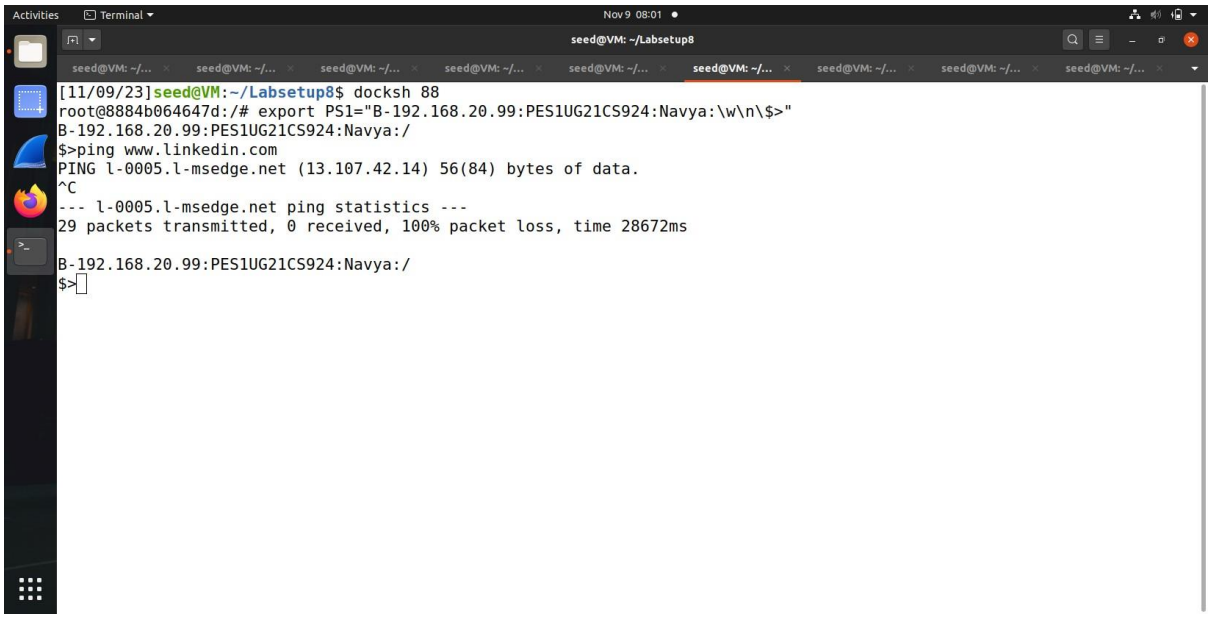
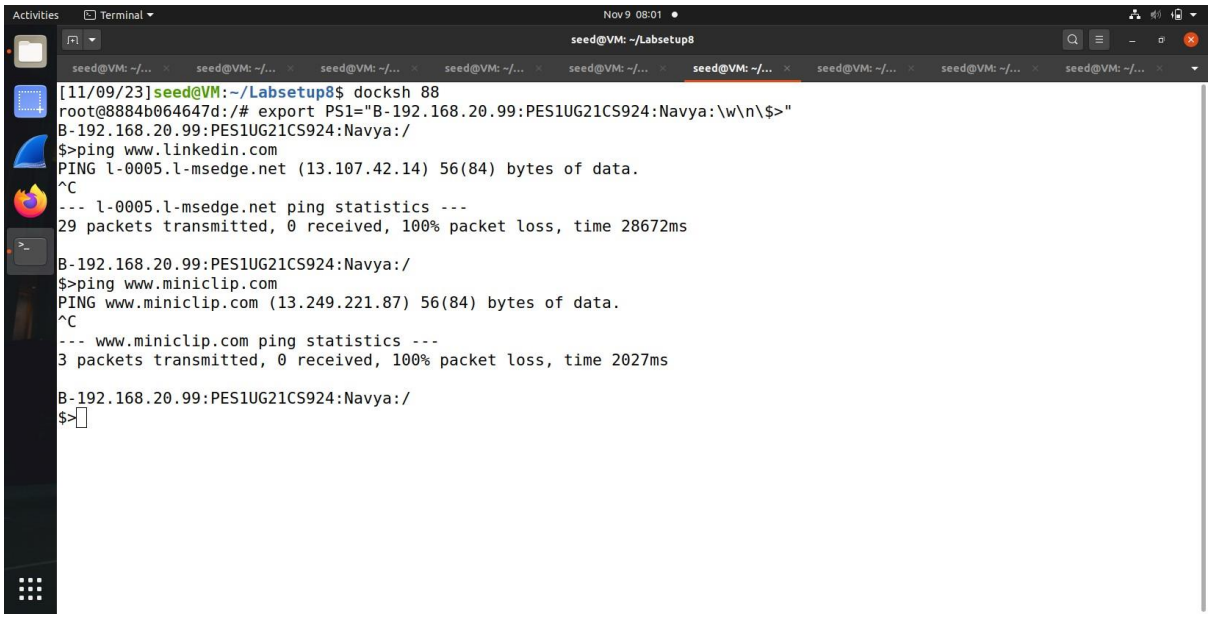
LAB 8

