School of Computing and Information Systems

COMP90074: Web Security

Assignment 3 - Project Plutus

Due date: No later than 11:59pm on Sunday 4th June 2023
Weight: 25% Marked out of 100
Note: All challenges have a flag in the format: FLAG{something_here}
Note: None of the challenges are related to each other. They are entirely independent.

Submission format

All students must submit a zip file with all their code. A PDF version of their penetration testing report and threat modelling report should be submitted separately. The PDF must be named <username>-assignment3-threat-modelling.pdf and <username>-assignment3-penetration-test-report.pdf. The zip must be named <username>-assignment3.zip (e.g. testuser1-assignment3.zip).

All code for each challenge must be clearly labelled and stored in a separate file, so it is not confused with the code for other challenges.

Finally, all code must be referenced within the report. This implies that there will be code in both the report and the separate code file for each task.

If you have any questions or queries, please feel free to reach out via the discussion board, or by contacting Sajeeb, Maleehah, or John.

Mandatory deliverable:

1. Threat modelling PDF report
2. Penetration test PDF report
3. Zip containing all code used

Scenario (Common for All Sections)

Bank of UniMelb has just come to market and provides an easy, self-service banking solution to its clients. Users can open new accounts (with proper identity verification measures), perform banking transactions (such as money transfers and bill payments) with existing accounts, or close their accounts all from the comfort of their own laptop or computer. Branch managers have higher privileges than users, and can communicate and interact with their clients (users assigned to them), see their accounts, and freeze them.

COVID-19 restrictions and the need to work from home have meant Bank of UniMelb wants a rapid migration to this new digital system. Executives have selected you to ensure the system is penetration tested and secure before it is deployed. They love the innovation you have brought to the penetration testing game with your business “We Test Pens Incorporated” and have high expectations of you.

(Note: the web application you are testing will not have these features implemented, but you can assume they exist and are functioning correctly.)

Threat Modelling (20%)

Using STRIDE, threat model the application and identify the possible vulnerabilities (at least two per letter of the acronym). Also, make sure you state who the relevant threat actor is (e.g. state actor, external attacker, bank client, internal employee, etc.). We expect some descriptions around the vulnerabilities, diving into some details, alongside the remediation for them. Should you require knowing the development stack for your remediations, please assume it is LAMP (Linux Apache MySQL PHP).

Required sections of the report:

1. Vulnerabilities / threats
2. Correlating threats to threat actors
3. Remediations

Please use the sample report template provided. There will be marks deducted for anyone who does not use this template.
There is no set word limit for this exercise, but an example has been provided in the template for you to use as a guide. (Please remove this example prior to submission.)

Testing Scenario (40%)

Bank of UniMelb has selected you for this task due to the high reputation of your cyber security degree, and a belief that you will perform with a very high degree of skill. Due to the aforementioned COVID restrictions, the organisation has a limited budget, limited time, and was not able to set up a full testing environment. You will be performing all your testing in a production environment and therefore must use great care and skill, performing only manual penetration testing, while being acutely aware of your behaviour in the organisation's environment to prevent potential denial of service attacks **(this means no automated scanning)**.

As you are now a professional, your goal is to present your findings in a high quality report for delivery at the end of this engagement. The quality of your work and the effort that you put in cannot be judged without a quality report detailing all your findings, potential consequences, and recommended remediations. Please see the “Submission format” section for a further explanation on what you must submit for this assignment to be marked.

Lastly, as a tip, you will be testing the full web application specified in the “Scope” section, and are expected to find the following vulnerabilities:

Vulnerability	Flag Format	Marks Weighting
Bypassing client-side authentication	FLAG{}	5
IDOR via a hidden parameter	FLAG{}	7.5
Authentication weakness leading to account takeover	FLAG{}	7.5
Privilege escalation	FLAG{}	10
Sensitive files / directories left behind during testing / development	FLAG{}	10

For this assignment some tasks will require automation. We recommend using Burp’s Intruder, but the same thing can be accomplished with Python.

The final vulnerability, Sensitive files / directories left behind during testing / development, will require the use of an automated tool to brute force directories. We will allow DirBuster, which can be cloned from https://gitlab.com/kalilinux/packages/dirbuster. You are free to use external wordlists but **you must limit the number of threads to 5**.

Please ensure you write up these findings in a suitable format in your report as you find them. **Also make sure to add in your own mitigation recommendations! The practicality of the remediation is very important (tailor the recommendations to the application).**

Note: There are branch managers created for each student with usernames in the format <username>-branch-manager. You should attempt to compromise these accounts and no others.

BONUS MARKS: If you are able to identify vulnerabilities that have not been listed, please report them for a chance at bonus marks. Bonus marks will be provided at the discretion of the lecturer based on complexity of the finding and quality of the writeup.

Out of Scope

The following vulnerabilities are known to the developer and must not be submitted within the penetration testing report (**no marks will be awarded for these findings**):

1. Insufficient password policies
2. Sensitive information over HTTP
3. Directory listing enabled

4. Lack of rate limiting on the login page
5. Server-status exposed

Scope

Testing must only be performed on http://assignment-plutus.unimelb.life/
Testing must be manual only. Manual tools may be used (Burp, Zap, etc), *however you may not use the automated scanning capabilities of these tools*.
No automated scanning or automated tools can be used.
No load testing, denial of service (DOS) or distributed denial of service (DDOS) attacks.
You may use Burp's Intruder, but use less than 30 payloads per minute.
You may also use DirBuster, but you must limit the number of threads to 5.

User Credentials

Use your **Melbourne University username** as the **username and password**, for logging into the application.

Report Writing (40%)

For this assignment, we expect a professionally written report, provided to the client (teaching staff), explaining and specifying each vulnerability you identified by discussing the vulnerability, the process of exploitation (steps to reproduce the exploits), the potential impact to the organisation, overall risk, and the remediation (making sure to tailor it to the application). The vulnerabilities must be listed in order of remediation priority, based on the risk posed. We expect an overall assessment of the risk posture for the application, using the findings from your penetration test. **Also, please ensure that the flag is displayed in a screenshot at the end of each challenge's writeup. We will not be accepting any flags that are not displayed in a screenshot.**

Please use the sample report template provided. There will be marks deducted for anyone who does not use this template.