width = '90%' align = 'center'";
19          html = html + '<tr>';
```

## CSRF.html, line 146 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | CSRF.html 146 . | | |
| **Sink:** | CSRF.html:146 | | |

```
144                         <div class="col-md-8">
145                             <div class="well well-sm">
146                                 <form class="attack-form" accept-charset="UNKNOWN"
id="csrf-feedback"
147                                       method="POST"
148                                       prepareData="feedback"
```

## CSRF.html, line 35 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | CSRF.html 35 . | | |
| **Sink:** | CSRF.html:35 | | |

```
33          </div>
34          <br/>
35          <form class="attack-form" accept-charset="UNKNOWN" id="confirm-flag-1"
36                method="POST" name="form2"
37                successCallback=""
```

## CSRF.html, line 213 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | CSRF.html 213 . | | |
| **Sink:** | CSRF.html:213 | | |

```
211             </i>
212         </div>
213         <form class="attack-form" accept-charset="UNKNOWN" id="confirm-flag-feedback"
214               method="POST" name="form2"
215               action="/WebGoat/csrf/feedback">
```

## SqlInjectionMitigations.html, line 96 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | SqlInjectionMitigations.html 96 . | | |
| **Sink:** | SqlInjectionMitigations.html:96 | | |

```
94      <div class="attack-container">
95          <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
96          <form class="attack-form" accept-charset="UNKNOWN"
97                method="POST" name="form"
98                action="/WebGoat/SqlOnlyInputValidationOnKeywords/attack"
```

## PasswordReset.html, line 104 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | PasswordReset.html 104 . | | |

ot Copyright 2024 Open Text.

| | |
|---|---|
| Sink: | PasswordReset.html:104 |

```
102                 <div class="attack-container">
103                     <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
104                     <form class="attack-form" accept-charset="UNKNOWN"
105                           method="POST"
106                           action="/WebGoat/PasswordReset/questions">
```

## password_reset.html, line 12 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | password_reset.html 12 . | | |
| Sink: | password_reset.html:12 | | |

```
10                 <div class="row">
11                     <div class="col-xs-12 col-sm-8 col-md-6 col-sm-offset-2 col-md-offset-3">
12                         <form role="form" method="POST"
action="/WebGoat/PasswordReset/reset/change-password" th:object="${form}"
novalidate="novalidate">
13                             <h2 class="sign_up_title">Reset your password</h2>
14                                 <div class="form-group"
th:classappend="${#fields.hasErrors('password')}? 'has-error'">
```

## text.js, line 270 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | text.js 270 HTTP . | | |
| Sink: | text.js:270 FunctionPointerCall: open() | | |

```
268                 text.get = function (url, callback, errback, headers) {
269                     var xhr = text.createXhr(), header;
270                     xhr.open('GET', url, true);
271
272                     //Allow plugins direct access to xhr headers
```

## SqlInjectionAdvanced.html, line 21 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | SqlInjectionAdvanced.html 21 . | | |
| Sink: | SqlInjectionAdvanced.html:21 | | |

```
19                 <div class="attack-container">
20                     <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
21                     <form class="attack-form" accept-charset="UNKNOWN"
22                           method="POST" name="form"
23                           action="/WebGoat/SqlInjectionAdvanced/attack6a">
```

## idor.js, line 14 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | idor.js 14 HTTP . | | |
| Sink: | idor.js:14 AssignmentStatement() | | |

```
12     console.warn("on view profile activated")
13     webgoat.customjs.jquery.ajax({
14         method: "GET",
15         url: "/WebGoat/IDOR/profile",
16         contentType: 'application/json; charset=UTF-8'
```

## SqlInjection.html, line 88 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |

ot Copyright 2024 Open Text.

| | |
|---|---|
| **Abstract:** | SqlInjection.html 88 . |
| **Sink:** | SqlInjection.html:88 |

```
86          <div class="attack-container">
87              <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
88              <form class="attack-form" accept-charset="UNKNOWN"
89                    method="POST" name="form"
90                    action="/WebGoat/SqlInjection/attack5"
```

## credentials.js, line 3 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | credentials.js 3 HTTP . | | |
| **Sink:** | credentials.js:3 FunctionPointerCall: open() | | |

```
1   function submit_secret_credentials() {
2       var xhttp = new XMLHttpRequest();
3       xhttp['open']('POST', 'InsecureLogin/login', true);
4    //sending the request is obfuscated, to descourage js reading
5    var
_0xb7f9=["\x43\x61\x70\x74\x61\x69\x6E\x4A\x61\x63\x6B","\x42\x6C\x61\x63\x6B\x50\x65\
x61\x72\x6C","\x73\x74\x72\x69\x6E\x67\x69\x66\x79","\x73\x65\x6E\x64"];xhttp[_0xb7f9[
3]](JSON[_0xb7f9[2]]({username:_0xb7f9[0],password:_0xb7f9[1]}))
```

## PathTraversal.html, line 223 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | PathTraversal.html 223 . | | |
| **Sink:** | PathTraversal.html:223 | | |

```
221             <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
222             <div class="upload-container">
223                 <form class="attack-form" accept-charset="UNKNOWN"
224                       method="POST" name="form"
225                       onsubmit='return false'
```

## CrossSiteScriptingStored.html, line 68 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | CrossSiteScriptingStored.html 68 . | | |
| **Sink:** | CrossSiteScriptingStored.html:68 | | |

```
66    <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
67
68    <form class="attack-form" accept-charset="UNKNOWN"
69       method="POST" name="DOMFollowUp"
70       action="/WebGoat/CrossSiteScripting/stored-xss-follow-up">
```

## Cryptography.html, line 113 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | Cryptography.html 113 . | | |
| **Sink:** | Cryptography.html:113 | | |

```
111   <div class="attack-container">
112    <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
113    <form class="attack-form" method="POST" name="form"
action="/WebGoat/crypto/secure/defaults">
114    What is the unencrypted message<br/>
115    <input name="secretText" value="" type="TEXT"/><br/>
```

## GoatRouter.js, line 68 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | GoatRouter.js 68 HTTP . | | |
| Sink: | GoatRouter.js:68 AssignmentStatement() | | |

```
66                      console.log('phoneHome invoked');
67                      webgoat.customjs.jquery.ajax({
68                          method: "POST",
69                          url: "/WebGoat/CrossSiteScripting/phone-home-xss",
70                          data: {param1: 42, param2: 24},
```

## InsecureLogin.html, line 18 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | InsecureLogin.html 18 . | | |
| Sink: | InsecureLogin.html:18 | | |

```
16                  <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
17                  <script th:src="@{/lesson_js/credentials.js}"></script>
18                  <form class="attack-form" accept-charset="UNKNOWN" name="task"
19                        method="POST"
20                        action="/WebGoat/InsecureLogin/task">
```

## XXE.html, line 164 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | XXE.html 164 . | | |
| Sink: | XXE.html:164 | | |

```
162             <img th:src="@{/images/wolf-enabled.png}" class="webwolf-enabled"/>
163             <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
164             <form class="attack-form" accept-charset="UNKNOWN"
165                   method="POST" name="form"
166                   prepareData="blindXXE"
```

## JWT.html, line 16 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | JWT.html 16 . | | |
| Sink: | JWT.html:16 | | |

```
14      <div class="attack-container">
15          <img th:src="@{/images/wolf-enabled.png}" class="webwolf-enabled"/>
16          <form id="decode" class="attack-form" method="POST" name="form"
action="/WebGoat/JWT/decode">
17              <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
18              <br>
```

## path_traversal.js, line 59 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | path_traversal.js 59 HTTP . | | |
| Sink: | path_traversal.js:59 FunctionPointerCall: get() | | |

```
57
58  function newRandomPicture() {
59      $.get("PathTraversal/random-picture", function (result, status) {
60          document.getElementById("randomCatPicture").src = "" +
result;
61      });
```

ot Copyright 2024 Open Text.

| WebWolfIntroduction.html, line 19 (Cross-Site Request Forgery) | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | WebWolfIntroduction.html 19 . | | |
| Sink: | WebWolfIntroduction.html:19 | | |

```
17
18
19          <form class="attack-form" accept-charset="UNKNOWN"
style="position:relative;top:150px"
20                  method="POST" name="form"
21                  action="/WebGoat/WebWolf/mail/">
```

| JWT.html, line 49 (Cross-Site Request Forgery) | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | JWT.html 49 . | | |
| Sink: | JWT.html:49 | | |

```
47          <div class="attack-output"></div>
48          <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
49          <form class="attack-form" accept-charset="UNKNOWN"
50                  method="POST"
51                  successCallback="jwtSigningCallback"
```

| ChromeDevTools.html, line 46 (Cross-Site Request Forgery) | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | ChromeDevTools.html 46 . | | |
| Sink: | ChromeDevTools.html:46 | | |

```
44      <div class="attack-container">
45          <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
46          <form class="attack-form" accept-charset="UNKNOWN"
47                  method="POST" name="form"
48                  action="/WebGoat/ChromeDevTools/network">
```

| PasswordReset.html, line 144 (Cross-Site Request Forgery) | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | PasswordReset.html 144 . | | |
| Sink: | PasswordReset.html:144 | | |

```
142     <div class="attack-container">
143         <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
144         <form class="attack-form" accept-charset="UNKNOWN"
145                 method="POST" name="form"
146                 action="/WebGoat/PasswordReset/SecurityQuestions">
```

| Challenge5.html, line 26 (Cross-Site Request Forgery) | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | Challenge5.html 26 . | | |
| Sink: | Challenge5.html:26 | | |

```
24                              <div class="row">
25                                  <div class="col-lg-12">
26                                      <form id="login-form" class="attack-form" accept-
charset="UNKNOWN"
27                                              method="POST" name="form"
```

```
28                                                    action="/WebGoat/challenge/5" role="form">
```

## webwolf-login.html, line 17 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | webwolf-login.html 17 . | | |

**Sink:** webwolf-login.html:17

```
15                  <div class="row" style="margin-top:20px">
16                      <div class="col-xs-12 col-sm-8 col-md-6 col-sm-offset-2 col-md-offset-3"
th:style="'background:url(' + @{/images/wolf.png} + ') no-repeat right;'">
17                          <form th:action="@{/login}" method="post">
18                              <fieldset>
19                                  <h2>Sign in</h2>
```

## LessonTemplate.html, line 48 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | LessonTemplate.html 48 . | | |

**Sink:** LessonTemplate.html:48

```
46
47                          <!-- modify the action to point to the intended endpoint and set other
attributes as desired -->
48                          <form class="attack-form" accept-charset="UNKNOWN"
49                                method="POST" name="form"
50                                action="/WebGoat/lesson-template/sample-attack">
```

## path_traversal.js, line 53 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | path_traversal.js 53 HTTP . | | |

**Sink:** path_traversal.js:53 FunctionPointerCall: get()

```
51
52  webgoat.customjs.profileUploadCallbackRetrieval = function () {
53      $.get("PathTraversal/profile-picture", function (result, status) {
54          document.getElementById("previewRetrieval").src = "" +
result;
55      });
```

## SqlInjectionMitigations.html, line 176 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | SqlInjectionMitigations.html 176 . | | |

**Sink:** SqlInjectionMitigations.html:176

```
174                         </div>
175                     </form>
176                     <form class="attack-form" method="POST" name="form"
action="/WebGoat/SqlInjectionMitigations/attack12a">
177                         <div class="form-group">
178                             <div class="input-group">
```

## CSRF.html, line 237 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | CSRF.html 237 . | | |

**Sink:** CSRF.html:237

```
235                         </i>
236                     </div>
237                     <form class="attack-form" accept-charset="UNKNOWN" id="confirm-flag-login"
238                           method="POST" name="form2"
```

ot Copyright 2024 Open Text.

```
239                         action="/WebGoat/csrf/login">
```

## HttpBasics.html, line 26 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | HttpBasics.html 26 . | | |

**Sink:** HttpBasics.html:26

```
24                      action="/WebGoat/HttpBasics/attack1">
25                      <div id="lessonContent">
26                       <form accept-charset="UNKNOWN" method="POST" name="form"
27                        action="#attack/307/100">
28                        Enter Your Name: <input name="person" value="" type="TEXT"/><input
```

## webwolf-login.html, line 17 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | webwolf-login.html 17 . | | |

**Sink:** webwolf-login.html:17

```
15                      <div class="row" style="margin-top:20px">
16                          <div class="col-xs-12 col-sm-8 col-md-6 col-sm-offset-2 col-md-offset-3"
th:style="'background:url(' + @{/images/wolf.png} + ') no-repeat right;'">
17                              <form th:action="@{/login}" method="post">
18                                  <fieldset>
19                                      <h2>Sign in</h2>
```

## HttpBasics.html, line 52 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | HttpBasics.html 52 . | | |

**Sink:** HttpBasics.html:52

```
50                      <!-- you can write your own custom forms, but standard form submission
will take you to your endpoint and outside of the WebGoat framework -->
51                      <!-- of course, you can write your own ajax submission /handling in
your own javascript if you like -->
52                      <form class="attack-form" accept-charset="UNKNOWN"
53                       method="POST" name="form"
54                       action="/WebGoat/HttpBasics/attack2">
```

## BypassRestrictions.html, line 66 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | BypassRestrictions.html 66 . | | |

**Sink:** BypassRestrictions.html:66

```
64                          <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
65
66                          <form class="attack-form" accept-charset="UNKNOWN" name="frontendValidation"
67                                id="frontendValidation"
68                                method="POST"
```

## ChromeDevTools.html, line 46 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | ChromeDevTools.html 46 . | | |

**Sink:** ChromeDevTools.html:46

```
44                      <div class="attack-container">
45                          <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
46                          <form class="attack-form" accept-charset="UNKNOWN"
```

ot Copyright 2024 Open Text.

```
47                          method="POST" name="form"
48                          action="/WebGoat/ChromeDevTools/network">
```

## bypass.js, line 11 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | bypass.js 11 HTTP . | | |
| **Sink:** | bypass.js:11 AssignmentStatement() | | |

```
9       console.warn("on view profile activated")
10      webgoat.customjs.jquery.ajax({
11          method: "GET",
12          url: "/WebGoat/IDOR/profile",
13          contentType: 'application/json; charset=UTF-8'
```

## IDOR.html, line 23 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | IDOR.html 23 . | | |
| **Sink:** | IDOR.html:23 | | |

```
21
22              <!-- modify the action to point to the intended endpoint -->
23              <form class="attack-form" accept-charset="UNKNOWN"
24                    method="POST" name="form"
25                    action="/WebGoat/IDOR/login">
```

## PathTraversal.html, line 70 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | PathTraversal.html 70 . | | |
| **Sink:** | PathTraversal.html:70 | | |

```
68              <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
69              <div class="upload-container">
70                  <form class="attack-form" accept-charset="UNKNOWN"
71                        method="POST" name="form"
72                        onsubmit='return false'
```

## Cryptography.html, line 90 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | Cryptography.html 90 . | | |
| **Sink:** | Cryptography.html:90 | | |

```
88          Now suppose you have the following private key:<br/>
89          <pre><div id="privatekey" ></div></pre><br/>
90          <form class="attack-form" method="POST" name="form"
action="/WebGoat/crypto/signing/verify">
91          Then what was the modulus of the public key
92          <input name="modulus" value="" type="TEXT"/>
```

## clientSideFilteringFree.js, line 41 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | clientSideFilteringFree.js 41 HTTP . | | |
| **Sink:** | clientSideFilteringFree.js:41 FunctionPointerCall: get() | | |

```
39      $(".checkoutCode").on("blur", function () {
40          var checkoutCode = $(".checkoutCode").val();
41          $.get("clientSideFiltering/challenge-store/coupons/" + checkoutCode, function
(result, status) {
```

ot Copyright 2024 Open Text.

```
42                          var discount = result.discount;
43                          if (discount > 0) {
```

## jquery.form.js, line 245 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | jquery.form.js 245 HTTP . | | |

Sink: jquery.form.js:245 FunctionPointerCall: get()

```
243                         // see:  http://groups.google.com/group/jquery-
dev/browse_thread/thread/36395b7ab510dd5d
244                         if (options.closeKeepAlive) {
245                             $.get(options.closeKeepAlive, function() {
246                                 jqxhr = fileUploadIframe(a);
247                             });
```

## credentials.js, line 3 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | credentials.js 3 HTTP . | | |

Sink: credentials.js:3 FunctionPointerCall: open()

```
1                   function submit_secret_credentials() {
2                       var xhttp = new XMLHttpRequest();
3                       xhttp['open']('POST', 'InsecureLogin/login', true);
4                    //sending the request is obfuscated, to descourage js reading
5                    var
_0xb7f9=["\x43\x61\x70\x74\x61\x69\x6E\x4A\x61\x63\x6B","\x42\x6C\x61\x63\x6B\x50\x65\
x61\x72\x6C","\x73\x74\x72\x69\x6E\x67\x69\x66\x79","\x73\x65\x6E\x64"];xhttp[_0xb7f9[
3]](JSON[_0xb7f9[2]]({username:_0xb7f9[0],password:_0xb7f9[1]}))
```

## CrossSiteScriptingMitigation.html, line 44 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | CrossSiteScriptingMitigation.html 44 . | | |

Sink: CrossSiteScriptingMitigation.html:44

```
42                   <div class="adoc-content"
th:replace="doc:lessons/xss/documentation/CrossSiteScripting_content8c.adoc"></div>
43                   <div class="attack-container" style="height: 100%; border: none !important;min-height:
450px;">
44                     <form id="codesubmit2" style="height: 100%; min-height: 350px;" class="attack-form"
accept-charset="UNKNOWN" method="POST" name="form"
action="/WebGoat/CrossSiteScripting/attack4">
45                       <div>
46                        <div id="editor2" style="position: absolute; top: 0; right: 0; bottom: 0; left: 0;
height: 350px;" name="editor2"></div>
```

## Cryptography.html, line 48 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | Cryptography.html 48 . | | |

Sink: Cryptography.html:48

```
46                     <div class="attack-container">
47                       <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
48                       <form class="attack-form" method="POST" name="form"
action="/WebGoat/crypto/encoding/xor">
49                       Suppose you found the database password encoded as {xor}Oz4rPj0+LDovPiwsKDAtOw==<br/>
50                       What would be the actual password
```

## Challenge5.html, line 69 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |

| | |
|---|---|
| **Abstract:** | Challenge5.html 69 . |
| **Sink:** | Challenge5.html:69 |

```
67                  </div>
68                  <br/>
69                  <form class="attack-form" method="POST" name="form"
action="/WebGoat/challenge/flag">
70                      <div class="form-group">
71                          <div class="input-group">
```

**SqlInjection.html, line 16 (Cross-Site Request Forgery)**

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | SqlInjection.html 16 . | | |
| **Sink:** | SqlInjection.html:16 | | |

```
14      <div class="attack-container">
15          <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
16          <form class="attack-form" accept-charset="UNKNOWN"
17                method="POST" name="form"
18                action="/WebGoat/SqlInjection/attack2"
```

**InsecureDeserialization.html, line 26 (Cross-Site Request Forgery)**

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | InsecureDeserialization.html 26 . | | |
| **Sink:** | InsecureDeserialization.html:26 | | |

```
24          <div class="attack-container">
25              <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
26              <form class="attack-form" accept-charset="UNKNOWN" name="task"
27                    method="POST"
28                    action="/WebGoat/InsecureDeserialization/task">
```

**Challenge6.html, line 65 (Cross-Site Request Forgery)**

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | Challenge6.html 65 . | | |
| **Sink:** | Challenge6.html:65 | | |

```
63                                          </div>
64                                      </form>
65                                      <form id="register-form" class="attack-form"
accept-charset="UNKNOWN"
66                                            method="PUT" name="form"
67                                            action="/WebGoat/challenge/6"
style="display: none;" role="form">
```

**password_reset.html, line 12 (Cross-Site Request Forgery)**

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | password_reset.html 12 . | | |
| **Sink:** | password_reset.html:12 | | |

```
10      <div class="row">
11          <div class="col-xs-12 col-sm-8 col-md-6 col-sm-offset-2 col-md-offset-3">
12              <form role="form" method="POST"
action="/WebGoat/PasswordReset/reset/change-password" th:object="${form}"
novalidate="novalidate">
13                  <h2 class="sign_up_title">Reset your password</h2>
14                      <div class="form-group"
th:classappend="${#fields.hasErrors('password')}? 'has-error'">
```

**challenge8.js, line 46 (Cross-Site Request Forgery)**

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | challenge8.js 46 HTTP . | | |

Sink: challenge8.js:46 FunctionPointerCall: get()

