Lab 2: Detect Session Hijacking

Lab Scenario

Session hijacking is very dangerous; it places the victim at risk of identity theft, fraud, and loss of sensitive information. All networks that use TCP/IP are vulnerable to different types of hijacking attacks. Moreover, these kinds of attacks are very difficult to detect, and often go unnoticed unless the attacker causes severe damage. However, following best practices can protect against session hijacking attacks.

As a professional ethical hacker or penetration tester, it is very important that you have the required knowledge to detect session hijacking attacks and protect your organization's system against them. Fortunately, there are various tools available that can help you to detect session hijacking attacks such as packet sniffers, IDSs, and SIEMs.

Lab Objectives

- Detect session hijacking using Wireshark

Overview of Detecting Session Hijacking

There are two primary methods that can be used to detect session hijacking:

- **Manual Method**: Involves using packet sniffing software such as Wireshark and SteelCentral Packet Analyzer to monitor session hijacking attacks; the packet sniffer captures packets being transferred across the network, which are then analyzed using various filtering tools
- **Automatic Method**: Involves using Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) to monitor incoming network traffic; if a packet matches any of the attack signatures in the internal database, the IDS generates an alert, and the IPS blocks the traffic from entering the database

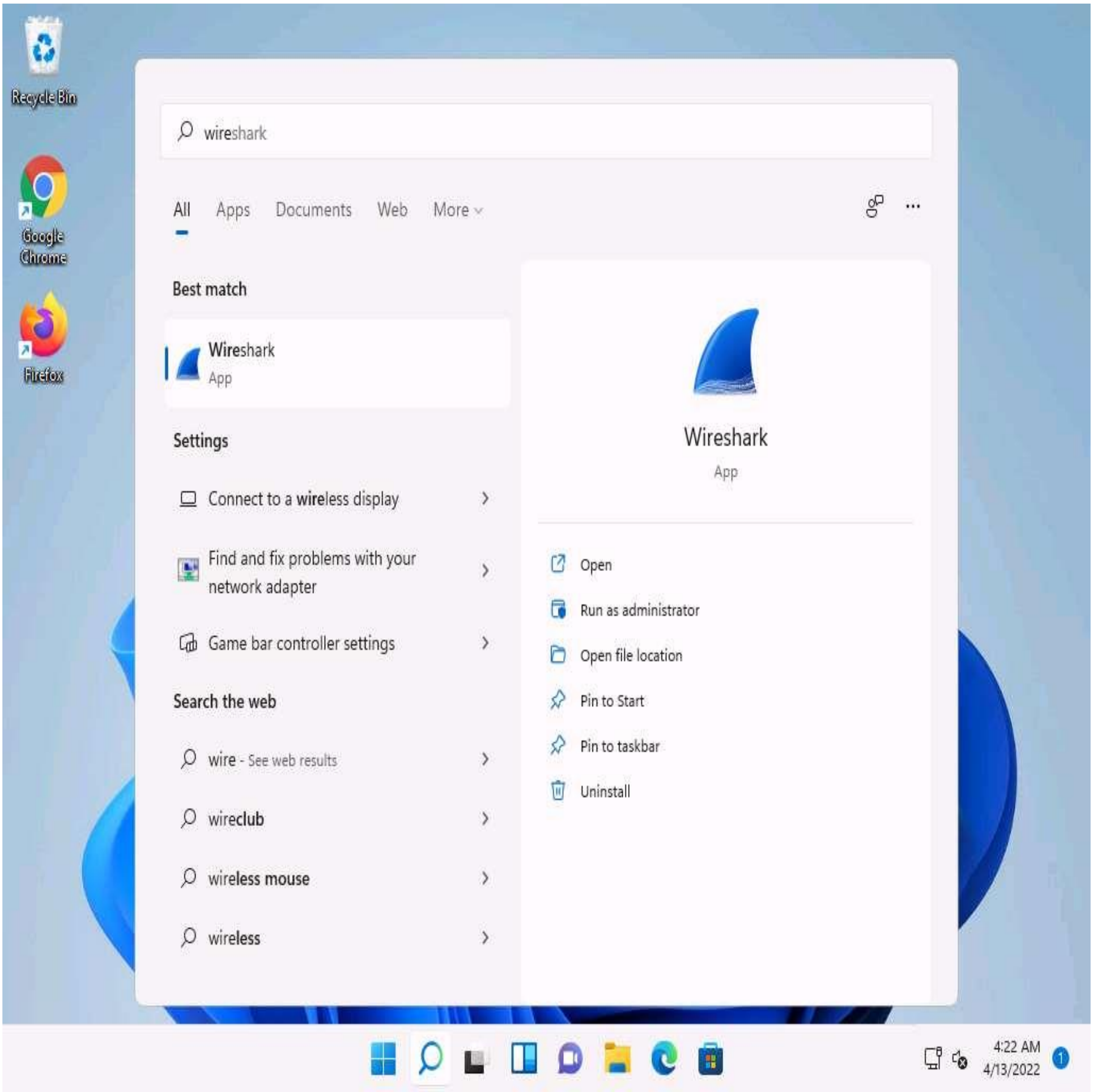
Task 1: Detect Session Hijacking using Wireshark

Wireshark allows you to capture and interactively browse the traffic running on a network. The tool uses WinPcap to capture packets, and so is only able to capture packets on networks that are supported by WinPcap. It captures live network traffic from Ethernet, IEEE 802.11, PPP/HDLC, ATM, Bluetooth, USB, Token Ring, Frame Relay, and FDDI networks. Security professionals can use Wireshark to monitor and detect session hijacking attempts.

