

**SECURE PROGRAMMING
PROJECT - PART 2**

**WEB APPLICATION FOR
COURSE GRADEBOOK**

Version 2.0
Date: 12/09/2018

Submitted by:
1. Shloak Agarwal
2. Aman Arora
3. Shweta Baskaran
4. Vishesh Kumar Dwivedi
5. Tejasvini Ravi Patil
6. Anirudh Raghavendrara Deshpande
7. Himanshu Ajay Singh
8. Balasubramaniam Theetharappan
9. Sarita Maruti Tigadi

Indexes

1- Vignettes of the System	3
2- Execution of Scenarios	8
3- Software Version Description	11
4- Deliverables form Part 1	12
5- Static Code Analysis Evidences	13
6- Project Management	14
7- Testing Report	16
8- Security Assessment/Compliance	34
9- References	42

Vignettes of the system

- Input/output validation
 - Regular Expression check:
Checks are in place for validating the format of the inputs entered by the user - the login ID and password.

```
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    HttpSession session = request.getSession();
    if (session.getAttribute(Constants.SP_USERNAME) != null
        && Constants.SP_USERTYPE_PROFESSOR.equals(session.getAttribute(Constants.SP_USERTYPE))) {
        ProfessorDao dao = new ProfessorDao();
        Integer profID = new Integer(session.getAttribute(Constants.SP_USERID).toString());
        String id = request.getParameter("ssid");
        String psid = request.getParameter("psid");
        String grade = request.getParameter("grade");
        if(id.matches("^\\d+$") &&
           grade.matches("^\\d{1}$")) {
            int ssid = Integer.parseInt(id);
            new UserDao().updateGrade(ssid, profID, grade);
            LOG.info("grade updated for studentid "+ssid+" by professor id "+profID);
            response.sendRedirect("/sp/professor/grades?psid="+psid);
        }
    }
}
```

- Existence in database:
Checks are made to ensure that the login details entered by the user are valid and exist in the database. If not, suitable error messages are displayed.

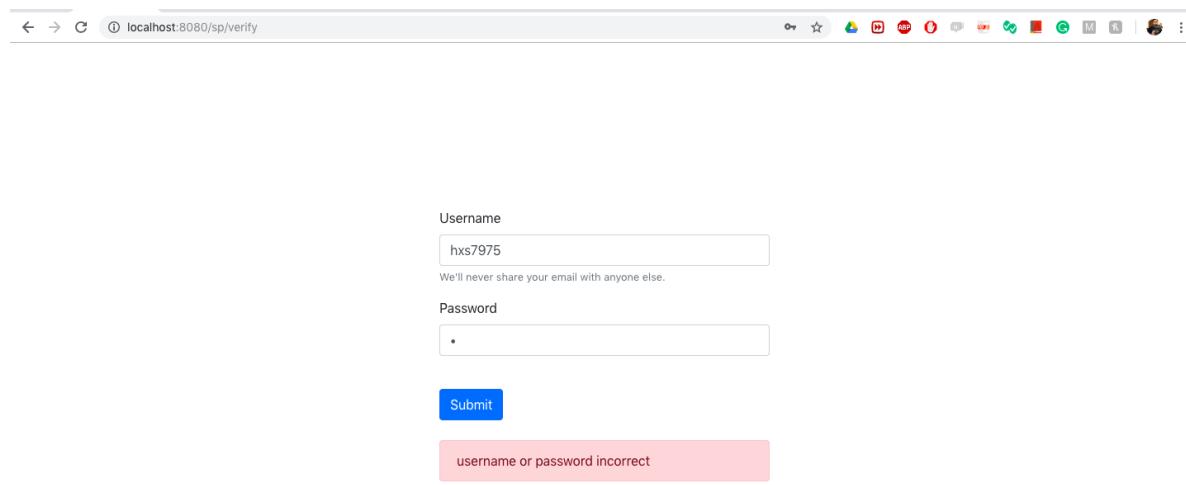


Figure 1

- Use of centralized logging and logging of various application events
A centralized login framework called log4j is used which is a logging package for Java. A centralized framework makes it easy to:
 - Provide one consistent view of the system reflected in the logs.
 - Facilitate changes, such as moving logging to another machine, switching from logging to a file to logging to a database, or updating validation or privacy measures.

- This is also preferred as it includes timestamps by default.

The events logged in the grade book are:

- User Login - authentication of user by Login ID and password
- Update of grades - by Professor
- User Logout - end of session
- This log is maintained on the server side and is not available to be accessed by students or professor. Only the database administrator and the website admin can access it.
- Both success as well as failure messages that are generated during the use of the grade book application is logged.

- List all events that your application specifically audits

The following is the list of events which are audited by the application using log4j.properties

- successful login
- grade updated
- successful logout

- Exception/error handling

Exceptions are properly handled by using the try-catch statement block. Appropriate ways of handling exceptions are implemented for each exception thrown.

```
protected int update(String query, Object[] data) {
    createConnection();
    PreparedStatement preparedStatement;
    int i = 0;
    try {
        preparedStatement = connection.prepareStatement(query);
        for (int j = 0; j < data.length; j++) {
            Object object = data[j];
            if (object instanceof Integer) {
                preparedStatement.setInt(j + 1, ((Integer) object));
            } else if (object instanceof String) {
                preparedStatement.setString(j + 1, object.toString());
            }
        }
        i = preparedStatement.executeUpdate();
    } catch (Exception e) {
        LOGGER.error(e);
    }
    closeConnection();
    return i;
}
```

- Sample log file entries

The centralized log maintained on the server side is updated whenever any important event takes place in the client end. The app.log file is generated by the application has the following log entries.

```

address => 0:0:0:0:0:0:1
2018-12-09 14:32:26 INFO  SPFFilter:38 - valid request from ip-
address => 0:0:0:0:0:0:1
2018-12-09 14:32:27 INFO  SPFFilter:70 - successfull login
request for ttj7975 from ip-address => 0:0:0:0:0:0:1
2018-12-09 14:32:27 INFO  SPFFilter:48 - valid request from ip-
address => 0:0:0:0:0:0:1
2018-12-09 14:32:29 INFO  SPFFilter:48 - valid request from ip-
address => 0:0:0:0:0:0:1
2018-12-09 14:32:34 INFO  SPFFilter:48 - valid request from ip-
address => 0:0:0:0:0:0:1
2018-12-09 14:32:34 INFO  ProfessorController:87 - grade
updated for studentid5 by professor id 1
2018-12-09 14:32:34 INFO  SPFFilter:48 - valid request from ip-
address => 0:0:0:0:0:0:1
2018-12-09 14:32:40 INFO  SPFFilter:48 - valid request from ip-
address => 0:0:0:0:0:0:1
2018-12-09 14:32:40 INFO  ProfessorController:87 - grade
updated for studentid10 by professor id 1
2018-12-09 14:32:40 INFO  SPFFilter:48 - valid request from ip-
address => 0:0:0:0:0:0:1
2018-12-09 16:56:28 ERROR SPFFilter:43 - invalid request from
ip-address => 0:0:0:0:0:0:1
2018-12-09 16:56:28 INFO  SPFFilter:38 - valid request from ip-
address => 0:0:0:0:0:0:1
2018-12-09 16:58:33 INFO  SPFFilter:38 - valid request from ip-
address => 0:0:0:0:0:0:1
2018-12-09 16:58:33 INFO  SPFFilter:70 - successfull login
request for ttj7975 from ip-address => 0:0:0:0:0:0:1
2018-12-09 16:58:33 INFO  SPFFilter:48 - valid request from ip-
```

```

- Session management

A new session begins every time a user is validated and the session identifier is generated on the server side. This session identifier is no longer valid and the session expires once the user logs out of the system.

A timestamp is set for maximum session lifetime and is fixed to be for 30 minutes. If inactivity on the web application occurs for more than 30 minutes, the session expires.

```

<!-- ===== Default Session Configuration ===== -->
<!-- You can set the default session timeout (in minutes) for all newly -->
<!-- created sessions by modifying the value below. -->

<session-config>
 <session-timeout>30</session-timeout>
</session-config>

```

```

} else {
 LOG.info("successful login request for "+clientUser.getUserName()+" from ip-address => "+request.getRemoteAddress());
 dbUser.setLoginAttempts(0);
 userDao.update(dbUser);
 request.setAttribute("error", "login successful");
 if(dbUser.getProfessorId()!=0) {
 Professor professor=new Professor();
 professor.setProfessorId(dbUser.getProfessorId());
 professor = new ProfessorDao().getOne(professor);
 HttpSession session = request.getSession();

 session.setAttribute(Constants.SP_USERNAME, professor.getName());
 session.setAttribute(Constants.SP_USERID, professor.getProfessorId());
 session.setAttribute(Constants.SP_USERTYPE, Constants.SP_USERTYPE_PROFESSOR);
 response.sendRedirect(request.getContextPath() + "/professor/dashboard");
 }else if(dbUser.getStudentId()!=0) {
 Student student=new Student();
 student.setStudentId(dbUser.getStudentId());
 student = new StudentDao().getOne(student);
 HttpSession session = request.getSession();

 session.setAttribute(Constants.SP_USERNAME, student.getName());
 session.setAttribute(Constants.SP_USERID, student.getStudentId());
 session.setAttribute(Constants.SP_USERTYPE, Constants.SP_USERTYPE_STUDENT);
 response.sendRedirect(request.getContextPath() + "/student/dashboard");
 }
}

```

- Authentication and authorization

A user is validated using the login page where user enters login ID and password. Depending on the login details (as verified from the backend database), further views of the user differs and different privileges are granted. For example, if the user login details match with an entry in the database that is of a Student's, the grade edit feature is disabled and the student is authorized only to view the grades. Also, each student is restricted to viewing only his/her grade and not grade of any other student. If the login details are authenticated to be of a Professor, he/she can view grades of all students and also edit/update the grades. These changes will be logged.

- Cryptography

SALT is used for encrypting the user password when it is stored and transmitted from client to server while user logs in. It is a random data that is used as an additional input to a one-way function that "hashes" data, a password or passphrase. Hence it is used to safeguard passwords in storage.

A new salt is randomly generated for each password.

The salt and the password (or its version after Key stretching) are concatenated and processed with a cryptographic hash function, and the resulting output (but not the original password) is stored with the salt in a user database. Hashing allows for later authentication without keeping and therefore risking the plaintext password in the event that the authentication data store is compromised.

Here, SHA1 is used for key generation.

```

public boolean check(User clientUserObject) throws Exception {
 String[] saltAndHash = password.split("\\$");
 if (saltAndHash.length != 2) {
 throw new IllegalStateException("The stored password must have the form 'salt$hash'");
 }
 String hashOfInput = hash(clientUserObject.getPassword(), Base64.decodeBase64(saltAndHash[0]));
 return hashOfInput.equals(saltAndHash[1]);
}

private String hash(String password, byte[] salt) throws Exception {
 if (password == null || password.length() == 0)
 throw new IllegalArgumentException("Empty passwords are not supported.");
 SecretKeyFactory f = SecretKeyFactory.getInstance("PBKDF2WithHmacSHA1");
 SecretKey key = f.generateSecret(new PBEKeySpec(password.toCharArray(), salt, iterations, desiredKeyLen));
 return Base64.encodeBase64String(key.getEncoded());
}

```

- Privilege management

Each user is associated with a Role ID which has a specific set of Privileges. User cannot perform operations for which he/she does not have privileges. Each time user logins into the application, the respective privileges are assigned to the user

The log files and database information will be accessible only by the system administrator and not the Student or Professor.

- Mitigations for buffer overflows

Java is designed to avoid the buffer overflow by checking the bounds of a buffer (like an array) and preventing any access beyond those bounds. Hence it is a safe language and takes care of buffer overflows by itself.

- Mitigations for Cross Site Request Forgeries (CSRF)

Cross-site request forgery, is a type of malicious exploit of a website where unauthorized commands are transmitted from a user that the web application trusts. In our application we are using strong encryption and functions like SHA-1 and SALT that will generate hashes for password and paraphrase and SHA-1 used for key generation before transmitting to server.

- Mitigations for injection attacks

SQL injection is a code injection technique, used to attack data-driven applications, in which nefarious SQL statements are inserted into an entry field for execution. We prevent this type of an attack in our application by making use of SQL prepared statements and incorporating whitelisting data on server side.

