MOC Security Investigation

By Tyler Sheehan and Dylan Stewart

dystewar@redhat.com

tyler_sheehan@student.uml.edu

Red Hat

Office of the CTO Interneship (Summer 2021)

Primary Focus of the Internship:

Running security audit for the Massachusetts Open Cloud(MOC)

What is the MOC?

“The Mass Open Cloud (MOC) is a production public cloud… the project is a unique collaborative effort between higher education, government, non-profit entities, and industry.”

- To use the MOC infrastructure you must first be cleared by an MOC admin and given a dedicated Project by said admin
- Each Project is a hosted by the MOC and provides an openStack environment where users can:
    - Spin up customizable virtual machines
    - Create virtual networks

https://massopen.cloud/

Presentation Topics:

Project 1: Network Enumeration

Project 2: Subdomain Enumeration

Project 3: Web Application Fuzzing

Project 4: URL Fuzzing

Project 5: Cookie/Session Manipulation

Project 6: Web Vulnerability Scans

Project 1: Network Enumeration

Network Enumeration Overview:

Basic network discovery to determine what the standard network layout looks like for an MOC project

Objectives:

- Use tools to run basic network scans
  - Host Discovery
  - Scan hosts for open ports
  - Scan host for Operating System

Tools:

- Nmap (network mapping tool)

Nmap: https://nmap.org/

Results:

Identified 3 different hosts in our network

1) Gateway router
2) 2 DNS servers

Each host was safely configured and no unexpected or unnecessary ports were open. All 3 are running Linux OS

For more Details: Network Enumeration

Project 2: Subdomain Enumeration

Red Hat

Subdomain Enumeration Overview:

Discover as many subdomains under massopen.cloud as possible

Objectives:

- Find and execute tool for subdomain enumeration
- Determine ways to get more info from subdomains we discovered
    - IP addresses where the sites are hosted
        - Info associated with these IP addresses

Tools:

1. Sublist3r (subdomain enumeration tool)
2. Dig command line tool (querying DNS nameservers)
3. Whois command line tool (identifies who owns a domain and their contact info)

The biggest issue we had was the amount of data we got back from the dig and whois commands

To help with processing the data we made several scripts

Sublist3r: https://github.com/aboul3la/Sublist3r
Dig: https://linuxize.com/post/how-to-use-dig-command-to-query-dns-in-linux/
Whois: https://www.liquidweb.com/kb/new-user-tutorial-whois/

Results:

- Compiled a list of 138 accessible subdomains for review
- Successfully found the IP address where each subdomain is being hosted (where possible) using dig
- Found more information about each ip using whois

For more details: Subdomain Enumeration Report

Project 3: Web Application Fuzzing

Red Hat

Project 3 Overview

Feed data into all the inputs on the MOC website and to see if any weird or harmful situations occur (server errors, unauthorized access, etc).

Issues of Project 3

- This type of testing requires more technological expertise than was initially expected. In addition due to time restraints we didn't have enough time to acquire said expertise.
- This type of testing can potentially harm the environment it's being used to test and the MOC doesn't have a testing environment so any harm would be directly to users.

Results of Project 3

This project was scrapped due to the issues mentioned before and a new project was developed. This new project was safer, faster, and required significantly less expertise and precision.

Project 4: URL Fuzzing

Project 4 Overview

- Due to the issues with Project 3 mentioned before we decided to use a more simple fuzzing method called URL Fuzzing.
- Took the URLS found during Project 2 and using the programs DirBuster and Dirb, to check the URLS for potentially unwanted websites.

Issues and Solutions of Project 4

Problem: After starting testing with DirBuster the **network provider** of the MOC blocked the IP address being used for testing. Then limited the testing to 250 requests every 300 seconds

Solution: Change from DirBuster, which can only do as low as 1 request every second to Dirb which can include a delay between tests. In addition making use of a smaller word list to decrease test time.

Results of Project 4

- All http and https portions of the subdomains were scanned with the new word list and no particularly vulnerable URLS were found. (No user specific data was found, no access to amdin protected resources, etc)
- We specifically looked for
  - 200 (OK): The request succeeded
  - 403 (Forbidden): The server understood the request, but is refusing to fulfill it
  - Any 500 HTTP codes (indicate server side errors)

For more details: URL Fuzzing Report

Project 5: Cookie/Session Manipulation

Red Hat

Cookie/Session Manipulation Overview:

- MOC uses session cookies keep track of user sessions.
- We want to see if we can capture a web request
- Try to use this request in another login session

Tools:

- Burpsuite (web application security tool)

Burpsuite: https://portswigger.net/burp

Results of Session Manipulation:

- Captured GET request using Burpsuite
- Sent the request in a new session
- Confirmed that the session cookies were functioning properly

For more details: Cookie/Session Manipulation Report

Project 6: Web Vulnerability Scans

Red Hat

Objectives

- Find a tool to perform vulnerability analysis
- Determine which urls are reachable from a web browser to run vulnerability scans
- Determine which urls had webpages that displayed login pages to test for SQL injection vulnerabilities

Tool for analysis

Nikto

- Nikto is a Perl based open source web server scanning tool which detects anything from outdated software to server misconfigurations and other types of vulnerabilities such as SQL injection
- Run from command line

Nikto scanner repository: https://github.com/sullo/nikto

Results

- No SQL injection vulnerabilities were found on any MOC associated login pages
- Some potential vulnerabilities we uncovered on MOC URLs were Cross Site Scripting and Cross Site Tracing as well as clickjacking
- Due to time constraints the results of these tests need to be further investigated

For more details: Nikto Vulnerability Scans Report

Nikto scanner repository: https://github.com/sullo/nikto

Future Tests and Topics/Results

Red Hat

- Web app analysis with Nessus
- Take a deeper look at attacking the MOC from the inside (within our own project space)
- Sit down with MOC engineers and see if there is anything interesting in the data we collected
- Attempt to break out of MOC containers to access shared resources

Raw Results:

- Github Repo

Comprehensive Results:

- Comprehensive Subdomain Spreadsheet

Acknowledgements

We would like to thank Fabrizio D'Angelo, Heidi Dempsey, and Sean Waite for all their guidance during this research project.

Red Hat