NAME: NAVYA PERAM

SRN: PES1UG21CS924

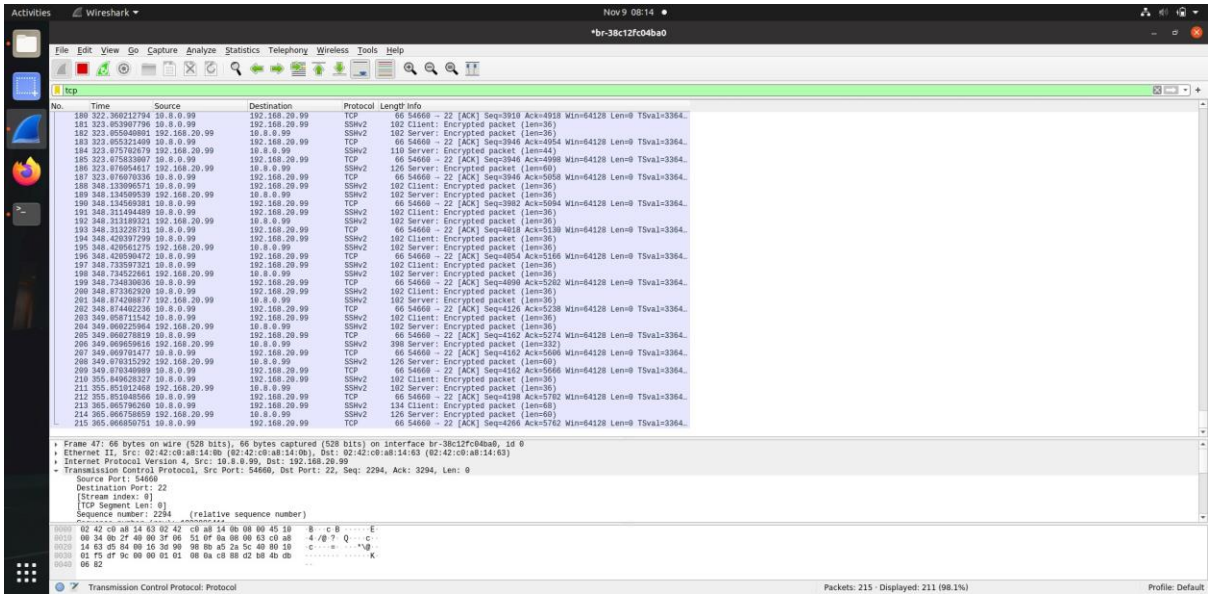


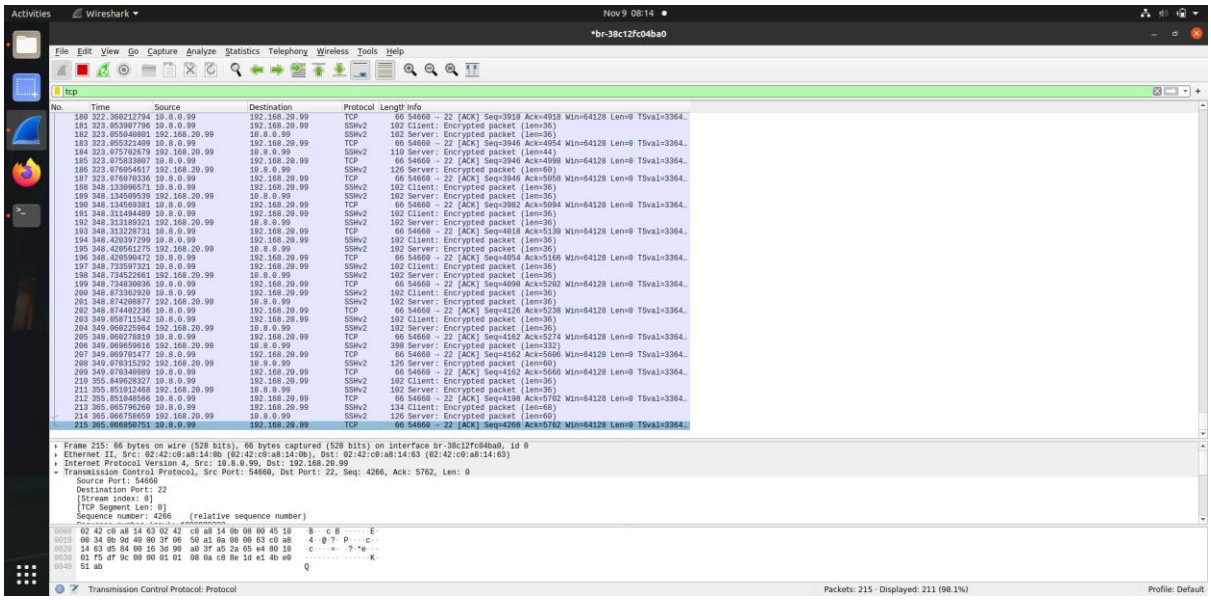


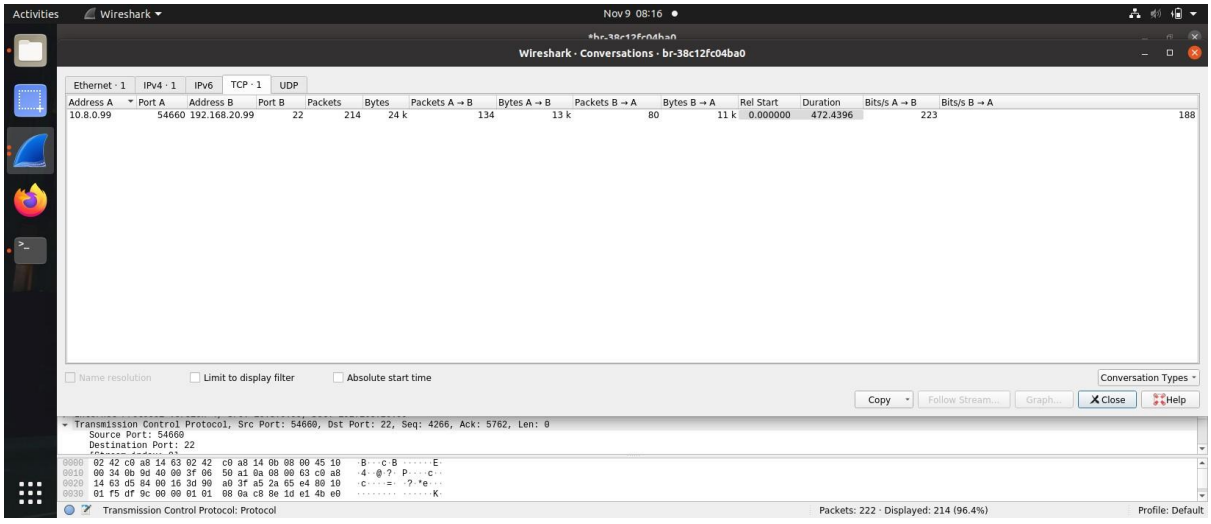
Task 1

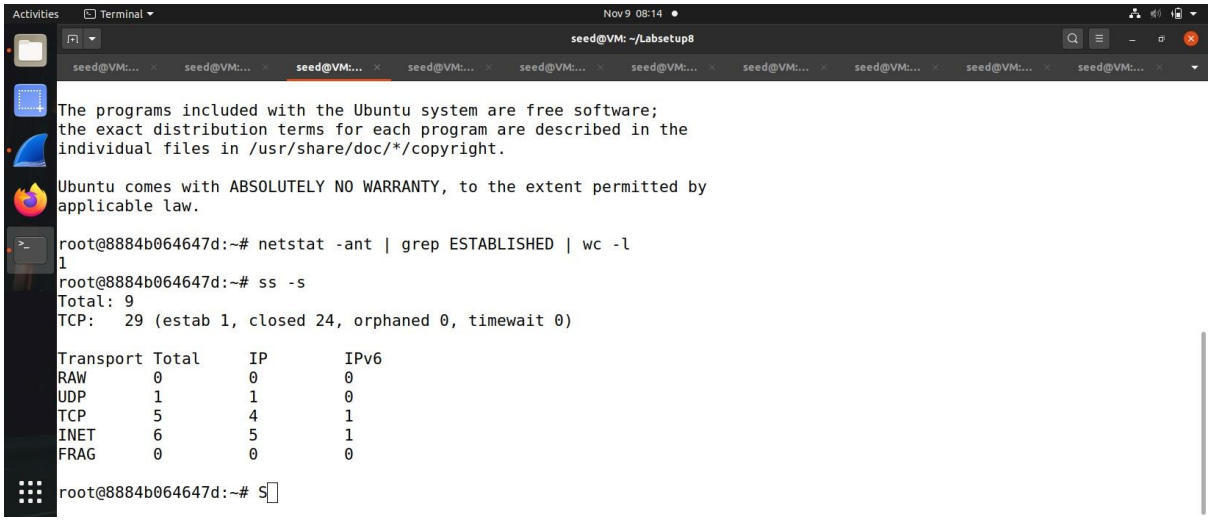
We are using the code ssh -L 0.0.0.0:8000:192.168.20.99:23 root@192.168.20.99 to create a tunnel to access services on the remote machine and to also bypass firewalls and other restrictions. In this case, the command will create a tunnel to the SSH server at 192.168.20.99, and then forward any traffic sent to port 8000 on the local machine to port 23 on the SSH server. By this we will be able to access the SSH server on port 8000 on the local machine, even if it is not accessible on port 23.

For A


Activities
Wireshark
Nov 9 08:14
*br-38c12fc04ba0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp
No. Time Source Destination Protocol Length Info
180 322.860212794 10.8.0.99 192.168.20.99 TCP 66 54660 → 22 [ACK] Seq=3910 Ack=4918 Win=64128 Len=0 TSval=3364…
181 323.053907796 10.8.0.99 192.168.20.99 SSHv2 102 Client: Encrypted packet (len=36)
182 323.055040801 192.168.20.99 10.8.0.99 SSHv2 102 Server: Encrypted packet (len=36)
183 323.055321409 10.8.0.99 192.168.20.99 TCP 66 54660 → 22 [ACK] Seq=3946 Ack=4954 Win=64128 Len=0 TSval=3364…
184 323.075702679 192.168.20.99 10.8.0.99 SSHv2 110 Server: Encrypted packet (len=44)
185 323.075833007 10.8.0.99 192.168.20.99 TCP 66 54660 → 22 [ACK] Seq=3946 Ack=4998 Win=64128 Len=0 TSval=3364…
186 323.076054617 192.168.20.99 10.8.0.99 SSHv2 126 Server: Encrypted packet (len=60)
187 323.076070336 10.8.0.99 192.168.20.99 TCP 66 54660 → 22 [ACK] Seq=3946 Ack=5058 Win=64128 Len=0 TSval=3364…
188 348.133096571 10.8.0.99 192.168.20.99 SSHv2 102 Client: Encrypted packet (len=36)
189 348.134509539 192.168.20.99 10.8.0.99 SSHv2 102 Server: Encrypted packet (len=36)
190 348.134569381 10.8.0.99 192.168.20.99 TCP 66 54660 → 22 [ACK] Seq=3982 Ack=5094 Win=64128 Len=0 TSval=3364…
191 348.311494489 10.8.0.99 192.168.20.99 SSHv2 102 Client: Encrypted packet (len=36)
192 348.313189321 192.168.20.99 10.8.0.99 SSHv2 102 Server: Encrypted packet (len=36)
193 348.313228731 10.8.0.99 192.168.20.99 TCP 66 54660 → 22 [ACK] Seq=4018 Ack=5130 Win=64128 Len=0 TSval=3364…
194 348.420397299 10.8.0.99 192.168.20.99 SSHv2 102 Client: Encrypted packet (len=36)
195 348.420561275 192.168.20.99 10.8.0.99 SSHv2 102 Server: Encrypted packet (len=36)
196 348.420590472 10.8.0.99 192.168.20.99 TCP 66 54660 → 22 [ACK] Seq=4054 Ack=5166 Win=64128 Len=0 TSval=3364…
197 348.733597321 10.8.0.99 192.168.20.99 SSHv2 102 Client: Encrypted packet (len=36)
198 348.734522661 192.168.20.99 10.8.0.99 SSHv2 102 Server: Encrypted packet (len=36)
199 348.734830036 10.8.0.99 192.168.20.99 TCP 66 54660 → 22 [ACK] Seq=4090 Ack=5202 Win=64128 Len=0 TSval=3364…
200 348.873362920 10.8.0.99 192.168.20.99 SSHv2 102 Client: Encrypted packet (len=36)
201 348.874208877 192.168.20.99 10.8.0.99 SSHv2 102 Server: Encrypted packet (len=36)
202 348.874402236 10.8.0.99 192.168.20.99 TCP 66 54660 → 22 [ACK] Seq=4126 Ack=5238 Win=64128 Len=0 TSval=3364…
203 349.058711542 10.8.0.99 192.168.20.99 SSHv2 102 Client: Encrypted packet (len=36)
204 349.060225964 192.168.20.99 10.8.0.99 SSHv2 102 Server: Encrypted packet (len=36)
205 349.060278819 10.8.0.99 192.168.20.99 TCP 66 54660 → 22 [ACK] Seq=4162 Ack=5274 Win=64128 Len=0 TSval=3364…
206 349.069659616 192.168.20.99 10.8.0.99 SSHv2 398 Server: Encrypted packet (len=332)
207 349.069761477 10.8.0.99 192.168.20.99 TCP 66 54660 → 22 [ACK] Seq=4162 Ack=5606 Win=64128 Len=0 TSval=3364…
208 349.070315292 192.168.20.99 10.8.0.99 SSHv2 126 Server: Encrypted packet (len=60)
209 349.070340989 10.8.0.99 192.168.20.99 TCP 66 54660 → 22 [ACK] Seq=4162 Ack=5666 Win=64128 Len=0 TSval=3364…
210 355.849628327 10.8.0.99 192.168.20.99 SSHv2 102 Client: Encrypted packet (len=36)
211 355.851012468 192.168.20.99 10.8.0.99 SSHv2 102 Server: Encrypted packet (len=36)
212 355.851048566 10.8.0.99 192.168.20.99 TCP 66 54660 → 22 [ACK] Seq=4198 Ack=5702 Win=64128 Len=0 TSval=3364…
213 365.865796260 10.8.0.99 192.168.20.99 SSHv2 134 Client: Encrypted packet (len=68)
214 365.866758659 192.168.20.99 10.8.0.99 SSHv2 126 Server: Encrypted packet (len=60)
215 365.866850751 10.8.0.99 192.168.20.99 TCP 66 54660 → 22 [ACK] Seq=4266 Ack=5762 Win=64128 Len=0 TSval=3364…
Frame 47: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface br-38c12fc04ba0, id 0
Ethernet II, Src: 02:42:c0:a8:14:0b (02:42:c0:a8:14:0b), Dst: 02:42:c0:a8:14:63 (02:42:c0:a8:14:63)
Internet Protocol Version 4, Src: 10.8.0.99, Dst: 192.168.20.99
Transmission Control Protocol, Src Port: 54660, Dst Port: 22, Seq: 2294, Ack: 3294, Len: 0
Source Port: 54660
Destination Port: 22
[Stream index: 0]
[TCP Segment Len: 0]
Sequence number: 2294 (relative sequence number)
0000 02 42 c0 a8 14 63 02 42 c0 a8 14 0b 08 00 45 10 ·B···c·B ····E·
0010 00 34 0b 2f 40 00 3f 06 51 0f 0a 08 00 63 c0 a8 ·4·/@·?· Q····c··
0020 14 63 d5 84 00 16 3d 90 98 0b a5 2a 5c 40 80 10 ·c····=· ···*\@··
0030 01 f5 df 9c 00 00 01 01 08 0a c8 88 d2 b8 4b db ·····K·
0040 06 82
Transmission Control Protocol: Protocol
Packets: 215 · Displayed: 211 (98.1%)
Profile: Default