```
44 function doVote(stars) {
45     $("#voteResultMsg").hide();
46     $.get("challenge/8/vote/" + stars, function (result) {
47         if (result["error"]) {
48             $("#voteResultMsg").addClass('alert-danger alert-dismissable');
```

## GoatRouter.js, line 68 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | GoatRouter.js 68 HTTP . | | |

Sink: GoatRouter.js:68 AssignmentStatement()

```
66                     console.log('phoneHome invoked');
67                     webgoat.customjs.jquery.ajax({
68                         method: "POST",
69                         url: "/WebGoat/CrossSiteScripting/phone-home-xss",
70                         data: {param1: 42, param2: 24},
```

## ChromeDevTools.html, line 67 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | ChromeDevTools.html 67 . | | |

Sink: ChromeDevTools.html:67

```
65         </form>
66
67         <form class="attack-form" accept-charset="UNKNOWN"
68               method="POST" name="form"
69               action="/WebGoat/ChromeDevTools/network">
```

## CIA.html, line 30 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | CIA.html 30 . | | |

Sink: CIA.html:30

```
28         <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
29         <div class="container-fluid">
30             <form id="quiz-form" class="attack-form" accept-charset="UNKNOWN"
31                   method="POST" name="form"
32                   action="cia/quiz" role="form">
```

## SqlInjectionMitigations.html, line 125 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | SqlInjectionMitigations.html 125 . | | |

Sink: SqlInjectionMitigations.html:125

```
123     <div class="attack-container">
124         <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
125         <form class="attack-form" accept-charset="UNKNOWN"
126               method="POST" name="form"
127               action="/WebGoat/SqlInjectionMitigations/attack12a">
```

## CrossSiteScriptingStored.html, line 68 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|

| Kingdom: | Encapsulation |
|---|---|
| Abstract: | CrossSiteScriptingStored.html 68 . |
| Sink: | CrossSiteScriptingStored.html:68 |

```
66                <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
67
68              <form class="attack-form" accept-charset="UNKNOWN"
69                  method="POST" name="DOMFollowUp"
70                  action="/WebGoat/CrossSiteScripting/stored-xss-follow-up">
```

## PasswordReset.html, line 48 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | PasswordReset.html 48 . | | |
| Sink: | PasswordReset.html:48 | | |

```
46                                  </div>
47                              </form>
48                              <form class="attack-form" accept-charset="UNKNOWN"
novalidate="novalidate"
49                                  method="POST"
50                                  action="/WebGoat/PasswordReset/simple-mail">
```

## path_traversal.js, line 75 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | path_traversal.js 75 HTTP . | | |
| Sink: | path_traversal.js:75 FunctionPointerCall: get() | | |

```
73
74  webgoat.customjs.profileZipSlipRetrieval = function () {
75      $.get("PathTraversal/zip-slip", function (result, status) {
76          document.getElementById("previewZipSlip").src = "" +
result;
77      });
```

## XXE.html, line 92 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | XXE.html 92 . | | |
| Sink: | XXE.html:92 | | |

```
90      <div class="attack-container">
91          <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
92          <form class="attack-form" accept-charset="UNKNOWN"
93                method="POST" name="form"
94                prepareData="contentTypeXXE"
```

## Challenge8.html, line 234 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | Challenge8.html 234 . | | |
| Sink: | Challenge8.html:234 | | |

```
232
233             <br/>
234             <form class="attack-form" method="POST" name="form"
action="/WebGoat/challenge/flag">
235                 <div class="form-group">
236                     <div class="input-group">
```

## Cryptography.html, line 31 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | Cryptography.html 31 . | | |

Sink: Cryptography.html:31

```
29                    Now suppose you have intercepted the following header:<br/>
30                    <div id="basicauthtoken" ></div><br/>
31                    <form class="attack-form" method="POST" name="form"
action="/WebGoat/crypto/encoding/basic-auth">
32                    Then what was the username
33                    <input name="answer_user" value="" type="TEXT"/>
```

## SSRF.html, line 35 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | SSRF.html 35 . | | |

Sink: SSRF.html:35

```
33                <div class="attack-container">
34                    <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
35                    <form class="attack-form" accept-charset="UNKNOWN"
36                          method="POST" name="form"
37                          action="/WebGoat/SSRF/task2">
```

## IDOR.html, line 135 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | IDOR.html 135 . | | |

Sink: IDOR.html:135

```
133
134                <!-- modify the action to point to the intended endpoint -->
135                <form class="attack-form" accept-charset="UNKNOWN" id="view-other"
136                      method="GET" name="view-other-profile"
137                      action="/WebGoat/IDOR/profile/{userId}">
```

## ace.js, line 4157 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | ace.js 4157 HTTP . | | |

Sink: ace.js:4157 FunctionPointerCall: open()

```
4155              exports.get = function (url, callback) {
4156                  var xhr = new XMLHttpRequest();
4157                  xhr.open('GET', url, true);
4158                  xhr.onreadystatechange = function () {
4159                      if (xhr.readyState === 4) {
```

## PasswordReset.html, line 223 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | PasswordReset.html 223 . | | |

Sink: PasswordReset.html:223

```
221                                    Forgot your password?
222                                </h4>
223                                <form class="attack-form" accept-charset="UNKNOWN"
224                                      method="POST" name="form"
225
action="/WebGoat/PasswordReset/ForgotPassword/create-password-reset-link"
```

## LogSpoofing.html, line 17 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | LogSpoofing.html 17 . | | |

Sink: LogSpoofing.html:17

```
15      <div class="attack-container">
16          <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
17          <form class="attack-form" accept-charset="UNKNOWN" name="task"
18                method="POST"
19                action="/WebGoat/LogSpoofing/log-spoofing">
```

## CrossSiteScripting.html, line 13 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | CrossSiteScripting.html 13 . | | |

Sink: CrossSiteScripting.html:13

```
11   <div class="attack-container">
12    <div id="lessonContent">
13     <form class="attack-form" accept-charset="UNKNOWN"
14        method="POST" name="form"
15        action="/WebGoat/CrossSiteScripting/attack1">
```

## csrf-review.js, line 35 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | csrf-review.js 35 HTTP . | | |

Sink: csrf-review.js:35 FunctionPointerCall: get()

```
33      function getChallenges() {
34          $("#list").empty();
35          $.get('csrf/review', function (result, status) {
36              for (var i = 0; i < result.length; i++) {
37                  var comment = html.replace('USER', result[i].user);
```

## CrossSiteScriptingMitigation.html, line 44 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | CrossSiteScriptingMitigation.html 44 . | | |

Sink: CrossSiteScriptingMitigation.html:44

```
42   <div class="adoc-content"
th:replace="doc:lessons/xss/documentation/CrossSiteScripting_content8c.adoc"></div>
43   <div class="attack-container" style="height: 100%; border: none !important;min-height:
450px;">
44     <form id="codesubmit2" style="height: 100%; min-height: 350px;" class="attack-form"
accept-charset="UNKNOWN" method="POST" name="form"
action="/WebGoat/CrossSiteScripting/attack4">
45     <div>
46      <div id="editor2" style="position: absolute; top: 0; right: 0; bottom: 0; left: 0;
height: 350px;" name="editor2"></div>
```

## JWT.html, line 16 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | JWT.html 16 . | | |

Sink: JWT.html:16

```
14      <div class="attack-container">
15          <img th:src="@{/images/wolf-enabled.png}" class="webwolf-enabled"/>
16          <form id="decode" class="attack-form" method="POST" name="form"
action="/WebGoat/JWT/decode">
17              <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
```

ot Copyright 2024 Open Text.

```
18                          <br>
```

## Challenge6.html, line 30 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | Challenge6.html 30 . | | |

**Sink:** Challenge6.html:30

```
28                                  <div class="row">
29                                      <div class="col-lg-12">
30                                          <form id="login-form" class="attack-form" accept-
charset="UNKNOWN"
31                                                  method="POST" name="form"
32                                                  action="/WebGoat/challenge/6" role="form">
```

## IDOR.html, line 81 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | IDOR.html 81 . | | |

**Sink:** IDOR.html:81

```
79              <div class="adoc-content"
th:replace="doc:lessons/idor/documentation/IDOR_whatDiffs.adoc"></div>
80              <!-- modify the action to point to the intended endpoint -->
81              <form class="attack-form"
82                    method="POST" name="diff-form"
83                    action="IDOR/diff-attributes">
```

## CrossSiteScripting.html, line 134 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | CrossSiteScripting.html 134 . | | |

**Sink:** CrossSiteScripting.html:134

```
132          <div class="attack-container">
133            <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
134            <form class="attack-form" accept-charset="UNKNOWN"
135               method="POST" name="DOMTestRoute"
136               action="/WebGoat/CrossSiteScripting/attack6a">
```

## VulnerableComponents.html, line 100 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | VulnerableComponents.html 100 . | | |

**Sink:** VulnerableComponents.html:100

```
98            <div class="attack-container">
99             <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
100            <form class="attack-form" accept-charset="UNKNOWN"
101             method="POST" name="form"
102             action="/WebGoat/VulnerableComponents/attack1">
```

## PathTraversal.html, line 223 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | PathTraversal.html 223 . | | |

**Sink:** PathTraversal.html:223

```
221             <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
222             <div class="upload-container">
223                 <form class="attack-form" accept-charset="UNKNOWN"
```

ot Copyright 2024 Open Text.

```
224                                 method="POST" name="form"
225                                 onsubmit='return false'
```

## path_traversal.js, line 13 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | path_traversal.js 13 HTTP . | | |
| **Sink:** | path_traversal.js:13 FunctionPointerCall: get() | | |

```
11
12                  webgoat.customjs.profileUploadCallback = function () {
13                      $.get("PathTraversal/profile-picture", function (result, status) {
14                          document.getElementById("preview").src = "" + result;
15                      });
```

## LogSpoofing.html, line 39 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | LogSpoofing.html 39 . | | |
| **Sink:** | LogSpoofing.html:39 | | |

```
37                      <div class="attack-container">
38                          <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
39                          <form class="attack-form" accept-charset="UNKNOWN" name="task"
40                                method="POST"
41                                action="/WebGoat/LogSpoofing/log-bleeding">
```

## InsecureLogin.html, line 26 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | InsecureLogin.html 26 . | | |
| **Sink:** | InsecureLogin.html:26 | | |

```
24                          </form>
25                          <br></br>
26                          <form class="attack-form" accept-charset="UNKNOWN" name="task"
27                                method="POST"
28                                action="/WebGoat/InsecureLogin/task">
```

## CSRF.html, line 237 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | CSRF.html 237 . | | |
| **Sink:** | CSRF.html:237 | | |

```
235                         </i>
236                     </div>
237                     <form class="attack-form" accept-charset="UNKNOWN" id="confirm-flag-login"
238                           method="POST" name="form2"
239                           action="/WebGoat/csrf/login">
```

## hijackform.html, line 3 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | hijackform.html 3 . | | |
| **Sink:** | hijackform.html:3 | | |

```
1                   <div class="row">
2                    <div class="col-md-4">
3                     <form class="attack-form" accept-charset="UNKNOWN" method="POST"
4                      action="/WebGoat/HijackSession/login">
5                      <div style="padding: 20px;" id="password-login">
```

## jwt.html, line 32 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | jwt.html 32 . | | |

Sink: jwt.html:32

```
30                  </form>
31              </div>
32              <form id="encodeForm">
33                  <div class="form-group">
34                      <label>Decoded</label>
```

## Challenge1.html, line 40 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | Challenge1.html 40 . | | |

Sink: Challenge1.html:40

```
38              </div>
39
40              <form class="attack-form" method="POST" name="form"
action="/WebGoat/challenge/flag">
41                  <div class="form-group">
42                      <div class="input-group">
```

## Challenge6.html, line 65 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | Challenge6.html 65 . | | |

Sink: Challenge6.html:65

```
63                                          </div>
64                                      </form>
65                                      <form id="register-form" class="attack-form"
accept-charset="UNKNOWN"
66                                            method="PUT" name="form"
67                                            action="/WebGoat/challenge/6"
style="display: none;" role="form">
```

## IDOR.html, line 81 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | IDOR.html 81 . | | |

Sink: IDOR.html:81

```
79              <div class="adoc-content"
th:replace="doc:lessons/idor/documentation/IDOR_whatDiffs.adoc"></div>
80              <!-- modify the action to point to the intended endpoint -->
81              <form class="attack-form"
82                    method="POST" name="diff-form"
83                    action="IDOR/diff-attributes">
```

## SqlInjection.html, line 217 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | SqlInjection.html 217 . | | |

Sink: SqlInjection.html:217

```
215         <div class="attack-container">
216             <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
217             <form class="attack-form" accept-charset="UNKNOWN"
218                   method="POST" name="form"
```

ot Copyright 2024 Open Text.

```
219                         action="/WebGoat/SqlInjection/attack8"
```

## SqlInjection.html, line 189 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | SqlInjection.html 189 . | | |

**Sink:** SqlInjection.html:189

```
187     <div class="attack-container">
188         <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
189         <form class="attack-form" accept-charset="UNKNOWN"
190               method="POST" name="form"
191               action="/WebGoat/SqlInjection/assignment5b">
```

## login.html, line 31 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | login.html 31 . | | |

**Sink:** login.html:31

```
29              </div>
30              <br/><br/>
31              <form th:action="@{/login}" method='POST' style="width: 200px;">
32                  <div class="form-group">
33                      <label for="exampleInputEmail1"
th:text="#{username}">Username</label>
```

## SqlInjectionMitigations.html, line 26 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | SqlInjectionMitigations.html 26 . | | |

**Sink:** SqlInjectionMitigations.html:26

```
24      <div class="attack-container">
25          <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
26          <form class="attack-form" accept-charset="UNKNOWN" method="POST" name="form"
action="/WebGoat/SqlInjectionMitigations/attack10a">
27              <div>
28                  <p>Connection conn = DriverManager.<input type="text" name="field1"
id="field1" />(DBURL, DBUSER, DBPW);</p>
```

## idor.js, line 14 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | idor.js 14 HTTP . | | |

**Sink:** idor.js:14 AssignmentStatement()

```
12      console.warn("on view profile activated")
13      webgoat.customjs.jquery.ajax({
14          method: "GET",
15          url: "/WebGoat/IDOR/profile",
16          contentType: 'application/json; charset=UTF-8'
```

## CrossSiteScripting.html, line 134 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | CrossSiteScripting.html 134 . | | |

**Sink:** CrossSiteScripting.html:134

```
132  <div class="attack-container">
133    <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
```

ot Copyright 2024 Open Text.

```
134                   <form class="attack-form" accept-charset="UNKNOWN"
135                      method="POST" name="DOMTestRoute"
136                      action="/WebGoat/CrossSiteScripting/attack6a">
```

## BypassRestrictions.html, line 66 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | BypassRestrictions.html 66 . | | |

**Sink:** BypassRestrictions.html:66

```
64                          <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
65
66                          <form class="attack-form" accept-charset="UNKNOWN" name="frontendValidation"
67                                id="frontendValidation"
68                                method="POST"
```

## Cryptography.html, line 48 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | Cryptography.html 48 . | | |

**Sink:** Cryptography.html:48

```
46                    <div class="attack-container">
47                     <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
48                     <form class="attack-form" method="POST" name="form"
action="/WebGoat/crypto/encoding/xor">
49                     Suppose you found the database password encoded as {xor}Oz4rPj0+LDovPiwsKDAtOw==<br/>
50                     What would be the actual password
```

## IDOR.html, line 58 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | IDOR.html 58 . | | |

**Sink:** IDOR.html:58

```
56
57                          <!-- modify the action to point to the intended endpoint -->
58                          <form class="attack-form" accept-charset="UNKNOWN"
59                                method="GET" name="form"
60                                action="/WebGoat/IDOR/profile">
```

## PasswordReset.html, line 176 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | PasswordReset.html 176 . | | |

**Sink:** PasswordReset.html:176

```
174                         <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
175
176                         <form class="attack-form" accept-charset="UNKNOWN"
177                               method="POST"
178                               action="/WebGoat/PasswordReset/reset/login">
```

## LessonController.js, line 147 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | LessonController.js 147 HTTP . | | |

**Sink:** LessonController.js:147 AssignmentStatement()

```
145                       $.ajax({
146                           url:'service/restartlesson.mvc',
```

ot Copyright 2024 Open Text.

```
147                                 method:'GET'
148                             }).done(function(lessonLink) {
149                                 self.loadLesson(self.name);
```

## XXE.html, line 92 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | XXE.html 92 . | | |

Sink: XXE.html:92

```
90          <div class="attack-container">
91              <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
92              <form class="attack-form" accept-charset="UNKNOWN"
93                    method="POST" name="form"
94                    prepareData="contentTypeXXE"
```

## ChromeDevTools.html, line 67 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | ChromeDevTools.html 67 . | | |

Sink: ChromeDevTools.html:67

```
65              </form>
66
67              <form class="attack-form" accept-charset="UNKNOWN"
68                    method="POST" name="form"
69                    action="/WebGoat/ChromeDevTools/network">
```

## HttpProxies.html, line 25 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | HttpProxies.html 25 . | | |

Sink: HttpProxies.html:25

```
23              <div class="attack-container">
24                  <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
25                  <form class="attack-form" accept-charset="UNKNOWN" name="intercept-
request"
26                        method="POST"
27                        action="/WebGoat/HttpProxies/intercept-request">
```

## SSRF.html, line 35 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | SSRF.html 35 . | | |

Sink: SSRF.html:35

```
33              <div class="attack-container">
34                  <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
35                  <form class="attack-form" accept-charset="UNKNOWN"
36                        method="POST" name="form"
37                        action="/WebGoat/SSRF/task2">
```

## MissingFunctionAC.html, line 54 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Encapsulation | | |
| Abstract: | MissingFunctionAC.html 54 . | | |

Sink: MissingFunctionAC.html:54

```
52
```

ot Copyright 2024 Open Text.

