MOP Security Team Progress T1 2023

Contents

This document the work that the security team has completed over the course of the 2023 T1 Trimester and allows for new juniors to obtain a brief understanding of what work was done, the results of the work, and the location of where the respective reports can be found. It will also list and detail work that will need to be completed in the future with scans and attacks that may be completed for further security testing. This is only a brief overview of the work that is completed, and it is recommended that juniors read the full reports of each piece of work for a higher understanding of what work was done and what the full scope of results were.

Completed Work

Port Scanning

Port scanning has been completed on the MOP webapp to check for any open ports that may be vulnerable to an attack. It had been found that TCP ports 80 and 443 were open and over 1000 UDP ports were open. Thile the two TCP ports that were open had been for certain web operations, the 1000 UDP ports posed an issue as only a few are somewhat necessary to be open while the rest should be shut for safety. The full report can be found at:

City of Melbourne > Files > Resources > Cyber Security > Scans > Using NMAP to find Vulnerabilities in TCP and UDP Ports.pdf

Vulnerability Scanning

Vulnerability has been completed on the MOP webapp to check for vulnerabilities in the network, webapp and server. This scan was completed using the Nessus suite with a variety of different scans being utilized to get a grasp of what may be impacting the MOP website's security. It found that there were very few vulnerabilities in the results of the scans with nothing that may pose a severe security risk. The full report can be found at:

City of Melbourne > Files > Resources > Cyber Security > Scans > Using Nessus to scan for vulnerabilities in the MOP web application.pdf

Stress Testing

Stress Testing has been completed on the MOP webapp to check the amount of people that the site can handle before issues start to arise. It was found that 30 concurrent users can be on the site for a length of 5 minutes with a 99.92% success rate which shows promise for how well the site can perform with high traffic. There is still room for optimization in the future with config file changes. The full report can be found at:

City of Melbourne > Files > Resources > Cyber Security > Scans > Stress_Testing Results.docx

Source Code Inspection

The source code of the MOP project was inspected to check for any security related errors that may be present. Through searching the source code, it was found that there were a few issues that were present that will need fixing by the other teams in future. There was a lax content security policy in the main initialisation file and credential leaks as credential data is not being hidden. These issues are for the development team to fix. The full report can be found at:

City of Melbourne > Files > Resources > Cyber Security > Report 1 – Source Code Inspection.docx

Failed Attacks

Traversal Attack

A traversal attack was attempted on the MOP webapp which would allow for an attacker to read files in a running application which aren't usually meant to be read. This attack failed as we were not able to read any hidden files during the attack. Also, any edited packet would be rejected or sanitized and would redirect to the homepage. This may mean that there are prevention methods in place that may need to be investigated further for testing. The full report can be found at:

City of Melbourne > Files > Resources > Cyber Security > Failed Attacks > Failed Attack(Traversal).docx

Denial of Service (DoS)

Many different denial of service flooding attacks were attempted on the MOP server to see if it can be shut down with a flood of packets being sent to it. SYN, UDP, ICMP, and TCP floods were attempted to see if any could cause issues for the network that MOP was running on. During the attack all of the packets were transmitted, but none of them were being received by the network. This may mean that it is being recognised as a DoS attack and preventing them from being received. The full report can be found at:

City of Melbourne > Files > Resources > Cyber Security > Failed Attacks > Failed Attack(Denial of Service).docx

SSL Stripping

SSL stripping was attempted on the MOP webapp to strip the secure connection to unencrypt it to make the website more vulnerable to further attacks. It takes advantage of unaware people to implement simple methods of attack against them. Bettercap was used to try stripping the security of the connection, however it was not functioning as intended as it was not logging any information or traffic from the MOP website when the attack was attempted. This is due to the HSTS method that prevents attacks like SSL stripping, so more research or attempts may be needed in order for this attack to properly work with these safeguards in place. This report can be found at:

City of Melbourne > Files > Resources > Cyber Security > Failed Attacks > Failed Attack(SSL Stripping).docx

Potential Attacks

These attacks can be carried out by the security team if anyone is stuck on ideas for attacks or things to do for each sprint. Some of these may be more complicated than others but it is a good starting point for searching for attacks that may be relevant on the MOP website or any other project. Even if an attack fails it is still good information as it allows us to know the extent of the project's security on these attacks and how we can potentially push further in the future.