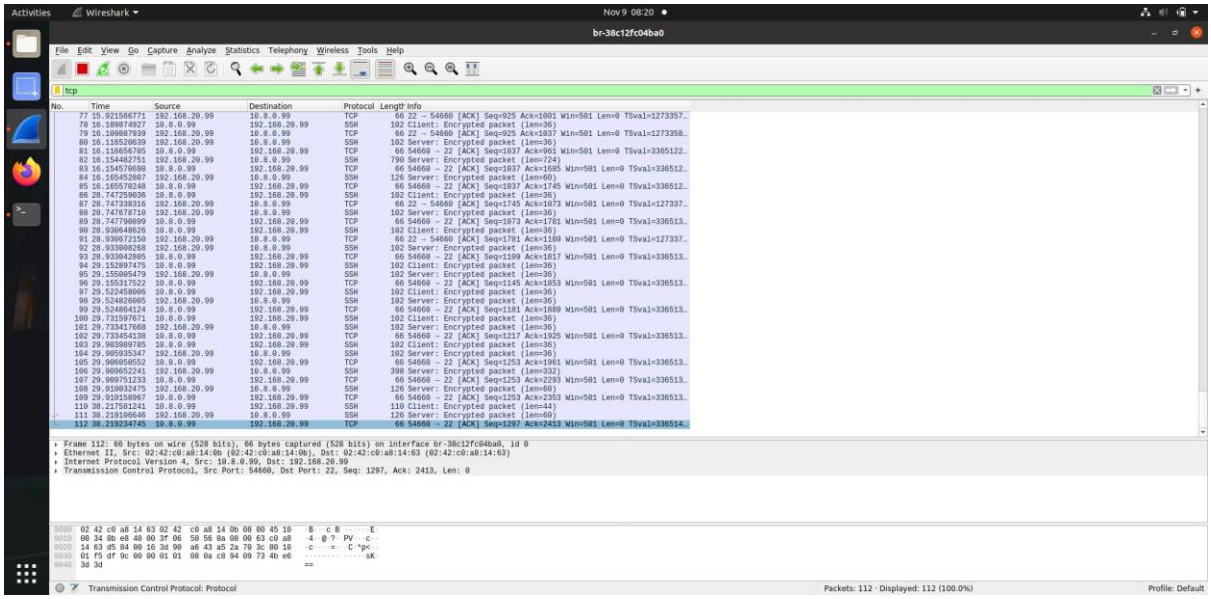

Activities
Wireshark
Nov 9 08:14
*br-38c12fc04ba0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp
No. Time Source Destination Protocol Length Info
180 322.860212794 10.8.0.99 192.168.20.99 TCP 66 54660 → 22 [ACK] Seq=3910 Ack=4918 Win=64128 Len=0 TSval=3364…
181 323.053907796 10.8.0.99 192.168.20.99 SSHv2 102 Client: Encrypted packet (len=36)
182 323.055040801 192.168.20.99 10.8.0.99 SSHv2 102 Server: Encrypted packet (len=36)
183 323.055321409 10.8.0.99 192.168.20.99 TCP 66 54660 → 22 [ACK] Seq=3946 Ack=4954 Win=64128 Len=0 TSval=3364…
184 323.075702679 192.168.20.99 10.8.0.99 SSHv2 110 Server: Encrypted packet (len=44)
185 323.075833007 10.8.0.99 192.168.20.99 TCP 66 54660 → 22 [ACK] Seq=3946 Ack=4998 Win=64128 Len=0 TSval=3364…
186 323.076054617 192.168.20.99 10.8.0.99 SSHv2 126 Server: Encrypted packet (len=60)
187 323.076070336 10.8.0.99 192.168.20.99 TCP 66 54660 → 22 [ACK] Seq=3946 Ack=5058 Win=64128 Len=0 TSval=3364…
188 348.133096571 10.8.0.99 192.168.20.99 SSHv2 102 Client: Encrypted packet (len=36)
189 348.134509539 192.168.20.99 10.8.0.99 SSHv2 102 Server: Encrypted packet (len=36)
190 348.134569381 10.8.0.99 192.168.20.99 TCP 66 54660 → 22 [ACK] Seq=3982 Ack=5094 Win=64128 Len=0 TSval=3364…
191 348.311494489 10.8.0.99 192.168.20.99 SSHv2 102 Client: Encrypted packet (len=36)
192 348.313189321 192.168.20.99 10.8.0.99 SSHv2 102 Server: Encrypted packet (len=36)
193 348.313228731 10.8.0.99 192.168.20.99 TCP 66 54660 → 22 [ACK] Seq=4018 Ack=5130 Win=64128 Len=0 TSval=3364…
194 348.420397299 10.8.0.99 192.168.20.99 SSHv2 102 Client: Encrypted packet (len=36)
195 348.420561275 192.168.20.99 10.8.0.99 SSHv2 102 Server: Encrypted packet (len=36)
196 348.420590472 10.8.0.99 192.168.20.99 TCP 66 54660 → 22 [ACK] Seq=4054 Ack=5166 Win=64128 Len=0 TSval=3364…
197 348.733597321 10.8.0.99 192.168.20.99 SSHv2 102 Client: Encrypted packet (len=36)
198 348.734522661 192.168.20.99 10.8.0.99 SSHv2 102 Server: Encrypted packet (len=36)
199 348.734830036 10.8.0.99 192.168.20.99 TCP 66 54660 → 22 [ACK] Seq=4090 Ack=5202 Win=64128 Len=0 TSval=3364…
200 348.873362920 10.8.0.99 192.168.20.99 SSHv2 102 Client: Encrypted packet (len=36)
201 348.874208877 192.168.20.99 10.8.0.99 SSHv2 102 Server: Encrypted packet (len=36)
202 348.874402236 10.8.0.99 192.168.20.99 TCP 66 54660 → 22 [ACK] Seq=4126 Ack=5238 Win=64128 Len=0 TSval=3364…
203 349.058711542 10.8.0.99 192.168.20.99 SSHv2 102 Client: Encrypted packet (len=36)
204 349.060225964 192.168.20.99 10.8.0.99 SSHv2 102 Server: Encrypted packet (len=36)
205 349.060278819 10.8.0.99 192.168.20.99 TCP 66 54660 → 22 [ACK] Seq=4162 Ack=5274 Win=64128 Len=0 TSval=3364…
206 349.069659616 192.168.20.99 10.8.0.99 SSHv2 398 Server: Encrypted packet (len=332)
207 349.069761477 10.8.0.99 192.168.20.99 TCP 66 54660 → 22 [ACK] Seq=4162 Ack=5606 Win=64128 Len=0 TSval=3364…
208 349.070315292 192.168.20.99 10.8.0.99 SSHv2 126 Server: Encrypted packet (len=60)
209 349.070340989 10.8.0.99 192.168.20.99 TCP 66 54660 → 22 [ACK] Seq=4162 Ack=5666 Win=64128 Len=0 TSval=3364…
210 355.849628327 10.8.0.99 192.168.20.99 SSHv2 102 Client: Encrypted packet (len=36)
211 355.851012468 192.168.20.99 10.8.0.99 SSHv2 102 Server: Encrypted packet (len=36)
212 355.851048566 10.8.0.99 192.168.20.99 TCP 66 54660 → 22 [ACK] Seq=4198 Ack=5702 Win=64128 Len=0 TSval=3364…
213 365.865796260 10.8.0.99 192.168.20.99 SSHv2 134 Client: Encrypted packet (len=68)
214 365.866758659 192.168.20.99 10.8.0.99 SSHv2 126 Server: Encrypted packet (len=60)
215 365.866850751 10.8.0.99 192.168.20.99 TCP 66 54660 → 22 [ACK] Seq=4266 Ack=5762 Win=64128 Len=0 TSval=3364…
Frame 215: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface br-38c12fc04ba0, id 0
Ethernet II, Src: 02:42:c0:a8:14:0b (02:42:c0:a8:14:0b), Dst: 02:42:c0:a8:14:63 (02:42:c0:a8:14:63)
Internet Protocol Version 4, Src: 10.8.0.99, Dst: 192.168.20.99
Transmission Control Protocol, Src Port: 54660, Dst Port: 22, Seq: 4266, Ack: 5762, Len: 0
Source Port: 54660
Destination Port: 22
[Stream index: 0]
[TCP Segment Len: 0]
Sequence number: 4266 (relative sequence number)
0000 02 42 c0 a8 14 63 02 42 c0 a8 14 0b 08 00 45 10 ·B···c·B ····E·
0010 00 34 0b 9d 40 00 3f 06 50 a1 0a 08 00 63 c0 a8 ·4··@·?· P····c··
0020 14 63 d5 84 00 16 3d 90 a0 3f a5 2a 65 e4 80 10 ·c····=· ·?·*e···
0030 01 f5 df 9c 00 00 01 01 08 0a c8 8e 1d e1 4b e0 ·····K·
0040 51 ab Q·
Transmission Control Protocol: Protocol
Packets: 215 · Displayed: 211 (98.1%)
Profile: Default

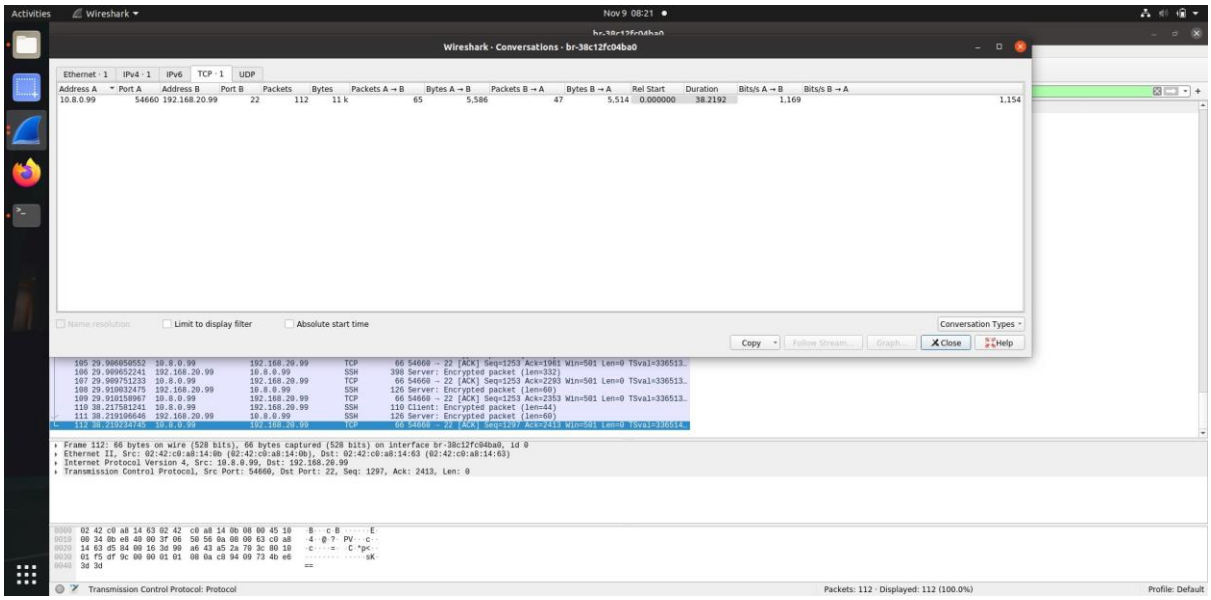

Activities
Wireshark
Nov 9 08:16
Wireshark · Conversations · br-38c12fc04ba0
Ethernet · 1 IPv4 · 1 IPv6 TCP · 1 UDP
Address A Port A Address B Port B Packets Bytes Packets A → B Bytes A → B Packets B → A Bytes B → A Rel Start Duration Bits/s A → B Bits/s B → A
10.8.0.99 54660 192.168.20.99 22 214 24 k 134 13 k 80 11 k 0.000000 472.4396 223 188
Name resolution
Limit to display filter
Absolute start time
Conversation Types
Copy
Follow Stream…
Graph…
Close
Help
Transmission Control Protocol, Src Port: 54660, Dst Port: 22, Seq: 4266, Ack: 5762, Len: 0
Source Port: 54660
Destination Port: 22
0000 02 42 c0 a8 14 63 02 42 c0 a8 14 0b 08 00 45 10 ·B···c·B ····E·
0010 00 34 0b 9d 40 00 3f 06 50 a1 0a 08 00 63 c0 a8 ·4··@·?· P····c··
0020 14 63 d5 84 00 16 3d 90 a0 3f a5 2a 65 e4 80 10 ·c····=· ·?·*e···
0030 01 f5 df 9c 00 00 01 01 08 0a c8 8e 1d e1 4b e0 ·····K·
Transmission Control Protocol: Protocol
Packets: 222 · Displayed: 214 (96.4%)
Profile: Default