```

protected int update(String query, Object[] data) {
 createConnection();
 PreparedStatement preparedStatement;
 int i = 0;
 try {
 preparedStatement = connection.prepareStatement(query);
 for (int j = 0; j < data.length; j++) {
 Object object = data[j];
 if (object instanceof Integer) {
 preparedStatement.setInt(j + 1, ((Integer) object));
 } else if (object instanceof String) {
 preparedStatement.setString(j + 1, object.toString());
 }
 }
 i = preparedStatement.executeUpdate();
 } catch (Exception e) {
 LOGGER.error(e);
 }
 closeConnection();
 return i;
}

```

- Handling of sensitive information

All the data with regards to the grade book application is stored securely on the database with only access to the system administrator. Passwords are encrypted using the SHA1 SALT encryption method so that passwords of all users are kept safe when stored and transmitted. Session ID is also a sensitive data and is randomly generated on successful authentication of the user.

## Execution of scenarios

- Scenarios:
  1. User (Student/Professor) will login to the system by entering the valid credentials.
  2. Student will login to the system to retrieve the course details (term, subjects, grades etc.).
  3. Professor will login to the system to retrieve and update course details.

- Screenshots of system execution:

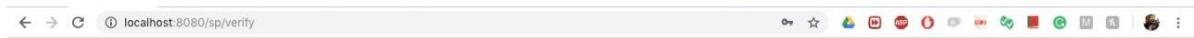
### LOGIN:

localhost:8080/sp/

Username  
hxs7975  
We'll never share your email with anyone else.

Password  
•

Submit



Username

hx57975

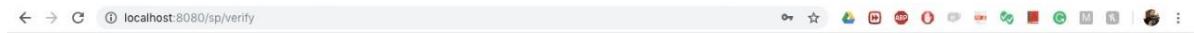
We'll never share your email with anyone else.

Password

\*

Submit

username or password incorrect



Username

hx57975

We'll never share your email with anyone else.

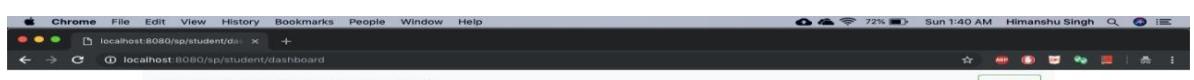
Password

\*

Submit

maximus limit exceeded, please contact admin

## STUDENT DASHBOARD:



FALL 18

No of Subjects 3

[View Grades](#)

[Logout](#)



Subject Name

SECURE PROGRAMMING

DBMS

SDP

Professor Name

Thomas Trey Johns

Thomas Trey Johns

Thomas Trey Johns

Grade

D

F

A

[Logout](#)

## PROFESSOR DASHBOARD:

The screenshot shows a grid-based dashboard for managing courses. The grid is organized into three columns and three rows. The first column contains 'FALL 18' and 'SPRING 19'. The second column contains 'SPRING 19', 'FALL 18', and an empty row. The third column contains 'FALL 18', 'DBMS', and 'JAVA'. Each cell in the grid contains course names and a 'View Grades' button.

|                             |                             |                             |
|-----------------------------|-----------------------------|-----------------------------|
| FALL 18                     | SPRING 19                   | FALL 18                     |
| SECURE PROGRAMMING          | SECURE PROGRAMMING          | DBMS                        |
| <a href="#">View Grades</a> | <a href="#">View Grades</a> | <a href="#">View Grades</a> |
| SPRING 19                   | FALL 18                     | SPRING 19                   |
| DBMS                        | SDP                         | JAVA                        |
| <a href="#">View Grades</a> | <a href="#">View Grades</a> | <a href="#">View Grades</a> |
| SPRING 19                   |                             |                             |
| NETWORKING                  |                             |                             |
| <a href="#">View Grades</a> |                             |                             |

The screenshot shows a modal dialog titled 'Update Grade' over a list of students. The student 'Himanshu Ajay Singh' is selected, showing details: Name (himanshuajay.singh@mavs.uta.edu), Email (Himanshu Ajay Singh), and Grade (A). The 'Edit' button is highlighted. The 'Update' button is visible at the bottom of the dialog.

| Student Name        | Email                           | Grade |
|---------------------|---------------------------------|-------|
| Himanshu Ajay Singh | himanshuajay.singh@mavs.uta.edu | A     |

Update Grade

Name: himanshuajay.singh@mavs.uta.edu  
Email: Himanshu Ajay Singh  
Grade: A

[Close](#) [Update](#)

The screenshot shows a modal dialog titled 'Update Grade' over a list of students. The student 'Himanshu Ajay Singh' is selected, showing details: Name (himanshuajay.singh@mavs.uta.edu), Email (Himanshu Ajay Singh), and Grade (F). The 'Edit' button is highlighted. The 'Update' button is visible at the bottom of the dialog.

| Student Name        | Email                           | Grade |
|---------------------|---------------------------------|-------|
| Himanshu Ajay Singh | himanshuajay.singh@mavs.uta.edu | F     |

Update Grade

Name: himanshuajay.singh@mavs.uta.edu  
Email: Himanshu Ajay Singh  
Grade: F

[Close](#) [Update](#)

The image consists of three vertically stacked screenshots of a web application running in a Chrome browser on a Mac OS X system. The application is a professor's grade management tool.

**Screenshot 1:** Shows a student record for "Himanshu Ajay Singh" with an email of "himanshuajay.singh@mavs.uta.edu" and a current grade of "F". A green "Edit" button is visible. A modal dialog titled "Update Grade" is open, showing the student's name and email, and a dropdown menu where the grade "A" is selected. Buttons for "Close" and "Update" are at the bottom of the modal.

**Screenshot 2:** Shows the same student record after the grade has been updated. The grade now displays as "A" instead of "F".

**Screenshot 3:** Shows the same student record again, but with the grade changed back to "F".

## Software Version Description

- Final product release version: There is only one branch which is the master branch.
- List of issues resolved since prior release  
The links for the following tasks is as shown below:  
<https://app.asana.com/0/939282246374097/939664138627054/f>

- List of known issues that weren't resolved  
There is a total of 27 issues generated by FindBugs which are 'of concern', which are not fixed
- Dependencies we have used for the application to operate and their version numbers
  - Version Control: GitHub
  - IDE: Eclipse ide as it supports Java EE and its open source
  - Database: MySQL (relational and open source)
  - Compilers: Java 1.8
  - Static analysis tool: find bug
  - Logging: log4j
  - Programming language: java 1.8, Html, JavaScript
  - Application server: Apache Tomcat 8
  - Technology: Java EE (servlets and JSP)
  - Frameworks: Bootstrap 4.1.3, Apache Standard Taglibs 1.2.3 (for avoiding script lets)

## Deliverables from part 1

The report of project part 1 is in Zip folder and is updated accordingly.

## Static code analysis evidence

- Manual code review

Static Code Analysis Evidence including the following:

| File name & line number         | Defect Description                                   | Disposition | Fix by          |
|---------------------------------|------------------------------------------------------|-------------|-----------------|
| ProfessorController,<br>Line 46 | Type casting using new Integer object, used parseInt | Fixed       | Himanshu Sing   |
| ProfessorDao,<br>Line 38        | Prepared statement object was not closed             | Fixed       | Tejasvini Patil |

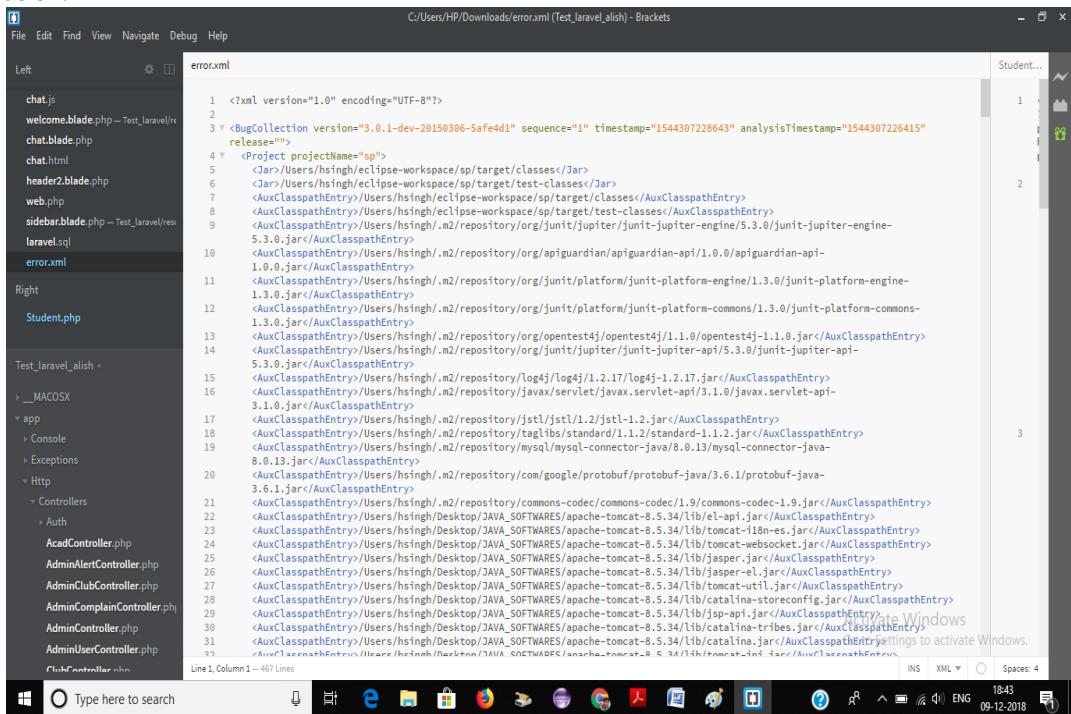
Snippet from ProfessorController.java:

- Automated

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                                       |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------|
| Bug: Null passed for non-null parameter of resultSetToArrayList(ResultSet) in com.uta.sp.dao.JdbcConnection.getMany(String) This method call passes a null value for a non-null method parameter. Either the parameter is annotated as a parameter that should always be non-null, or analysis has shown that it will always be dereferenced.<br>Rank: Scary (8), confidence: Normal Pattern:<br>NP_NULL_PARAM_DEREF Type: NP, Category: CORRECTNESS (Correctness) | fixed                                 |
| Bug: HTTP parameter directly written to HTTP header output in com.uta.sp.controller.ProfessorController.doPost(HttpServletRequest, HttpServletResponse)                                                                                                                                                                                                                                                                                                            | fixed                                 |
| Bug: HTTP parameter directly written to HTTP header output in com.uta.sp.controller.ProfessorController.doPost(HttpServletRequest, HttpServletResponse)                                                                                                                                                                                                                                                                                                            | fixed                                 |
| Bug: Unresolvable reference to javax.crypto.SecretKey by com.uta.sp.dao.JdbcConnectionTest                                                                                                                                                                                                                                                                                                                                                                         | In the test file so no need to change |
| Bug: Unresolvable reference to javax.crypto.SecretKey by com.uta.sp.dao.JdbcConnectionTest                                                                                                                                                                                                                                                                                                                                                                         | In the test file so no                |

|                                                                                       |                                       |
|---------------------------------------------------------------------------------------|---------------------------------------|
|                                                                                       | need to change                        |
| Unresolvable reference to javax.crypto.SecretKey by com.uta.sp.dao.JdbcConnectionTest | In the test file so no need to change |

The following figure is the screenshot of **error.xml** file, it is a vulnerability report of the system. The report is generated using **FindBugs** vulnerability tool.



The screenshot shows the Brackets IDE interface with the file 'error.xml' open. The code content is as follows:

```

<?xml version="1.0" encoding="UTF-8"?>
<BugCollection version="3.0.1-dev-20150306-5afe4d1" sequence="1" timestamp="1544307228643" analysisTimestamp="1544307226415" release="">
 <Project projectName="sp">
 <Jar>/Users/hsingh/eclipse-workspace/sp/target/classes</Jar>
 <Jar>/Users/hsingh/eclipse-workspace/sp/target/test-classes</Jar>
 <AuxClasspathEntry>/Users/hsingh/eclipse-workspace/sp/target/classes</AuxClasspathEntry>
 <AuxClasspathEntry>/Users/hsingh/eclipse-workspace/sp/target/test-classes</AuxClasspathEntry>
 <AuxClasspathEntry>/Users/hsingh/.m2/repository/org/junit/jupiter/junit-jupiter-engine/5.3.0/jar</AuxClasspathEntry>
 <AuxClasspathEntry>/Users/hsingh/.m2/repository/org/apiguardian/apiguardian-api/1.0.0/apiguardian-api-1.0.0.jar</AuxClasspathEntry>
 <AuxClasspathEntry>/Users/hsingh/.m2/repository/org/junit/platform/junit-platform-engine/1.3.0/junit-platform-engine-1.3.0.jar</AuxClasspathEntry>
 <AuxClasspathEntry>/Users/hsingh/.m2/repository/org/junit/platform/junit-platform-commons/1.3.0/junit-platform-commons-1.3.0.jar</AuxClasspathEntry>
 <AuxClasspathEntry>/Users/hsingh/.m2/repository/org/opentest4j/opentest4j/1.1.0/opentest4j-1.1.0.jar</AuxClasspathEntry>
 <AuxClasspathEntry>/Users/hsingh/.m2/repository/org/junit/jupiter/junit-jupiter-api/5.3.0/junit-jupiter-api-5.3.0.jar</AuxClasspathEntry>
 <AuxClasspathEntry>/Users/hsingh/.m2/repository/log4j/log4j/1.2.17/log4j-1.2.17.jar</AuxClasspathEntry>
 <AuxClasspathEntry>/Users/hsingh/.m2/repository/java/servlet-api/3.1.0/java-servlet-api-3.1.0.jar</AuxClasspathEntry>
 <AuxClasspathEntry>/Users/hsingh/.m2/repository/com/google/protobuf/protobuf-java/3.6.1/protobuf-java-3.6.1.jar</AuxClasspathEntry>
 <AuxClasspathEntry>/Users/hsingh/.m2/repository/commons-codec/commons-codec/1.9/commons-codec-1.9.jar</AuxClasspathEntry>
 <AuxClasspathEntry>/Users/hsingh/Desktop/JAVA_SOFTWARES/apache-tomcat-8.5.34/lib/el-api.jar</AuxClasspathEntry>
 <AuxClasspathEntry>/Users/hsingh/Desktop/JAVA_SOFTWARES/apache-tomcat-8.5.34/lib/tomcat-18n-es.jar</AuxClasspathEntry>
 <AuxClasspathEntry>/Users/hsingh/Desktop/JAVA_SOFTWARES/apache-tomcat-8.5.34/lib/tomcat-websocket.jar</AuxClasspathEntry>
 <AuxClasspathEntry>/Users/hsingh/Desktop/JAVA_SOFTWARES/apache-tomcat-8.5.34/lib/jasper.jar</AuxClasspathEntry>
 <AuxClasspathEntry>/Users/hsingh/Desktop/JAVA_SOFTWARES/apache-tomcat-8.5.34/lib/jasper-el.jar</AuxClasspathEntry>
 <AuxClasspathEntry>/Users/hsingh/Desktop/JAVA_SOFTWARES/apache-tomcat-8.5.34/lib/tomcat-util.jar</AuxClasspathEntry>
 <AuxClasspathEntry>/Users/hsingh/Desktop/JAVA_SOFTWARES/apache-tomcat-8.5.34/lib/catalina-storeconfig.jar</AuxClasspathEntry>
 <AuxClasspathEntry>/Users/hsingh/Desktop/JAVA_SOFTWARES/apache-tomcat-8.5.34/lib/jsp-api.jar</AuxClasspathEntry>
 <AuxClasspathEntry>/Users/hsingh/Desktop/JAVA_SOFTWARES/apache-tomcat-8.5.34/lib/catalina-tribes.jar</AuxClasspathEntry>
 <AuxClasspathEntry>/Users/hsingh/Desktop/JAVA_SOFTWARES/apache-tomcat-8.5.34/lib/catalina.jar</AuxClasspathEntry>
 </Project>

```

The error.xml file is included in the zip file.

## Project management

Asana is use in project management part. It is an application designed to help teams organize, track, and manage their work. Below are the screenshots of the dashboard. As the project was scheduled, task was assigned and was completed on accordingly to the time assigned.

The screenshot shows the Asana interface in 'Board' mode for a project titled 'Secure Programming'. The board has four columns: 'To-Do', 'In Progress', 'Testing', and 'Done'. Each column contains several tasks with their names, descriptions, and due dates. The tasks are color-coded by owner.

| To-Do                    | In Progress                              | Testing | Done                  |
|--------------------------|------------------------------------------|---------|-----------------------|
| Database Design Checking | Static Analysis                          | HTML    | Requirement Gathering |
| SQL Checking             |                                          |         | Application Planning  |
| Testing Code             | Security Feature Report (Project Report) |         |                       |
| Report                   | Database Design                          |         |                       |
| Security                 | Back-end                                 |         |                       |

Different column were made like To-Do, In Progress, Testing and Done.

This screenshot shows the Asana interface in 'Board' mode for the same 'Secure Programming' project. The board structure is identical to the first one, with four columns: 'To-Do', 'In Progress', 'Testing', and 'Done'. The tasks listed are different, reflecting the progression of work.

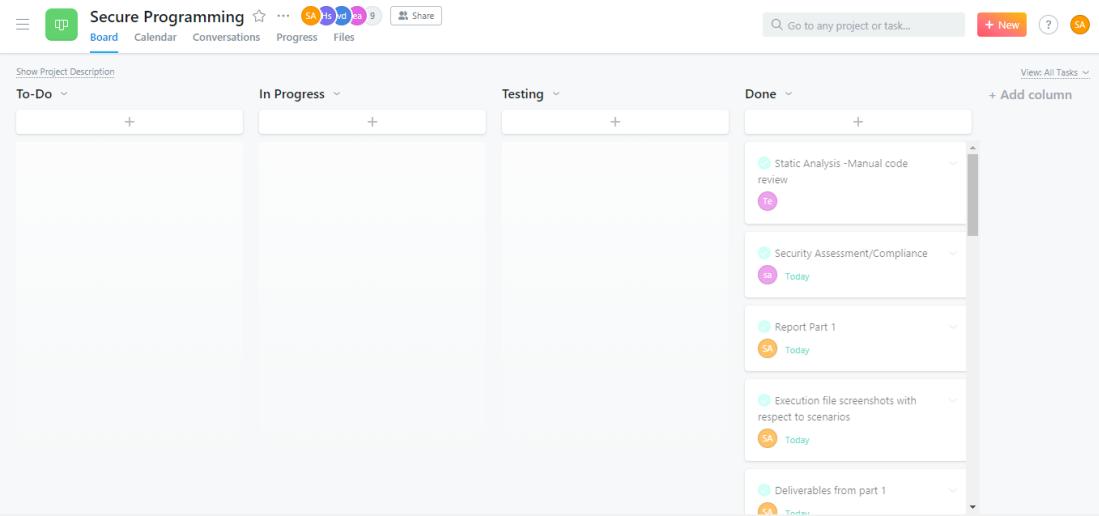
| To-Do                                                | In Progress                        | Testing             | Done                               |
|------------------------------------------------------|------------------------------------|---------------------|------------------------------------|
| Vulnerability scan Report                            | Final Project Report               | Penetration Testing | Requirement Gathering              |
| Load Testing                                         | Deliverables from part 1           | Static Analysis     | Security vignette (part of Report) |
| Coding                                               | Software Version Description (SVD) | HTML                | Application Planning               |
| Testing report - Load Testing                        | Report Part 1                      | Security            |                                    |
| Execution file screenshots with respect to scenarios | ui bug fix and logging             | Back-end            |                                    |

As the task proceeds, the task goes to different columns.

This screenshot shows the Asana interface in 'Board' mode for the 'Secure Programming' project. The board now includes a fifth column, 'Done', which contains tasks that have been completed. The tasks in the other columns are the same as the previous screenshots, showing the progression of work.

| To-Do                                                | In Progress                        | Testing                       | Done                 |
|------------------------------------------------------|------------------------------------|-------------------------------|----------------------|
| Vulnerability scan Report                            | Final Project Report               | Penetration Testing           | Application Planning |
| Load Testing                                         | Deliverables from part 1           | Static Analysis               | Database Design      |
| Coding                                               | Software Version Description (SVD) | HTML                          | HTML                 |
| Testing report - Load Testing                        |                                    | Security                      | Back-end             |
| Execution file screenshots with respect to scenarios |                                    | Testing report - Load Testing | Coding               |

In last, task completes the owner mark it as complete.



## Testing report

- Security testing result

We have done all component testing of the system and here as a measure of security, we have included security snippet codes. For example, One of the most important part of the system as a user and as a security perspective is entering the system (signing in to the system).For that we have applied regular expression check on the input given by the user .

```
/*
protected void doPost(HttpServletRequest request, HttpServletResponse response)
 throws ServletException, IOException {
 HttpSession session = request.getSession();
 if (session.getAttribute(Constants.SP_USERNAME) != null
 && Constants.SP_USERTYPE_PROFESSOR.equals(session.getAttribute(Constants.SP_USERTYPE))) {
 ProfessorDao dao = new ProfessorDao();
 Integer profID = new Integer(session.getAttribute(Constants.SP_USERID).toString());
 String id = request.getParameter("ssid");
 String psid = request.getParameter("psid");
 String grade = request.getParameter("grade");
 if(id.matches("[0-9]+") &&
 grade.matches("[A-F]{1}")) {
 int ssid = Integer.parseInt(id);
 new UserDAO().updateGrade(ssid, profID, grade);
 LOG.info("grade updated for studentid "+ssid+" by professor id "+profID);
 response.sendRedirect("/sp/professor/grades?psid="+psid);
 }
 }
}
```

The regular expression check will help in proper input validation. We already did the penetration testing, preventing the brute-force attack and we have the result below in the penetration testing section below. Still, in this section, validation part of the system is shown by the following figures.

A screenshot of a web browser window. The address bar shows the URL `localhost:8080/sp/verify`. The page displays a login form with two fields: 'Username' containing 'hxs7975' and 'Password' containing a single asterisk ('\*'). Below the fields is a blue 'Submit' button. A pink error message box at the bottom contains the text 'username or password incorrect'.

A screenshot of a web browser window. The address bar shows the URL `localhost:8080/sp/verify`. The page displays a login form with two fields: 'Username' containing 'hxs7975' and 'Password' containing a single asterisk ('\*'). Below the fields is a blue 'Submit' button. A pink error message box at the bottom contains the text 'maximus limit exceeded, please contact admin'.

- Penetration testing result

#### Penetration Testing using Burp Suite

With this we test the grade book application for the login protection by carrying out the brute force attack. Turning on the intercept after passing the credentials in our application we see the post request being intercepted. The two values listed are username and password, thus we brute force these values. In the burp suite we make use of the intruder which allows us to edit and manipulate the requests.

The screenshot shows the Burp Suite interface. In the top right, a browser window displays a login page for 'localhost:8081/sp/' with fields for 'Username' (containing 'aaa') and 'Password' (containing '\*\*\*\*'). Below the browser is a status bar showing 'Apache Tomcat/8.5.35'. In the bottom left, the Burp Suite interface has tabs for Target, Proxy, Spider, Scanner, Intruder, Repeater, and Sequencer. The Proxy tab is selected, showing a list of requests. One request is highlighted with the URL 'http://localhost:8081 [127.0.0.1]'. Below the list are buttons for Forward, Drop, Intercept is on, and Action. Underneath these buttons are tabs for Raw, Params, Headers, and Hex. The Raw tab displays a POST request with the following headers and body:

```

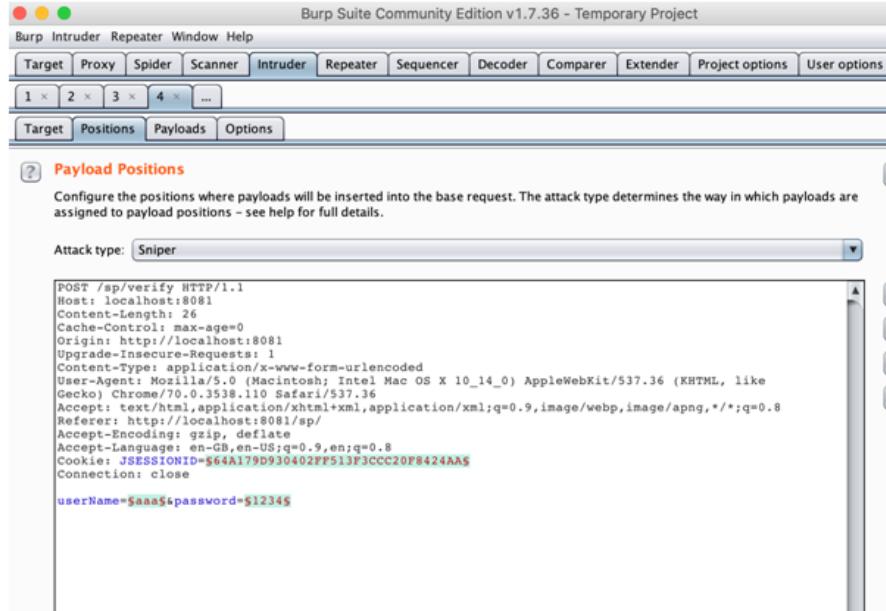
POST /sp/verify HTTP/1.1
Host: localhost:8081
Content-Length: 26
Cache-Control: max-age=0
Origin: http://localhost:8081
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_0) AppleWebKit/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Cookie: JSESSIONID=64A179D930402FF513F3CCC20F8424AA
Connection: close

```

The body of the request contains the parameters 'username=aaa&password=1234'.

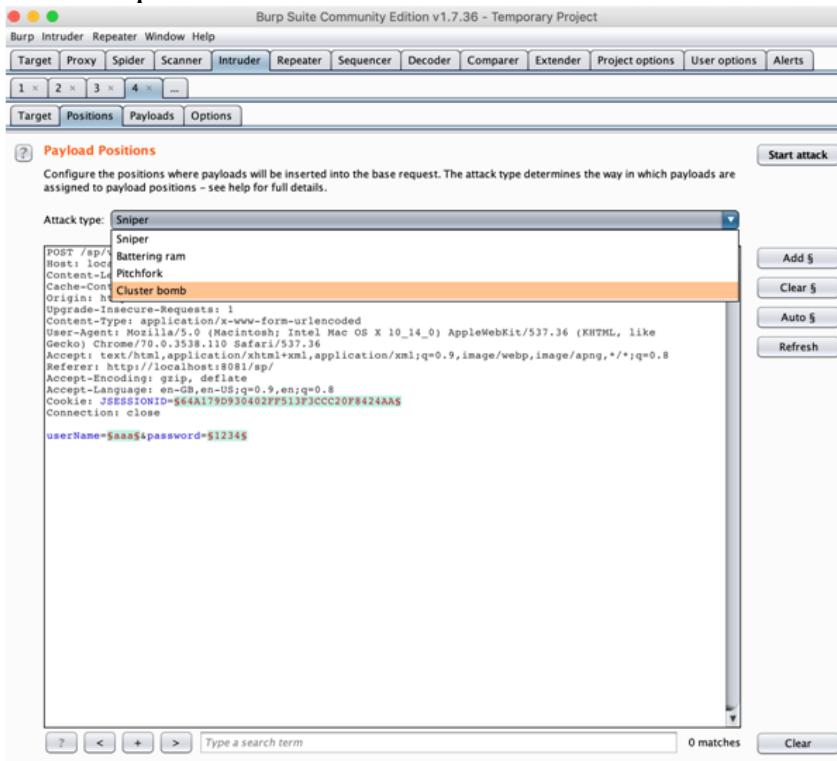
Sending the request to the intruder and hit forward.

The screenshot shows the Burp Suite interface with the same setup as the previous one. The 'Proxy' tab is selected, and the same POST request is highlighted. A context menu is open over the request, with 'Send to Intruder' selected. The menu also includes options like 'Send to Spider', 'Do an active scan', 'Send to Repeater', 'Send to Sequencer', 'Send to Comparer', 'Send to Decoder', 'Request in browser', 'Engagement tools [Pro version only]', 'Change request method', and 'Change body encoding'. The menu has a dark background with white text and icons.



In the positions inside the intruder we see that we get the desired request that we intercepted and also, we see the payloads being highlighted for which we can carry out the brute force.

Here we just use the username and password to carry out the attack and we use the cluster bomb attack type as we are brute forcing against two values and these are required in combination.



In the payloads we see that we two payloads set and, in the payload, set 1 we add the word list of user ids and the payload 2 for the passwords.

Burp Intruder Repeater Window Help  
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts  
2 × 3 × 4 × ...  
Target Positions Payloads Options

**Payload Sets**

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1 Payload count: 0  
Payload type: Simple list Request count: 0

**Payload Options [Simple list]**

This payload type lets you configure a simple list of strings that are used as payloads.

Paste Load ... Remove Clear  
Add Enter a new item Add from list ... [Pro version only]

Look In: Desktop  
File Name: logins.txt  
Files of Type: All Files

**Payload Processing**

You can define rules to perform various processing tasks on each payload before it is used.

Add Enabled Rule

**Payload Options [Simple list]**

This payload type lets you configure a simple list of strings that are used as payloads.

Paste Load ... Remove Clear  
Add Enter a new item  
Add from list ... [Pro version only]