```
53                  <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
54                  <form class="attack-form" accept-charset="UNKNOWN"
55                        method="POST" name="form"
56                        action="/WebGoat/access-control/hidden-menu">
```

## JWT.html, line 188 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | JWT.html 188 . | | |

**Sink:** JWT.html:188

```
186     <div class="attack-container">
187         <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
188         <form class="attack-form" accept-charset="UNKNOWN"
189               method="POST"
190               additionalHeaders="addBearerToken"
```

## PasswordReset.html, line 24 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | PasswordReset.html 24 . | | |

**Sink:** PasswordReset.html:24

```
22
23
24                      <form class="attack-form" accept-charset="UNKNOWN"
novalidate="novalidate"
25                            method="POST"
26                            action="/WebGoat/PasswordReset/simple-mail/reset">
```

## LogSpoofing.html, line 39 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | LogSpoofing.html 39 . | | |

**Sink:** LogSpoofing.html:39

```
37      <div class="attack-container">
38          <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
39          <form class="attack-form" accept-charset="UNKNOWN" name="task"
40                method="POST"
41                action="/WebGoat/LogSpoofing/log-bleeding">
```

## SqlInjectionAdvanced.html, line 80 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | SqlInjectionAdvanced.html 80 . | | |

**Sink:** SqlInjectionAdvanced.html:80

```
78                          <div class="row">
79                              <div class="col-lg-12">
80                                  <form id="login-form" class="attack-form" accept-
charset="UNKNOWN"
81                                        method="POST" name="form"
82
action="/WebGoat/SqlInjectionAdvanced/challenge_Login"
```

## quiz.js, line 15 (Cross-Site Request Forgery)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Encapsulation | | |
| **Abstract:** | quiz.js 15 HTTP . | | |

ot Copyright 2024 Open Text.

| Sink: | quiz.js:15 FunctionPointerCall: open() |
|---|---|

```
13      var client = new XMLHttpRequest();
14      var quiz_id = document.getElementById("quiz_id").getAttribute("data-quiz_id");
15      client.open('GET', '/WebGoat/lesson_js/questions_' + quiz_id + '.json');
16      client.onreadystatechange = function() {
17          if (this.readyState == 4 && this.status == 200) {
```

## PasswordReset.html, line 24 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | PasswordReset.html 24 . | | |

| Sink: | PasswordReset.html:24 |
|---|---|

```
22
23
24                          <form class="attack-form" accept-charset="UNKNOWN"
novalidate="novalidate"
25                                  method="POST"
26                                  action="/WebGoat/PasswordReset/simple-mail/reset">
```

## Challenge6.html, line 102 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | Challenge6.html 102 . | | |

| Sink: | Challenge6.html:102 |
|---|---|

```
100             </div>
101             <br/>
102             <form class="attack-form" method="POST" name="form"
action="/WebGoat/challenge/flag">
103                 <div class="form-group">
104                     <div class="input-group">
```

## XXE.html, line 25 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | XXE.html 25 . | | |

| Sink: | XXE.html:25 |
|---|---|

```
23              <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
24
25              <form class="attack-form" accept-charset="UNKNOWN"
26                    method="POST" name="form"
27                    prepareData="simpleXXE"
```

## SqlInjectionAdvanced.html, line 21 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | SqlInjectionAdvanced.html 21 . | | |

| Sink: | SqlInjectionAdvanced.html:21 |
|---|---|

```
19      <div class="attack-container">
20          <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
21          <form class="attack-form" accept-charset="UNKNOWN"
22                method="POST" name="form"
23                action="/WebGoat/SqlInjectionAdvanced/attack6a">
```

## LogSpoofing.html, line 17 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | LogSpoofing.html 17 . | | |

ot Copyright 2024 Open Text.

| Sink: | LogSpoofing.html:17 |
|---|---|

```
15          <div class="attack-container">
16              <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
17              <form class="attack-form" accept-charset="UNKNOWN" name="task"
18                    method="POST"
19                    action="/WebGoat/LogSpoofing/log-spoofing">
```

## xxe.js, line 72 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | xxe.js 72 HTTP . | | |
| Sink: | xxe.js:72 FunctionPointerCall: get() | | |

```
70
71          function getComments(field) {
72              $.get("xxe/comments", function (result, status) {
73                  $(field).empty();
74                  for (var i = 0; i < result.length; i++) {
```

## ace.js, line 4157 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | ace.js 4157 HTTP . | | |
| Sink: | ace.js:4157 FunctionPointerCall: open() | | |

```
4155        exports.get = function (url, callback) {
4156            var xhr = new XMLHttpRequest();
4157            xhr.open('GET', url, true);
4158            xhr.onreadystatechange = function () {
4159                if (xhr.readyState === 4) {
```

## BypassRestrictions.html, line 19 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | BypassRestrictions.html 19 . | | |
| Sink: | BypassRestrictions.html:19 | | |

```
17              <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
18              <div class="container-fluid">
19                  <form class="attack-form" accept-charset="UNKNOWN"
name="fieldRestrictions"
20                        method="POST"
21                        action="/WebGoat/BypassRestrictions/FieldRestrictions">
```

## SqlInjection.html, line 144 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | SqlInjection.html 144 . | | |
| Sink: | SqlInjection.html:144 | | |

```
142         <div class="attack-container">
143             <div class="assignment-success"><i class="fa fa-2 fa-check hidden" aria-
hidden="true"></i></div>
144             <form class="attack-form" accept-charset="UNKNOWN"
145                   method="POST" name="form"
146                   action="/WebGoat/SqlInjection/assignment5a">
```

## CSRF.html, line 93 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | CSRF.html 93 . | | |

ot Copyright 2024 Open Text.

| Sink: | CSRF.html:93 |
|---|---|

```
91                                  <div class="post-footer">
92                                      <div class="input-group">
93                                          <form class="attack-form" accept-charset="UNKNOWN"
id="csrf-review"
94                                                  method="POST" name="review-form"
95                                                  successCallback=""
```

## goatApp.js, line 21 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | goatApp.js 21 HTTP . | | |
| Sink: | goatApp.js:21 FunctionPointerCall: getJSON() | | |

```
19                  initApp: function () {
20                      var locale = localStorage.getItem('locale') || 'en';
21                      $.getJSON('service/labels.mvc', function(data) {
22                          window.polyglot = new Polyglot({phrases: data});//i18n polyglot
labels
23                          asyncErrorHandler.init();
```

## Challenge8.html, line 234 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | Challenge8.html 234 . | | |
| Sink: | Challenge8.html:234 | | |

```
232
233                 <br/>
234                 <form class="attack-form" method="POST" name="form"
action="/WebGoat/challenge/flag">
235                     <div class="form-group">
236                         <div class="input-group">
```

## registration.html, line 15 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | registration.html 15 . | | |
| Sink: | registration.html:15 | | |

```
13      <fieldset>
14          <legend th:text="#{register.title}">Please Sign Up</legend>
15          <form class="form-horizontal" action="#" th:action="@{/register.mvc}"
th:object="${userForm}"
16                  method='POST'>
```

## assignment13.js, line 43 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | assignment13.js 43 HTTP . | | |
| Sink: | assignment13.js:43 FunctionPointerCall: get() | | |

```
41
42  function getServers(column) {
43      $.get("SqlInjectionMitigations/servers?column=" + column, function (result,
status) {
44          $("#servers").empty();
45          for (var i = 0; i < result.length; i++) {
```

## PathTraversal.html, line 192 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | PathTraversal.html 192 . | | |

ot Copyright 2024 Open Text.

| Sink: | PathTraversal.html:192 |
|---|---|

```
190
191                     <br/>
192                     <form class="attack-form" method="POST" name="form"
action="/WebGoat/PathTraversal/random">
193                         <div class="assignment-success"><i class="fa fa-2 fa-check hidden"
aria-hidden="true"></i></div>
194                         <div class="form-group">
```

## GoatUtils.js, line 56 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | GoatUtils.js 56 HTTP . | | |
| Sink: | GoatUtils.js:56 FunctionPointerCall: get() | | |

```
54
55                          showLessonCookiesAndParams: function() {
56                              $.get(goatConstants.cookieService, {}, function(reply) {
57                                  $("#lesson_cookies").html(reply);
58                              }, "html");
```

## path_traversal.js, line 29 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | path_traversal.js 29 HTTP . | | |
| Sink: | path_traversal.js:29 FunctionPointerCall: get() | | |

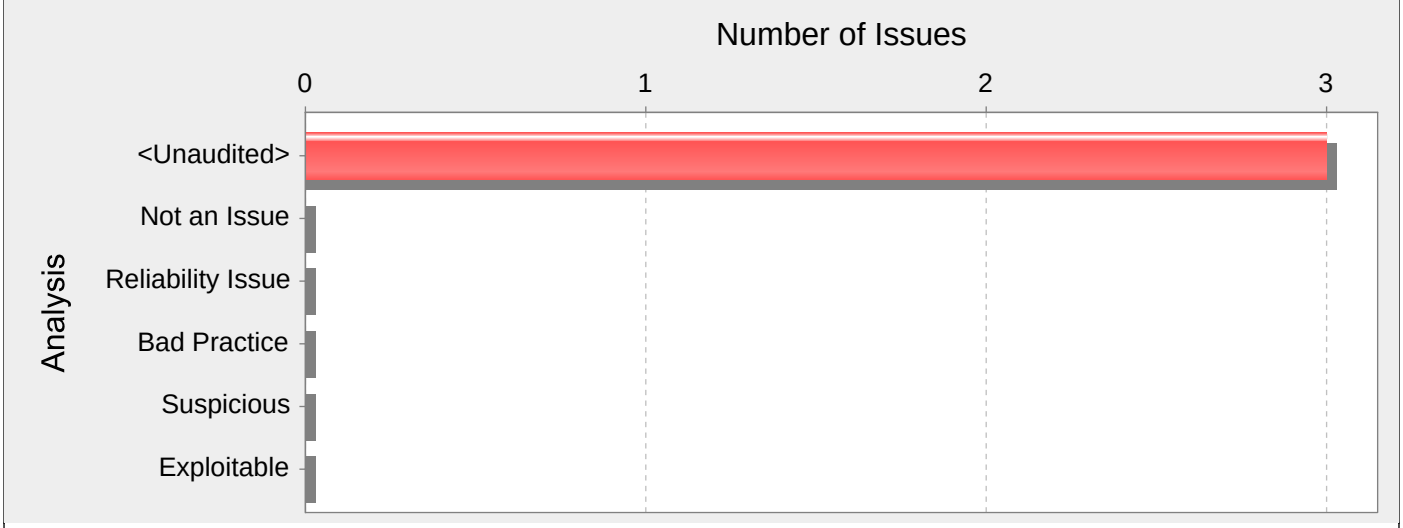
```
27
28  webgoat.customjs.profileUploadCallbackFix = function () {
29      $.get("PathTraversal/profile-picture", function (result, status) {
30          document.getElementById("previewFix").src = "" + result;
31      });
```

## registration.html, line 26 (Cross-Site Request Forgery)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Encapsulation | | |
| Abstract: | registration.html 26 . | | |
| Sink: | registration.html:26 | | |

```
24                      <fieldset>
25                          <legend th:text="#{register.title}">Please Sign Up</legend>
26                          <form class="form-horizontal" action="#" th:action="@{/register.mvc}"
th:object="${userForm}"
27                                method='POST'>
```

## 49: 01.08. XML (3 Issues)




**Abstract:**

CommentsCache.java:102 XML . XML .

**Explanation:**

XML XML . XML . XML URI . XML URI (: ) . XML (XXE) , denial of service , , , denial of service .

XML XXE .

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE foo [
<!ELEMENT foo ANY >
<!ENTITY xxe SYSTEM "file:///dev/random" >]><foo>&xxe;</foo>
```

XML /dev/random (UNIX ).

**Recommendations:**

XML unmarshaller XML .

XXE injection XML java.io.File, java.io.Reader java.io.InputStream unmarshal . XML unmarshal .

```
DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
dbf.setFeature("http://apache.org/xml/features/disallow-doctype-decl", true);
DocumentBuilder db = dbf.newDocumentBuilder();
Document document = db.parse(<XML Source>);
Model model = (Model) u.unmarshal(document);
```

StAX .

- :

```
xmlInputFactory.setProperty("javax.xml.stream.isSupportingExternalEntities", false);
```

- DOCTYPE .

```
xmlInputFactory.setProperty("javax.xml.stream.supportDTD", false);
```

- (resolver) XML .

```
xmlInputFactory.setXMLResolver(mySafeResolver);
```

### CommentsCache.java, line 102 (XML External Entity Injection)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | CommentsCache.java:102 XML . XML . | | |

**Source:** SimpleXXE.java:68 createNewComment(1)

```
66    @PostMapping(path = "xxe/simple", consumes = ALL_VALUE, produces =
APPLICATION_JSON_VALUE)
67    @ResponseBody
```

```
68                  public AttackResult createNewComment(HttpServletRequest request, @RequestBody
String commentStr) {
69                      String error = "";
70                      try {
```

Sink: CommentsCache.java:102 javax.xml.stream.XMLInputFactory.createXMLStreamReader()

```
100                     }
101
102                     var xsr = xif.createXMLStreamReader(new StringReader(xml));
103
104                     var unmarshaller = jc.createUnmarshaller();
```

## CommentsCache.java, line 102 (XML External Entity Injection)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | CommentsCache.java:102 XML . XML . | | |

Source: BlindSendFileAssignment.java:80 addComment(0)

```
78                  @PostMapping(path = "xxe/blind", consumes = ALL_VALUE, produces =
APPLICATION_JSON_VALUE)
79                  @ResponseBody
80                  public AttackResult addComment(@RequestBody String commentStr) {
81                      var fileContentsForUser =
userToFileContents.getOrDefault(getWebSession().getUser(), "");
```

Sink: CommentsCache.java:102 javax.xml.stream.XMLInputFactory.createXMLStreamReader()

```
100                     }
101
102                     var xsr = xif.createXMLStreamReader(new StringReader(xml));
103
104                     var unmarshaller = jc.createUnmarshaller();
```

## CommentsCache.java, line 102 (XML External Entity Injection)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | CommentsCache.java:102 XML . XML . | | |

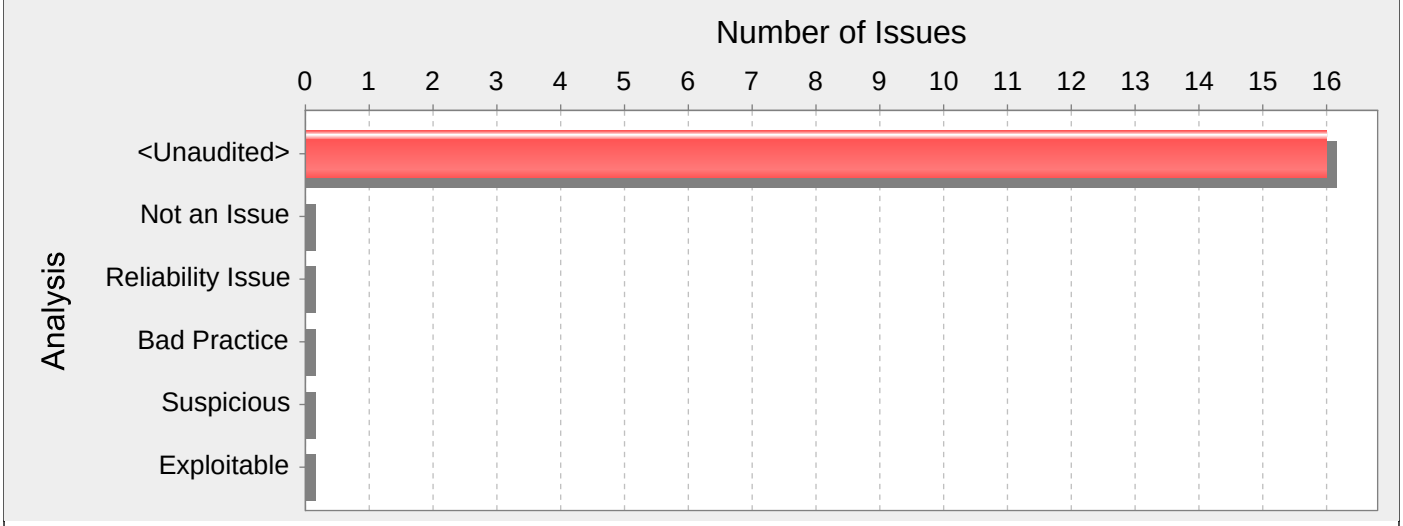
Source: ContentTypeAssignment.java:60 createNewUser(1)

```
58                  @PostMapping(path = "xxe/content-type")
59                  @ResponseBody
60                  public AttackResult createNewUser(HttpServletRequest request, @RequestBody String
commentStr, @RequestHeader("Content-Type") String contentType) throws Exception {
61                      AttackResult attackResult = failed(this).build();
```

Sink: CommentsCache.java:102 javax.xml.stream.XMLInputFactory.createXMLStreamReader()

```
100                     }
101
102                     var xsr = xif.createXMLStreamReader(new StringReader(xml));
103
104                     var unmarshaller = jc.createUnmarshaller();
```

## 49: 01.07. URL (16 Issues)




### Abstract:

ProfileUploadRetrieval.java 96 HTTP . URL .

### Explanation:

. , . URL open redirection .

Open Redirection URL . URL URL URL . open redirection .

1: JSP dest URL .

```
<%
...
String strDest = request.getParameter("dest");
pageContext.forward(strDest);
...
%>
```

“http://trusted.example.com/ecommerce/redirect.asp?dest=www.wilyhacker.com” , . Example 1 “http://www.wilyhacker.com” .

URL . URL "http://trusted.example.com/ecommerce/redirect.asp?dest=%77%69%6C%79%68%61%63%6B%65%72%2E%63%6F%6D" .

### Recommendations:

URL . . , URL . URL .

2: URL . URL .

```
<%
...
try {
int strDest = Integer.parseInt(request.getParameter("dest"));
if((strDest >= 0) && (strDest <= strURLArray.length -1 ))
{
strFinalURL = strURLArray[strDest];
pageContext.forward(strFinalURL);
}
}
catch (NumberFormatException nfe) {
// Handle exception
...
}
...
```

ot Copyright 2024 Open Text.

%>

, URL . , .

**Tips:**

1. (Struts Spring MVC ). , Fortify Fortify Static Code Analyzer . Context-Sensitive Ranking( ) . Fortify , Fortify Software Security Research Group .

## path_traversal.js, line 60 (Open Redirect)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | path_traversal.js 60 HTTP . URL . | | |

**Source:** path_traversal.js:59 lambda(0)

```
57
58                  function newRandomPicture() {
59                      $.get("PathTraversal/random-picture", function (result, status) {
60                          document.getElementById("randomCatPicture").src = "" +
                    result;
61                      });
```

**Sink:** path_traversal.js:60 Assignment to src()

```
58                  function newRandomPicture() {
59                      $.get("PathTraversal/random-picture", function (result, status) {
60                          document.getElementById("randomCatPicture").src = "" +
                    result;
61                      });
62                  }
```

## path_traversal.js, line 54 (Open Redirect)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | path_traversal.js 54 HTTP . URL . | | |

**Source:** path_traversal.js:53 lambda(0)

```
51
52                  webgoat.customjs.profileUploadCallbackRetrieval = function () {
53                      $.get("PathTraversal/profile-picture", function (result, status) {
54                          document.getElementById("previewRetrieval").src = "" +
                    result;
55                      });
```

**Sink:** path_traversal.js:54 Assignment to src()

```
52                  webgoat.customjs.profileUploadCallbackRetrieval = function () {
53                      $.get("PathTraversal/profile-picture", function (result, status) {
54                          document.getElementById("previewRetrieval").src = "" +
                    result;
55                      });
56                  }
```

## path_traversal.js, line 30 (Open Redirect)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | path_traversal.js 30 HTTP . URL . | | |

**Source:** path_traversal.js:29 lambda(0)

```
27
28                  webgoat.customjs.profileUploadCallbackFix = function () {
29                      $.get("PathTraversal/profile-picture", function (result, status) {
30                          document.getElementById("previewFix").src = "" + result;
31                      });
```

**Sink:** path_traversal.js:30 Assignment to src()

```
28                  webgoat.customjs.profileUploadCallbackFix = function () {
29                      $.get("PathTraversal/profile-picture", function (result, status) {
30                          document.getElementById("previewFix").src = "" + result;
31                      });
```

ot Copyright 2024 Open Text.

```
32                  }
```

## path_traversal.js, line 47 (Open Redirect)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | path_traversal.js 47 HTTP . URL . | | |

Source: path_traversal.js:46 lambda(0)

```
44
45                  webgoat.customjs.profileUploadCallbackRemoveUserInput = function () {
46                      $.get("PathTraversal/profile-picture", function (result, status) {
47                          document.getElementById("previewRemoveUserInput").src =
"" + result;
48                      });
```

Sink: path_traversal.js:47 Assignment to src()

```
45                  webgoat.customjs.profileUploadCallbackRemoveUserInput = function () {
46                      $.get("PathTraversal/profile-picture", function (result, status) {
47                          document.getElementById("previewRemoveUserInput").src =
"" + result;
48                      });
49                  }
```

## path_traversal.js, line 60 (Open Redirect)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | path_traversal.js 60 HTTP . URL . | | |

Source: path_traversal.js:59 lambda(0)

```
57
58                  function newRandomPicture() {
59                      $.get("PathTraversal/random-picture", function (result, status) {
60                          document.getElementById("randomCatPicture").src = "" +
result;
61                      });
```

Sink: path_traversal.js:60 Assignment to src()

```
58                  function newRandomPicture() {
59                      $.get("PathTraversal/random-picture", function (result, status) {
60                          document.getElementById("randomCatPicture").src = "" +
result;
61                      });
62                  }
```

## path_traversal.js, line 14 (Open Redirect)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | path_traversal.js 14 HTTP . URL . | | |

Source: path_traversal.js:13 lambda(0)

```
11
12                  webgoat.customjs.profileUploadCallback = function () {
13                      $.get("PathTraversal/profile-picture", function (result, status) {
14                          document.getElementById("preview").src = "" + result;
15                      });
```

Sink: path_traversal.js:14 Assignment to src()

```
12                  webgoat.customjs.profileUploadCallback = function () {
13                      $.get("PathTraversal/profile-picture", function (result, status) {
14                          document.getElementById("preview").src = "" + result;
15                      });
16                  }
```

## path_traversal.js, line 76 (Open Redirect)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Input Validation and Representation | | |

ot Copyright 2024 Open Text.

| | |
|---|---|
| **Abstract:** | path_traversal.js 76 HTTP . URL . |
| **Source:** | path_traversal.js:75 lambda(0) |

```
73
74                webgoat.customjs.profileZipSlipRetrieval = function () {
75                    $.get("PathTraversal/zip-slip", function (result, status) {
76                        document.getElementById("previewZipSlip").src = "" +
result;
77                    });
```

| | |
|---|---|
| **Sink:** | path_traversal.js:76 Assignment to src() |

```
74                webgoat.customjs.profileZipSlipRetrieval = function () {
75                    $.get("PathTraversal/zip-slip", function (result, status) {
76                        document.getElementById("previewZipSlip").src = "" +
result;
77                    });
78                }
```