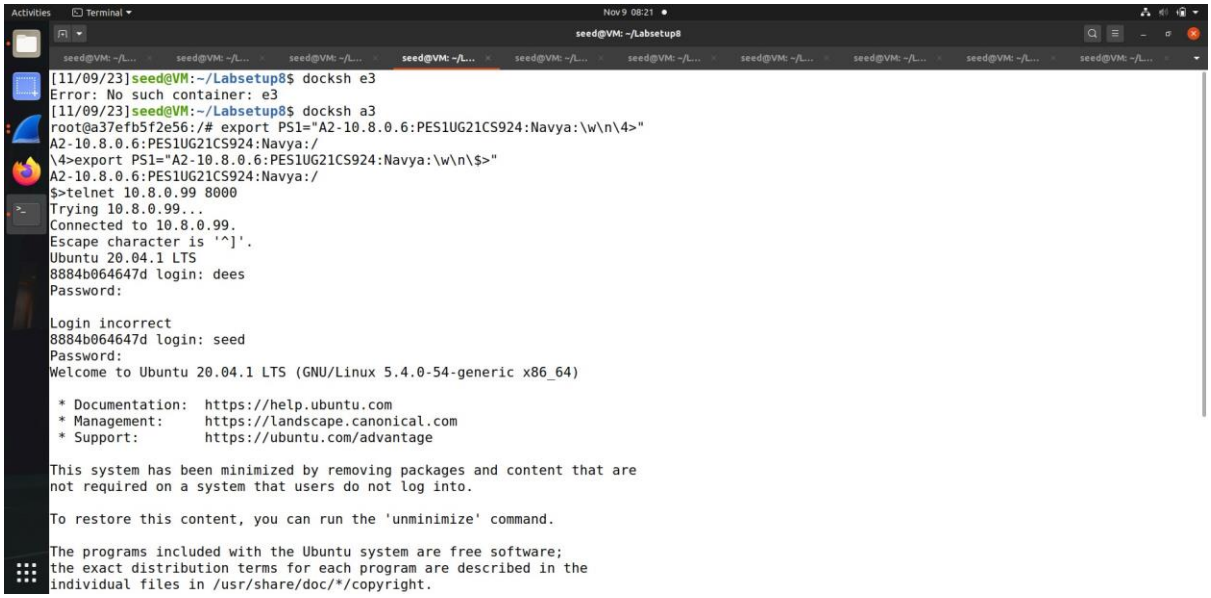


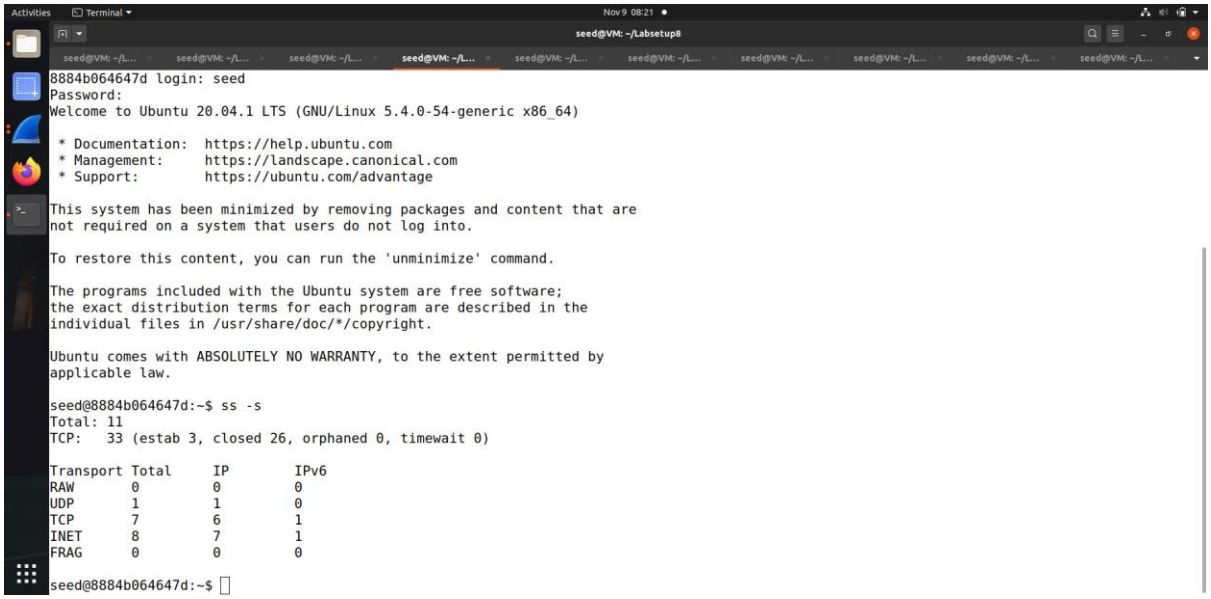
The number of tcp connections with 1 established and 24 closed.

For A2



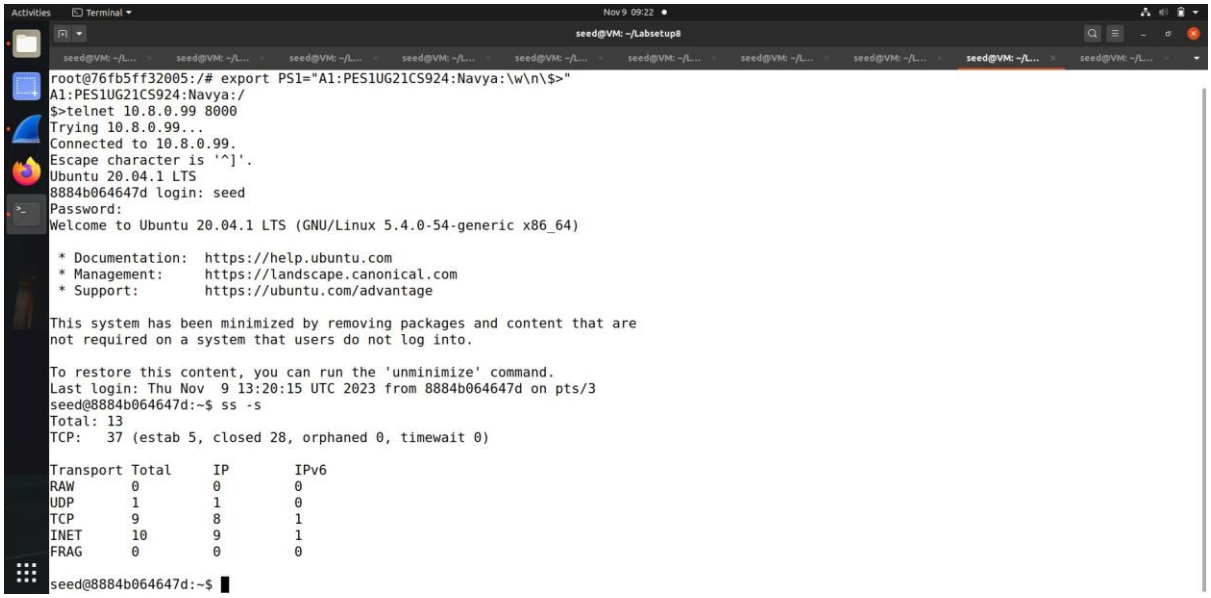




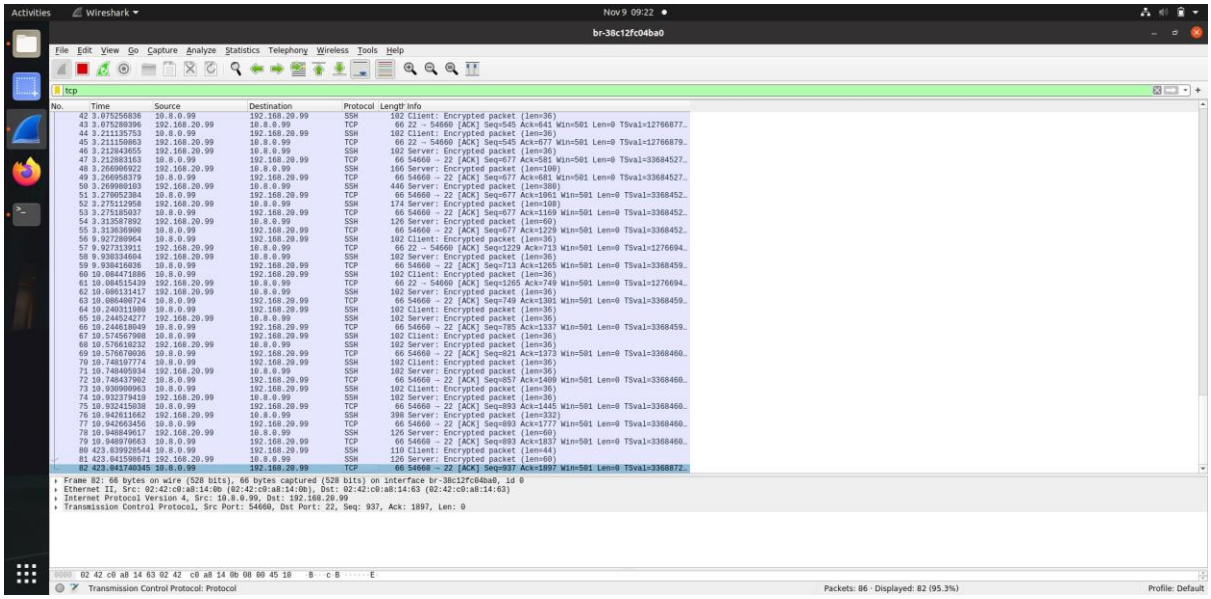


The number of TCP connections with 3 established and 26 closed.