```
xcellent
!!^$@#)&%&
!@#$
!@#$AHmeD
!staugust89
"1chuda*#"
##Jkk11
##shivramsingh@@
#Project123
#h9870051052
```

Burp Suite Community Edition v1.7.36 - temporary Project

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alert

2 x 3 x 4 x ...

Target Positions Payloads Options

**Payload Sets**

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 2 Payload count: 0

Payload type: Simple list Request count: 0

**Payload Options [Simple]**

This payload type lets you configure a simple list of strings that are used as payloads.

Look In: Passwords

File Name: 10-million-password-list-top-1000000.txt

Files of Type: All Files

Paste Load ... Remove Clear Add Enter a new item Add from list ... [Pro version o]

Open Cancel

**Payload Processing**

You can define rules to perform various processing tasks on each payload before it is used.

Add Enabled Rule

Edit Remove Up

Burp Suite Community Edition v1.7.36 - Temporary Project

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts

2 x 3 x 4 x ...

Target Positions Payloads Options

**Payload Sets**

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 2 Payload count: 999,999

Payload type: Simple list Request count: 494,084,733

**Payload Options [Simple list]**

This payload type lets you configure a simple list of strings that are used as payloads.

123456 password 12345678 qwerty 123456789 12345 1234 111111 1234567 dragon

Paste Load ... Remove Clear Add Enter a new item Add from list ... [Pro version only]

**Payload Processing**

You can define rules to perform various processing tasks on each payload before it is used.

Add Enabled Rule

Edit Remove Up Down

Now we start the attack and it will go through all the combinations, the application is tested against 494,084,733 username password combinations.

| Request | Payload1          | Payload2 | Status | Error | Timeout | Length | Comment |
|---------|-------------------|----------|--------|-------|---------|--------|---------|
| 0       |                   | 123456   | 200    |       |         | 1395   |         |
| 1       | xcellent          | 123456   | 200    |       |         | 1403   |         |
| 2       | !!^\$@#)&%%       | 123456   | 200    |       |         | 1394   |         |
| 3       | l@#\$             | 123456   | 200    |       |         | 1398   |         |
| 4       | !@#\$AHmed        | 123456   | 200    |       |         | 1403   |         |
| 5       | Istaugust89       | 123456   | 200    |       |         | 1405   |         |
| 6       | "1chuda#"         | 123456   | 200    |       |         | 1404   |         |
| 7       | #!Jkk11           | 123456   | 200    |       |         | 1402   |         |
| 8       | ##\$hivramsingh@@ | 123456   | 200    |       |         | 1410   |         |
| 9       | #Project123       | 123456   | 200    |       |         | 1405   |         |
| 10      | #b9870051052      | 123456   | 200    |       |         | 1406   |         |
| 11      | #bhanu29          | 123456   | 200    |       |         | 1402   |         |
| 12      | #include#         | 123456   | 200    |       |         | 1403   |         |
| 13      | terorreno         | 123456   | 200    |       |         | 1404   |         |

132 of 494084733

EDIT  
Remove  
Up  
Down

When we scan the combinations for different lengths and check the response and in render, we see that the potential combinations fail for our application. Thus, making our application stringent against the login attacks.

Result 2 | Intruder attack 3

Payload 1: !!^\$@#)&%%&  
Payload 2: 123456  
Status: 200  
Length: 1394  
Timer: 32

Previous  
Next  
Action

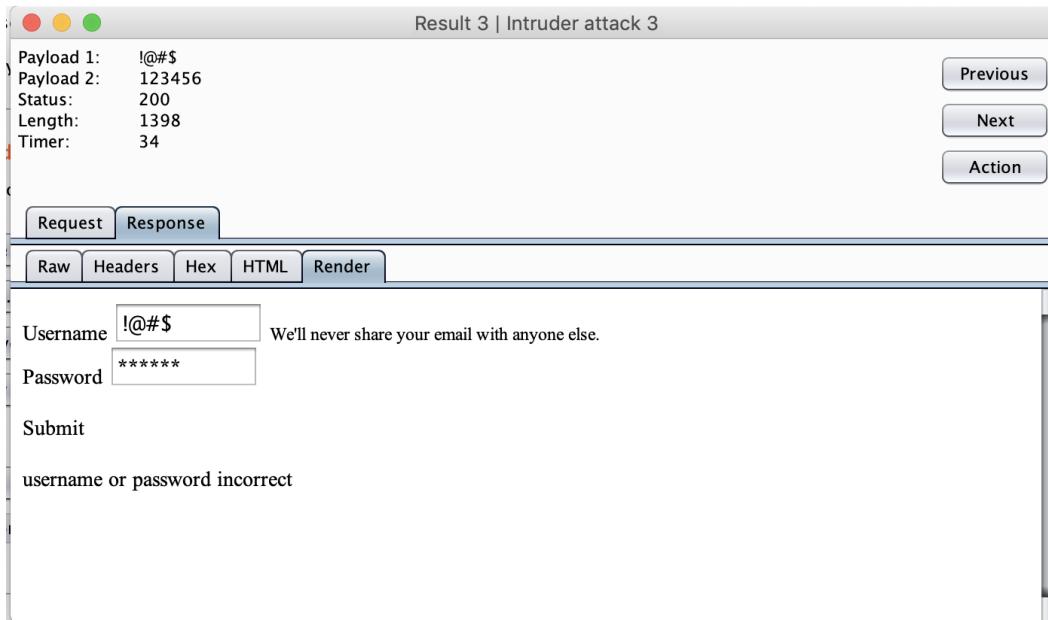
Request Response

Raw Headers Hex HTML Render

Username  We'll never share your email with anyone else.  
Password  \*\*\*\*\*

Submit

username or password incorrect



## Target and Spidering

Here we have our target at <http://localhost:8081> in our site map which is the structure of the web page and we add it to the scope which allows us to define our automated spidering.

| Host                  | Method | URL                     | Params | Status | Length | MIME type |
|-----------------------|--------|-------------------------|--------|--------|--------|-----------|
| http://localhost:8081 | GET    | /                       |        | 200    | 11401  | HTML      |
| http://localhost:8081 | GET    | /asf-logo-wide.svg      |        | 200    | 27459  | XML       |
| http://localhost:8081 | GET    | /sp/                    |        | 200    | 1223   | HTML      |
| http://localhost:8081 | GET    | /sp/professor/dashb...  |        | 200    | 1606   | HTML      |
| http://localhost:8081 | GET    | /sp/professor/grades... | ✓      | 200    | 4029   | HTML      |
| http://localhost:8081 | GET    | /sp/professor/grades... | ✓      | 200    | 4029   | HTML      |
| http://localhost:8081 | GET    | /sp/professor/grades... | ✓      | 200    | 4029   | HTML      |
| http://localhost:8081 | GET    | /sp/professor/grades... | ✓      | 200    | 4029   | HTML      |
| http://localhost:8081 | GET    | /sp/student/dashboard   |        | 200    | 1053   | HTML      |
| http://localhost:8081 | GET    | /sp/student/grades?...  | ✓      | 200    | 1220   | HTML      |
| http://localhost:8081 | GET    | /sp/student/grades?...  | ✓      | 200    | 932    | HTML      |
| http://localhost:8081 | GET    | /sp/student/grades?...  | ✓      | 200    | 932    | HTML      |

Burp Suite Community Edition v1.7.36 - Temporary Project

Burp Intruder Repeater Window Help

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts

Site map Scope

Filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders

| Host                  | Method | URL                         | Params | Status | Length | MIME ty |
|-----------------------|--------|-----------------------------|--------|--------|--------|---------|
| http://localhost:8081 | GET    | /                           |        | 200    | 11401  | HTML    |
| http://localhost:8081 | GET    | /asf-logo-wide.svg          |        | 200    | 27459  | XML     |
| http://localhost:8081 | GET    | /sp/                        |        | 200    | 1223   | HTML    |
| http://localhost:8081 | GET    | /sp/professor/dashboard     |        | 200    | 1606   | HTML    |
| http://localhost:8081 | GET    | /sp/professor/grades...     | ✓      | 200    | 4029   | HTML    |
| http://localhost:8081 | GET    | /sp/professor/grades...     | ✓      | 200    | 4029   | HTML    |
| http://localhost:8081 | GET    | /sp/professor/grades...     | ✓      | 200    | 4029   | HTML    |
| http://localhost:8081 | GET    | /sp/student/dashboard       |        | 200    | 1053   | HTML    |
| http://localhost:8081 | GET    | /sp/student/grades?sort=asc | ✓      | 200    | 1220   | HTML    |
| http://localhost:8081 | GET    | /sp/student/grades?sort=asc | ✓      | 200    | 932    | HTML    |
| http://localhost:8081 | GET    | /sp/student/grades?sort=asc | ✓      | 200    | 932    | HTML    |

http://localhost:8081

- /
- asf-logo
- ▶ docs
- ▶ examples
- ▶ host-man
- ▶ manager
- sp
- ▶ sp

Add to scope

Spider this host  
Actively scan this host  
Passively scan this host

Engagement tools [Pro version only]

- Compare site maps
- Expand branch
- Expand requested items
- Collapse branch
- Delete host
- Copy URLs in this host
- Copy links in this host
- Save selected items
- Show new site map window
- Site map help

HTTP/1.1  
Host: localhost:8081  
Insecure-Requests: 1  
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_14\_0)  
Accept-Language: en-US,en;q=0.9  
Accept: application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,  
Accept-Encoding: gzip, deflate  
Accept-Charset: ISO-8859-1,utf-8;q=0.7,\*;q=0.3  
Connection: close

?

<

+

>

Type a search term

0 matches

Spidering allows crawling through the website getting us idea how it is structured and how we can break through it. Thus, now we spider this host.

Burp Suite Community Edition v1.7.36 - Temporary Project

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts

Site map Scope

Logging of out-of-scope Proxy traffic is disabled [Re-enable](#)

Filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders

| Host                                  | Method | URL                     | Params | Status | Length | MIME type |
|---------------------------------------|--------|-------------------------|--------|--------|--------|-----------|
| <a href="#">http://localhost:8081</a> | GET    | /                       |        | 200    | 11401  | HTML      |
|                                       | GET    | /asf-logo-wide.svg      |        | 200    | 27459  | XML       |
|                                       | GET    | /sp/                    |        | 200    | 1223   | HTML      |
|                                       | GET    | /sp/professor/dashboard |        | 200    | 1606   | HTML      |
|                                       | GET    | /sp/professor/grades... | ✓      | 200    | 4029   | HTML      |
|                                       | GET    | /sp/professor/grades... | ✓      | 200    | 4029   | HTML      |
|                                       | GET    | /sp/professor/grades... | ✓      | 200    | 4029   | HTML      |
|                                       | GET    | /sp/student/dashboard   |        | 200    | 1053   | HTML      |
|                                       | GET    | /sp/student/grades?...  | ✓      | 200    | 1220   | HTML      |
|                                       | GET    | /sp/student/grades?...  | ✓      | 200    | 932    | HTML      |
|                                       | GET    | /sp/student/grades?...  | ✓      | 200    | 932    | HTML      |

http://localhost:8081

- /
- asf-logo-wide.svg
- ▶ docs
- ▶ examples
- ▶ host-manager
- ▶ manager
- sp
- ▶ sp

http://maxcdn.bootstrapcdncdn.com

http://r2---sn-ax5go-q4fs.gvt1.com

http://r3---sn-ax5go-q4fl.gvt1.com

http://redirector.gvt1.com

https://redirector.gvt1.com

https://svn.apache.org

https://tomcat.apache.org

https://twitter.com

http://update.googleapis.com

https://wiki.apache.org

https://wordpress.org

http://www.apache.org

http://www.apple.com

https://www.facebook.com

http://www.google.com

https://www.google.com

https://www.googletagmanager.com

http://www.gstatic.com

https://www.linkedin.com

https://www.reviewexp.com

http://www.w3.org

https://www.youtube.com

Spider this host

Actively scan this host

Passively scan this host

Engagement tools [Pro version only]

- Compare site maps
- Expand branch
- Expand requested items
- Collapse branch
- Delete host
- Copy URLs in this host
- Copy links in this host
- Save selected items
- Show new site map window
- Site map help

intosh; Intel Mac OS X 10\_14\_0) like Gecko) Chrome/70.0.3538.110 Safari/537.36  
xml,application/xml;q=0.9,image/webp,image/apng,  
te;  
;q=0.9,en;q=0.8

Type a search term

0 matches

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts

Site map Scope

Logging of out-of-scope Proxy traffic is disabled Re-enable

Filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders

|   | Host                  | Method | URL                | Params | Status | Length | MIME type |
|---|-----------------------|--------|--------------------|--------|--------|--------|-----------|
| ▶ | http://localhost:8081 | GET    | /                  |        | 200    | 11401  | HTML      |
| ▶ | http://localhost:8081 | GET    | /asf-logo-wide.svg |        | 200    | 27459  | XML       |
| ▶ | http://localhost:8081 | GET    | /docs/             |        | 200    | 17365  | HTML      |
| ▶ | http://localhost:8081 | GET    | /docs/BUILDING.txt |        | 200    | 19760  | text      |

Burp Spider - Submit Form

Burp Spider needs your guidance to submit a login form. Please choose the value of each form field which should be used when submitting the form. You can control how Burp handles forms in the Spider options tab.

Action URL: http://localhost:8081/sp/verify  
Method: POST

| Type     | Name     | Value |
|----------|----------|-------|
| Password | password |       |
| Text     | userName |       |

Submit form Ignore form

http://127.0.0.1:8088

http://addldap.sourceforge.net

https://ant.apache.org

https://apr.apache.org

http://blogs.msdn.com

https://bugzilla.org

http://bz.apache.org

https://bz.apache.org

https://cd.mozilla.org

http://cheatography.com

http://clieck.it

https://coolsite.com

https://comicspot.com

https://comicspot.com

https://coolsite.com

https://cviegi.com

https://cw.com

http://dl.google.com

https://dl.google.com

http://docspdf.com

http://findpdf.com

https://forrst.com

https://git.com

https://hirerush.com

https://hs.com

https://httptoolkit.com

https://httptoolkit.com

http://init.com

http://init.com

http://init.ess.apple.com

https://instagram.com

http://jakarta.apache.org

https://jakarta.apache.org

http://java.sun.com

http://javamail.java.net

http://jcp.org

https://jcp.org

http://localhost

http://localhost:1977

http://localhost:8080

http://localhost:8081

/

asf-logo-wide.svg

docs/

Burp Suite Community Edition v1.7.36 - Temporary Project

Burp Intruder Repeater Window Help

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts

Site map Scope

Logging of out-of-scope Proxy traffic is disabled Re-enable

Filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders

| Host                                  | Method | URL                | Params | Status | Length | MIME type |
|---------------------------------------|--------|--------------------|--------|--------|--------|-----------|
| <a href="#">http://localhost:8081</a> | GET    | /                  |        | 200    | 11401  | HTML      |
| <a href="#">http://localhost:8081</a> | GET    | /asf-logo-wide.svg |        | 200    | 27459  | XML       |
| <a href="#">http://localhost:8081</a> | GET    | /docs/             |        | 200    | 17365  | HTML      |
| <a href="#">http://localhost:8081</a> | GET    | /docs/BUILDING.txt |        | 200    | 19760  | text      |

Burp Spider - Submit Form

Burp Spider needs your guidance to submit a login form. Please choose the value of each form field which should be used when submitting the form. You can control how Burp handles forms in the Spider options tab.

Action URL: [http://localhost:8081/examples/jsp/security/protected/j\\_security\\_check](http://localhost:8081/examples/jsp/security/protected/j_security_check)

Method: POST

| Type     | Name       | Value |
|----------|------------|-------|
| Password | j_password |       |
| Text     | j_username |       |

Submit form Ignore form

http://127.0.0.1:8088

http://adldap.sourceforge.net

https://ant.apache.org

https://apr.apache.org

http://blogs.msdn.com

https://bu

http://bug

https://bz

https://cd

http://che

http://cli

https://co

https://co

https://co

https://co

https://cv

https://cw

http://dl.g

https://dl.

http://doc

http://find

https://for

https://git

https://hir

https://hs

https://ht

https://ht

http://init

http://init

http://init.ess.apple.com

https://instagram.com

http://jakarta.apache.org

https://jakarta.apache.org

http://java.sun.com

http://javamail.java.net

http://jcp.org

https://jcp.org

http://localhost

http://localhost:1977

http://localhost:8080

http://localhost:8081

- /
- asf-logo-wide.svg
- docs

Type a search term 0 matches

Now going into the spider, we see the requests it made and number of bytes it transferred. We further go into the application login and try to do sql injection and we again perform the spidering

Burp Suite Community Edition v1.7.36 - Temporary Project

Burp Intruder Repeater Window Help

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts

Control Options

**Spider Status**

Use these settings to monitor and control Burp Spider. To begin spidering, browse to the target application, then right-click one or more nodes in the target site map, and choose "Spider this host / branch".

Spider is running Clear queues

Requests made: 411  
Bytes transferred: 3,755,108  
Requests queued: 0  
Forms queued: 0

**Spider Scope**

Use suite scope [defined in Target tab] (checked)  
Use custom scope

---

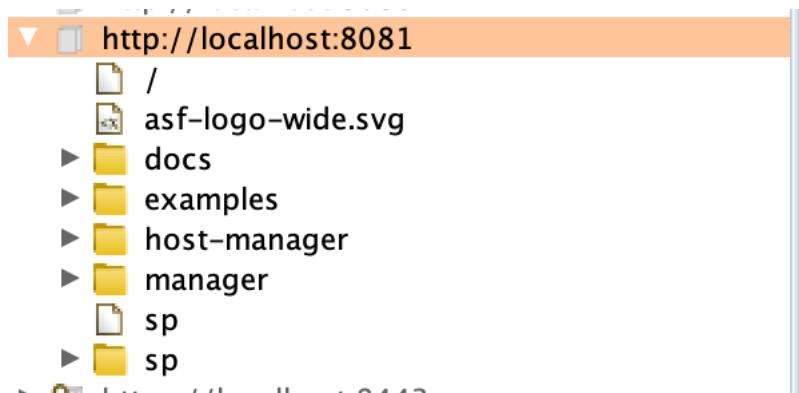
**Application Login**

These settings control how the Spider submits login forms.

Don't submit login forms  
 Prompt for guidance  
 Handle as ordinary forms  
 Automatically submit these credentials:

Username: admin' or 1=1--  
Password:

We have the structure of the application



## Finding the hidden files using ZAP

The screenshot shows the OWASP Zed Attack Proxy (ZAP) interface in Standard Mode. The left sidebar lists contexts and sites, with 'Sites' expanded to show a site at 'http://localhost:8081'. The main pane displays a 'Welcome to the OWASP Zed Attack Proxy (ZA)' message and a table of requests. One request is listed:

| ID | Req. Timestamp      | Method | URL                        | Code | Reason           | RTT    | Size | Resp. Body | Highest Alert  | Note | Tags |
|----|---------------------|--------|----------------------------|------|------------------|--------|------|------------|----------------|------|------|
| 1  | 9/12/18 6:07:39 ... | GET    | http://localhost:8081/s/p  | 200  | 6... 1,089 bytes | Medium |      |            | Form, Password |      |      |
| 3  | 9/12/18 6:08:32 ... | GET    | http://localhost:8081/s/p/ | 200  | 1... 1,089 bytes | Medium |      |            | Form, Password |      |      |

To the right, there is a login form with fields for 'Username' (placeholder 'Enter username') and 'Password', and a 'Submit' button.

The screenshot shows the OWASP Zed Attack Proxy (ZAP) interface in Standard Mode. The left sidebar lists contexts and sites, with 'Sites' expanded to show a site at 'http://localhost:8081'. A context menu is open over a URL entry in the list, specifically for 'GET:s/p'. The 'Attack' submenu is visible, containing options like Spider..., Active Scan..., Forced Browse site, etc. The main pane shows the same welcome message and request table as the first screenshot. To the right, there is a login form with fields for 'Username' (placeholder 'Enter username') and 'Password', and a 'Submit' button.

The screenshot shows the OWASP Zed Attack Proxy (ZAP) version 2.7.0 interface. The top navigation bar includes 'Standard Mode', 'Sites' (selected), 'Request', 'Response', and a '+' button. The left sidebar displays a tree view of the 'Sites' section, listing various URLs such as http://localhost:8080/docs, http://localhost:8080/examples, and http://localhost:8080/manager. The main central area features a large 'Welcome to the OWASP Zed Attack Proxy (ZA)' header, followed by a brief introduction about ZAP's purpose and usage. Below this is a 'URL to attack:' input field containing 'http://', an 'Attack' button, and a 'Stop' button. A progress indicator shows 'Not started'. At the bottom, there is a note about exploring the application using a browser or automatically. The bottom navigation bar includes tabs for 'History', 'Search', 'Alerts' (selected), 'Output', 'Spider' (selected), and a '+' button. The bottom-most section displays a table of processed URLs, showing columns for 'Processed', 'Method', 'URI', and 'Flags'. The table lists several GET requests to various endpoints like /search/, /selenium/, and /script.js. The bottom status bar shows 'Progress: 0: http://localhost:8081', 'Current Scans: 0', 'URLs Found: 1014', 'Nodes Added: 370', and an 'Export' button.

Carrying out the attack using Forced browser directory: -

Standard Mode

Sites +

Contexts Default Context

Sites ▾ http://localhost:8080 http://localhost:8081

Welcome to the OWASP Zed Attack Proxy (ZAP)

ZAP is an easy to use integrated penetration testing tool for finding vulnerabilities in web applications.

Please be aware that you should only attack applications that you have been specifically given permission to test.

To quickly test an application, enter its URL below and press 'Attack'.

URL to attack: http:// Select...

Attack Stop

Progress: Not started

For a more in depth test you should explore your application using your browser or automated regression tests while proxying through ZAP.

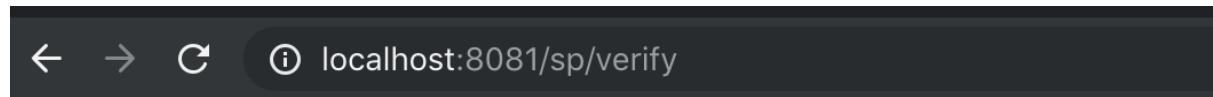
Explore your application: Launch Browser JxBrowser ↗

History Search Alerts Output Spider Forced Browse +

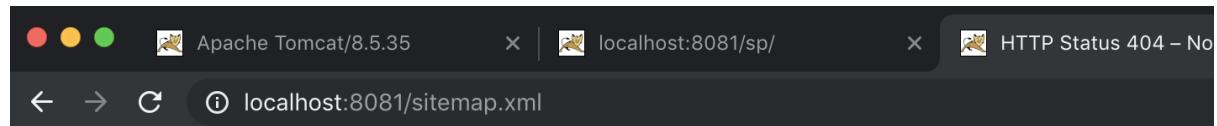
Site: localhost:8081 List: directory-list-1.0.txt 12% Current Scans:1 Num Requests:17539

| Req. Timestamp     | Resp. Timestamp    | Method | URL                                            | Code | Reason    | Size Resp. Header | Size Resp. Body |
|--------------------|--------------------|--------|------------------------------------------------|------|-----------|-------------------|-----------------|
| 9/12/18 6:13:49 PM | 9/12/18 6:13:49 PM | GET    | http://localhost:8081/                         | 200  | 120 bytes | 11,266 bytes      |                 |
| 9/12/18 6:13:49 PM | 9/12/18 6:13:49 PM | GET    | http://localhost:8081/sp/                      | 200  | 114 bytes | 1,089 bytes       |                 |
| 9/12/18 6:13:50 PM | 9/12/18 6:13:50 PM | GET    | http://localhost:8081/docs/                    | 200  | 200 bytes | 17,145 bytes      |                 |
| 9/12/18 6:13:50 PM | 9/12/18 6:13:50 PM | GET    | http://localhost:8081/docs/config/             | 200  | 198 bytes | 9,151 bytes       |                 |
| 9/12/18 6:13:50 PM | 9/12/18 6:13:50 PM | GET    | http://localhost:8081/examples/                | 200  | 198 bytes | 1,126 bytes       |                 |
| 9/12/18 6:13:50 PM | 9/12/18 6:13:50 PM | GET    | http://localhost:8081/docs/security-howto.html | 200  | 200 bytes | 35,746 bytes      |                 |
| 9/12/18 6:13:50 PM | 9/12/18 6:13:50 PM | GET    | http://localhost:8081/sp/verify                | 200  | 73 bytes  | 14 bytes          |                 |
| 9/12/18 6:13:51 PM | 9/12/18 6:13:51 PM | GET    | http://localhost:8081/docs/manager-howto.html  | 200  | 200 bytes | 75,405 bytes      |                 |
| 9/12/18 6:13:51 PM | 9/12/18 6:13:51 PM | GET    | http://localhost:8081/docs/cluster-howto.html  | 200  | 200 bytes | 43,705 bytes      |                 |
| 9/12/18 6:13:51 PM | 9/12/18 6:13:51 PM | GET    | http://localhost:8081/manager/                 | 302  | 122 bytes | 0 bytes           |                 |
| 9/12/18 6:13:51 PM | 9/12/18 6:13:51 PM | GET    | http://localhost:8081/manager/status           | 401  | 243 bytes | 2,473 bytes       |                 |
| 9/12/18 6:13:51 PM | 9/12/18 6:13:51 PM | GET    | http://localhost:8081/examples/servlets        | 302  | 103 bytes | 0 bytes           |                 |
| 9/12/18 6:13:51 PM | 9/12/18 6:13:51 PM | GET    | http://localhost:8081/manager/html             | 401  | 243 bytes | 2,473 bytes       |                 |
| 9/12/18 6:13:51 PM | 9/12/18 6:13:51 PM | GET    | http://localhost:8081/examples/iso             | 302  | 98 bytes  | 0 bytes           |                 |

## Accessing the hidden files



Served at: /sp



## HTTP Status 404 – Not Found

Type Status Report

Message /sitemap.xml

Description The origin server did not find a current representation for the target resource or is not willing to disclose that one exists.

Apache Tomcat/8.5.35

Hidden files don't give the direct access to the pages and the sensitive parts of the application in complete structure of the application are not obtained both in burp suite and Zap thus making the application secure.

- Unit testing result  
Here is the unit testing report (actual screenshot) using **Junit unit testing framework**.

Eclipse File Edit Navigate Search Project Run Window Help

eclipse-workspace - /Users/hsingh/eclipse-workspace/.metadata/plugins/edu.umd.cs.findbugs.plugin.eclipse.sp.fbwarnings.xml - Eclipse IDE

Problems Javadoc Declaration Search Console Bug Explorer Bug Info Call Hierarchy SonarLint Report SonarLint Rule Description Coverage Debug Servers