## ProfileUploadRetrieval.java, line 96 (Open Redirect)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | ProfileUploadRetrieval.java 96 HTTP . URL . | | |
| **Source:** | ProfileUploadRetrieval.java:85 javax.servlet.ServletRequest.getParameter() | | |

```
83                }
84                try {
85                    var id = request.getParameter("id");
86                    var catPicture = new File(catPicturesDirectory, (id == null ?
RandomUtils.nextInt(1, 11) : id) + ".jpg");
```

| | |
|---|---|
| **Sink:** | ProfileUploadRetrieval.java:96 org.springframework.http.ResponseEntity.HeadersBuilder.location() |

```
94                    return ResponseEntity.ok()
95
.contentType(MediaType.parseMediaType(MediaType.IMAGE_JPEG_VALUE))
96                            .location(new URI("/PathTraversal/random-picture?id=" +
catPicture.getName()))
97
.body(Base64.getEncoder().encode(FileCopyUtils.copyToByteArray(catPicture)));
98                }
```

## ProfileUploadRetrieval.java, line 96 (Open Redirect)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | ProfileUploadRetrieval.java 96 HTTP . URL . | | |
| **Source:** | ProfileUploadRetrieval.java:46 ProfileUploadRetrieval(0) | | |

```
44    private final File catPicturesDirectory;
45
46    public ProfileUploadRetrieval(@Value("${webgoat.server.directory}") String
webGoatHomeDirectory) {
47        this.catPicturesDirectory = new File(webGoatHomeDirectory, "/PathTraversal/" +
"/cats");
48        this.catPicturesDirectory.mkdirs();
```

| | |
|---|---|
| **Sink:** | ProfileUploadRetrieval.java:96 org.springframework.http.ResponseEntity.HeadersBuilder.location() |

```
94                    return ResponseEntity.ok()
95
.contentType(MediaType.parseMediaType(MediaType.IMAGE_JPEG_VALUE))
96                            .location(new URI("/PathTraversal/random-picture?id=" +
catPicture.getName()))
97
.body(Base64.getEncoder().encode(FileCopyUtils.copyToByteArray(catPicture)));
98                }
```

## path_traversal.js, line 47 (Open Redirect)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Input Validation and Representation | | |

| | |
|---|---|
| **Abstract:** | path_traversal.js 47 HTTP . URL . |
| **Source:** | path_traversal.js:46 lambda(0) |

```
44
45  webgoat.customjs.profileUploadCallbackRemoveUserInput = function () {
46      $.get("PathTraversal/profile-picture", function (result, status) {
47          document.getElementById("previewRemoveUserInput").src =
"" + result;
48      });
```

| | |
|---|---|
| **Sink:** | path_traversal.js:47 Assignment to src() |

```
45  webgoat.customjs.profileUploadCallbackRemoveUserInput = function () {
46      $.get("PathTraversal/profile-picture", function (result, status) {
47          document.getElementById("previewRemoveUserInput").src =
"" + result;
48      });
49  }
```

## path_traversal.js, line 14 (Open Redirect)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | path_traversal.js 14 HTTP . URL . | | |
| **Source:** | path_traversal.js:13 lambda(0) | | |

```
11
12  webgoat.customjs.profileUploadCallback = function () {
13      $.get("PathTraversal/profile-picture", function (result, status) {
14          document.getElementById("preview").src = "" + result;
15      });
```

| | |
|---|---|
| **Sink:** | path_traversal.js:14 Assignment to src() |

```
12  webgoat.customjs.profileUploadCallback = function () {
13      $.get("PathTraversal/profile-picture", function (result, status) {
14          document.getElementById("preview").src = "" + result;
15      });
16  }
```

## path_traversal.js, line 30 (Open Redirect)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | path_traversal.js 30 HTTP . URL . | | |
| **Source:** | path_traversal.js:29 lambda(0) | | |

```
27
28  webgoat.customjs.profileUploadCallbackFix = function () {
29      $.get("PathTraversal/profile-picture", function (result, status) {
30          document.getElementById("previewFix").src = "" + result;
31      });
```

| | |
|---|---|
| **Sink:** | path_traversal.js:30 Assignment to src() |

```
28  webgoat.customjs.profileUploadCallbackFix = function () {
29      $.get("PathTraversal/profile-picture", function (result, status) {
30          document.getElementById("previewFix").src = "" + result;
31      });
32  }
```

## ProfileUploadRetrieval.java, line 100 (Open Redirect)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | ProfileUploadRetrieval.java 100 HTTP . URL . | | |
| **Source:** | ProfileUploadRetrieval.java:46 ProfileUploadRetrieval(0) | | |

```
44      private final File catPicturesDirectory;
45
46      public ProfileUploadRetrieval(@Value("${webgoat.server.directory}") String
webGoatHomeDirectory) {
```

```
47                      this.catPicturesDirectory = new File(webGoatHomeDirectory, "/PathTraversal/" +
"/cats");
48                      this.catPicturesDirectory.mkdirs();
```

Sink: ProfileUploadRetrieval.java:100 org.springframework.http.ResponseEntity.HeadersBuilder.location()

```
98                          }
99                          return ResponseEntity.status(HttpStatus.NOT_FOUND)
100                                 .location(new URI("/PathTraversal/random-picture?id=" +
catPicture.getName()))
101
.body(StringUtils.arrayToCommaDelimitedString(catPicture.getParentFile().listFiles()).
getBytes());
102                     } catch (IOException | URISyntaxException e) {
```

## path_traversal.js, line 54 (Open Redirect)

| Fortify Priority: | Critical | Folder | Critical |
|---|---|---|---|
| Kingdom: | Input Validation and Representation | | |

Abstract: path_traversal.js 54 HTTP . URL .

Source: path_traversal.js:53 lambda(0)

```
51
52  webgoat.customjs.profileUploadCallbackRetrieval = function () {
53      $.get("PathTraversal/profile-picture", function (result, status) {
54          document.getElementById("previewRetrieval").src = "" +
result;
55      });
```

Sink: path_traversal.js:54 Assignment to src()

```
52  webgoat.customjs.profileUploadCallbackRetrieval = function () {
53      $.get("PathTraversal/profile-picture", function (result, status) {
54          document.getElementById("previewRetrieval").src = "" +
result;
55      });
56  }
```

## path_traversal.js, line 76 (Open Redirect)

| Fortify Priority: | Critical | Folder | Critical |
|---|---|---|---|
| Kingdom: | Input Validation and Representation | | |

Abstract: path_traversal.js 76 HTTP . URL .

Source: path_traversal.js:75 lambda(0)

```
73
74  webgoat.customjs.profileZipSlipRetrieval = function () {
75      $.get("PathTraversal/zip-slip", function (result, status) {
76          document.getElementById("previewZipSlip").src = "" +
result;
77      });
```

Sink: path_traversal.js:76 Assignment to src()

```
74  webgoat.customjs.profileZipSlipRetrieval = function () {
75      $.get("PathTraversal/zip-slip", function (result, status) {
76          document.getElementById("previewZipSlip").src = "" +
result;
77      });
78  }
```

## ProfileUploadRetrieval.java, line 100 (Open Redirect)

| Fortify Priority: | Critical | Folder | Critical |
|---|---|---|---|
| Kingdom: | Input Validation and Representation | | |

Abstract: ProfileUploadRetrieval.java 100 HTTP . URL .

Source: ProfileUploadRetrieval.java:85 javax.servlet.ServletRequest.getParameter()

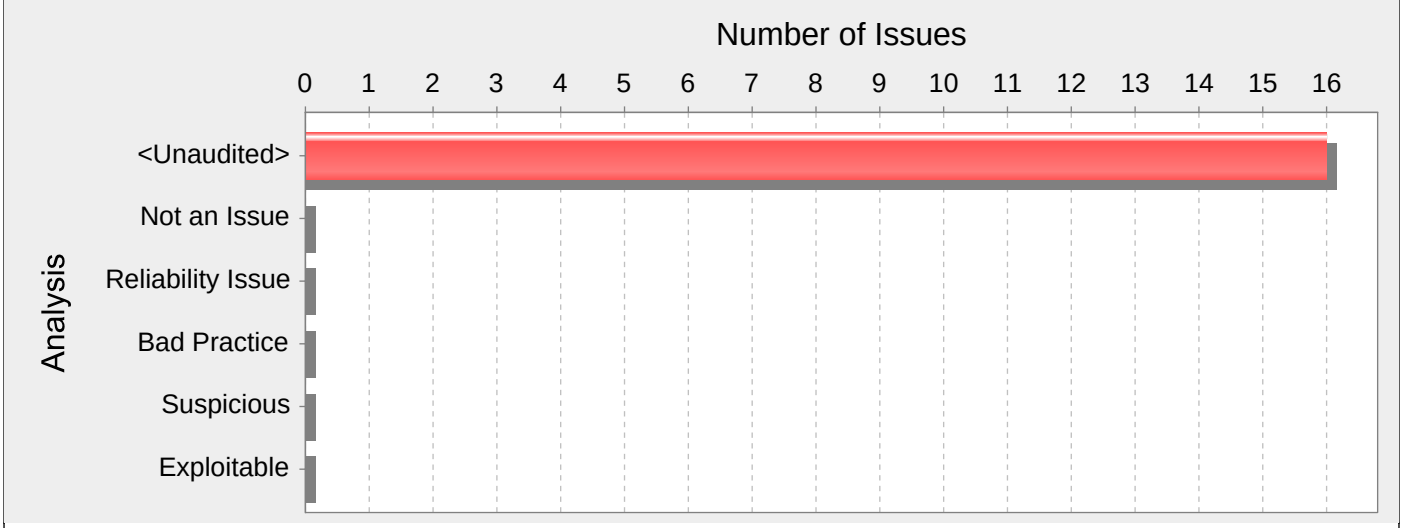
```
83                      }
84                      try {
85                          var id = request.getParameter("id");
86                          var catPicture = new File(catPicturesDirectory, (id == null ?
RandomUtils.nextInt(1, 11) : id) + ".jpg");
```

ot Copyright 2024 Open Text.

| Sink: | ProfileUploadRetrieval.java:100<br>org.springframework.http.ResponseEntity.HeadersBuilder.location() |
|---|---|

```
98                      }
99                      return ResponseEntity.status(HttpStatus.NOT_FOUND)
100                             .location(new URI("/PathTraversal/random-picture?id=" +
catPicture.getName()))
101
.body(StringUtils.arrayToCommaDelimitedString(catPicture.getParentFile().listFiles()).
getBytes());
102                 } catch (IOException | URISyntaxException e) {
```

## 49: 01.06. (16 Issues)




**Abstract:**
ProfileUpload.java 28 org.springframework.web.multipart.MultipartFile Spring MVC . .

**Explanation:**
. (: JSP/ASPX/PHP) .
: Spring MVC .

```
@Controller
public class MyFormController {
...
@RequestMapping("/test")
public String uploadFile (org.springframework.web.multipart.MultipartFile file) {
...
}  ...
}
```

. path manipulation, command injection dangerous file inclusion , .

**Recommendations:**
. , . . . , . . , , .

### files.html, line 40 (Often Misused: File Upload)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Medium | **Folder** | Medium |
| **Kingdom:** | API Abuse | | |

**Abstract:** .

**Sink:** files.html:40

```
38                          <div class="form-inline">
39                              <div class="form-group">
40                                  <input type="file" name="file"/>
41                              </div>
42                              <button type="submit" class="btn btn-md btn-primary">Upload
files</button>
```

### PathTraversal.html, line 140 (Often Misused: File Upload)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Medium | **Folder** | Medium |
| **Kingdom:** | API Abuse | | |

**Abstract:** .

**Sink:** PathTraversal.html:140

```
138                         <div class="browse-button">
139                             <i class="fa fa-pencil"></i>
140                             <input class="browse-input" type="file" required
name="uploadedFile"
141                                     id="uploadedFileRemoveUserInput"/>
```

ot Copyright 2024 Open Text.

```
142                                 </div>
```

## FileServer.java, line 72 (Often Misused: File Upload)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Medium | **Folder** | Medium |
| **Kingdom:** | API Abuse | | |
| **Abstract:** | FileServer.java 72 org.springframework.web.multipart.MultipartFile Spring MVC . . | | |
| **Sink:** | FileServer.java:72 Function: importFile() | | |

```
70
71                      @PostMapping(value = "/fileupload")
72                      public ModelAndView importFile(@RequestParam("file") MultipartFile myFile) throws
IOException {
73                          var user = (WebGoatUser)
SecurityContextHolder.getContext().getAuthentication().getPrincipal();
74                          var destinationDir = new File(fileLocation, user.getUsername());
```

## PathTraversal.html, line 85 (Often Misused: File Upload)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Medium | **Folder** | Medium |
| **Kingdom:** | API Abuse | | |
| **Abstract:** | . | | |
| **Sink:** | PathTraversal.html:85 | | |

```
83                              <div class="browse-button">
84                                  <i class="fa fa-pencil"></i>
85                                  <input class="browse-input" type="file" required
name="uploadedFile" id="uploadedFileFix"/>
86                              </div>
87                              <span class="Error"></span>
```

## ProfileUploadRemoveUserInput.java, line 26 (Often Misused: File Upload)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Medium | **Folder** | Medium |
| **Kingdom:** | API Abuse | | |
| **Abstract:** | ProfileUploadRemoveUserInput.java 26 org.springframework.web.multipart.MultipartFile Spring MVC . . | | |
| **Sink:** | ProfileUploadRemoveUserInput.java:26 Function: uploadFileHandler() | | |

```
24                      @PostMapping(value = "/PathTraversal/profile-upload-remove-user-input", consumes =
ALL_VALUE, produces = APPLICATION_JSON_VALUE)
25                      @ResponseBody
26                      public AttackResult uploadFileHandler(@RequestParam("uploadedFileRemoveUserInput")
MultipartFile file) {
27                          return super.execute(file, file.getOriginalFilename());
28                      }
```

## PathTraversal.html, line 238 (Often Misused: File Upload)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Medium | **Folder** | Medium |
| **Kingdom:** | API Abuse | | |
| **Abstract:** | . | | |
| **Sink:** | PathTraversal.html:238 | | |

```
236                             <div class="browse-button">
237                                 <i class="fa fa-pencil"></i>
238                                 <input class="browse-input" type="file" required
name="uploadedFile"
239                                         id="uploadedFileZipSlip"/>
240                             </div>
```

## PathTraversal.html, line 31 (Often Misused: File Upload)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Medium | **Folder** | Medium |
| **Kingdom:** | API Abuse | | |
| **Abstract:** | . | | |
| **Sink:** | PathTraversal.html:31 | | |

ot Copyright 2024 Open Text.

```
29                                  <div class="browse-button">
30                                      <i class="fa fa-pencil"></i>
31                                      <input class="browse-input" type="file" required
name="uploadedFile" id="uploadedFile"/>
32                                  </div>
33                                  <span class="Error"></span>
```

**ProfileZipSlip.java, line 45 (Often Misused: File Upload)**

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Medium | **Folder** | Medium |
| **Kingdom:** | API Abuse | | |

**Abstract:** ProfileZipSlip.java 45 org.springframework.web.multipart.MultipartFile Spring MVC . .

**Sink:** ProfileZipSlip.java:45 Function: processZipUpload()

```
43
44      @SneakyThrows
45      private AttackResult processZipUpload(MultipartFile file) {
46          var tmpZipDirectory =
Files.createTempDirectory(getWebSession().getUserName());
47          var uploadDirectory = new File(getWebGoatHomeDirectory(), "/PathTraversal/" +
getWebSession().getUserName());
```

**PathTraversal.html, line 31 (Often Misused: File Upload)**

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Medium | **Folder** | Medium |
| **Kingdom:** | API Abuse | | |

**Abstract:** .

**Sink:** PathTraversal.html:31

```
29                                  <div class="browse-button">
30                                      <i class="fa fa-pencil"></i>
31                                      <input class="browse-input" type="file" required
name="uploadedFile" id="uploadedFile"/>
32                                  </div>
33                                  <span class="Error"></span>
```

**ProfileUploadFix.java, line 28 (Often Misused: File Upload)**

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Medium | **Folder** | Medium |
| **Kingdom:** | API Abuse | | |

**Abstract:** ProfileUploadFix.java 28 org.springframework.web.multipart.MultipartFile Spring MVC . .

**Sink:** ProfileUploadFix.java:28 Function: uploadFileHandler()

```
26      @PostMapping(value = "/PathTraversal/profile-upload-fix", consumes = ALL_VALUE,
produces = APPLICATION_JSON_VALUE)
27      @ResponseBody
28      public AttackResult uploadFileHandler(
29              @RequestParam("uploadedFileFix") MultipartFile file,
30              @RequestParam(value = "fullNameFix", required = false) String fullName) {
```

**PathTraversal.html, line 140 (Often Misused: File Upload)**

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Medium | **Folder** | Medium |
| **Kingdom:** | API Abuse | | |

**Abstract:** .

**Sink:** PathTraversal.html:140

```
138                                 <div class="browse-button">
139                                     <i class="fa fa-pencil"></i>
140                                     <input class="browse-input" type="file" required
name="uploadedFile"
141                                             id="uploadedFileRemoveUserInput"/>
142                                 </div>
```

**ProfileZipSlip.java, line 36 (Often Misused: File Upload)**

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Medium | **Folder** | Medium |

ot Copyright 2024 Open Text.

| Kingdom: | API Abuse |
|---|---|
| Abstract: | ProfileZipSlip.java 36 org.springframework.web.multipart.MultipartFile Spring MVC . . |

Sink: ProfileZipSlip.java:36 Function: uploadFileHandler()

```
34      @PostMapping(value = "/PathTraversal/zip-slip", consumes = ALL_VALUE, produces = APPLICATION_JSON_VALUE)
35      @ResponseBody
36      public AttackResult uploadFileHandler(@RequestParam("uploadedFileZipSlip") MultipartFile file) {
37          if (!file.getOriginalFilename().toLowerCase().endsWith(".zip")) {
38              return failed(this).feedback("path-traversal-zip-slip.no-zip").build();
```

## PathTraversal.html, line 238 (Often Misused: File Upload)

| Fortify Priority: | Medium | Folder | Medium |
|---|---|---|---|
| Kingdom: | API Abuse | | |
| Abstract: | . | | |

Sink: PathTraversal.html:238

```
236                         <div class="browse-button">
237                             <i class="fa fa-pencil"></i>
238                             <input class="browse-input" type="file" required name="uploadedFile"
239                                    id="uploadedFileZipSlip"/>
240                         </div>
```

## ProfileUpload.java, line 28 (Often Misused: File Upload)

| Fortify Priority: | Medium | Folder | Medium |
|---|---|---|---|
| Kingdom: | API Abuse | | |
| Abstract: | ProfileUpload.java 28 org.springframework.web.multipart.MultipartFile Spring MVC . . | | |

Sink: ProfileUpload.java:28 Function: uploadFileHandler()

```
26      @PostMapping(value = "/PathTraversal/profile-upload", consumes = ALL_VALUE, produces = APPLICATION_JSON_VALUE)
27      @ResponseBody
28      public AttackResult uploadFileHandler(@RequestParam("uploadedFile") MultipartFile file, @RequestParam(value = "fullName", required = false) String fullName) {
29          return super.execute(file, fullName);
30      }
```

## PathTraversal.html, line 85 (Often Misused: File Upload)

| Fortify Priority: | Medium | Folder | Medium |
|---|---|---|---|
| Kingdom: | API Abuse | | |
| Abstract: | . | | |

Sink: PathTraversal.html:85

```
83                          <div class="browse-button">
84                              <i class="fa fa-pencil"></i>
85                              <input class="browse-input" type="file" required name="uploadedFile" id="uploadedFileFix"/>
86                          </div>
87                          <span class="Error"></span>
```

## files.html, line 40 (Often Misused: File Upload)

| Fortify Priority: | Medium | Folder | Medium |
|---|---|---|---|
| Kingdom: | API Abuse | | |
| Abstract: | . | | |

Sink: files.html:40

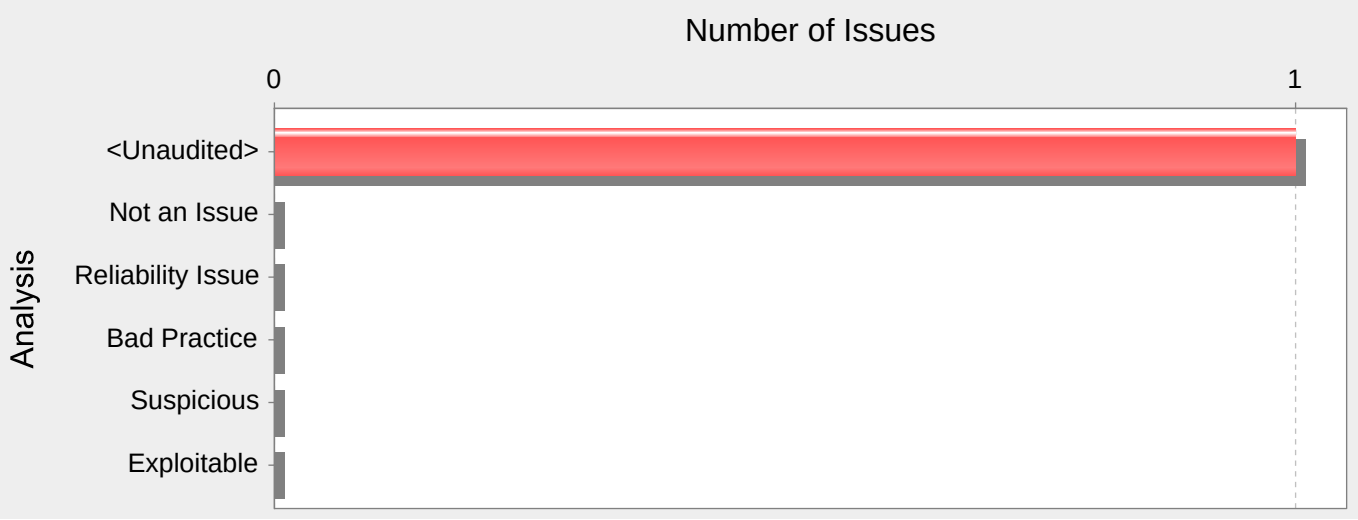
```
38                      <div class="form-inline">
39                          <div class="form-group">
40                              <input type="file" name="file"/>
41                          </div>
```

Copyright 2024 Open Text.

```
42                                  <button type="submit" class="btn btn-md btn-primary">Upload
                    files</button>
```

ot Copyright 2024 Open Text.