Cross Site Scripting (XSS)

Cross-site scripting (XSS) is a type of web security vulnerability that allows an attacker to inject malicious code into a web page viewed by other users. The attack occurs when a web application does not properly validate user input and allows the attacker to inject malicious code that is executed by unsuspecting users who view the page. There are three types of XSS attacks:

- Reflected XSS attacks occur when an attacker injects malicious code into a URL that is then reflected back to the user in the page's response. The attacker can then use this code to steal sensitive data or perform other malicious actions.
- Stored XSS attacks occur when an attacker injects malicious code into a web application's database. This code is then served to other users who view the affected page. The attacker can then use this code to steal sensitive data or perform other malicious actions.
- DOM-based XSS attacks occur when an attacker injects malicious code into the Document Object Model (DOM) of a web page, which can then be executed by the user's web browser.

There is no software to carry this out for you, you must use specific malicious code that is to be injected into the source code to initiate this attack.

SQL Injection

SQL injection is a type of web security vulnerability that allows an attacker to inject malicious SQL code into a web application's database. The attack occurs when a web application does not properly validate or sanitize user input and allows the attacker to manipulate the SQL queries that are executed by the application.

SQL injection attacks can be completed in several ways. One common method is through the use of input fields such as search boxes, forms, or login screens. Attackers can use these input fields to inject malicious SQL code that alters the behaviour of the database and allows them to perform unauthorized actions such as accessing or modifying sensitive data. Another method of SQL injection is through the use of URL parameters. Attackers can modify the URL parameters that are sent to a web application in order to inject malicious SQL code and perform unauthorized actions.

This may be useful in the future when the MOP webpage is further developed with more functions such as the login screens and search boxes. Currently it may not be of great use due to the lack of user functionality on the website, but it will become relevant in the future.

File Inclusion Attacks

File inclusion attacks, also known as directory traversal attacks or path traversal attacks, are a type of web security vulnerability that allows an attacker to access and execute files that are stored outside the web application's root directory. The attack occurs when a web application includes user input in a file path without proper validation or sanitization. This can allow an attacker to manipulate the file path and access files on the server that should not be accessible, such as configuration files, password files, or other sensitive data.

File inclusion attacks can be completed in several ways, including through the use of input fields such as search boxes, forms, or file upload screens. Attackers can use these input fields to inject malicious file paths that allow them to access and execute files on the server.

Again, not as relevant currently due to lack of implementation of the functionality used to complete the attack in the first place but may be relevant in the future when such functions are added to the website. It is important regardless to recognise these attacks as they will need to be tested once certain functionality is added to test new parts of the website.

Retry Failed Attacks

It is important to always retry failed attacks again in the future as even if they didn’t work now, they may become working again in the future. Updates in the source code and the website as a whole may leave room for failed attacks to work again. Small changes can leave gaps for vulnerabilities the can be exploited by a wide range of attacks. Even trying attacks that did work in the future may prove to yield good results as scans can show more results that may have not been there in the past. It is more important to test new things instead of this, but it may be good to try something that didn’t work and approach it from a different angle such as trying new methods, software, or tools to try make the attack work as this is what attackers would do when attempting to bypass security.

Complete Attacks on other Projects

As we complete more and more attacks on the MOP project, we will eventually run out of things to do to test the security of the website. Once we have exhausted our options and understood the security of the website, we will eventually have to move to testing different projects in Chameleon while we wait for the MOP project to implement more features that will open up more avenues for attack. Extending our reach to different projects will prove extremely beneficial for Chameleon as a whole as we can test the security of all projects in the company to ensure everything is properly secured as one breach could lead to cascading issues for each other project. This idea will most likely be explored in T2 as I believe it will be very important to start looking into that then.