```
<terminated> /Library/Java/JavaVirtualMachines/jdk1.8.0_181.jdk/Contents/Home/bin/java (Dec 9, 2018, 3:58:00 PM)
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ sp ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-surefire-plugin:2.22.0:test (default-test) @ sp ---
[INFO]
[INFO] -----
[INFO] T E S T S
[INFO] -----
[INFO] [INFO] Running com.uta.sp.dao.JdbcConnectionTest
Connection Successful
[{PNAME=Thomas Trey Johns, GRADE=K, NAME=SECURE PROGRAMMING}, {PNAME=Thomas Trey Johns, GRADE=F, NAME=DBMS}, {PNAME=Thomas Trey Johns, GRADE=A, NAME=SDP}]
Connection Successful
Connection Successful
Student [studentId=1, name=Himanshu Ajay Singh, email=himanshuajay.singh@mavs.uta.edu]
true
Connection Successful
[INFO] Tests run: 5, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 1.16 s - in com.uta.sp.dao.JdbcConnectionTest
[INFO] [INFO] Running com.uta.sp.controller.ConfigTest
2018-12-09 15:58:07 INFO ConfigTest:11 - Test Log
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0 s - in com.uta.sp.controller.ConfigTest
[INFO]
[INFO] Results:
[INFO] Tests run: 6, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] -----
[INFO] --- maven-war-plugin:2.2:war (default-war) @ sp ---
[INFO] Packaging webapp
[INFO] Assembling webapp [sp] in [/Users/hsingh/eclipse-workspace/sp/target/sp]
[INFO] Processing war project
[INFO] Copying Webapp resources [/Users/hsingh/eclipse-workspace/sp/src/main/webapp]
[INFO] Webapp assembled in [177 msec]
[INFO] Building war: /Users/hsingh/eclipse-workspace/sp/target/sp.war
[INFO] WEB-INF/web.xml already added, skipping
[INFO]
[INFO] -----
[INFO] --- maven-install-plugin:2.4:install (default-install) @ sp ---
[INFO] Installing /Users/hsingh/eclipse-workspace/sp/target/sp.war to /Users/hsingh/.m2/repository/com/uta/sp/0.0.1-SNAPSHOT/sp-0.0.1-SNAPSHOT.war
[INFO] Installing /Users/hsingh/eclipse-workspace/sp/pom.xml to /Users/hsingh/.m2/repository/com/uta/sp/0.0.1-SNAPSHOT/sp-0.0.1-SNAPSHOT.pom
[INFO]
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 5.880 s
[INFO] Finished at: 2018-12-09T15:58:08-06:00
[INFO]
```

Package Explorer Type Hierarchy JUnit

Finished after 1.311 seconds

Runs: 5/5 Errors: 0 Failures: 0

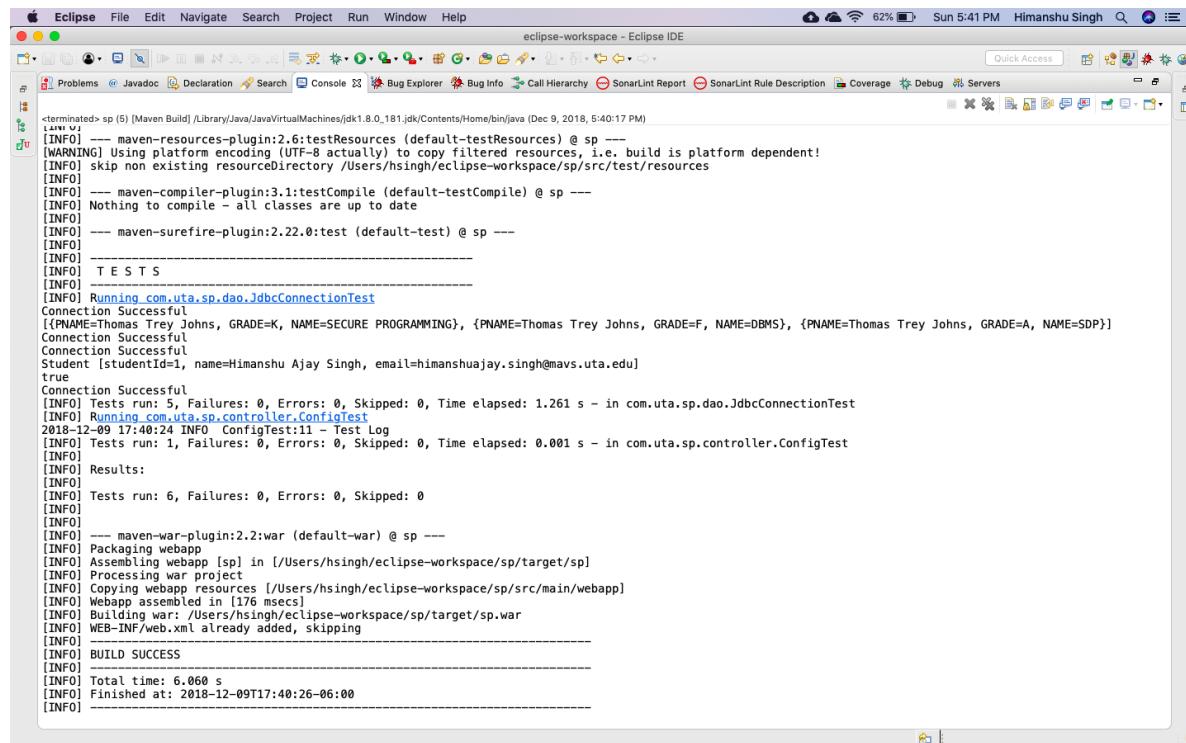
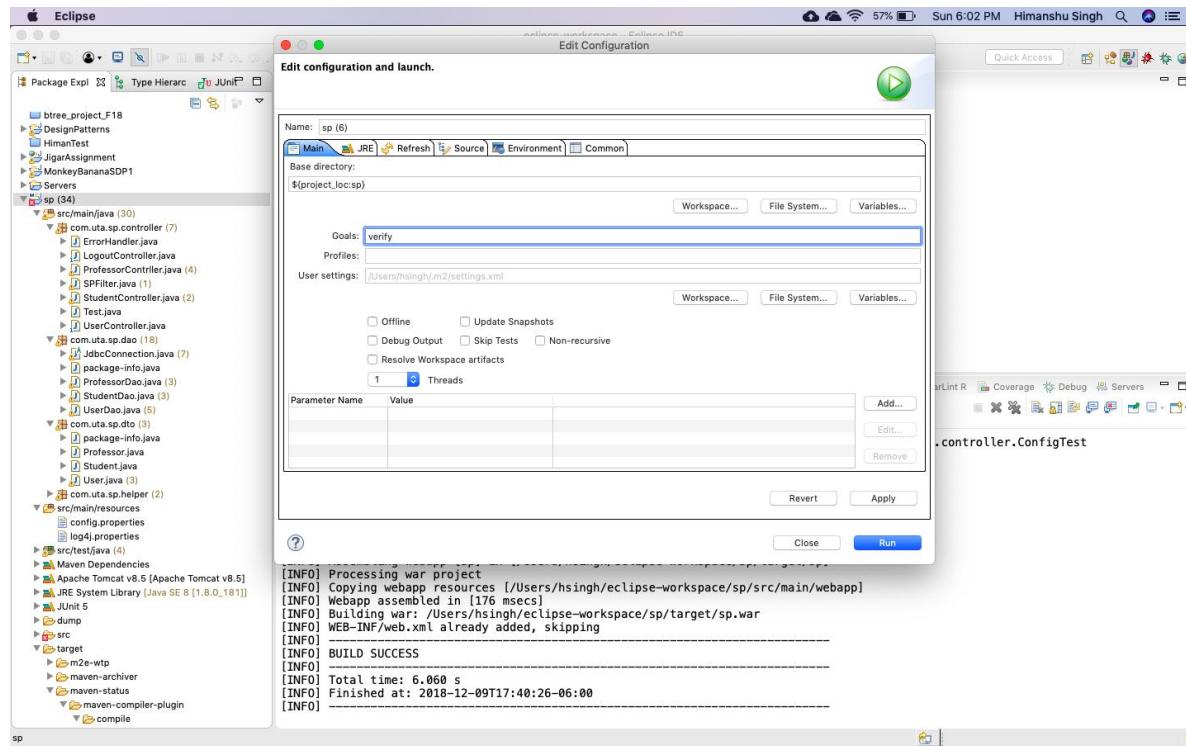
JdbcConnectionTest [Runner: JUnit 5] (0.946 s)

- studentGetGrades() (0.622 s)
- psidTest() (0.025 s)
- studentGetOneTest() (0.030 s)
- passwordTest() (0.155 s)
- testUpdate() (0.114 s)

Failure Trace

- Integration testing results

The integration testing is done with maven configuration file using **maven verifies command**.



## Security assessment/Compliance

- STIG checklist

### **1. Oracle Linux 5 Security Technical Implementation Guide: Version 1, Release: 13 Benchmark Date: 26 Oct 2018**

**Vul ID:** V-791      **Rule ID:** SV-64509r1\_rule    **STIG ID:** GEN001360

**Severity:** CAT II      **Classification:** Unclass

We have two files configuration. Properties and log4j.properties change the mode using # chmod 0755 <filename> for the above two files to prevent unauthorized modification of these files

The screenshot shows a software interface for managing security vulnerabilities. On the left is a list of vulnerabilities with columns for Status, Vul ID, and Rule Name. The row for Vul ID V-791 is highlighted. The main pane displays detailed information for this rule, including its title, discussion, check text, fix text, procedure, references, and findings.

| Status | Vul ID       | Rule Name        |
|--------|--------------|------------------|
| NR     | V-785        | GEN001160        |
| NR     | V-786        | GEN001180        |
| NR     | V-787        | GEN001260        |
| NR     | V-788        | GEN001800        |
| NR     | V-789        | GEN001320        |
| NR     | V-790        | GEN001340        |
|        | <b>V-791</b> | <b>GEN001360</b> |
| NR     | V-792        | GEN001280        |
| NR     | V-793        | GEN001300        |
| NR     | V-794        | GEN001200        |
| NR     | V-795        | GEN001220        |
| NR     | V-796        | GEN001240        |
| NR     | V-797        | GEN001400        |
| NR     | V-798        | GEN001380        |
| NR     | V-800        | GEN001420        |
| NR     | V-801        | GEN002380        |
| NR     | V-802        | GEN002440        |
| NR     | V-803        | GEN002400        |
| NR     | V-804        | GEN002460        |
| NR     | V-805        | GEN002420        |
| NR     | V-806        | GEN002500        |
| NR     | V-807        | GEN002520        |
| NR     | V-808        | GEN002560        |
| NR     | V-810        | GEN002640        |

**Oracle Linux 5 Security Technical Implementation Guide :: Version 1, Release: 13 Benchmark Date: 26 Oct 2018**

**Vul ID:** V-791    **Rule ID:** SV-64509r1\_rule    **STIG ID:** GEN001360

**Severity:** CAT II    **Classification:** Unclass

**Rule Title:** The NIS/NIS+/yp command files must have mode 0755 or less permissive.

**Discussion:** NIS/NIS+/yp files are part of the system's identification and authentication processes and are critical to system security. Unauthorized modification of these files could compromise these processes and the system.

**Check Text:** Perform the following to check NIS file permissions.