## 49: 01.05. (1 Issues)




**Abstract:**
VulnerableTaskHolder.java readObject() exec()() . .

**Explanation:**
Command injection .

- . .
- . .

, . command injection .

1. .
2. .
3. .

: . /var/yp make .

...
System.Runtime.getRuntime().exec("make");
...

make Runtime.exec() . $PATH make . . , make .

**Recommendations:**
. . . , . .

. , (world-writable) . ID . , .

. . .

**Tips:**
1. Windows Windows . . COMMAND.COM . BREAK, CALL, CHCP, CHDIR(CD), CLS, COPY, CTTY, DATE, DEL(ERASE), DIR, ECHO, EXIT, FOR, GOTO, IF, MKDIR(MD), PATH, PAUSE, PROMPT, REM, RENAME(REN), RMDIR(RD), SET, SHIFT, TIME, TYPE, VER, VERIFY, VOL . .

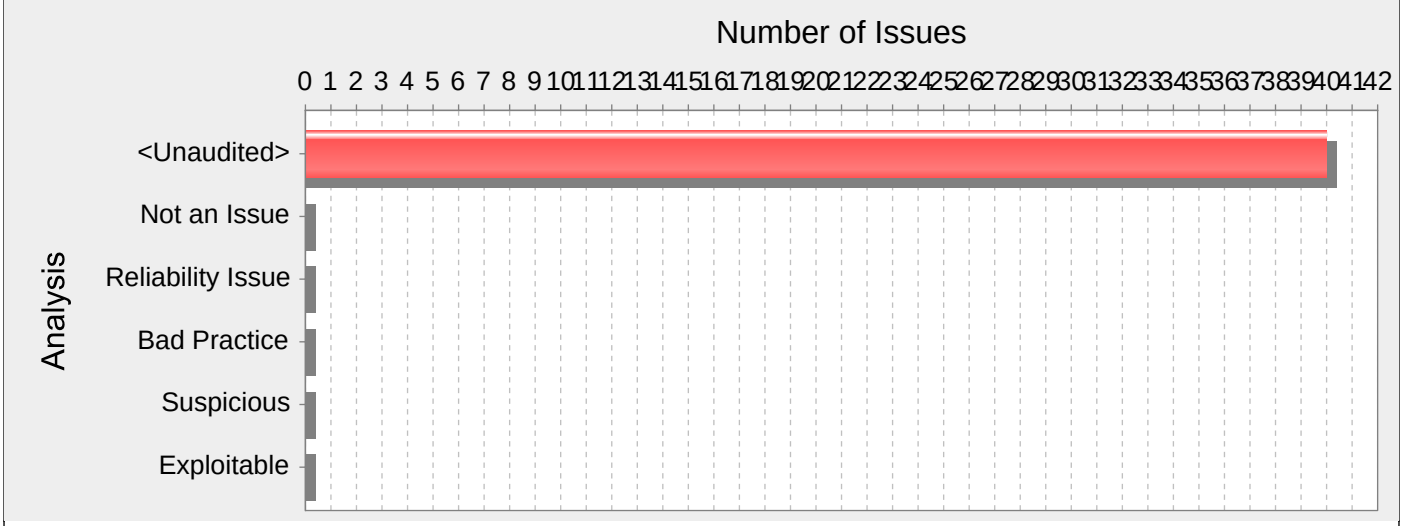
### VulnerableTaskHolder.java, line 60 (Command Injection)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | VulnerableTaskHolder.java readObject() exec()() . . | | |
| **Sink:** | VulnerableTaskHolder.java:60 exec(0) | | |

```
58                log.info("about to execute: {}", taskAction);
59                try {
60                        Process p = Runtime.getRuntime().exec(taskAction);
61                        BufferedReader in = new BufferedReader(
62                                        new InputStreamReader(p.getInputStream()));
```

Copyright 2024 Open Text.

## 49: 01.04. (40 Issues)




**Abstract:**

MissingFunctionACUsers.java usersService() 75 .

**Explanation:**

XSS(Cross-site scripting) .

1. . Persistent(Stored ) XSS , Reflected XSS .

2. .

JavaScript HTML, Flash . XSS , .

1: JSP ID .

```
<%...
Statement stmt = conn.createStatement();
ResultSet rs = stmt.executeQuery("select * from emp where id="+eid);
if (rs != null) {
rs.next();
String name = rs.getString("name");
}
%>
Employee Name: <%= name %>
```

name . name . name . . Persistent( Stored) XSS . XSS "" . JavaScript .

2: JSP HTTP ID eid .

```
<% String eid = request.getParameter("eid"); %>
...
Employee ID: <%= eid %>
```

Example 1 eid . eid , HTTP .

. URL ? URL URL . . Reflected XSS .

Cross-Site Scripting . . . .

3: Android WebView JavaScript ( JavaScript ) Android .

```
...
WebView webview = (WebView) findViewById(R.id.webview);
webview.getSettings().setJavaScriptEnabled(true);
String url = this.getIntent().getExtras().getString("url");
webview.loadUrl(url);
```

ot Copyright 2024 Open Text.

...

url javascript: JavaScript WebView .

, XSS HTTP . XSS .

- Example 1 . . Persistent XSS . . . .

- Example 2 HTTP HTTP . XSS . . URL . URL URL . , .

- Example 3 .

(Struts Spring MVC ). , Fortify Fortify Static Code Analyzer . Context-Sensitive Ranking( ) . Fortify , Fortify Software Security Research Group .

**Recommendations:**

XSS .

XSS . ( ) . XSS .

SQL injection . XSS . XSS . . . , XSS .

XSS HTTP . , 0-9 . . HTML .

. . . HTML HTML . XSS . SEI(Software Engineering Institute) CERT(R) Coordination Center [1].

Block-level element ( ):

- "<" .

- "&" .

- ">" "<" .

.

- .

- .

- .

- "&" .

, URL . URL .

- , URL .

- "&" CGI .

- ASCII (, ISO-8859-1 127 ) URL .

- "%" HTTP . , "%" "%68%65%6C%6C%6F" "hello" .

<SCRIPT> </SCRIPT> :

- , , .

:

- (!) (") .

:

- UTF-7 "<" '+ADw-' . ( , UTF-7) .

XSS . , . . .

, . ISO 8859-1 HTML [2].

Cross-Site Scripting HTTP Cross-Site Scripting . . . .

**Tips:**

1. Fortify Secure Coding Rulepacks SQL Injection , XSS . , DATABASE . .

2. URL XSS , JavaScript DOM(Document Object Model) . Rulepacks Cross-Site Scripting URL . URL Fortify Cross-Site Scripting: Poor Validation .

## stored-xss.js, line 40 (Cross-Site Scripting: DOM)

**Fortify Priority:** Critical **Folder** Critical

| Kingdom: | Input Validation and Representation |
|---|---|
| Abstract: | stored-xss.js lambda() 40 . |
| Source: | stored-xss.js:35 lambda(0) |

```
33                  function getChallenges() {
34                      $("#list").empty();
35                      $.get('CrossSiteScripting/stored-xss', function (result, status) {
36                          for (var i = 0; i < result.length; i++) {
37                              var comment = html.replace('USER', result[i].user);
```

| Sink: | stored-xss.js:40 ~JS_Generic.append() |
|---|---|

```
38                              comment = comment.replace('DATETIME', result[i].dateTime);
39                              comment = comment.replace('COMMENT', result[i].text);
40                              $("#list").append(comment);
41                          }
```

## stored-xss.js, line 40 (Cross-Site Scripting: DOM)

| Fortify Priority: | Critical | Folder | Critical |
|---|---|---|---|
| Kingdom: | Input Validation and Representation | | |
| Abstract: | stored-xss.js lambda() 40 . | | |
| Source: | stored-xss.js:35 lambda(0) | | |

```
33                  function getChallenges() {
34                      $("#list").empty();
35                      $.get('CrossSiteScripting/stored-xss', function (result, status) {
36                          for (var i = 0; i < result.length; i++) {
37                              var comment = html.replace('USER', result[i].user);
```

| Sink: | stored-xss.js:40 ~JS_Generic.append() |
|---|---|

```
38                              comment = comment.replace('DATETIME', result[i].dateTime);
39                              comment = comment.replace('COMMENT', result[i].text);
40                              $("#list").append(comment);
41                          }
```

## clientSideFilteringFree.js, line 41 (Cross-Site Scripting: Self)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Input Validation and Representation | | |
| Abstract: | clientSideFilteringFree.js lambda() 41 . | | |
| Source: | clientSideFilteringFree.js:40 ~JS_Generic.val() | | |

```
38      })
39      $(".checkoutCode").on("blur", function () {
40          var checkoutCode = $(".checkoutCode").val();
41          $.get("clientSideFiltering/challenge-store/coupons/" + checkoutCode, function
(result, status) {
42              var discount = result.discount;
```

| Sink: | clientSideFilteringFree.js:41 ~JS_Generic.get() |
|---|---|

```
39      $(".checkoutCode").on("blur", function () {
40          var checkoutCode = $(".checkoutCode").val();
41          $.get("clientSideFiltering/challenge-store/coupons/" + checkoutCode, function
(result, status) {
42              var discount = result.discount;
43              if (discount > 0) {
```

## GoatUtils.js, line 57 (Cross-Site Scripting: DOM)

| Fortify Priority: | Critical | Folder | Critical |
|---|---|---|---|
| Kingdom: | Input Validation and Representation | | |
| Abstract: | GoatUtils.js lambda() 57 . | | |
| Source: | GoatUtils.js:56 lambda(0) | | |

```
54
55                      showLessonCookiesAndParams: function() {
56                          $.get(goatConstants.cookieService, {}, function(reply) {
57                              $("#lesson_cookies").html(reply);
58                          }, "html");
```

| | |
|---|---|
| **Sink:** | GoatUtils.js:57 ~JS_Generic.html() |

```
55                          showLessonCookiesAndParams: function() {
56                              $.get(goatConstants.cookieService, {}, function(reply) {
57                                  $("#lesson_cookies").html(reply);
58                              }, "html");
59                          },
```

## path_traversal.js, line 47 (Cross-Site Scripting: DOM)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | path_traversal.js lambda() 47 . | | |
| **Source:** | path_traversal.js:46 lambda(0) | | |

```
44
45              webgoat.customjs.profileUploadCallbackRemoveUserInput = function () {
46                  $.get("PathTraversal/profile-picture", function (result, status) {
47                      document.getElementById("previewRemoveUserInput").src =
"" + result;
48                  });
```

| | |
|---|---|
| **Sink:** | path_traversal.js:47 Assignment to src() |

```
45              webgoat.customjs.profileUploadCallbackRemoveUserInput = function () {
46                  $.get("PathTraversal/profile-picture", function (result, status) {
47                      document.getElementById("previewRemoveUserInput").src =
"" + result;
48                  });
49              }
```

## path_traversal.js, line 76 (Cross-Site Scripting: DOM)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | path_traversal.js lambda() 76 . | | |
| **Source:** | path_traversal.js:75 lambda(0) | | |

```
73
74              webgoat.customjs.profileZipSlipRetrieval = function () {
75                  $.get("PathTraversal/zip-slip", function (result, status) {
76                      document.getElementById("previewZipSlip").src = "" +
result;
77                  });
```

| | |
|---|---|
| **Sink:** | path_traversal.js:76 Assignment to src() |

```
74              webgoat.customjs.profileZipSlipRetrieval = function () {
75                  $.get("PathTraversal/zip-slip", function (result, status) {
76                      document.getElementById("previewZipSlip").src = "" +
result;
77                  });
78              }
```

## path_traversal.js, line 14 (Cross-Site Scripting: DOM)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | path_traversal.js lambda() 14 . | | |
| **Source:** | path_traversal.js:13 lambda(0) | | |

```
11
12              webgoat.customjs.profileUploadCallback = function () {
13                  $.get("PathTraversal/profile-picture", function (result, status) {
14                      document.getElementById("preview").src = "" + result;
15                  });
```

| | |
|---|---|
| **Sink:** | path_traversal.js:14 Assignment to src() |

```
12              webgoat.customjs.profileUploadCallback = function () {
13                  $.get("PathTraversal/profile-picture", function (result, status) {
14                      document.getElementById("preview").src = "" + result;
15                  });
```

ot Copyright 2024 Open Text.

```
16                  }
```

## path_traversal.js, line 76 (Cross-Site Scripting: DOM)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | path_traversal.js lambda() 76 . | | |

**Source:** path_traversal.js:75 lambda(0)

```
73
74                  webgoat.customjs.profileZipSlipRetrieval = function () {
75                      $.get("PathTraversal/zip-slip", function (result, status) {
76                          document.getElementById("previewZipSlip").src = "" +
result;
77                      });
```

**Sink:** path_traversal.js:76 Assignment to src()

```
74                  webgoat.customjs.profileZipSlipRetrieval = function () {
75                      $.get("PathTraversal/zip-slip", function (result, status) {
76                          document.getElementById("previewZipSlip").src = "" +
result;
77                      });
78                  }
```

## MissingFunctionACUsers.java, line 75 (Cross-Site Scripting: Persistent)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | MissingFunctionACUsers.java usersService() 75 . | | |

**Source:** MissingAccessControlUserRepository.java:23 org.springframework.jdbc.core.namedparam.NamedParameterJdbcTemplate.query()

```
21
22                      public List<User> findAllUsers() {
23                          return jdbcTemplate.query("select username, password, admin from
access_control_users", mapper);
24                      }
```

**Sink:** MissingFunctionACUsers.java:75 org.springframework.http.ResponseEntity.ok()

```
73                      @ResponseBody
74                      public ResponseEntity<List<DisplayUser>> usersService() {
75                          return ResponseEntity.ok(userRepository.findAllUsers().stream().map(user ->
new DisplayUser(user, PASSWORD_SALT_SIMPLE)).collect(Collectors.toList()));
76                      }
```

## clientSideFiltering.js, line 17 (Cross-Site Scripting: Self)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | clientSideFiltering.js ajaxFunction() 17 . | | |

**Source:** clientSideFiltering.js:12 Read value()

```
10                      if (!dataFetched) {
11                          dataFetched = true;
12                          ajaxFunction(document.getElementById("userID").value);
13                      }
14                  }
```

**Sink:** clientSideFiltering.js:17 ~JS_Generic.get()

```
15
16                  function ajaxFunction(userId) {
17                      $.get("clientSideFiltering/salaries?userId=" + userId, function (result, status) {
18                          var html = "<table border = '1' width = '90%' align = 'center'";
19                          html = html + '<tr>';
```

## csrf-review.js, line 41 (Cross-Site Scripting: DOM)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |

ot Copyright 2024 Open Text.

| | |
|---|---|
| **Kingdom:** | Input Validation and Representation |
| **Abstract:** | csrf-review.js lambda() 41 . |
| **Source:** | csrf-review.js:35 lambda(0) |

```
33                  function getChallenges() {
34                      $("#list").empty();
35                      $.get('csrf/review', function (result, status) {
36                          for (var i = 0; i < result.length; i++) {
37                              var comment = html.replace('USER', result[i].user);
```

| | |
|---|---|
| **Sink:** | csrf-review.js:41 ~JS_Generic.append() |

```
39                              comment = comment.replace('COMMENT', result[i].text);
40                              comment = comment.replace('STARS', result[i].stars)
41                              $("#list").append(comment);
42                          }
```

## assignment13.js, line 57 (Cross-Site Scripting: DOM)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | assignment13.js lambda() 57 . | | |
| **Source:** | assignment13.js:43 lambda(0) | | |

```
41
42              function getServers(column) {
43                  $.get("SqlInjectionMitigations/servers?column=" + column, function (result,
status) {
44                      $("#servers").empty();
45                      for (var i = 0; i < result.length; i++) {
```

| | |
|---|---|
| **Sink:** | assignment13.js:57 ~JS_Generic.append() |

```
55                          server = server.replace('MAC', result[i].mac);
56                          server = server.replace('DESCRIPTION', result[i].description);
57                          $("#servers").append(server);
58                      }
```

## MissingFunctionACUsers.java, line 83 (Cross-Site Scripting: Persistent)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | MissingFunctionACUsers.java usersFixed() 83 . | | |
| **Source:** | MissingAccessControlUserRepository.java:23 org.springframework.jdbc.core.namedparam.NamedParameterJdbcTemplate.query() | | |

```
21
22              public List<User> findAllUsers() {
23                  return jdbcTemplate.query("select username, password, admin from
access_control_users", mapper);
24              }
```

| | |
|---|---|
| **Sink:** | MissingFunctionACUsers.java:83 org.springframework.http.ResponseEntity.ok() |

```
81                  var currentUser = userRepository.findByUsername(webSession.getUserName());
82                  if (currentUser != null && currentUser.isAdmin()) {
83                      return ResponseEntity.ok(userRepository.findAllUsers().stream().map(user -
> new DisplayUser(user, PASSWORD_SALT_ADMIN)).collect(Collectors.toList()));
84                  }
85                  return ResponseEntity.status(HttpStatus.FORBIDDEN).build();
```

## xxe.js, line 78 (Cross-Site Scripting: DOM)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | xxe.js lambda() 78 . | | |
| **Source:** | xxe.js:72 lambda(0) | | |

```
70
71              function getComments(field) {
```

ot Copyright 2024 Open Text.

```
72                  $.get("xxe/comments", function (result, status) {
73                      $(field).empty();
74                      for (var i = 0; i < result.length; i++) {
```

Sink: xxe.js:78 ~JS_Generic.append()

```
76                          comment = comment.replace('DATETIME', result[i].dateTime);
77                          comment = comment.replace('COMMENT', result[i].text);
78                          $(field).append(comment);
79                      }
```

## xxe.js, line 78 (Cross-Site Scripting: DOM)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | xxe.js lambda() 78 . | | |

Source: xxe.js:72 lambda(0)

```
70
71  function getComments(field) {
72      $.get("xxe/comments", function (result, status) {
73          $(field).empty();
74          for (var i = 0; i < result.length; i++) {
```

Sink: xxe.js:78 ~JS_Generic.append()

```
76                          comment = comment.replace('DATETIME', result[i].dateTime);
77                          comment = comment.replace('COMMENT', result[i].text);
78                          $(field).append(comment);
79                      }
```

## path_traversal.js, line 60 (Cross-Site Scripting: DOM)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | path_traversal.js lambda() 60 . | | |

Source: path_traversal.js:59 lambda(0)

```
57
58  function newRandomPicture() {
59      $.get("PathTraversal/random-picture", function (result, status) {
60          document.getElementById("randomCatPicture").src = "" +
    result;
61      });
```

Sink: path_traversal.js:60 Assignment to src()

```
58  function newRandomPicture() {
59      $.get("PathTraversal/random-picture", function (result, status) {
60          document.getElementById("randomCatPicture").src = "" +
    result;
61      });
62  }
```

## jquery.form.js, line 346 (Cross-Site Scripting: DOM)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | jquery.form.js fileUploadXhr() 346 . | | |

Source: jquery.form.js:115 Read window.location()

```
113
114             url = (typeof action === 'string') ? $.trim(action) : '';
115             url = url || window.location.href || '';
116             if (url) {
117                 // clean url (don't include hash vaue)
```

Sink: jquery.form.js:346 ~JS_Generic.ajax()

```
344                 }
345             };
346             return $.ajax(s);
347         }
```

## challenge8.js, line 52 (Cross-Site Scripting: DOM)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | challenge8.js lambda() 52 . | | |

