BSc (Hons) in Computing in Software Development

Group Project Assessment Brief: Penetration Testing Project

Dr. Muhammad Adil Raja

October 3, 2025

Module: Secure Systems, COMP I8029
Assessment Weighting: 25% of Module
Assessment Type: Group Assessment
Report and Software Artifact (if any) Submission Deadline: 10st November, 2025

Warning:
1. Academic Integrity: This assessment is to be completed individually, and any form of collusion or cheating will be subject to severe academic penalties.
2. Prohibited Tools: Do not use ChatGPT or any other AI-based text generation tools or any unfair means to complete this assessment.
3. No Plagiarism: Do not copy content from external sources without proper citation. Plagiarism will result in a significant reduction of marks.

Table of contents:

Project Summary

As global maritime operations become digitally dependent, securing ocean-going vessels from cyber threats is essential. This project focuses on the **"Ocean Guardian,"** a high-tech cargo ship exposed to international cyber risks, including sabotage and data breaches.

The core objective is a comprehensive vulnerability assessment. We will execute a multi-faceted evaluation of the ship's network security, focusing on its IT infrastructure, critical operational technology (OT), navigation systems, and communication protocols. Our methodology includes **penetration testing, deep vulnerability scanning, exploitation simulation, and formal risk analysis** to precisely identify and document weaknesses available to a cyber adversary.

Getting Started

For the purposes of this project, you have been provided one virtual machine: **Metasploitable**. The captain of the ship has given you written permission to both scan and attempt to actively exploit this machine for the purposes of penetration testing, but wasn't able to provide much in the way of technical details, saying they were "not really a computer person".

Download the virtual machine and import it into VMware (or Virtual Box) alongside your Kali VM.

Tools Required

- Victim OS: Metaploitable operating system.
- OS for analysis and exploitation: Kali Linux.
- Tools for pen testing: Nmap, Metasploit, OpenVAS, Burp Suite, OWASP ZAP.

Testing Requirements

For the purposes for this course project, you can consider your penetration test to be complete if you achieve the following goals:

- Run port/version scans using Nmap and vulnerability scans using OpenVAS and/or Nessus
- Discover at least **two exploits** that provide for remote command-line access.
    - At least one method, either directly, or through privilege escalation, must allow you to gain full control over the machine, equivalent to the root, Administrator, or SYSTEM user.
- Discover at least **one existing user account** (username and password) that provides for remote command-line access to the system. The existing password must be provided in plaintext.
- Perform penetration testing of the OWASP juice shop using Burp Suite according to the tasks listed here: https://github.com/0xrajneesh/Ethical-Hacking-Projects-for-beginners/blob/main/Project-2-Penetration-Testing-a-Vulnerable-Web-Application.md

Notes:
(1) Attacks requiring physical machine access to be successful are outside the scope of this project. In this penetration test, you are only interacting with the Ocean Guardian system over the network.

Moreover, install the OWASP juice shop on your system. https://owasp.org/www-project-juice-shop/

Documentation Requirements

Your deliverable for this penetration test should be a **formal written report**.

Note: Your report should be **narrative in style**, with human explanation and commentary. A "report" that is *merely* a collection of screenshots and data dumps will be **graded poorly.**

Your report should be structured as follows:

Section 1: Introduction / Purpose of Penetration Test

Section 2: Executive Summary of Results

What methods of access did you discover? What logins did you discover?

Section 3: Application Scanning Results

Document the applications running on the server. Provide a table summarizing the protocol (TCP/UDP), port number, application name, application version, and any other information you find relevant. In addition, describe *how* the results were obtained, i.e. your *methodology*.

Section 4, 5, 6, ... : Access via Exploit *(EACH exploit gets its OWN section)*

For *each* exploit or privilege escalation method that you **personally verified** (and not *merely* theorized from some automated scan tool), provide a section describing that method in **detail**. Include the following information:

1. What network service or software component is vulnerable? Specify its name and version number.
2. What port number does that service listen on? (if applicable)
3. What is the CVE number of the vulnerability? (if applicable)
4. What are the exact steps you took to *discover* the vulnerability?
5. What are the exact steps you took to *exploit* the vulnerability?
6. What level of access is granted by the vulnerability? (Commands run as a specific user? Does that user have full control?)
7. How can you *prove* that you have the level of access you claim? Provide screenshots documenting your key accomplishments.
8. How does the vulnerability function? (this may require external research on your part)
9. What steps should a system administrator take to mitigate this vulnerability?

Section *n*: Access via User Logins

Document the user login(s) (usernames and passwords) that you discovered. Include the following information:

1. Username and passwords for each login
2. How did you discover and verify each of these logins?
3. What steps should a system administrator take (or policies that should be enforced) to reduce the likelihood of attackers obtaining these logins through similar methods?

Conclusion

Appendix: Vulnerability Scan Results

As an appendix to your written report, submit the results of an automated vulnerability scan tool such as Nessus or OpenVAS. In this appendix, it is acceptable to have vulnerabilities listed that you have not personally had the time to verify in this project. This section should contain the **complete** scan results with all the vulnerability details, not just a summary. Export the scan results as a PDF, and concatenate it to your report PDF.

Resources

- All the class labs to-date! :)
- Exploit Proof-of-Concepts (POCs)

- Linux Post-Exploit Cheat Sheet
- Linux Enumeration Cheat Sheet

Grading

Checkpoint 1 (5 pts)

For the first checkpoint, submit a progress report of your penetration test to date. By this point, you should have conducted port and version scans of the target using Nmap, and vulnerability scans of the target using either OpenVAS or Nessus. The progress report must include the following elements:

1. Application scanning results for server. This should be a table summarizing the protocol (tcp/udp), port number, application name, application version, and any other information you find relevant.
2. A discussion of how you obtained the application scanning results
3. A discussion of at least three technical steps (tests, scans, exploit attempts, etc) that you intend to take next
4. Appendix: OpenVAS scan or Nessus scan showing complete scan results with all vulnerability details

Checkpoint 2 (10 pts)

For the second checkpoint, submit a progress report of your penetration test to date. The progress report must include the following elements:

1. Application scanning results for server.
2. A summary of the vulnerabilities proven to exist. **For full credit, you must have exploited at least one vulnerability.** Include a screenshot of each successful exploit as "proof of work" in addition to the written commentary.
3. A discussion of at least three technical steps (tests, scans, exploit attempts, etc) that you intend to take next

Checkpoint 3 (10 pts)

1. For the third checkpoint, submit a progress report of your penetration testing of the OWASP juice shop. There are two marks for each of the following tasks:

   a) Identify open ports.

   b) SQL injection.

   c) Cross Site Scripting (XSS).

   d) File Upload Vulnerability.

   e) Directory Traversal.

   f) Cross-Site Request Forgery (CSRF).

Penetration Test Report - Technical Content (40pts)

- 2+ exploits to gain shell access to the target system, including proof of access - 30pts
- 1+ logins to access the target system, including proof of access - 10pts

Penetration Test Report - Human Explanation of Technical Content (35 pts)

- Clear discussion of each step taken to obtain an exploit or login - 20 pts
- Clear discussion of how each exploit functions - 5 pts
- Clear discussion of steps that a system administrator should take to mitigate each vulnerability - 10 pts

Submission

Submit your report in PDF format to Moodle. Please also submit a score sheet of how much each person contributed to each of the deliverables mentioned in the previous section. You should mention your contribution on a scale from 1-5 for each member of the group that how much they contributed to the CA. 1 means poor contribution and 5 means excellent contribution.