```
ls -la /var/yp/*
```

If the file's mode is more permissive than 0755, this is a finding.

**Fix Text:** Change the mode of NIS/NIS+/yp command files to 0755 or less permissive.

**Procedure (example):**

```
chmod 0755 <filename>
```

**References**

**CCI:** CCI-000225: The organization employs the concept of least privilege, allowing only authorized accesses for users (and processes acting on behalf of users) which are necessary to accomplish assigned tasks in accordance with organizational missions and business functions.  
NIST SP 800-53 :: AC-6  
NIST SP 800-53A :: AC-6.1  
NIET CO-001 C3 Revision A - AC-6

**Finding Details**

**Comments**

```
[Himanshus-Air-a:0a:classes hsingh$ ls -l
total 16
drwxr-x--- 3 hsingh staff 96 Dec 9 14:32 com
-rw-r----- 1 hsingh staff 74 Dec 9 02:25 config.properties
-rw-r----- 1 hsingh staff 714 Dec 9 01:16 log4j.properties
[Himanshus-Air-a70a:classes hsingh$ chmod 0755 com/
com/ config.properties
[Himanshus-Air-a70a:classes hsingh$ chmod 0755 config.properties
[Himanshus-Air-a70a:classes hsingh$ ls -l
total 16
drwxr-x--- 3 hsingh staff 96 Dec 9 14:32 com
-rwxr-xr-x 1 hsingh staff 74 Dec 9 02:25 config.properties
-rw-r----- 1 hsingh staff 714 Dec 9 01:16 log4j.properties
Himanshus-Air-a70a:classes hsingh$]
```

## 2. Oracle Linux 5 Security Technical Implementation Guide: Version 1, Release: 13 Benchmark Date: 26 Oct 2018

**Vul ID:** V-766      **Rule ID:** SV-63383r1\_rule    **STIG ID:** GEN000460

**Severity:** CAT II      **Classification:** Unclass

Disabling accounts after a limited number of unsuccessful login attempts  
improves protection against password guessing attacks

| Status | Vul ID       | Rule Name        |
|--------|--------------|------------------|
| NR     | V-756        | GEN000020        |
| NR     | V-760        | GEN000280        |
| NR     | V-761        | GEN000300        |
| NR     | V-762        | GEN000320        |
| NR     | V-763        | GEN000400        |
| NR     | V-765        | GEN000440        |
| D      | <b>V-766</b> | <b>GEN000460</b> |
| NR     | V-768        | GEN000480        |
| NR     | V-769        | GEN000520        |
| NR     | V-770        | GEN000560        |
| NR     | V-773        | GEN000880        |
| NR     | V-774        | GEN000900        |
| NR     | V-775        | GEN000920        |
| NR     | V-776        | GEN000940        |
| NR     | V-777        | GEN000960        |
| NR     | V-778        | GEN000980        |
| NR     | V-780        | GEN000360        |
| NR     | V-781        | GEN000380        |
| NR     | V-782        | GEN006480        |
| NR     | V-783        | GEN000120        |
| NR     | V-784        | GEN001140        |
| NR     | V-785        | GEN001160        |
| NR     | V-786        | GEN001180        |
| NR     | V-787        | GEN001260        |

**Oracle Linux 5 Security Technical Implementation Guide :: Version 1, Release: 13 Benchmark Date: 26 Oct 2018**

**Vul ID:** V-766    **Rule ID:** SV-63383r1\_rule    **STIG ID:** GEN000460

**Severity:** CAT II    **Classification:** Unclass

**Rule Title:** The system must disable accounts after three consecutive unsuccessful login attempts.

**Discussion:** Disabling accounts after a limited number of unsuccessful login attempts improves protection against password guessing attacks.

**Check Text:** Check the pam\_tally configuration.  
# more /etc/pam.d/system-auth  
Confirm the following line is configured, before any "auth sufficient" lines:  
auth required pam\_tally2.so deny=3  
If no such line is found, this is a finding.

**Fix Text:** By default link /etc/pam.d/system-auth points to /etc/pam.d/system-auth-ac which is the file maintained by the authconfig utility. In order to add pam options other than those available via the utility create /etc/pam.d/system-auth-local with the options and including system-auth-ac. In order to set the account lockout to three failed attempts the content should be similar to:

```
auth required pam_access.so
auth required pam_tally2.so deny=3
auth include system-auth-ac
account required pam_tally2.so
account include system-auth-ac
password include system-auth-ac
session include system-auth-ac
```

**Finding Details**

**Comments**

Fix: The below screenshot shows the fix which is implemented

localhost:8080/tp/verify

Username: hcs7975  
We'll never share your email with anyone else.

Password:

Submit

maximus limit exceeded, please contact admin

### 3. Oracle Linux 5 Security Technical Implementation Guide: Version 1, Release: 13 Benchmark Date: 26 Oct 2018

Vul ID: V-765      Rule ID: SV-63363r1\_rule    STIG ID: GEN000440

Severity: CAT II      Classification: Unclass

The rule says that all successful and unsuccessful logins and logouts must be logged

The screenshot shows a software interface for managing security rules. On the left is a list of vulnerabilities (Vul ID) from V-756 to V-787. The row for V-765 is highlighted. The main pane displays the following information:

**Oracle Linux 5 Security Technical Implementation Guide :: Version 1, Release: 13 Benchmark Date: 26 Oct 2018**

**Vul ID:** V-765    **Rule ID:** SV-63363r1\_rule    **STIG ID:** GEN000440

**Severity:** CAT II    **Classification:** Unclass

**Rule Title:** Successful and unsuccessful logins and logouts must be logged.

**Discussion:** Monitoring and recording successful and unsuccessful logins assists in tracking unauthorized access to the system. Without this logging, the ability to track unauthorized activity to specific user accounts may be diminished.

**Check Text:** Determine if all logon attempts are being logged.

**Procedure:**  
Verify successful logins are being logged:  
# last -R | more  
If the command does not return successful logins, this is a finding.

Verify if unsuccessful logons are being logged:  
# lastb -R | more  
If the command does not return unsuccessful logins, this is a finding.

**Fix Text:** Make sure the collection files exist.

**Procedure:**  
If there are no successful logins being returned from the "last" command, create /var/log/wtmp:  
# touch /var/log/wtmp

If there are no unsuccessful logins being returned from the "lastb" command, create /var/log/btmp:

**Finding Details**

**Comments**

We have fixed the above rule by logging in all these details:

```
2018-12-09 14:32:20 INFO SPFFilter:30 - valid request from ip-address => 0.0.0.0.0.0.1
2018-12-09 14:32:27 INFO SPFFilter:70 - successfull login request for ttj7975 from ip-address
=> 0:0:0:0:0:0:1
2018-12-09 14:32:27 INFO SPFFilter:48 - valid request from ip-address => 0:0:0:0:0:0:1
2018-12-09 14:32:29 INFO SPFFilter:48 - valid request from ip-address => 0:0:0:0:0:0:1
2018-12-09 14:32:34 INFO SPFFilter:48 - valid request from ip-address => 0:0:0:0:0:0:1
2018-12-09 14:32:34 INFO ProfessorController:87 - grage updated for studentid5 by professor id
1
2018-12-09 14:32:34 INFO SPFFilter:48 - valid request from ip-address => 0:0:0:0:0:0:1
2018-12-09 14:32:40 INFO SPFFilter:48 - valid request from ip-address => 0:0:0:0:0:0:1
2018-12-09 14:32:40 INFO ProfessorController:87 - grage updated for studentid10 by professor
id 1
2018-12-09 14:32:40 INFO SPFFilter:48 - valid request from ip-address => 0:0:0:0:0:0:1
2018-12-09 16:56:28 ERROR SPFFilter:43 - invalid request from ip-address => 0:0:0:0:0:0:1
2018-12-09 16:56:28 INFO SPFFilter:38 - valid request from ip-address => 0:0:0:0:0:0:1
2018-12-09 16:58:33 INFO SPFFilter:38 - valid request from ip-address => 0:0:0:0:0:0:1
2018-12-09 16:58:33 INFO SPFFilter:70 - successfull login request for ttj7975 from ip-address
=> 0:0:0:0:0:0:1
2018-12-09 16:58:33 INFO SPFFilter:48 - valid request from ip-address => 0:0:0:0:0:0:1
2018-12-09 16:59:22 INFO SPFFilter:48 - valid request from ip-address => 0:0:0:0:0:0:1
2018-12-09 16:59:22 INFO LogoutController:38 - Thomas Trey Johns =>User successfull
logged out
2018-12-09 16:59:23 INFO SPFFilter:38 - valid request from ip-address => 0:0:0:0:0:0:1
```

## 4. Oracle Linux 5 Security Technical Implementation Guide: Version 1, Release: 13 Benchmark Date: 26 Oct 2018

Vul ID: V-761      Rule ID: SV-63251r1\_rule    STIG ID: GEN000300

Severity: CAT II      Classification: Unclass

All accounts on the system must have unique user or account names.

The screenshot shows a software interface for managing security vulnerabilities. On the left is a table of vulnerabilities with columns: Status, Vul ID, and Rule Name. The table lists various entries, with V-761 highlighted. The main pane displays detailed information for rule V-761:

**Oracle Linux 5 Security Technical Implementation Guide :: Version 1, Release: 13 Benchmark Date: 26 Oct 2018**

**Vul ID:** V-761    **Rule ID:** SV-63251r1\_rule    **STIG ID:** GEN000300  
**Severity:** CAT II    **Classification:** Unclass

**Rule Title:** All accounts on the system must have unique user or account names.

**Discussion:** A unique user name is the first part of the identification and authentication process. If user names are not unique, there can be no accountability on the system for auditing purposes. Multiple accounts sharing the same name could result in the denial of service to one or both of the accounts or unauthorized access to files or privileges.

**Check Text:** Check the system for duplicate account names.

**Example:**  
# pwck -r

If any duplicate account names are found, this is a finding.

**Fix Text:** Change user account names, or delete accounts, so each account has a unique name.

**References**

CCI: CCI-000764: The information system uniquely identifies and authenticates organizational users (or processes acting on behalf of organizational users).  
NIST SP 800-53 :: IA-2  
NIST SP 800-53A :: IA-2.1  
NIST SP 800-53 Revision 4 :: IA-2

**Finding Details**

**Comments**

Fix: In the database we have implemented this by using unique keyword for all user names:

The screenshot shows the MySQL Workbench interface displaying the schema of a table named 'user'. The table has the following structure:

| Field          | Type         | Null | Key | Default | Extra          |
|----------------|--------------|------|-----|---------|----------------|
| USER_ID        | int(11)      | NO   | PRI | NULL    | auto_increment |
| NAME           | varchar(45)  | NO   | UNI | NULL    |                |
| PASSWORD       | varchar(500) | NO   |     | NULL    |                |
| ROLE_ID        | int(11)      | YES  | MUL | NULL    |                |
| PROFESSOR_ID   | int(11)      | YES  | MUL | NULL    |                |
| STUDENT_ID     | int(11)      | YES  | MUL | NULL    |                |
| LOGIN_ATTEMPTS | int(11)      | YES  |     | 0       |                |

The 'NAME' column is defined with a unique key ('UNI').

## 5. Oracle Linux 5 Security Technical Implementation Guide: Version 1, Release: 13 Benchmark Date: 26 Oct 2018

Vul ID: V-770      Rule ID: SV-63787r1\_rule    STIG ID: GEN000560

Severity: CAT I      Classification: Unclass

The system must not have accounts configured with blank or null passwords.

The screenshot shows a software interface for managing security vulnerabilities. On the left is a table of vulnerabilities with columns: Status, Vul ID, and Rule Name. The row for V-770 is highlighted in green. The main pane displays detailed information for V-770, including its status as 'Not A Finding', severity as 'CAT I', and the rule title: 'The system must not have accounts configured with blank or null passwords'. It also includes a discussion about password authentication, a check text command to verify nullok settings, a fix text command to edit pam.d, and a reference section listing CCI-000366 and NIST SP 800-53 requirements. Below the main pane are sections for 'Finding Details' and 'Comments'.

| Status | Vul ID       | Rule Name        |
|--------|--------------|------------------|
| NR     | V-756        | GEN000020        |
| NR     | V-760        | GEN000280        |
| O      | V-761        | GEN000300        |
| NR     | V-762        | GEN000320        |
| NR     | V-763        | GEN000400        |
| O      | V-765        | GEN000440        |
| O      | V-766        | GEN000460        |
| NR     | V-768        | GEN000480        |
| NR     | V-769        | GEN000520        |
| NF     | <b>V-770</b> | <b>GEN000560</b> |
| NR     | V-773        | GEN000880        |
| NR     | V-774        | GEN000900        |
| NR     | V-775        | GEN000920        |
| NR     | V-776        | GEN000940        |
| NR     | V-777        | GEN000960        |
| NR     | V-778        | GEN000980        |
| NR     | V-780        | GEN000360        |
| NR     | V-781        | GEN000380        |
| NR     | V-782        | GEN006480        |
| NR     | V-783        | GEN000120        |
| NR     | V-784        | GEN001140        |
| NR     | V-785        | GEN001160        |
| NR     | V-786        | GEN001180        |
| NR     | V-787        | GEN001260        |

Fix: In the database we have implemented this by specifying No null in password field

The screenshot shows the 'Result Grid' of MySQL Workbench displaying the structure of a table. The table has columns: Field, Type, Null, Key, Default, and Extra. The 'PASSWORD' column is defined as varchar(500) with a 'NO' nullability constraint, which corresponds to the 'No null in password field' mentioned in the fix text. Other columns include USER\_ID, NAME, ROLE\_ID, PROFESSOR\_ID, STUDENT\_ID, and LOGIN\_ATTEMPTS.

| Field          | Type         | Null | Key | Default | Extra          |
|----------------|--------------|------|-----|---------|----------------|
| ► USER_ID      | int(11)      | NO   | PRI | NULL    | auto_increment |
| NAME           | varchar(45)  | NO   | UNI | NULL    |                |
| PASSWORD       | varchar(500) | NO   |     | NULL    |                |
| ROLE_ID        | int(11)      | YES  | MUL | NULL    |                |
| PROFESSOR_ID   | int(11)      | YES  | MUL | NULL    |                |
| STUDENT_ID     | int(11)      | YES  | MUL | NULL    |                |
| LOGIN_ATTEMPTS | int(11)      | YES  |     | 0       |                |

- Changes made in web server, application server for security hardening. The session-timeout is updated from the default value of 30 to 15 in tomcat/config/web.xml
 

```
<session-config>
 <session-timeout>15</session-timeout>
 </session-config>
```
- Vulnerability scan reports
 Reports were generated for listing vulnerabilities in network and web application using Nessus. Screenshots of overall vulnerabilities are as below and reports are included in the file.

Nmap Script Suite

Scans Settings

Certificate error https://localhost:8834/#/scans/reports/5/hosts

**Vulnerability Scan**

Back to My Scans

Configure Audit Trail Launch Export

FOLDERS My Scans All Scans Trash

RESOURCES Policies Plugin Rules Customized Reports Scanners

Hosts 1 Vulnerabilities 23 Notes 1 History 5

Filter Search Hosts 1 Host

Host	Vulnerabilities
127.0.0.1	37

**Scan Details**

Name: Vulnerability Scan Status: Completed Policy: Basic Network Scan Scanner: Local Scanner Start: Today at 5:56 PM End: Today at 6:02 PM Elapsed: 7 minutes

**Vulnerabilities**

Critical (Red) 0, High (Orange) 0, Medium (Yellow) 1, Low (Green) 0, Info (Blue) 26

127.0.0.1

0	0	1	0	26
CRITICAL	HIGH	MEDIUM	LOW	INFO

Total: 27

SEVERITY	CVSS	PLUGIN	NAME
MEDIUM	5.0	57608	SMB Signing not required
INFO	N/A	21186	AJP Connector Detection
INFO	N/A	12634	Authenticated Check : OS Name and Installed Package Enumeration
INFO	N/A	45590	Common Platform Enumeration (CPE)
INFO	N/A	10736	DCE Services Enumeration
INFO	N/A	10052	Daytime Service Detection
INFO	N/A	54615	Device Type
INFO	N/A	11367	Discard Service Detection
INFO	N/A	10061	Echo Service Detection
INFO	N/A	12053	Host Fully Qualified Domain Name (FQDN) Resolution
INFO	N/A	117886	Local Checks Not Enabled (info)
INFO	N/A	92366	Microsoft Windows Last Boot Time
INFO	N/A	42410	Microsoft Windows NTLMSSP Authentication Request Remote Network Name Disclosure

**Vulnerability\_Scan\_91u4lw(1).pdf**

Severity	Port	ID	Description
INFO	N/A	42410	Microsoft Windows NTLMSSP Authentication Request Remote Network Name Disclosure
INFO	N/A	11011	Microsoft Windows SMB Service Detection
INFO	N/A	100871	Microsoft Windows SMB Versions Supported (remote check)
INFO	N/A	106716	Microsoft Windows SMB2 Dialects Supported (remote check)
INFO	N/A	10719	MySQL Server Detection
INFO	N/A	19506	Nessus Scan Information
INFO	N/A	110723	No Credentials Provided

Severity	Port	ID	Description
INFO	N/A	11936	OS Identification
INFO	N/A	97993	OS Identification and Installed Software Enumeration over SSH v2 (Using New SSH Library)
INFO	N/A	10198	Quote of the Day (QOTD) Service Detection
INFO	N/A	56468	Time of Last System Startup
INFO	N/A	35711	Universal Plug and Play (UPnP) Protocol Detection
INFO	N/A	20301	VMware ESX/GSX Server detection
INFO	N/A	35712	Web Server UPnP Detection

**Nessus Professional / Scans**

Certificate error https://localhost:8834/#/scans/reports/12/config/credentials

**Scans** Settings

**FOLDERS**

- My Scans
- All Scans
- Trash

**RESOURCES**

- Policies
- Plugin Rules
- Customized Reports
- Scanners

If the keywords %USER% and %PASS% are used, they will be substituted with the username and password provided above.

Check authentication on page: http://127.0.0.1:8090/sp/professor/dashboard

Regex to verify successful authentication: welcome

**Global Credential Settings**

Login method: POST

Re-authenticate delay (seconds): 0

Follow 30x redirections (# of levels): 0

Invert authenticated regex:

Use authenticated regex on HTTP headers:

Case insensitive authenticated regex:

Save Cancel

Nessus Professional / Fc New tab + ▾ Certificate error https://localhost:8634/#/scans/reports/12/vulnerabilities

Scans Settings

Web Test [Back to My Scans](#)

Hosts 0 Vulnerabilities 0 History 8

Filter Search Vulnerabilities 0 Vulnerabilities

No records found.

Scan Details

Name:	Web Test
Status:	Completed
Policy:	Web Application Tests
Scanner:	Local Scanner
Start:	Today at 7:38 PM
End:	Today at 7:40 PM
Elapsed:	2 minutes

Nessus Professional / Sc + ▾ Certificate error https://localhost:8834/#/scans/reports/12/config/credentials

Scans Settings

FOLDERS My Scans All Scans Trash

RESOURCES Policies Plugin Rules Customized Reports Scanners

CATEGORIES Plaintext Authentication Filter Credentials

All credentials in use

HTTP Method: HTTP login form, User: hxs7975

Authentication method: HTTP login form

Username: hxs7975

Password: \*\*\*\*\*

Login page: 127.0.0.1:8090/sp/

Login submission page: 127.0.0.1:8090/sp/verify

Login parameters: 127.0.0.1:8090/sp/verify?userName=tjj7975&pass

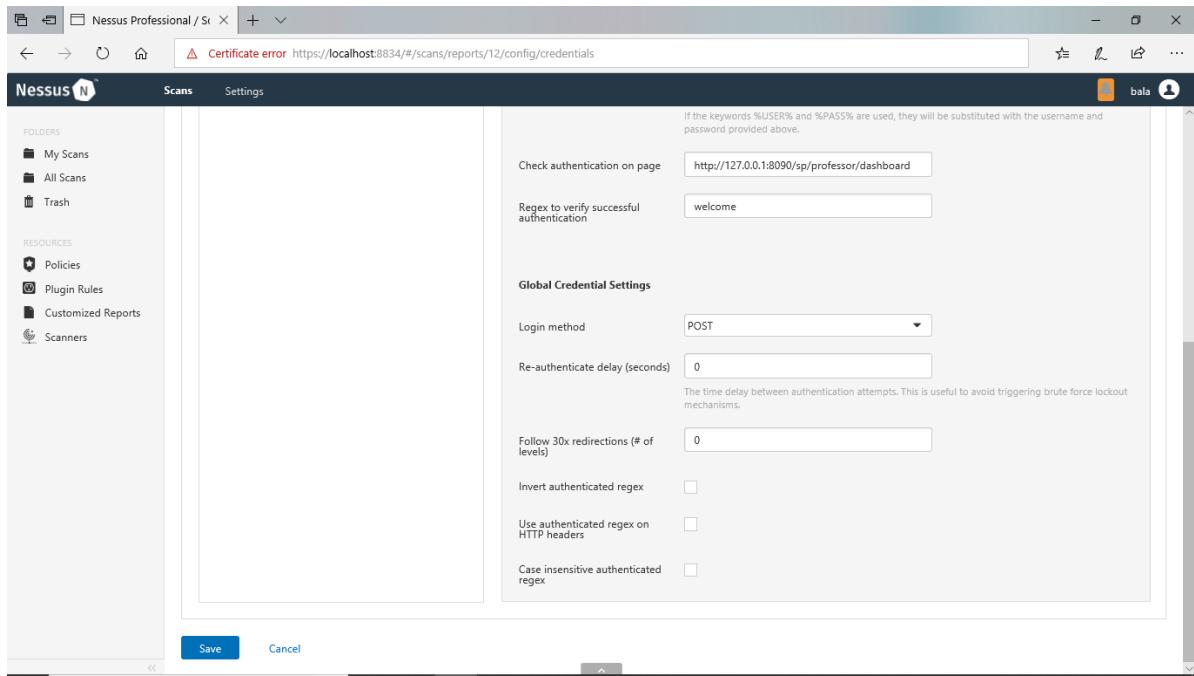
If the keywords %USER% and %PASS% are used, they will be substituted with the username and password provided above.

Check authentication on page: http://127.0.0.1:8090/sp/professor/dashboard

Regex to verify successful authentication: welcome

Global Credential Settings

Login method: POST



## References

1. [https://www.owasp.org/images/5/57/OWASP\\_Proactive\\_Controls\\_2.pdf](https://www.owasp.org/images/5/57/OWASP_Proactive_Controls_2.pdf)
2. <http://web.cerritos.edu/dwhitney/SitePages/CIS201/LectureNotesOnTalonNet/Chapter12Lecture.pdf>
3. <https://www.symantec.com/avcenter/reference/attack.surface.analysis.of.blackberry.devices.pdf>
4. <http://ethesis.nitrl.ac.in/5793/1/E-9.pdf>
5. <https://google.github.io/styleguide/javaguide.htm>