Source: challenge8.js:46 lambda(0)

```
44  function doVote(stars) {
45      $("#voteResultMsg").hide();
46      $.get("challenge/8/vote/" + stars, function (result) {
47          if (result["error"]) {
48              $("#voteResultMsg").addClass('alert-danger alert-dismissable');
```

Sink: challenge8.js:52 ~JS_Generic.html()

```
50              $("#voteResultMsg").addClass('alert-success alert-dismissable');
51          }
52          $("#voteResultMsg").html(result["message"]);
53          $("#voteResultMsg").show();
54      })
```

## path_traversal.js, line 60 (Cross-Site Scripting: DOM)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | path_traversal.js lambda() 60 . | | |

Source: path_traversal.js:59 lambda(0)

```
57
58  function newRandomPicture() {
59      $.get("PathTraversal/random-picture", function (result, status) {
60          document.getElementById("randomCatPicture").src = "" +
    result;
61      });
```

Sink: path_traversal.js:60 Assignment to src()

```
58  function newRandomPicture() {
59      $.get("PathTraversal/random-picture", function (result, status) {
60          document.getElementById("randomCatPicture").src = "" +
    result;
61      });
62  }
```

## clientSideFiltering.js, line 17 (Cross-Site Scripting: Self)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | clientSideFiltering.js ajaxFunction() 17 . | | |

Source: clientSideFiltering.js:12 Read value()

```
10      if (!dataFetched) {
11          dataFetched = true;
12          ajaxFunction(document.getElementById("userID").value);
13      }
14  }
```

Sink: clientSideFiltering.js:17 ~JS_Generic.get()

```
15
16  function ajaxFunction(userId) {
17      $.get("clientSideFiltering/salaries?userId=" + userId, function (result, status) {
18          var html = "<table border = '1' width = '90%' align = 'center'";
19          html = html + '<tr>';
```

## path_traversal.js, line 54 (Cross-Site Scripting: DOM)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | path_traversal.js lambda() 54 . | | |

Source: path_traversal.js:53 lambda(0)

ot Copyright 2024 Open Text.

```
51
52                  webgoat.customjs.profileUploadCallbackRetrieval = function () {
53                      $.get("PathTraversal/profile-picture", function (result, status) {
54                          document.getElementById("previewRetrieval").src = "" +
                    result;
55                      });
```

**Sink:** path_traversal.js:54 Assignment to src()

```
52                  webgoat.customjs.profileUploadCallbackRetrieval = function () {
53                      $.get("PathTraversal/profile-picture", function (result, status) {
54                          document.getElementById("previewRetrieval").src = "" +
                    result;
55                      });
56                  }
```

## GoatUtils.js, line 57 (Cross-Site Scripting: DOM)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | GoatUtils.js lambda() 57 . | | |

**Source:** GoatUtils.js:56 lambda(0)

```
54
55                              showLessonCookiesAndParams: function() {
56                                  $.get(goatConstants.cookieService, {}, function(reply) {
57                                      $("#lesson_cookies").html(reply);
58                                  }, "html");
```

**Sink:** GoatUtils.js:57 ~JS_Generic.html()

```
55                              showLessonCookiesAndParams: function() {
56                                  $.get(goatConstants.cookieService, {}, function(reply) {
57                                      $("#lesson_cookies").html(reply);
58                                  }, "html");
59                              },
```

## path_traversal.js, line 30 (Cross-Site Scripting: DOM)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | path_traversal.js lambda() 30 . | | |

**Source:** path_traversal.js:29 lambda(0)

```
27
28                  webgoat.customjs.profileUploadCallbackFix = function () {
29                      $.get("PathTraversal/profile-picture", function (result, status) {
30                          document.getElementById("previewFix").src = "" + result;
31                      });
```

**Sink:** path_traversal.js:30 Assignment to src()

```
28                  webgoat.customjs.profileUploadCallbackFix = function () {
29                      $.get("PathTraversal/profile-picture", function (result, status) {
30                          document.getElementById("previewFix").src = "" + result;
31                      });
32                  }
```

## challenge8.js, line 18 (Cross-Site Scripting: DOM)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Critical | **Folder** | Critical |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | challenge8.js lambda() 18 . | | |

**Source:** challenge8.js:7 lambda(0)

```
5
6                   function loadVotes() {
7                       $.get("challenge/8/votes/", function (votes) {
8                               var totalVotes = 0;
9                               for (var i = 1; i <= 5; i++) {
```

**Sink:** challenge8.js:18 ~JS_Generic.html()

ot Copyright 2024 Open Text.

```
16                              var progressBar = $('#progressBar' + i);
17                              progressBar.width(Math.round(percent) * 2 + '%');
18                              $("#nrOfVotes" + i).html(votes[i]);
19
20                          }
```

## jwt-voting.js, line 63 (Cross-Site Scripting: DOM)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | jwt-voting.js lambda() 63 . | | |

Source: jwt-voting.js:43 lambda(0)

```
41  function getVotings() {
42      $("#votesList").empty();
43      $.get("JWT/votings", function (result, status) {
44          for (var i = 0; i < result.length; i++) {
45              var voteTemplate = html.replace('IMAGE_SMALL', result[i].imageSmall);
```

Sink: jwt-voting.js:63 ~JS_Generic.append()

```
61              voteTemplate = voteTemplate.replace(/HIDDEN_VIEW_RATING/g, hidden);
62
63              $("#votesList").append(voteTemplate);
64          }
65      })
```

## jquery.form.js, line 346 (Cross-Site Scripting: DOM)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | jquery.form.js fileUploadXhr() 346 . | | |

Source: jquery.form.js:115 Read window.location()

```
113
114     url = (typeof action === 'string') ? $.trim(action) : '';
115     url = url || window.location.href || '';
116     if (url) {
117         // clean url (don't include hash vaue)
```

Sink: jquery.form.js:346 ~JS_Generic.ajax()

```
344             }
345         };
346         return $.ajax(s);
347     }
```

## clientSideFiltering.js, line 38 (Cross-Site Scripting: DOM)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | clientSideFiltering.js lambda() 38 . | | |

Source: clientSideFiltering.js:17 lambda(0)

```
15
16  function ajaxFunction(userId) {
17      $.get("clientSideFiltering/salaries?userId=" + userId, function (result, status) {
18          var html = "<table border = '1' width = '90%' align = 'center'";
19          html = html + '<tr>';
```

Sink: clientSideFiltering.js:38 Assignment to newdiv.innerHTML()

```
36
37          var newdiv = document.createElement("div");
38          newdiv.innerHTML = html;
39          var container = document.getElementById("hiddenEmployeeRecords");
40          container.appendChild(newdiv);
```

## path_traversal.js, line 30 (Cross-Site Scripting: DOM)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |

Copyright 2024 Open Text.

| | |
|---|---|
| Kingdom: | Input Validation and Representation |
| Abstract: | path_traversal.js lambda() 30 . |
| Source: | path_traversal.js:29 lambda(0) |

```
27
28                webgoat.customjs.profileUploadCallbackFix = function () {
29                    $.get("PathTraversal/profile-picture", function (result, status) {
30                        document.getElementById("previewFix").src = "" + result;
31                    });
```

Sink: path_traversal.js:30 Assignment to src()

```
28                webgoat.customjs.profileUploadCallbackFix = function () {
29                    $.get("PathTraversal/profile-picture", function (result, status) {
30                        document.getElementById("previewFix").src = "" + result;
31                    });
32                }
```

## path_traversal.js, line 47 (Cross-Site Scripting: DOM)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | path_traversal.js lambda() 47 . | | |
| Source: | path_traversal.js:46 lambda(0) | | |

```
44
45                webgoat.customjs.profileUploadCallbackRemoveUserInput = function () {
46                    $.get("PathTraversal/profile-picture", function (result, status) {
47                        document.getElementById("previewRemoveUserInput").src =
"" + result;
48                    });
```

Sink: path_traversal.js:47 Assignment to src()

```
45                webgoat.customjs.profileUploadCallbackRemoveUserInput = function () {
46                    $.get("PathTraversal/profile-picture", function (result, status) {
47                        document.getElementById("previewRemoveUserInput").src =
"" + result;
48                    });
49                }
```

## jquery.form.js, line 346 (Cross-Site Scripting: DOM)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | jquery.form.js fileUploadXhr() 346 . | | |
| Source: | jquery.form.js:115 Read window.location() | | |

```
113
114                   url = (typeof action === 'string') ? $.trim(action) : '';
115                   url = url || window.location.href || '';
116                   if (url) {
117                       // clean url (don't include hash vaue)
```

Sink: jquery.form.js:346 ~JS_Generic.ajax()

```
344                       }
345                   };
346                   return $.ajax(s);
347               }
```

## path_traversal.js, line 14 (Cross-Site Scripting: DOM)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | path_traversal.js lambda() 14 . | | |
| Source: | path_traversal.js:13 lambda(0) | | |

```
11
12                webgoat.customjs.profileUploadCallback = function () {
13                    $.get("PathTraversal/profile-picture", function (result, status) {
14                        document.getElementById("preview").src = "" + result;
```

ot Copyright 2024 Open Text.

```
15                  });
```

Sink: path_traversal.js:14 Assignment to src()

```
12  webgoat.customjs.profileUploadCallback = function () {
13      $.get("PathTraversal/profile-picture", function (result, status) {
14          document.getElementById("preview").src = "" + result;
15      });
16  }
```

## jquery.form.js, line 346 (Cross-Site Scripting: DOM)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | jquery.form.js fileUploadXhr() 346 . | | |

Source: jquery.form.js:115 Read window.location()

```
113
114     url = (typeof action === 'string') ? $.trim(action) : '';
115     url = url || window.location.href || '';
116     if (url) {
117         // clean url (don't include hash vaue)
```

Sink: jquery.form.js:346 ~JS_Generic.ajax()

```
344             }
345         };
346         return $.ajax(s);
347     }
```

## assignment13.js, line 57 (Cross-Site Scripting: DOM)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | assignment13.js lambda() 57 . | | |

Source: assignment13.js:43 lambda(0)

```
41
42  function getServers(column) {
43      $.get("SqlInjectionMitigations/servers?column=" + column, function (result,
    status) {
44          $("#servers").empty();
45          for (var i = 0; i < result.length; i++) {
```

Sink: assignment13.js:57 ~JS_Generic.append()

```
55              server = server.replace('MAC', result[i].mac);
56              server = server.replace('DESCRIPTION', result[i].description);
57              $("#servers").append(server);
58          }
```

## csrf-review.js, line 41 (Cross-Site Scripting: DOM)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | csrf-review.js lambda() 41 . | | |

Source: csrf-review.js:35 lambda(0)

```
33      function getChallenges() {
34          $("#list").empty();
35          $.get('csrf/review', function (result, status) {
36              for (var i = 0; i < result.length; i++) {
37                  var comment = html.replace('USER', result[i].user);
```

Sink: csrf-review.js:41 ~JS_Generic.append()

```
39                  comment = comment.replace('COMMENT', result[i].text);
40                  comment = comment.replace('STARS', result[i].stars)
41                  $("#list").append(comment);
42              }
```

## path_traversal.js, line 54 (Cross-Site Scripting: DOM)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |

| | |
|---|---|
| Kingdom: | Input Validation and Representation |
| Abstract: | path_traversal.js lambda() 54 . |
| Source: | path_traversal.js:53 lambda(0) |

```
51
52                webgoat.customjs.profileUploadCallbackRetrieval = function () {
53                    $.get("PathTraversal/profile-picture", function (result, status) {
54                        document.getElementById("previewRetrieval").src = "" +
result;
55                    });
```

| | |
|---|---|
| Sink: | path_traversal.js:54 Assignment to src() |

```
52                webgoat.customjs.profileUploadCallbackRetrieval = function () {
53                    $.get("PathTraversal/profile-picture", function (result, status) {
54                        document.getElementById("previewRetrieval").src = "" +
result;
55                    });
56                }
```

## jwt-voting.js, line 63 (Cross-Site Scripting: DOM)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | jwt-voting.js lambda() 63 . | | |
| Source: | jwt-voting.js:43 lambda(0) | | |

```
41                function getVotings() {
42                    $("#votesList").empty();
43                    $.get("JWT/votings", function (result, status) {
44                        for (var i = 0; i < result.length; i++) {
45                            var voteTemplate = html.replace('IMAGE_SMALL', result[i].imageSmall);
```

| | |
|---|---|
| Sink: | jwt-voting.js:63 ~JS_Generic.append() |

```
61                            voteTemplate = voteTemplate.replace(/HIDDEN_VIEW_RATING/g, hidden);
62
63                            $("#votesList").append(voteTemplate);
64                        }
65                    })
```

## challenge8.js, line 52 (Cross-Site Scripting: DOM)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | challenge8.js lambda() 52 . | | |
| Source: | challenge8.js:46 lambda(0) | | |

```
44                function doVote(stars) {
45                    $("#voteResultMsg").hide();
46                    $.get("challenge/8/vote/" + stars, function (result) {
47                        if (result["error"]) {
48                            $("#voteResultMsg").addClass('alert-danger alert-dismissable');
```

| | |
|---|---|
| Sink: | challenge8.js:52 ~JS_Generic.html() |

```
50                            $("#voteResultMsg").addClass('alert-success alert-dismissable');
51                        }
52                        $("#voteResultMsg").html(result["message"]);
53                        $("#voteResultMsg").show();
54                    })
```

## clientSideFiltering.js, line 38 (Cross-Site Scripting: DOM)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | clientSideFiltering.js lambda() 38 . | | |
| Source: | clientSideFiltering.js:17 lambda(0) | | |

```
15
16                function ajaxFunction(userId) {
17                    $.get("clientSideFiltering/salaries?userId=" + userId, function (result, status) {
```

ot Copyright 2024 Open Text.

```
18                  var html = "<table border = '1' width = '90%' align = 'center'";
19                  html = html + '<tr>';
```

Sink: clientSideFiltering.js:38 Assignment to newdiv.innerHTML()

```
36
37                  var newdiv = document.createElement("div");
38                  newdiv.innerHTML = html;
39                  var container = document.getElementById("hiddenEmployeeRecords");
40                  container.appendChild(newdiv);
```

## clientSideFilteringFree.js, line 41 (Cross-Site Scripting: Self)

| Fortify Priority: | Low | Folder | Low |
|---|---|---|---|
| Kingdom: | Input Validation and Representation | | |
| Abstract: | clientSideFilteringFree.js lambda() 41 . | | |

Source: clientSideFilteringFree.js:40 ~JS_Generic.val()

```
38      })
39      $(".checkoutCode").on("blur", function () {
40          var checkoutCode = $(".checkoutCode").val();
41          $.get("clientSideFiltering/challenge-store/coupons/" + checkoutCode, function
(result, status) {
42              var discount = result.discount;
```

Sink: clientSideFilteringFree.js:41 ~JS_Generic.get()

```
39      $(".checkoutCode").on("blur", function () {
40          var checkoutCode = $(".checkoutCode").val();
41          $.get("clientSideFiltering/challenge-store/coupons/" + checkoutCode, function
(result, status) {
42              var discount = result.discount;
43              if (discount > 0) {
```

## challenge8.js, line 18 (Cross-Site Scripting: DOM)

| Fortify Priority: | Critical | Folder | Critical |
|---|---|---|---|
| Kingdom: | Input Validation and Representation | | |
| Abstract: | challenge8.js lambda() 18 . | | |

Source: challenge8.js:7 lambda(0)

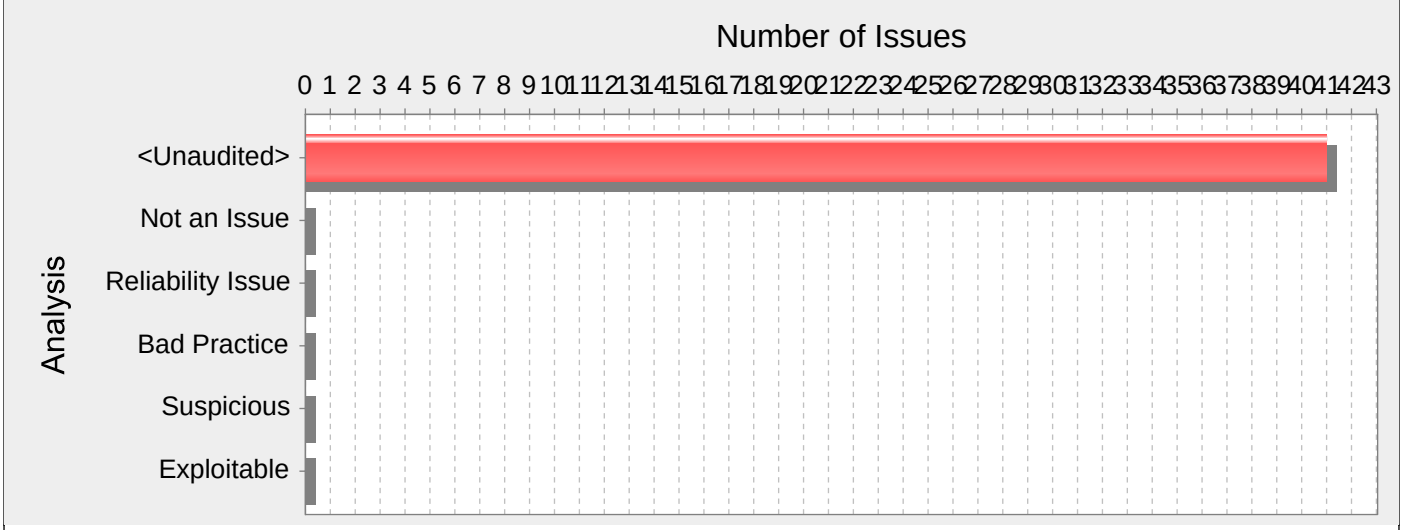
```
5
6   function loadVotes() {
7       $.get("challenge/8/votes/", function (votes) {
8               var totalVotes = 0;
9               for (var i = 1; i <= 5; i++) {
```

Sink: challenge8.js:18 ~JS_Generic.html()

```
16                  var progressBar = $('#progressBar' + i);
17                  progressBar.width(Math.round(percent) * 2 + '%');
18                  $("#nrOfVotes" + i).html(votes[i]);
19
20              }
```

Copyright 2024 Open Text.

## 49: 01.03. (41 Issues)




### Abstract:

MavenWrapperDownloader.java 50 File() . .

### Explanation:

Path manipulation .

1. .
2. .

, .

1: HTTP . "../../tomcat/conf/server.xml" .

```
String rName = request.getParameter("reportName");
File rFile = new File("/usr/local/apfr/reports/" + rName);
...
rFile.delete();
```

2: . , .txt .

```
fis = new FileInputStream(cfg.getProperty("sub")+".txt");
amt = fis.read(arr);
out.println(arr);
```

Path manipulation . . . .

3: Example 1 Android .

```
...
String rName = this.getIntent().getExtras().getString("reportName");
File rFile = getBaseContext().getFileStreamPath(rName);
...
rFile.delete();
...
```

### Recommendations:

Path Manipulation . .

. . . . .

### Tips:

1. Fortify Custom Rules Editor .
2. . . , . .
3. (Struts Spring MVC ). , Fortify Fortify Static Code Analyzer . Context-Sensitive Ranking( ) . Fortify , Fortify Software Security Research Group .