For A1



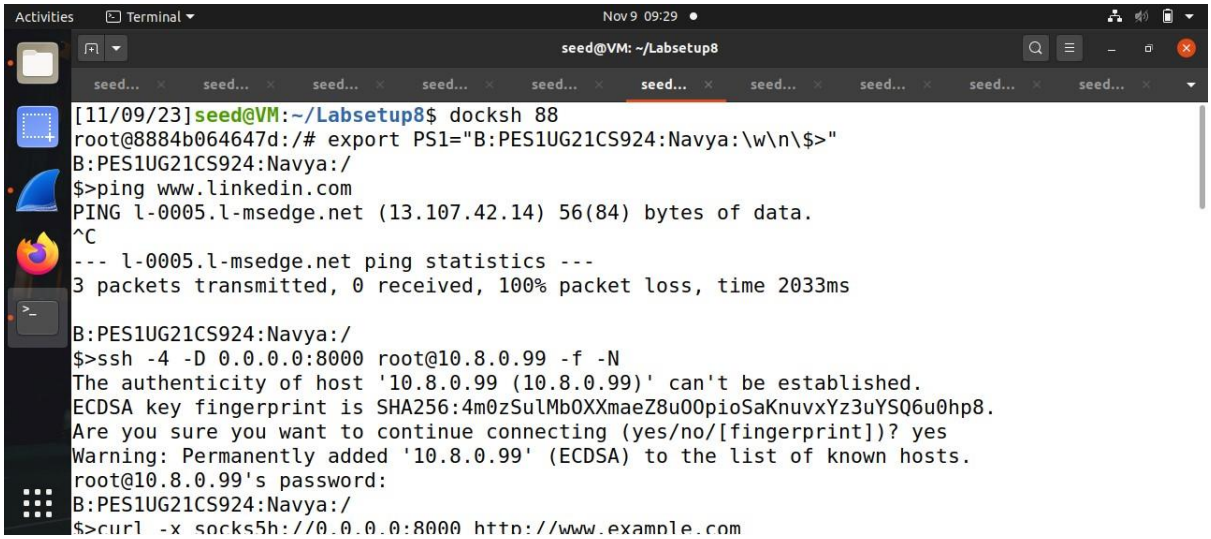
The number of TCP connections with 5 established and 28 closed.

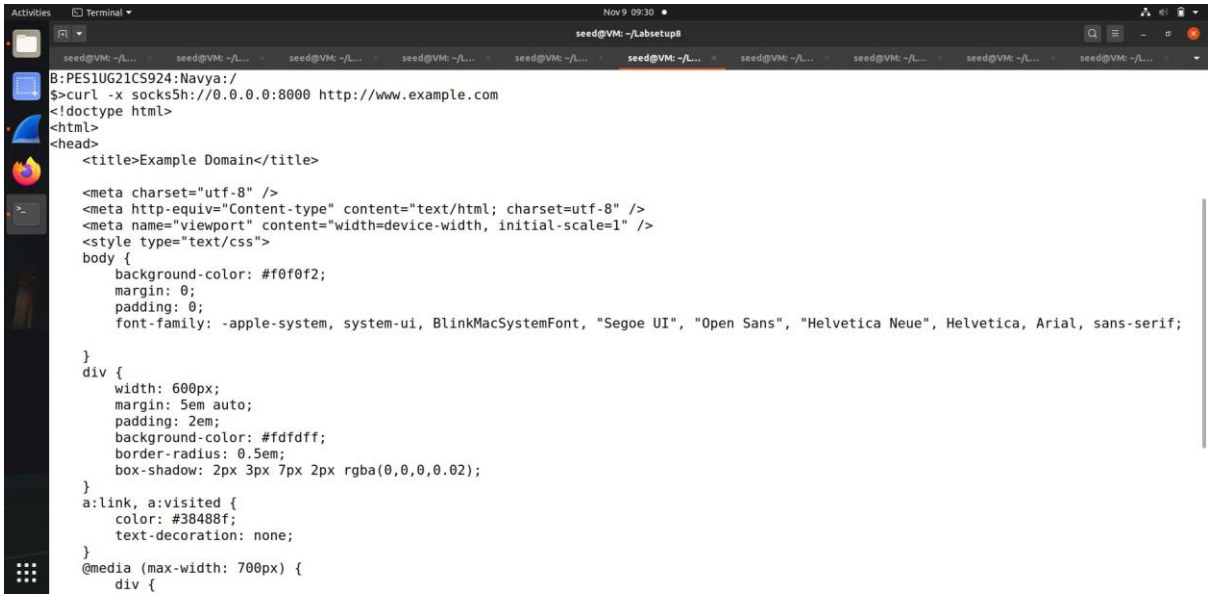


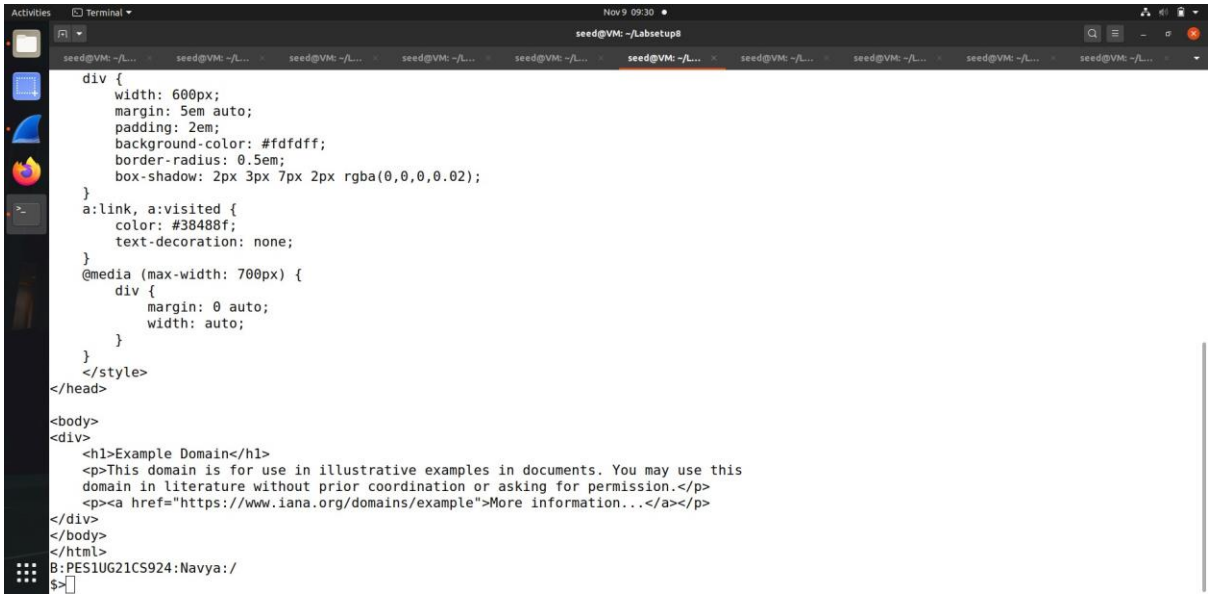
1) The total number of tcp connections in the entire process are 37.
2) The tunnel specified in the lab setup can successfully help users evade the firewall rule because it creates a direct connection between the client and the SSH server. The firewall rule only blocks traffic on port 22, but the tunnel forwards traffic on port 8000 to port 23 on the SSH server. This means that the traffic can bypass the firewall rule and reach the SSH server. Once the traffic reaches the SSH server, it is encrypted and forwarded to the remote host. This means that the firewall cannot see the content of the traffic, and therefore cannot block it. The client sends traffic to port 8000 on localhost. This traffic is forwarded to port 23 on the SSH server. The SSH server encrypts the traffic and forwards it to the remote host on port 22.The firewall cannot see the content of the traffic because it is encrypted. Therefore, the firewall cannot block the traffic. This type of tunnel is often used to bypass firewalls or other restrictions. It can also be used to create a secure connection to a remote host.

Task 2.1

Here, the command ssh -4 -D 0.0.0.0:8000 root@10.8.0.99 -f -N establishes a dynamic port forward from port 8000 on the local computer to port 22 on the server at IP address 10.8.0.99.The 0.0.0.0 part of the command specifies that the local host is any interface on the computer. This means that traffic sent to port 8000 on any interface on the computer will be forwarded to the remote server. The root@10.8.0.99 part of the command specifies that the remote host is the server at IP address 10.8.0.99. The -f flag puts the SSH session into the background. This means that the command will continue to run even after the closing of the terminal window.