## FileServer.java, line 77 (Path Manipulation)

| Fortify Priority: | Critical | Folder | Critical |
|---|---|---|---|
| Kingdom: | Input Validation and Representation | | |
| Abstract: | FileServer.java 77 File() . . | | |

Source: FileServer.java:72 importFile(0)

```
70
71      @PostMapping(value = "/fileupload")
72      public ModelAndView importFile(@RequestParam("file") MultipartFile myFile) throws
IOException {
73          var user = (WebGoatUser)
SecurityContextHolder.getContext().getAuthentication().getPrincipal();
74          var destinationDir = new File(fileLocation, user.getUsername());
```

Sink: FileServer.java:77 java.io.File.File()

```
75          destinationDir.mkdirs();
76          myFile.transferTo(new File(destinationDir, myFile.getOriginalFilename()));
77          log.debug("File saved to {}", new File(destinationDir,
myFile.getOriginalFilename()));
78
79          return new ModelAndView(
```

## ProfileUploadRetrieval.java, line 47 (Path Manipulation)

| Fortify Priority: | High | Folder | High |
|---|---|---|---|
| Kingdom: | Input Validation and Representation | | |
| Abstract: | ProfileUploadRetrieval.java 47 File() . . | | |

Source: ProfileUploadRetrieval.java:46 ProfileUploadRetrieval(0)

```
44      private final File catPicturesDirectory;
45
46      public ProfileUploadRetrieval(@Value("${webgoat.server.directory}") String
webGoatHomeDirectory) {
47          this.catPicturesDirectory = new File(webGoatHomeDirectory, "/PathTraversal/" +
"/cats");
48          this.catPicturesDirectory.mkdirs();
```

Sink: ProfileUploadRetrieval.java:47 java.io.File.File()

```
45
46      public ProfileUploadRetrieval(@Value("${webgoat.server.directory}") String
webGoatHomeDirectory) {
47          this.catPicturesDirectory = new File(webGoatHomeDirectory, "/PathTraversal/" +
"/cats");
48          this.catPicturesDirectory.mkdirs();
49      }
```

## ProfileZipSlip.java, line 54 (Path Manipulation)

| Fortify Priority: | Critical | Folder | Critical |
|---|---|---|---|
| Kingdom: | Input Validation and Representation | | |
| Abstract: | ProfileZipSlip.java 54 toFile() . . | | |

Source: ProfileZipSlip.java:36 uploadFileHandler(0)

```
34      @PostMapping(value = "/PathTraversal/zip-slip", consumes = ALL_VALUE, produces =
APPLICATION_JSON_VALUE)
35      @ResponseBody
36      public AttackResult uploadFileHandler(@RequestParam("uploadedFileZipSlip")
MultipartFile file) {
37          if (!file.getOriginalFilename().toLowerCase().endsWith(".zip")) {
38              return failed(this).feedback("path-traversal-zip-slip.no-zip").build();
```

Sink: ProfileZipSlip.java:54 java.nio.file.Path.toFile()

```
52          try {
53              var uploadedZipFile = tmpZipDirectory.resolve(file.getOriginalFilename());
54              FileCopyUtils.copy(file.getBytes(), uploadedZipFile.toFile());
55
56              ZipFile zip = new ZipFile(uploadedZipFile.toFile());
```

## FileServer.java, line 74 (Path Manipulation)

| Fortify Priority: | High | Folder | High |
|---|---|---|---|

| | |
|---|---|
| Kingdom: | Input Validation and Representation |
| Abstract: | FileServer.java 74 File() . . |

Source: FileServer.java:74 Read this.fileLocation()

```
72      public ModelAndView importFile(@RequestParam("file") MultipartFile myFile) throws
IOException {
73          var user = (WebGoatUser)
SecurityContextHolder.getContext().getAuthentication().getPrincipal();
74          var destinationDir = new File(fileLocation, user.getUsername());
75          destinationDir.mkdirs();
76          myFile.transferTo(new File(destinationDir, myFile.getOriginalFilename()));
```

Sink: FileServer.java:74 java.io.File.File()

```
72      public ModelAndView importFile(@RequestParam("file") MultipartFile myFile) throws
IOException {
73          var user = (WebGoatUser)
SecurityContextHolder.getContext().getAuthentication().getPrincipal();
74          var destinationDir = new File(fileLocation, user.getUsername());
75          destinationDir.mkdirs();
76          myFile.transferTo(new File(destinationDir, myFile.getOriginalFilename()));
```

## BlindSendFileAssignmentTest.java, line 113 (Path Manipulation)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | BlindSendFileAssignmentTest.java 113 File() . . | | |

Source: BlindSendFileAssignmentTest.java:113 Read this.webGoatHomeDirectory()

```
111     @Test
112     public void solveOnlyParamReferenceEntityInExternalDTD() throws Exception {
113         File targetFile = new File(webGoatHomeDirectory, "/XXE/" +
webSession.getUserName() + "/secret.txt");
114         //Host DTD on WebWolf site
115         String dtd = "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n" +
```

Sink: BlindSendFileAssignmentTest.java:113 java.io.File.File()

```
111     @Test
112     public void solveOnlyParamReferenceEntityInExternalDTD() throws Exception {
113         File targetFile = new File(webGoatHomeDirectory, "/XXE/" +
webSession.getUserName() + "/secret.txt");
114         //Host DTD on WebWolf site
115         String dtd = "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n" +
```

## ProfileUploadBase.java, line 36 (Path Manipulation)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | ProfileUploadBase.java 36 File() . . | | |

Source: ProfileUploadFix.java:22 ProfileUploadFix(0)

```
20  public class ProfileUploadFix extends ProfileUploadBase {
21
22      public ProfileUploadFix(@Value("${webgoat.server.directory}") String
webGoatHomeDirectory, WebSession webSession) {
23          super(webGoatHomeDirectory, webSession);
24      }
```

Sink: ProfileUploadBase.java:36 java.io.File.File()

```
34          }
35
36          var uploadDirectory = new File(this.webGoatHomeDirectory, "/PathTraversal/" +
webSession.getUserName());
37          if (uploadDirectory.exists()) {
38              FileSystemUtils.deleteRecursively(uploadDirectory);
```

## ProfileUploadRetrieval.java, line 96 (Resource Injection)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | ProfileUploadRetrieval.java 96 URI() ID . . | | |

Source: ProfileUploadRetrieval.java:46 ProfileUploadRetrieval(0)

```
44                  private final File catPicturesDirectory;
45
46                  public ProfileUploadRetrieval(@Value("${webgoat.server.directory}") String
webGoatHomeDirectory) {
47                      this.catPicturesDirectory = new File(webGoatHomeDirectory, "/PathTraversal/" +
"/cats");
48                      this.catPicturesDirectory.mkdirs();
```

Sink: ProfileUploadRetrieval.java:96 java.net.URI.URI()

```
94                              return ResponseEntity.ok()
95
.contentType(MediaType.parseMediaType(MediaType.IMAGE_JPEG_VALUE))
96                                      .location(new URI("/PathTraversal/random-picture?id=" +
catPicture.getName()))
97
.body(Base64.getEncoder().encode(FileCopyUtils.copyToByteArray(catPicture)));
98                          }
```

## Salaries.java, line 76 (Path Manipulation)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Input Validation and Representation | | |

Abstract: Salaries.java 76 File() . .

Source: Salaries.java:76 Read this.webGoatHomeDirectory()

```
74                  public List<Map<String, Object>> invoke() {
75                      NodeList nodes = null;
76                      File d = new File(webGoatHomeDirectory, "ClientSideFiltering/employees.xml");
77                      XPathFactory factory = XPathFactory.newInstance();
78                      XPath path = factory.newXPath();
```

Sink: Salaries.java:76 java.io.File.File()

```
74                  public List<Map<String, Object>> invoke() {
75                      NodeList nodes = null;
76                      File d = new File(webGoatHomeDirectory, "ClientSideFiltering/employees.xml");
77                      XPathFactory factory = XPathFactory.newInstance();
78                      XPath path = factory.newXPath();
```

## ProfileUploadBase.java, line 43 (Path Manipulation)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Input Validation and Representation | | |

Abstract: ProfileUploadBase.java 43 File() . .

Source: ProfileUploadFix.java:30 uploadFileHandler(1)

```
28                  public AttackResult uploadFileHandler(
29                          @RequestParam("uploadedFileFix") MultipartFile file,
30                          @RequestParam(value = "fullNameFix", required = false) String fullName) {
31                      return super.execute(file, fullName != null ? fullName.replace("../", "") :
"");
32                  }
```

Sink: ProfileUploadBase.java:43 java.io.File.File()

```
41                      try {
42                          uploadDirectory.mkdirs();
43                          var uploadedFile = new File(uploadDirectory, fullName);
44                          uploadedFile.createNewFile();
45                          FileCopyUtils.copy(file.getBytes(), uploadedFile);
```

## ProfileUploadBase.java, line 43 (Path Manipulation)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Input Validation and Representation | | |

Abstract: ProfileUploadBase.java 43 File() . .

Source: ProfileUpload.java:28 uploadFileHandler(1)

```
26                  @PostMapping(value = "/PathTraversal/profile-upload", consumes = ALL_VALUE,
produces = APPLICATION_JSON_VALUE)
27                  @ResponseBody
```

```
28                  public AttackResult uploadFileHandler(@RequestParam("uploadedFile") MultipartFile
file, @RequestParam(value = "fullName", required = false) String fullName) {
29                      return super.execute(file, fullName);
30                  }
```

Sink: ProfileUploadBase.java:43 java.io.File.File()

```
41                      try {
42                          uploadDirectory.mkdirs();
43                          var uploadedFile = new File(uploadDirectory, fullName);
44                          uploadedFile.createNewFile();
45                          FileCopyUtils.copy(file.getBytes(), uploadedFile);
```

## MvcConfiguration.java, line 61 (Path Manipulation)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Input Validation and Representation | | |

Abstract: MvcConfiguration.java 61 File() . .

Source: MvcConfiguration.java:61 Read this.fileLocation()

```
59      @PostConstruct
60      public void createDirectory() {
61          File file = new File(fileLocation);
62          if (!file.exists()) {
63              file.mkdirs();
```

Sink: MvcConfiguration.java:61 java.io.File.File()

```
59      @PostConstruct
60      public void createDirectory() {
61          File file = new File(fileLocation);
62          if (!file.exists()) {
63              file.mkdirs();
```

## BlindSendFileAssignmentTest.java, line 89 (Path Manipulation)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Input Validation and Representation | | |

Abstract: BlindSendFileAssignmentTest.java 89 File() . .

Source: BlindSendFileAssignmentTest.java:89 Read this.webGoatHomeDirectory()

```
87      @Test
88      public void solve() throws Exception {
89          File targetFile = new File(webGoatHomeDirectory, "/XXE/" +
webSession.getUserName() + "/secret.txt");
90          //Host DTD on WebWolf site
91          String dtd = "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n" +
```

Sink: BlindSendFileAssignmentTest.java:89 java.io.File.File()

```
87      @Test
88      public void solve() throws Exception {
89          File targetFile = new File(webGoatHomeDirectory, "/XXE/" +
webSession.getUserName() + "/secret.txt");
90          //Host DTD on WebWolf site
91          String dtd = "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n" +
```

## WebGoat.java, line 56 (Path Manipulation)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Input Validation and Representation | | |

Abstract: WebGoat.java 56 File() . .

Source: WebGoat.java:55 pluginTargetDirectory(0)

```
53
54      @Bean(name = "pluginTargetDirectory")
55      public File pluginTargetDirectory(@Value("${webgoat.user.directory}") final String
webgoatHome) {
56          return new File(webgoatHome);
57      }
```

Sink: WebGoat.java:56 java.io.File.File()

```
54      @Bean(name = "pluginTargetDirectory")
```

ot Copyright 2024 Open Text.

```
55                      public File pluginTargetDirectory(@Value("${webgoat.user.directory}") final String
webgoatHome) {
56                          return new File(webgoatHome);
57                      }
```

## ProfileUploadBase.java, line 43 (Path Manipulation)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | ProfileUploadBase.java 43 File() . . | | |

Source: ProfileUploadRemoveUserInput.java:26 uploadFileHandler(0)

```
24                      @PostMapping(value = "/PathTraversal/profile-upload-remove-user-input", consumes =
ALL_VALUE, produces = APPLICATION_JSON_VALUE)
25                      @ResponseBody
26                      public AttackResult uploadFileHandler(@RequestParam("uploadedFileRemoveUserInput")
MultipartFile file) {
27                          return super.execute(file, file.getOriginalFilename());
28                      }
```

Sink: ProfileUploadBase.java:43 java.io.File.File()

```
41                          try {
42                              uploadDirectory.mkdirs();
43                              var uploadedFile = new File(uploadDirectory, fullName);
44                              uploadedFile.createNewFile();
45                              FileCopyUtils.copy(file.getBytes(), uploadedFile);
```

## ProfileUploadBase.java, line 75 (Path Manipulation)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | ProfileUploadBase.java 75 File() . . | | |

Source: ProfileUpload.java:22 ProfileUpload(0)

```
20                  public class ProfileUpload extends ProfileUploadBase {
21
22                      public ProfileUpload(@Value("${webgoat.server.directory}") String
webGoatHomeDirectory, WebSession webSession) {
23                          super(webGoatHomeDirectory, webSession);
24                      }
```

Sink: ProfileUploadBase.java:75 java.io.File.File()

```
73
74                      protected byte[] getProfilePictureAsBase64() {
75                          var profilePictureDirectory = new File(this.webGoatHomeDirectory,
"/PathTraversal/" + webSession.getUserName());
76                          var profileDirectoryFiles = profilePictureDirectory.listFiles();
```

## MavenWrapperDownloader.java, line 50 (Path Manipulation)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | MavenWrapperDownloader.java 50 File() . . | | |

Source: MavenWrapperDownloader.java:48 main(0)

```
46                      private static final String PROPERTY_NAME_WRAPPER_URL = "wrapperUrl";
47
48                      public static void main(String args[]) {
49                          System.out.println("- Downloader started");
50                          File baseDirectory = new File(args[0]);
```

Sink: MavenWrapperDownloader.java:50 java.io.File.File()

```
48                      public static void main(String args[]) {
49                          System.out.println("- Downloader started");
50                          File baseDirectory = new File(args[0]);
51                          System.out.println("- Using base directory: " +
baseDirectory.getAbsolutePath());
```

## Assignment7.java, line 67 (Resource Injection)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |

ot Copyright 2024 Open Text.

| | |
|---|---|
| Kingdom: | Input Validation and Representation |
| Abstract: | Assignment7.java 67 URI() ID . . |
| Source: | Assignment7.java:67 javax.servlet.http.HttpServletRequest.getRequestURL() |

```
65                  String username = email.substring(0, email.indexOf("@"));
66                  if (StringUtils.hasText(username)) {
67                      URI uri = new URI(request.getRequestURL().toString());
68                      Email mail = Email.builder()
69                              .title("Your password reset link for challenge 7")
```

| | |
|---|---|
| Sink: | Assignment7.java:67 java.net.URI.URI() |

```
65                  String username = email.substring(0, email.indexOf("@"));
66                  if (StringUtils.hasText(username)) {
67                      URI uri = new URI(request.getRequestURL().toString());
68                      Email mail = Email.builder()
69                              .title("Your password reset link for challenge 7")
```

## ProfileUploadBase.java, line 36 (Path Manipulation)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | ProfileUploadBase.java 36 File() . . | | |
| Source: | ProfileUpload.java:22 ProfileUpload(0) | | |

```
20  public class ProfileUpload extends ProfileUploadBase {
21
22      public ProfileUpload(@Value("${webgoat.server.directory}") String
    webGoatHomeDirectory, WebSession webSession) {
23          super(webGoatHomeDirectory, webSession);
24      }
```

| | |
|---|---|
| Sink: | ProfileUploadBase.java:36 java.io.File.File() |

```
34          }
35
36          var uploadDirectory = new File(this.webGoatHomeDirectory, "/PathTraversal/" +
    webSession.getUserName());
37          if (uploadDirectory.exists()) {
38              FileSystemUtils.deleteRecursively(uploadDirectory);
```

## ProfileUploadRetrieval.java, line 86 (Path Manipulation)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | ProfileUploadRetrieval.java 86 File() . . | | |
| Source: | ProfileUploadRetrieval.java:85 javax.servlet.ServletRequest.getParameter() | | |

```
83          }
84          try {
85              var id = request.getParameter("id");
86              var catPicture = new File(catPicturesDirectory, (id == null ?
    RandomUtils.nextInt(1, 11) : id) + ".jpg");
```

| | |
|---|---|
| Sink: | ProfileUploadRetrieval.java:86 java.io.File.File() |

```
84          try {
85              var id = request.getParameter("id");
86              var catPicture = new File(catPicturesDirectory, (id == null ?
    RandomUtils.nextInt(1, 11) : id) + ".jpg");
87
88              if (catPicture.getName().toLowerCase().contains("path-traversal-
    secret.jpg")) {
```

## ProfileUploadBase.java, line 36 (Path Manipulation)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | ProfileUploadBase.java 36 File() . . | | |
| Source: | ProfileUploadRemoveUserInput.java:20 ProfileUploadRemoveUserInput(0) | | |

```
18  public class ProfileUploadRemoveUserInput extends ProfileUploadBase {
19
```

```
20                      public ProfileUploadRemoveUserInput(@Value("${webgoat.server.directory}") String
webGoatHomeDirectory, WebSession webSession) {
21                          super(webGoatHomeDirectory, webSession);
22                      }
```

Sink: ProfileUploadBase.java:36 java.io.File.File()

```
34                          }
35
36                          var uploadDirectory = new File(this.webGoatHomeDirectory, "/PathTraversal/" +
webSession.getUserName());
37                          if (uploadDirectory.exists()) {
38                              FileSystemUtils.deleteRecursively(uploadDirectory);
```

## ProfileUploadBase.java, line 75 (Path Manipulation)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Input Validation and Representation | | |

Abstract: ProfileUploadBase.java 75 File() . .

Source: ProfileUploadFix.java:22 ProfileUploadFix(0)

```
20  public class ProfileUploadFix extends ProfileUploadBase {
21
22      public ProfileUploadFix(@Value("${webgoat.server.directory}") String
webGoatHomeDirectory, WebSession webSession) {
23          super(webGoatHomeDirectory, webSession);
24      }
```

Sink: ProfileUploadBase.java:75 java.io.File.File()

```
73
74      protected byte[] getProfilePictureAsBase64() {
75          var profilePictureDirectory = new File(this.webGoatHomeDirectory,
"/PathTraversal/" + webSession.getUserName());
76          var profileDirectoryFiles = profilePictureDirectory.listFiles();
```

## FileServer.java, line 97 (Path Manipulation)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Input Validation and Representation | | |

Abstract: FileServer.java 97 File() . .

Source: FileServer.java:97 Read this.fileLocation()

```
95          WebGoatUser user = (WebGoatUser)
SecurityContextHolder.getContext().getAuthentication().getPrincipal();
96          String username = user.getUsername();
97          File destinationDir = new File(fileLocation, username);
98
99          ModelAndView modelAndView = new ModelAndView();
```

Sink: FileServer.java:97 java.io.File.File()

```
95          WebGoatUser user = (WebGoatUser)
SecurityContextHolder.getContext().getAuthentication().getPrincipal();
96          String username = user.getUsername();
97          File destinationDir = new File(fileLocation, username);
98
99          ModelAndView modelAndView = new ModelAndView();
```

## Salaries.java, line 61 (Path Manipulation)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Input Validation and Representation | | |

Abstract: Salaries.java 61 File() . .