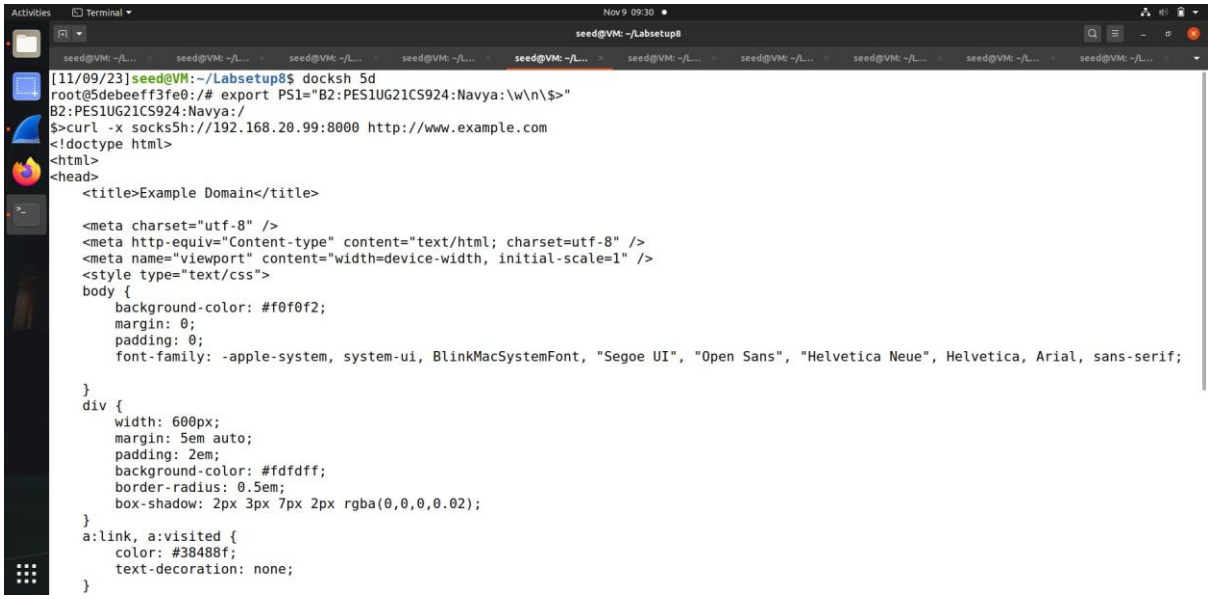


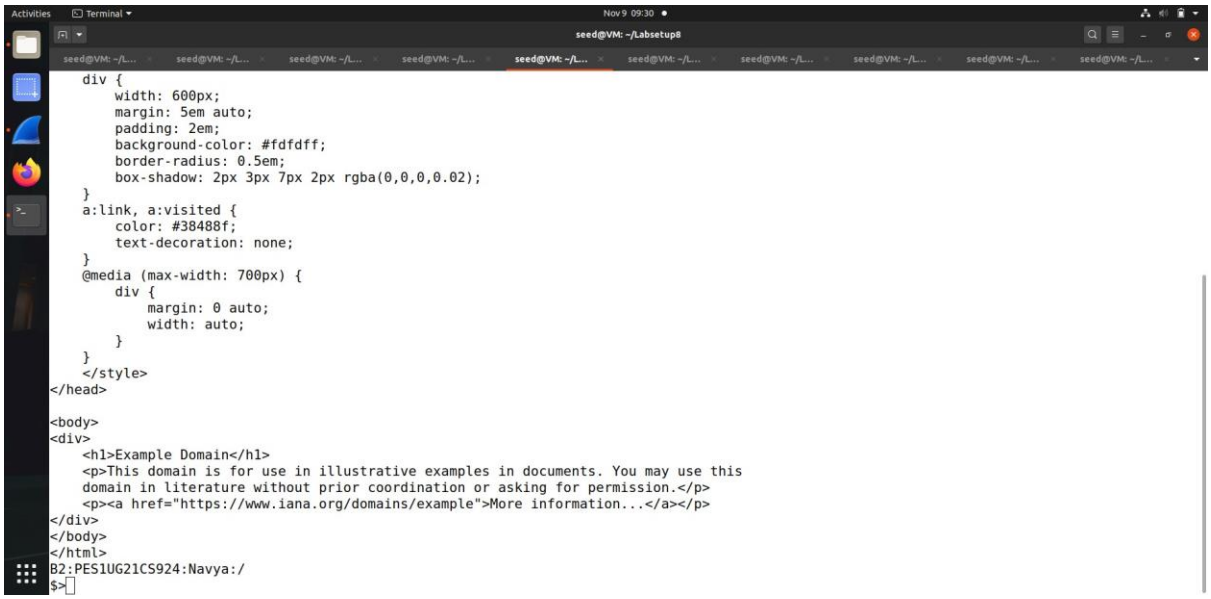



Activities
Terminal
Nov 9 09:30
seed@VM: ~/Labsetup8
div {
width: 600px;
margin: 5em auto;
padding: 2em;
background-color: #fdfdff;
border-radius: 0.5em;
box-shadow: 2px 3px 7px 2px rgba(0,0,0,0.02);
}
a:link, a:visited {
color: #38488f;
text-decoration: none;
}
@media (max-width: 700px) {
div {
margin: 0 auto;
width: auto;
}
}
</style>
</head>
<body>

<h1>Example Domain</h1>
This domain is for use in illustrative examples in documents. You may use this
domain in literature without prior coordination or asking for permission.
More information...

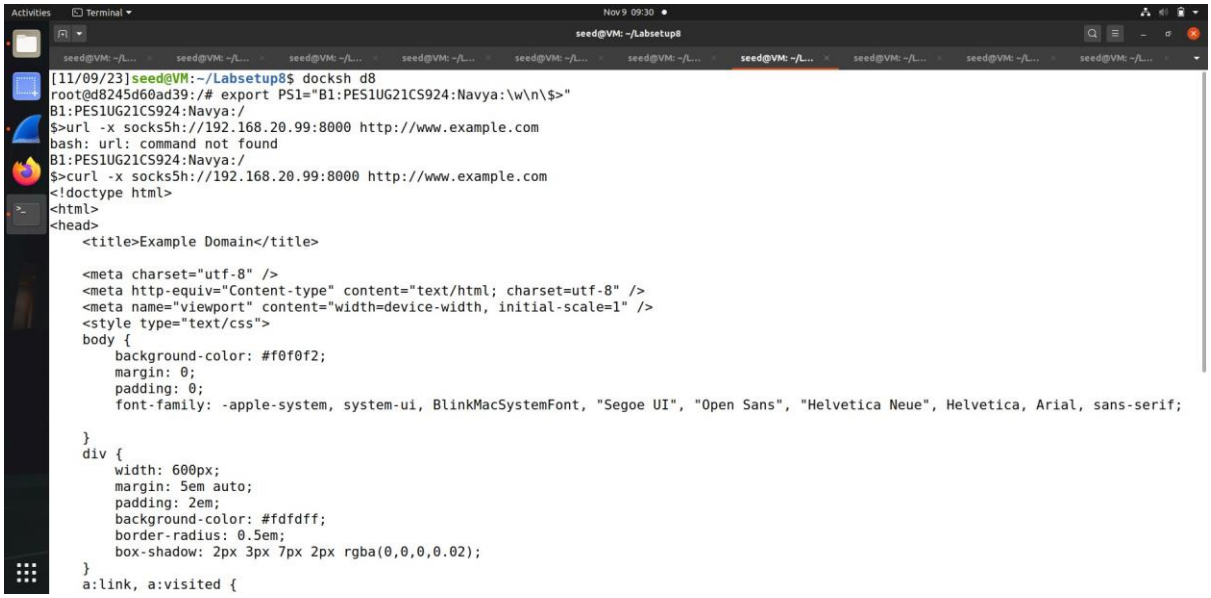
</body>
</html>
B:PES1UG21CS924:Navya:/
$>

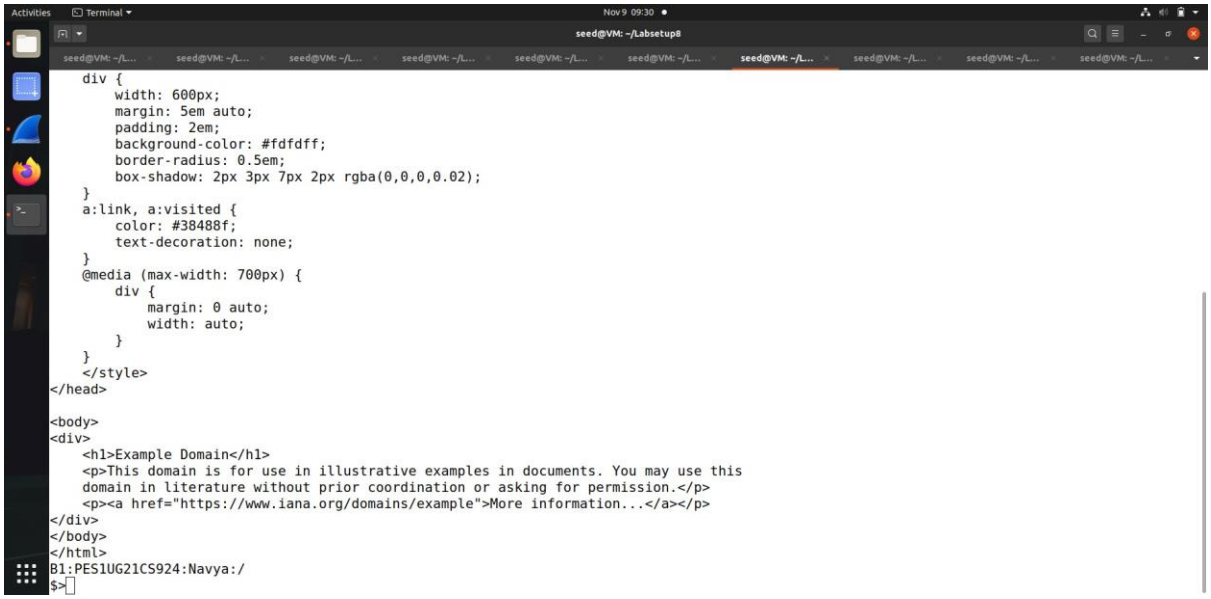

Activities
Terminal
Nov 9 09:30
seed@VM: ~/Labsetup8
[11/09/23]seed@VM:~/Labsetup8$ docksh 5d
root@5debeeff3fe0:/# export PS1="B2:PES1UG21CS924:Navya:\w\n\$>"
B2:PES1UG21CS924:Navya:/
$>curl -x socks5h://192.168.20.99:8000 http://www.example.com
<!doctype html>
<html>
<head>
<title>Example Domain</title>
<meta charset="utf-8" />
<meta http-equiv="Content-type" content="text/html; charset=utf-8" />
<meta name="viewport" content="width=device-width, initial-scale=1" />
<style type="text/css">
body {
background-color: #f0f0f2;
margin: 0;
padding: 0;
font-family: -apple-system, system-ui, BlinkMacSystemFont, "Segoe UI", "Open Sans", "Helvetica Neue", Helvetica, Arial, sans-serif;
}
div {
width: 600px;
margin: 5em auto;
padding: 2em;
background-color: #fdfdff;
border-radius: 0.5em;
box-shadow: 2px 3px 7px 2px rgba(0,0,0,0.02);
}
a:link, a:visited {
color: #38488f;
text-decoration: none;
}


Activities
Terminal
Nov 9 09:30
seed@VM: ~/Labsetup8
div {
width: 600px;
margin: 5em auto;
padding: 2em;
background-color: #fdfdff;
border-radius: 0.5em;
box-shadow: 2px 3px 7px 2px rgba(0,0,0,0.02);
}
a:link, a:visited {
color: #38488f;
text-decoration: none;
}
@media (max-width: 700px) {
div {
margin: 0 auto;
width: auto;
}
}
</style>
</head>
<body>

<h1>Example Domain</h1>
This domain is for use in illustrative examples in documents. You may use this
domain in literature without prior coordination or asking for permission.
More information...

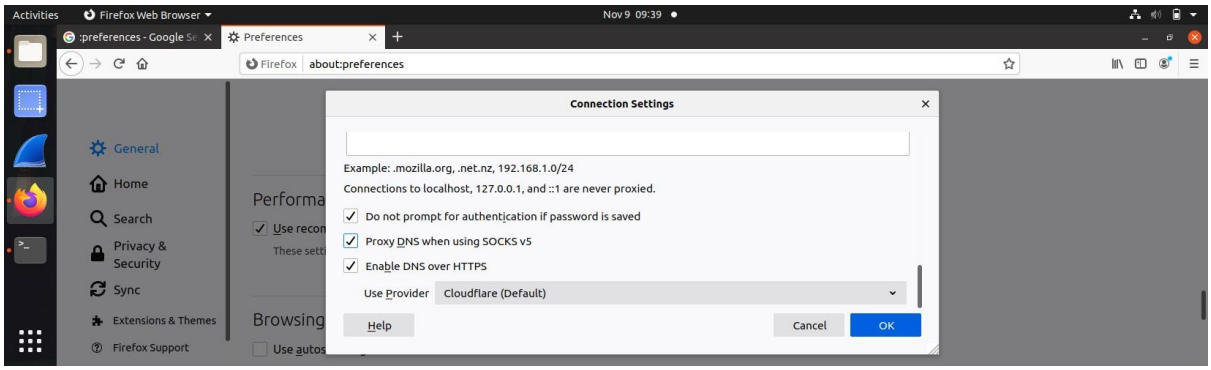
</body>
</html>
B2:PES1UG21CS924:Navya:/
$>



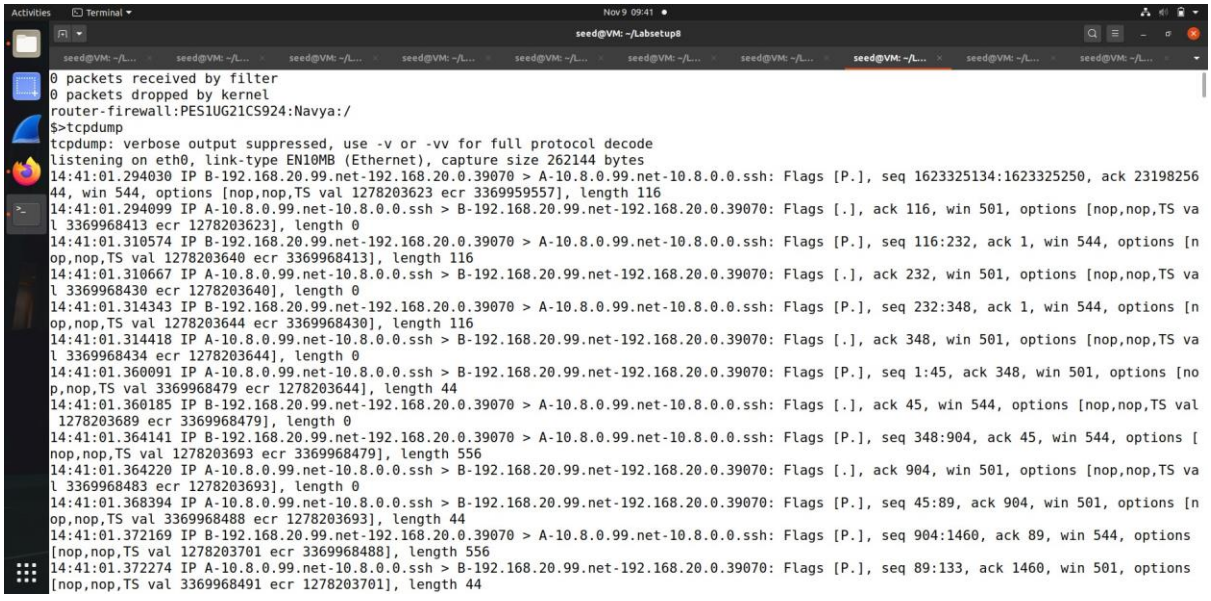


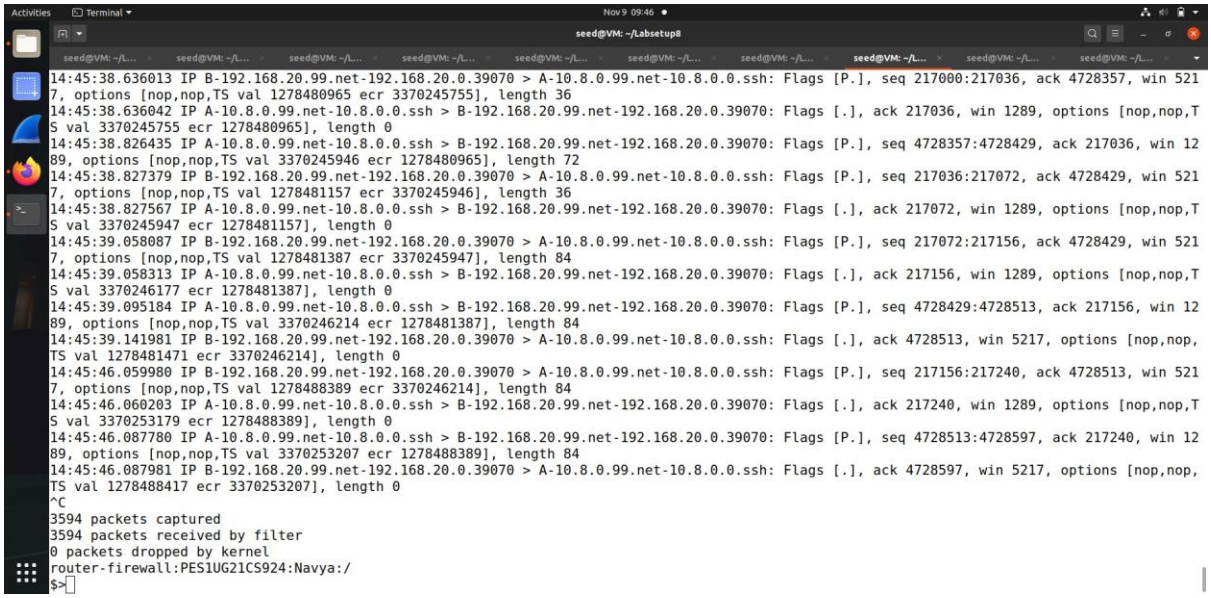
1) Here b1, b2 are only forwarding traffic to the intended web server, they do not establish an actual connection with the web server. The actual connection is established by the client computer that is sending the request to the web server. They are performing the task by SSH port forwarding. SSH port forwarding is a way to tunnel traffic through a secure SSH connection. This can be useful for bypassing firewalls or other restrictions, or for accessing services that are only accessible over SSH. They are using SSH port forwarding to forward traffic from port 8000 on the local machine to port 22 on the remote machine. This means that any traffic that is sent to port 8000 on the local machine will be forwarded to the remote machine. B will establish a connection directly with the intended web server. The computer sends its traffic to the SSH tunnel on computer B. The SSH tunnel forwards the traffic to the intended web server. The intended web server sends its responses back to the SSH tunnel, which then forwards them back to the computer.
2) The computer knows which server to connect to based on the destination port that is specified in the SSH port forwarding command. The destination port is 8000. This means that the computer will forward all traffic on port 8000 to the remote server.

Task 2.2


Activities
Firefox Web Browser
Nov 9 09:39
:preferences - Google Se
Preferences
Firefox
about:preferences
General
Home
Search
Privacy & Security
Sync
Extensions & Themes
Firefox Support
Performa
Use recon
These setti
Browsing
Use autos
Connection Settings
Example: .mozilla.org, .net.nz, 192.168.1.0/24
Connections to localhost, 127.0.0.1, and ::1 are never proxied.
Do not prompt for authentication if password is saved
Proxy DNS when using SOCKS v5
Enable DNS over HTTPS
Use Provider
Cloudflare (Default)
Help
Cancel
OK

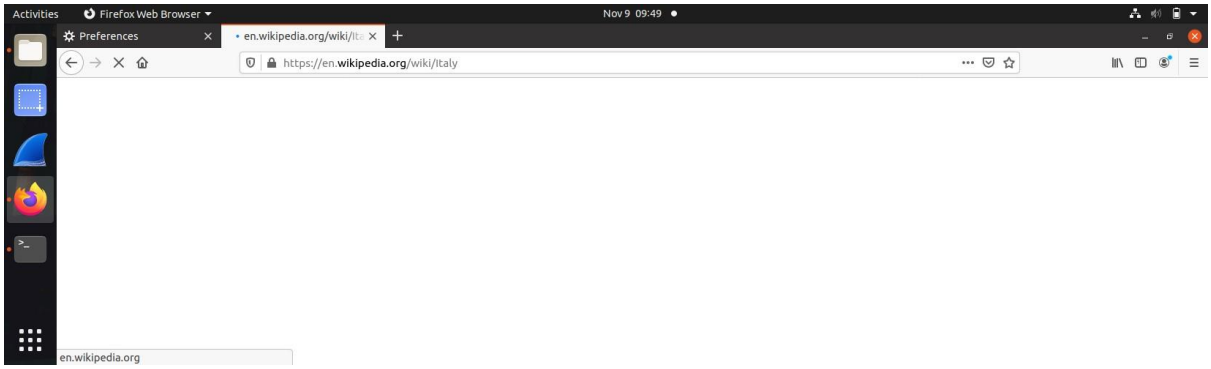

Activities
Firefox Web Browser
Nov 9 09:41
:preferences - Google Se
Preferences
Italy - Wikipedia
https://en.wikipedia.org/wiki/Italy
WIKIPEDIA
The Free Encyclopedia
Search Wikipedia
Search
Create account
Log in
Italy
310 languages
Article
Talk
Read
View source
View history
Tools
From Wikipedia, the free encyclopedia
Coordinates: 43°N 12°E
"Italia" redirects here. For other uses, see Italy (disambiguation) and Italia (disambiguation).
This article may be too long to read and navigate comfortably. Its current readable prose size is 140 kilobytes. Please consider splitting content into sub-articles, condensing it, or adding subheadings. Please discuss this issue on the article's talk page. (June 2023)
Contents [hide]
(Top)
Name
History
Geography
Politics
Economy
Demographics
Culture
See also
Notes
References
Bibliography
External links
Republic of Italy
Repubblica Italiana (Italian)
Flag
Emblem
Anthem: "Il Canto degli Italiani"
"The Song of the Italians"
1:39


Activities
Terminal
Nov 9 09:41
seed@VM: ~/Labsetup8
0 packets received by filter
0 packets dropped by kernel
router-firewall:PES1UG21CS924:Navya:/
$>tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
14:41:01.294030 IP B-192.168.20.99.net-192.168.20.0.39070 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [P.], seq 1623325134:1623325250, ack 2319825644, win 544, options [nop,nop,TS val 1278203623 ecr 3369959557], length 116
14:41:01.294099 IP A-10.8.0.99.net-10.8.0.0.ssh > B-192.168.20.99.net-192.168.20.0.39070: Flags [.], ack 116, win 501, options [nop,nop,TS val 3369968413 ecr 1278203623], length 0
14:41:01.310574 IP B-192.168.20.99.net-192.168.20.0.39070 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [P.], seq 116:232, ack 1, win 544, options [nop,nop,TS val 1278203640 ecr 3369968413], length 116
14:41:01.310667 IP A-10.8.0.99.net-10.8.0.0.ssh > B-192.168.20.99.net-192.168.20.0.39070: Flags [.], ack 232, win 501, options [nop,nop,TS val 3369968430 ecr 1278203640], length 0
14:41:01.314343 IP B-192.168.20.99.net-192.168.20.0.39070 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [P.], seq 232:348, ack 1, win 544, options [nop,nop,TS val 1278203644 ecr 3369968430], length 116
14:41:01.314418 IP A-10.8.0.99.net-10.8.0.0.ssh > B-192.168.20.99.net-192.168.20.0.39070: Flags [.], ack 348, win 501, options [nop,nop,TS val 3369968434 ecr 1278203644], length 0
14:41:01.360091 IP A-10.8.0.99.net-10.8.0.0.ssh > B-192.168.20.99.net-192.168.20.0.39070: Flags [P.], seq 1:45, ack 348, win 501, options [nop,nop,TS val 3369968479 ecr 1278203644], length 44
14:41:01.360185 IP B-192.168.20.99.net-192.168.20.0.39070 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [.], ack 45, win 544, options [nop,nop,TS val 1278203689 ecr 3369968479], length 0
14:41:01.364141 IP B-192.168.20.99.net-192.168.20.0.39070 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [P.], seq 348:904, ack 45, win 544, options [nop,nop,TS val 1278203693 ecr 3369968479], length 556
14:41:01.364220 IP A-10.8.0.99.net-10.8.0.0.ssh > B-192.168.20.99.net-192.168.20.0.39070: Flags [.], ack 904, win 501, options [nop,nop,TS val 3369968483 ecr 1278203693], length 0
14:41:01.368394 IP A-10.8.0.99.net-10.8.0.0.ssh > B-192.168.20.99.net-192.168.20.0.39070: Flags [P.], seq 45:89, ack 904, win 501, options [nop,nop,TS val 3369968488 ecr 1278203693], length 44
14:41:01.372169 IP B-192.168.20.99.net-192.168.20.0.39070 > A-10.8.0.99.net-10.8.0.0.ssh: Flags [P.], seq 904:1460, ack 89, win 544, options [nop,nop,TS val 1278203701 ecr 3369968488], length 556
14:41:01.372274 IP A-10.8.0.99.net-10.8.0.0.ssh > B-192.168.20.99.net-192.168.20.0.39070: Flags [P.], seq 89:133, ack 1460, win 501, options [nop,nop,TS val 3369968491 ecr 1278203701], length 44