Source: Salaries.java:61 Read this.webGoatHomeDirectory()

```
59      public void copyFiles() {
60          ClassPathResource classPathResource = new
ClassPathResource("lessons/employees.xml");
61          File targetDirectory = new File(webGoatHomeDirectory, "/ClientSideFiltering");
62          if (!targetDirectory.exists()) {
63              targetDirectory.mkdir();
```

Sink: Salaries.java:61 java.io.File.File()

Copyright 2024 Open Text.

```
59                      public void copyFiles() {
60                          ClassPathResource classPathResource = new
ClassPathResource("lessons/employees.xml");
61                          File targetDirectory = new File(webGoatHomeDirectory, "/ClientSideFiltering");
62                          if (!targetDirectory.exists()) {
63                              targetDirectory.mkdir();
```

## MD5.java, line 54 (Path Manipulation)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | MD5.java 54 File() . . | | |

Source: MD5.java:48 main(0)

```
46                       * @since ostermillerutils 1.00.00
47                       */
48                      public static void main(String[] args) {
49                          if (args.length == 0) {
50                              System.err.println("Please specify a file.");
```

Sink: MD5.java:54 java.io.File.File()

```
52                              for (String element : args) {
53                                  try {
54                                      System.out.println(MD5.getHashString(new File(element)) + " " +
element);
55                                  } catch (IOException x) {
56                                      System.err.println(x.getMessage());
```

## ProfileZipSlip.java, line 56 (Path Manipulation)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | ProfileZipSlip.java 56 toFile() . . | | |

Source: ProfileZipSlip.java:36 uploadFileHandler(0)

```
34                      @PostMapping(value = "/PathTraversal/zip-slip", consumes = ALL_VALUE, produces =
APPLICATION_JSON_VALUE)
35                      @ResponseBody
36                      public AttackResult uploadFileHandler(@RequestParam("uploadedFileZipSlip")
MultipartFile file) {
37                          if (!file.getOriginalFilename().toLowerCase().endsWith(".zip")) {
38                              return failed(this).feedback("path-traversal-zip-slip.no-zip").build();
```

Sink: ProfileZipSlip.java:56 java.nio.file.Path.toFile()

```
54                              FileCopyUtils.copy(file.getBytes(), uploadedZipFile.toFile());
55
56                              ZipFile zip = new ZipFile(uploadedZipFile.toFile());
57                              Enumeration<? extends ZipEntry> entries = zip.entries();
58                              while (entries.hasMoreElements()) {
```

## ProfileUploadRetrieval.java, line 100 (Resource Injection)

| | | | |
|---|---|---|---|
| Fortify Priority: | Low | Folder | Low |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | ProfileUploadRetrieval.java 100 URI() ID . . | | |

Source: ProfileUploadRetrieval.java:46 ProfileUploadRetrieval(0)

```
44                      private final File catPicturesDirectory;
45
46                      public ProfileUploadRetrieval(@Value("${webgoat.server.directory}") String
webGoatHomeDirectory) {
47                          this.catPicturesDirectory = new File(webGoatHomeDirectory, "/PathTraversal/" +
"/cats");
48                          this.catPicturesDirectory.mkdirs();
```

Sink: ProfileUploadRetrieval.java:100 java.net.URI.URI()

```
98                              }
99                              return ResponseEntity.status(HttpStatus.NOT_FOUND)
100                                     .location(new URI("/PathTraversal/random-picture?id=" +
catPicture.getName()))
```

ot Copyright 2024 Open Text.

```
101
                    .body(StringUtils.arrayToCommaDelimitedString(catPicture.getParentFile().listFiles()).
                    getBytes());
102                 } catch (IOException | URISyntaxException e) {
```

**ProfileUploadRetrieval.java, line 100 (Resource Injection)**

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Input Validation and Representation | | |

**Abstract:** ProfileUploadRetrieval.java 100 URI() ID . .

**Source:** ProfileUploadRetrieval.java:85 javax.servlet.ServletRequest.getParameter()

```
83                  }
84                  try {
85                      var id = request.getParameter("id");
86                      var catPicture = new File(catPicturesDirectory, (id == null ?
                    RandomUtils.nextInt(1, 11) : id) + ".jpg");
```

**Sink:** ProfileUploadRetrieval.java:100 java.net.URI.URI()

```
98                  }
99                  return ResponseEntity.status(HttpStatus.NOT_FOUND)
100                         .location(new URI("/PathTraversal/random-picture?id=" +
                    catPicture.getName()))
101
                    .body(StringUtils.arrayToCommaDelimitedString(catPicture.getParentFile().listFiles()).
                    getBytes());
102                 } catch (IOException | URISyntaxException e) {
```

**ProfileUploadRetrieval.java, line 96 (Resource Injection)**

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Input Validation and Representation | | |

**Abstract:** ProfileUploadRetrieval.java 96 URI() ID . .

**Source:** ProfileUploadRetrieval.java:85 javax.servlet.ServletRequest.getParameter()

```
83                  }
84                  try {
85                      var id = request.getParameter("id");
86                      var catPicture = new File(catPicturesDirectory, (id == null ?
                    RandomUtils.nextInt(1, 11) : id) + ".jpg");
```

**Sink:** ProfileUploadRetrieval.java:96 java.net.URI.URI()

```
94                      return ResponseEntity.ok()
95
                    .contentType(MediaType.parseMediaType(MediaType.IMAGE_JPEG_VALUE))
96                          .location(new URI("/PathTraversal/random-picture?id=" +
                    catPicture.getName()))
97
                    .body(Base64.getEncoder().encode(FileCopyUtils.copyToByteArray(catPicture)));
98                  }
```

**BlindSendFileAssignmentTest.java, line 79 (Path Manipulation)**

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Input Validation and Representation | | |

**Abstract:** BlindSendFileAssignmentTest.java 79 File() . .

**Source:** BlindSendFileAssignmentTest.java:79 Read this.webGoatHomeDirectory()

```
77      @Test
78      public void simpleXXEShouldNotWork() throws Exception {
79          File targetFile = new File(webGoatHomeDirectory, "/XXE/" +
                    webSession.getUserName() + "/secret.txt");
80          String content = "<?xml version=\"1.0\" standalone=\"yes\" ?><!DOCTYPE user
                    [<!ENTITY root SYSTEM \"file:///%s\"> ]><comment><text>&root;</text></comment>";
81          mockMvc.perform(MockMvcRequestBuilders.post("/xxe/blind")
```

**Sink:** BlindSendFileAssignmentTest.java:79 java.io.File.File()

```
77      @Test
78      public void simpleXXEShouldNotWork() throws Exception {
79          File targetFile = new File(webGoatHomeDirectory, "/XXE/" +
                    webSession.getUserName() + "/secret.txt");
80          String content = "<?xml version=\"1.0\" standalone=\"yes\" ?><!DOCTYPE user
                    [<!ENTITY root SYSTEM \"file:///%s\"> ]><comment><text>&root;</text></comment>";
```

```
81                  mockMvc.perform(MockMvcRequestBuilders.post("/xxe/blind")
```

## BlindSendFileAssignment.java, line 67 (Path Manipulation)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Input Validation and Representation | | |

**Abstract:** BlindSendFileAssignment.java 67 File() . .

**Source:** BlindSendFileAssignment.java:59 BlindSendFileAssignment(0)

```
57      private final Map<WebGoatUser, String> userToFileContents = new HashMap<>();
58
59      public BlindSendFileAssignment(@Value("${webgoat.user.directory}") String
webGoatHomeDirectory, CommentsCache comments) {
60          this.webGoatHomeDirectory = webGoatHomeDirectory;
61          this.comments = comments;
```

**Sink:** BlindSendFileAssignment.java:67 java.io.File.File()

```
65          var fileContents = "WebGoat 8.0 rocks... (" + randomAlphabetic(10) + ")";
66          userToFileContents.put(user, fileContents);
67          File targetDirectory = new File(webGoatHomeDirectory, "/XXE/" +
user.getUsername());
68          if (!targetDirectory.exists()) {
69              targetDirectory.mkdirs();
```

## LandingAssignment.java, line 61 (Resource Injection)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Input Validation and Representation | | |

**Abstract:** LandingAssignment.java 61 URI() ID . .

**Source:** LandingAssignment.java:61 javax.servlet.http.HttpServletRequest.getRequestURL()

```
59      @GetMapping("/WebWolf/landing/password-reset")
60      public ModelAndView openPasswordReset(HttpServletRequest request) throws
URISyntaxException {
61          URI uri = new URI(request.getRequestURL().toString());
62          ModelAndView modelAndView = new ModelAndView();
63          modelAndView.addObject("webwolfUrl", landingPageUrl);
```

**Sink:** LandingAssignment.java:61 java.net.URI.URI()

```
59      @GetMapping("/WebWolf/landing/password-reset")
60      public ModelAndView openPasswordReset(HttpServletRequest request) throws
URISyntaxException {
61          URI uri = new URI(request.getRequestURL().toString());
62          ModelAndView modelAndView = new ModelAndView();
63          modelAndView.addObject("webwolfUrl", landingPageUrl);
```

## ProfileUploadBase.java, line 75 (Path Manipulation)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Input Validation and Representation | | |

**Abstract:** ProfileUploadBase.java 75 File() . .

**Source:** ProfileZipSlip.java:30 ProfileZipSlip(0)

```
28  public class ProfileZipSlip extends ProfileUploadBase {
29
30      public ProfileZipSlip(@Value("${webgoat.server.directory}") String
webGoatHomeDirectory, WebSession webSession) {
31          super(webGoatHomeDirectory, webSession);
32      }
```

**Sink:** ProfileUploadBase.java:75 java.io.File.File()

```
73
74      protected byte[] getProfilePictureAsBase64() {
75          var profilePictureDirectory = new File(this.webGoatHomeDirectory,
"/PathTraversal/" + webSession.getUserName());
76          var profileDirectoryFiles = profilePictureDirectory.listFiles();
```

## Ping.java, line 52 (Path Manipulation)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |

| | |
|---|---|
| **Kingdom:** | Input Validation and Representation |
| **Abstract:** | Ping.java 52 File() . . |
| **Source:** | Ping.java:52 Read this.webGoatHomeDirectory() |

```
50          String logLine = String.format("%s %s %s", "GET", userAgent, text);
51          log.debug(logLine);
52          File logFile = new File(webGoatHomeDirectory, "/XXE/log" +
webSession.getUserName() + ".txt");
53          try {
54              try (PrintWriter pw = new PrintWriter(logFile)) {
```

| | |
|---|---|
| **Sink:** | Ping.java:52 java.io.File.File() |

```
50          String logLine = String.format("%s %s %s", "GET", userAgent, text);
51          log.debug(logLine);
52          File logFile = new File(webGoatHomeDirectory, "/XXE/log" +
webSession.getUserName() + ".txt");
53          try {
54              try (PrintWriter pw = new PrintWriter(logFile)) {
```

## MavenWrapperDownloader.java, line 108 (Resource Injection)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | MavenWrapperDownloader.java 108 URL() ID . . | | |
| **Source:** | MavenWrapperDownloader.java:62 java.util.Properties.load() | | |

```
60                  mavenWrapperPropertyFileInputStream = new
FileInputStream(mavenWrapperPropertyFile);
61                  Properties mavenWrapperProperties = new Properties();
62                  mavenWrapperProperties.load(mavenWrapperPropertyFileInputStream);
63                  url = mavenWrapperProperties.getProperty(PROPERTY_NAME_WRAPPER_URL,
url);
64              } catch (IOException e) {
```

| | |
|---|---|
| **Sink:** | MavenWrapperDownloader.java:108 java.net.URL.URL() |

```
106             });
107         }
108         URL website = new URL(urlString);
109         ReadableByteChannel rbc;
110         rbc = Channels.newChannel(website.openStream());
```

## ProfileZipSlip.java, line 47 (Path Manipulation)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | High | **Folder** | High |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | ProfileZipSlip.java 47 File() . . | | |
| **Source:** | ProfileZipSlip.java:30 ProfileZipSlip(0) | | |

```
28  public class ProfileZipSlip extends ProfileUploadBase {
29
30      public ProfileZipSlip(@Value("${webgoat.server.directory}") String
webGoatHomeDirectory, WebSession webSession) {
31          super(webGoatHomeDirectory, webSession);
32      }
```

| | |
|---|---|
| **Sink:** | ProfileZipSlip.java:47 java.io.File.File() |

```
45      private AttackResult processZipUpload(MultipartFile file) {
46          var tmpZipDirectory =
Files.createTempDirectory(getWebSession().getUserName());
47          var uploadDirectory = new File(getWebGoatHomeDirectory(), "/PathTraversal/" +
getWebSession().getUserName());
48          var currentImage = getProfilePictureAsBase64();
```

## SSRFTask2.java, line 53 (Resource Injection)

| | | | |
|---|---|---|---|
| **Fortify Priority:** | Low | **Folder** | Low |
| **Kingdom:** | Input Validation and Representation | | |
| **Abstract:** | SSRFTask2.java 53 URL() ID . . | | |
| **Source:** | SSRFTask2.java:46 completed(0) | | |

ot Copyright 2024 Open Text.

```
44                      @PostMapping("/SSRF/task2")
45                      @ResponseBody
46                      public AttackResult completed(@RequestParam String url) {
47                          return furBall(url);
48                      }
```

Sink: SSRFTask2.java:53 java.net.URL.URL()

```
51                          if (url.matches("http://ifconfig.pro")) {
52                              String html;
53                              try (InputStream in = new URL(url).openStream()) {
54                                  html = new String(in.readAllBytes(), StandardCharsets.UTF_8)
55                                          .replaceAll("\n","<br>"); // Otherwise the \n gets escaped in
the response
```

## ProfileZipSlip.java, line 60 (Path Manipulation)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Input Validation and Representation | | |

Abstract: ProfileZipSlip.java 60 File() . .

Source: ProfileZipSlip.java:57 java.util.zip.ZipFile.entries()

```
55
56                          ZipFile zip = new ZipFile(uploadedZipFile.toFile());
57                          Enumeration<? extends ZipEntry> entries = zip.entries();
58                          while (entries.hasMoreElements()) {
59                              ZipEntry e = entries.nextElement();
```

Sink: ProfileZipSlip.java:60 java.io.File.File()

```
58                          while (entries.hasMoreElements()) {
59                              ZipEntry e = entries.nextElement();
60                              File f = new File(tmpZipDirectory.toFile(), e.getName());
61                              InputStream is = zip.getInputStream(e);
62                              Files.copy(is, f.toPath(), StandardCopyOption.REPLACE_EXISTING);
```

## ProfileZipSlip.java, line 62 (Path Manipulation: Zip Entry Overwrite)

| | | | |
|---|---|---|---|
| Fortify Priority: | Medium | Folder | Medium |
| Kingdom: | Input Validation and Representation | | |

Abstract: ProfileZipSlip.java 62 copy()() .

Source: ProfileZipSlip.java:60 java.util.zip.ZipEntry.getName()

```
58                          while (entries.hasMoreElements()) {
59                              ZipEntry e = entries.nextElement();
60                              File f = new File(tmpZipDirectory.toFile(), e.getName());
61                              InputStream is = zip.getInputStream(e);
62                              Files.copy(is, f.toPath(), StandardCopyOption.REPLACE_EXISTING);
```

Sink: ProfileZipSlip.java:62 java.nio.file.Files.copy()

```
60                              File f = new File(tmpZipDirectory.toFile(), e.getName());
61                              InputStream is = zip.getInputStream(e);
62                              Files.copy(is, f.toPath(), StandardCopyOption.REPLACE_EXISTING);
63                          }
```

## FileServer.java, line 76 (Path Manipulation)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Input Validation and Representation | | |

Abstract: FileServer.java 76 File() . .

Source: FileServer.java:72 importFile(0)

```
70
71                      @PostMapping(value = "/fileupload")
72                      public ModelAndView importFile(@RequestParam("file") MultipartFile myFile) throws
IOException {
73                          var user = (WebGoatUser)
SecurityContextHolder.getContext().getAuthentication().getPrincipal();
74                          var destinationDir = new File(fileLocation, user.getUsername());
```

Sink: FileServer.java:76 java.io.File.File()

```
74                          var destinationDir = new File(fileLocation, user.getUsername());
```

ot Copyright 2024 Open Text.

```
75                  destinationDir.mkdirs();
76                  myFile.transferTo(new File(destinationDir, myFile.getOriginalFilename()));
77                  log.debug("File saved to {}", new File(destinationDir,
myFile.getOriginalFilename()));
```

## BlindSendFileAssignment.java, line 67 (Path Manipulation)

| | | | |
|---|---|---|---|
| Fortify Priority: | High | Folder | High |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | BlindSendFileAssignment.java 67 File() . . | | |

Source: UserService.java:30 org.owasp.webgoat.container.users.UserRepository.findByUsername()

```
28      @Override
29      public WebGoatUser loadUserByUsername(String username) throws
UsernameNotFoundException {
30          WebGoatUser webGoatUser = userRepository.findByUsername(username);
31          if (webGoatUser == null) {
32              throw new UsernameNotFoundException("User not found");
```

Sink: BlindSendFileAssignment.java:67 java.io.File.File()

```
65          var fileContents = "WebGoat 8.0 rocks... (" + randomAlphabetic(10) + ")";
66          userToFileContents.put(user, fileContents);
67          File targetDirectory = new File(webGoatHomeDirectory, "/XXE/" +
user.getUsername());
68          if (!targetDirectory.exists()) {
69              targetDirectory.mkdirs();
```

## ProfileZipSlip.java, line 53 (Path Manipulation)

| | | | |
|---|---|---|---|
| Fortify Priority: | Critical | Folder | Critical |
| Kingdom: | Input Validation and Representation | | |
| Abstract: | ProfileZipSlip.java 53 resolve() . . | | |

Source: ProfileZipSlip.java:36 uploadFileHandler(0)

```
34      @PostMapping(value = "/PathTraversal/zip-slip", consumes = ALL_VALUE, produces =
APPLICATION_JSON_VALUE)
35      @ResponseBody
36      public AttackResult uploadFileHandler(@RequestParam("uploadedFileZipSlip")
MultipartFile file) {
37          if (!file.getOriginalFilename().toLowerCase().endsWith(".zip")) {
38              return failed(this).feedback("path-traversal-zip-slip.no-zip").build();
```

Sink: ProfileZipSlip.java:53 java.nio.file.Path.resolve()

```
51
52          try {
53              var uploadedZipFile = tmpZipDirectory.resolve(file.getOriginalFilename());
54              FileCopyUtils.copy(file.getBytes(), uploadedZipFile.toFile());
```

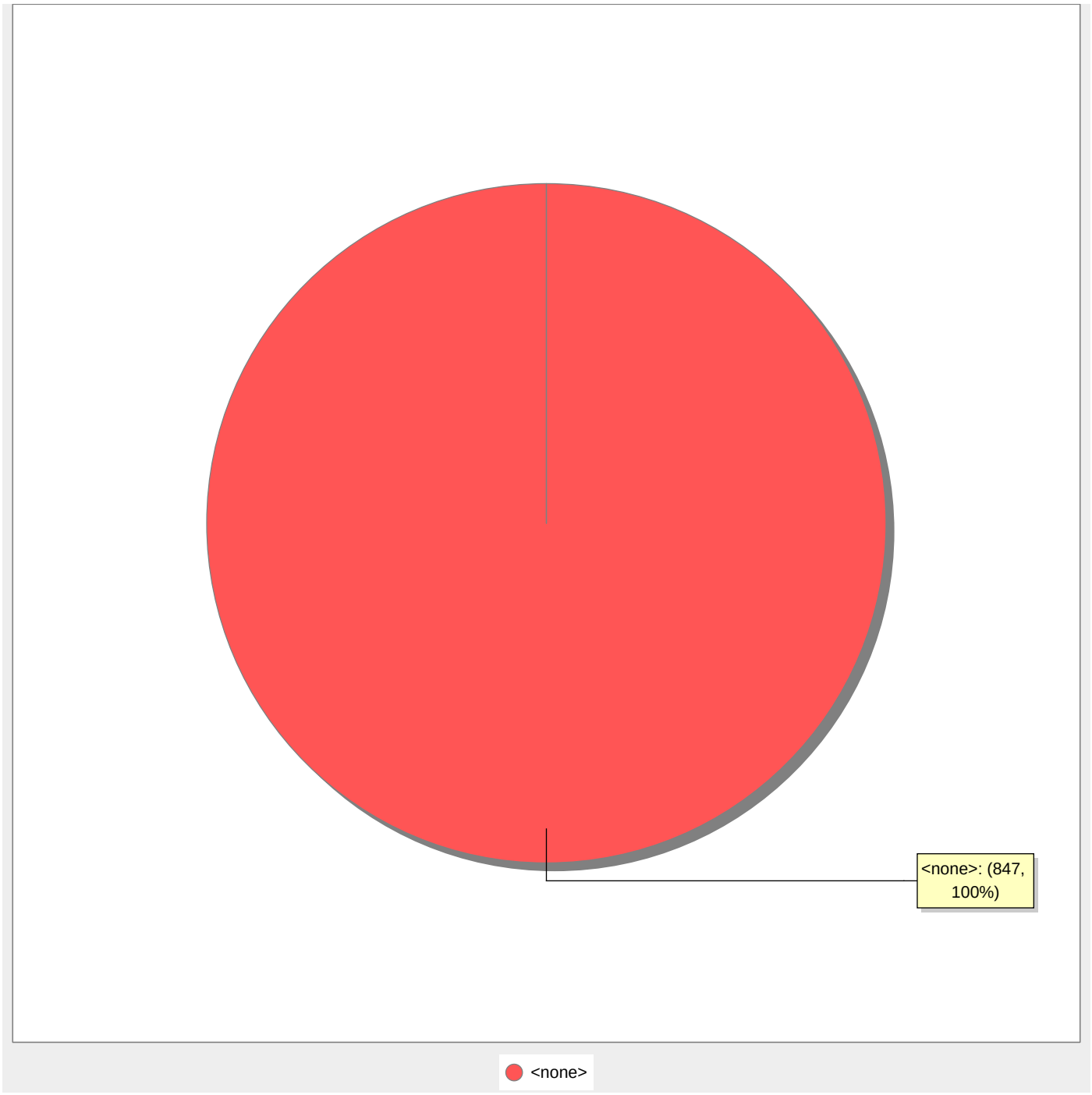
ot Copyright 2024 Open Text.

| Issue Count by Category | |
|---|---|
| Issues by 49 | |
| <none> | 223 |
| 07.02. API | 3 |
| 05.05. | 2 |
| 04.03. | 11 |
| 04.01. , 06.02. | 50 |
| 04.01. | 31 |
| 02.15. | 1 |
| 02.11. | 2 |
| 02.09. | 2 |
| 02.08. | 19 |
| 02.06. | 52 |
| 02.05. | 17 |
| 02.04. | 1 |
| 02.03. | 15 |
| 01.15. , 07.01. DNS lookup | 1 |
| 01.12. | 2 |
| 01.11. | 298 |
| 01.08. XML | 3 |
| 01.07. URL | 16 |
| 01.06. | 16 |
| 01.05. | 1 |
| 01.04. | 40 |
| 01.03. | 41 |

Copyright 2024 Open Text.

# Issue Breakdown by Analysis

## Issues by Analysis




Copyright 2024 Open Text.