Since we can access the website on the firefox we can say that the the tunnel is working accurately.

1) The traffic involved in the port forwarding process is: Computer in the container --> Local machine (port 8000) --> SSH tunnel --> Remote machine --> Intended web server

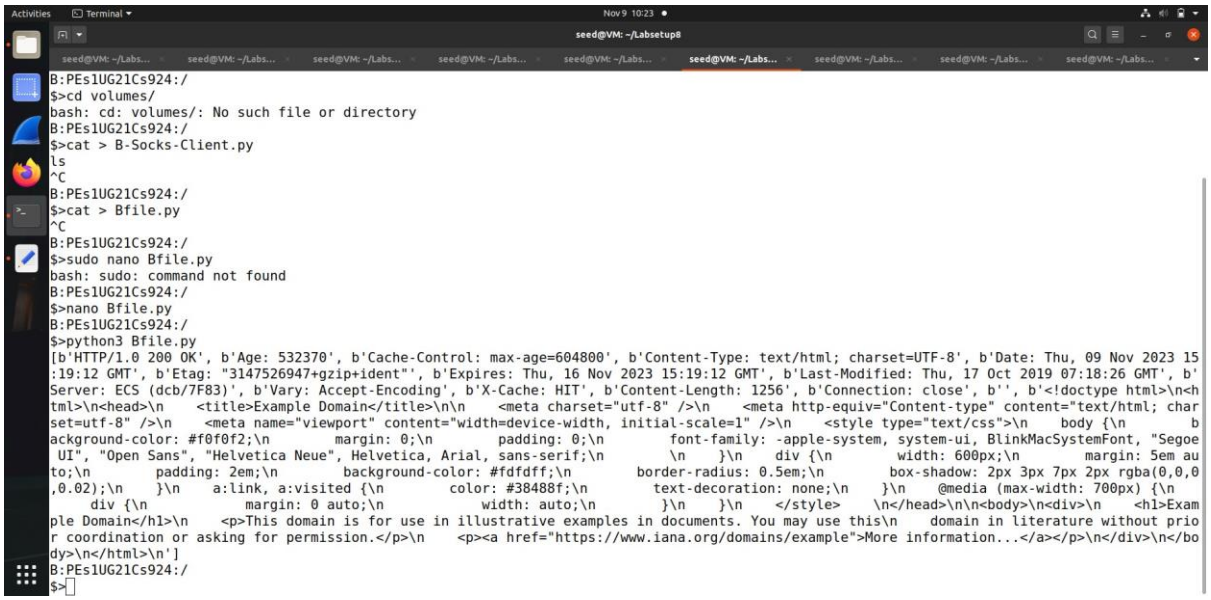


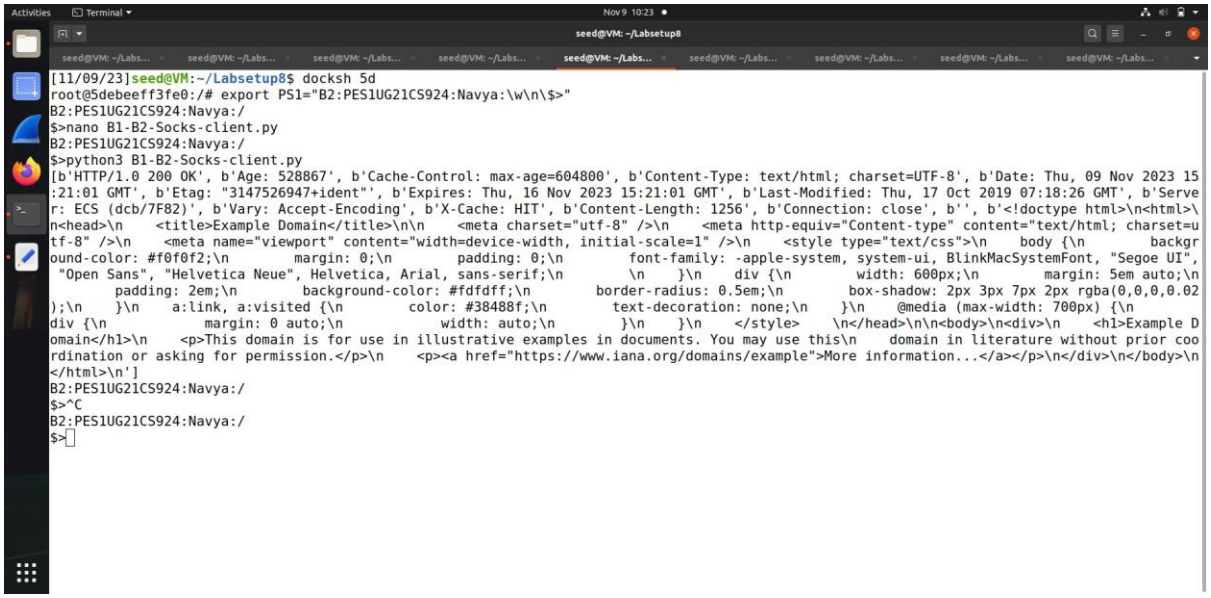
2) Once the SSH session is killed, the SSH tunnel will be broken and all traffic that is being forwarded through the tunnel will be dropped. If we try to browse a website after the SSH tunnel has been broken, we will receive an error message. This is because the computer will not be able to connect to the intended web server. This is because Firefox is trying to connect to Google directly, but the SSH tunnel has been broken and the computer cannot connect to the internet directly.

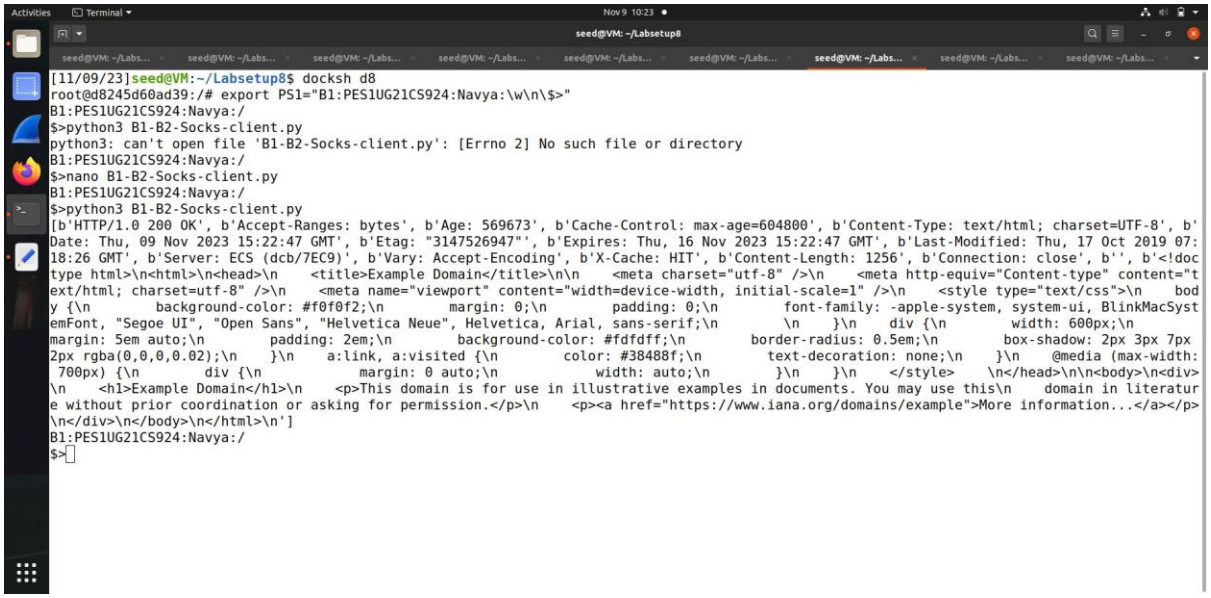
Task 2.3

Here , we use the command ssh -4 -D 0.0.0.0:8000 root@10.8.0.99 -f -N, which creates an SSH tunnel with dynamic port forwarding enabled. This means that all traffic sent to port 8000 on the local machine will be forwarded to port 22 on the remote machine at 10.8.0.99.The -4 flag refers to the usage of the IPv4 protocol. The -D flag enables dynamic port forwarding. The 0.0.0.0:8000 flag specifies forward traffic from port 8000 on the local machine. The root@10.8.0.99 flag specifies the

user and IP address of the remote machine to forward the traffic to. The -f flag puts the SSH session in the background. The -N flag prevents the SSH session from requesting a shell.







We are able to access the website example.com using the above codes.

Task 3

In the case of Socks5:

Pros –

Faster than VPNs because it does not encrypt traffic.

More flexible than other types of proxies, such as HTTP proxies.

Can be used with any type of traffic, including TCP, UDP, and ICMP.

Relatively easy to set up and use.

Cons –

Does not encrypt traffic, so it is less secure than VPNs.

May not be able to bypass firewalls or geo-restrictions.

May log user activity.

In case of VPN:

Pros-

Encrypts all traffic, making it more secure than SOCKS5 proxies.

Can bypass firewalls and geo-restrictions.

Can hide your IP address and location.

May offer additional features, such as kill switches and ad blockers.

Cons –

Slower than SOCKS5 proxies because it encrypts traffic.

More difficult to set up and use than SOCKS5 proxies.

May be more expensive than SOCKS5 proxies.