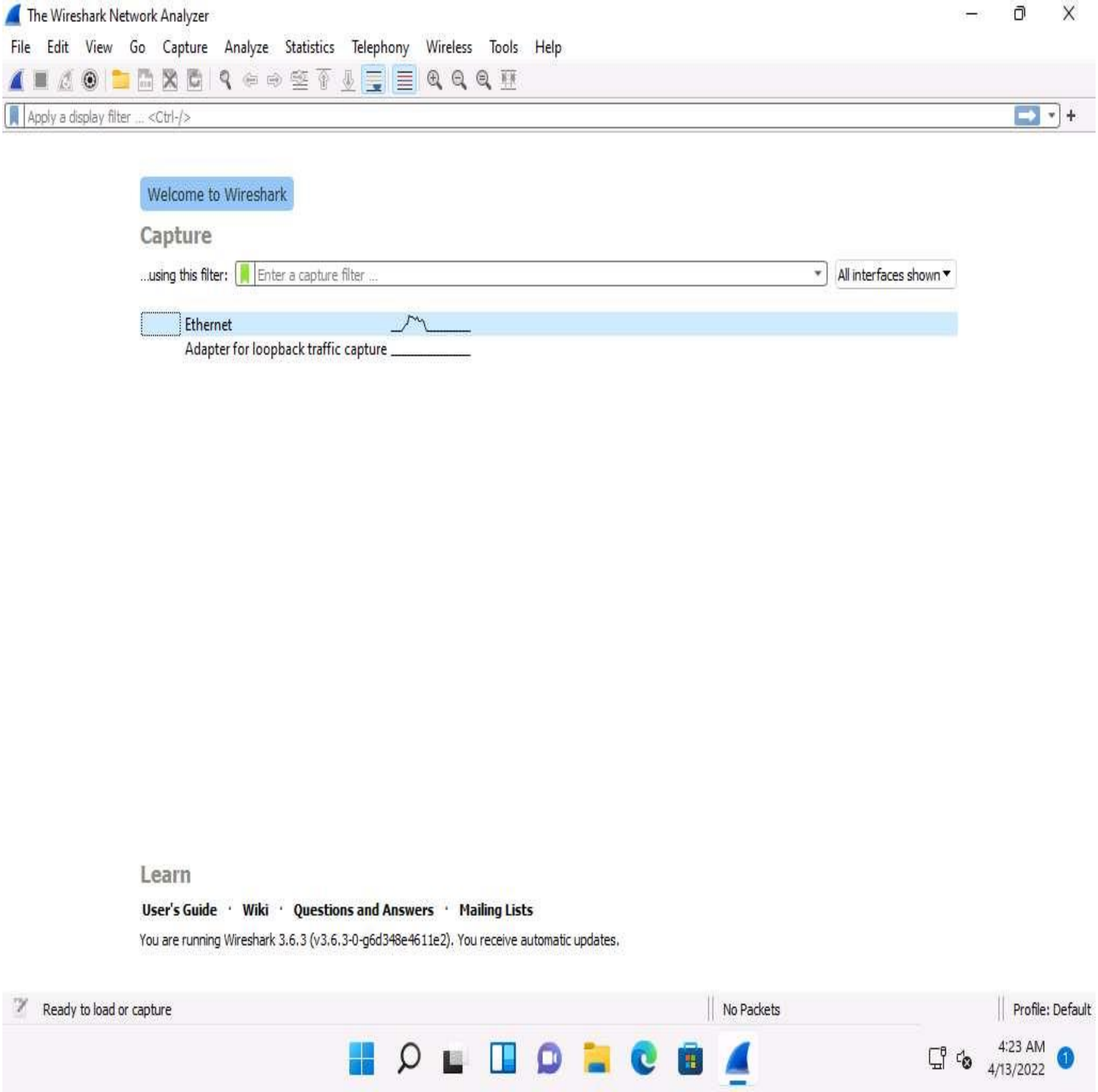
Here, we will use the Wireshark tool to detect session hijacking attacks manually on the target system.

We will use the **Parrot Security** (**10.10.1.13**) machine to carry out a session hijacking attack on the **Windows 11** (**10.10.1.11**) machine.

1. □ Click Windows 11 to switch to the **Windows 11** machine.
2. □ Click **Search** icon (🔍) on the **Desktop**. Type **wire** in the search field, the **Wireshark** appears in the result, click **Open** to launch it.



3. □ **The Wireshark Network Analyzer** window opens. Double-click the primary network interface (in this case, **Ethernet**) to start capturing network traffic.

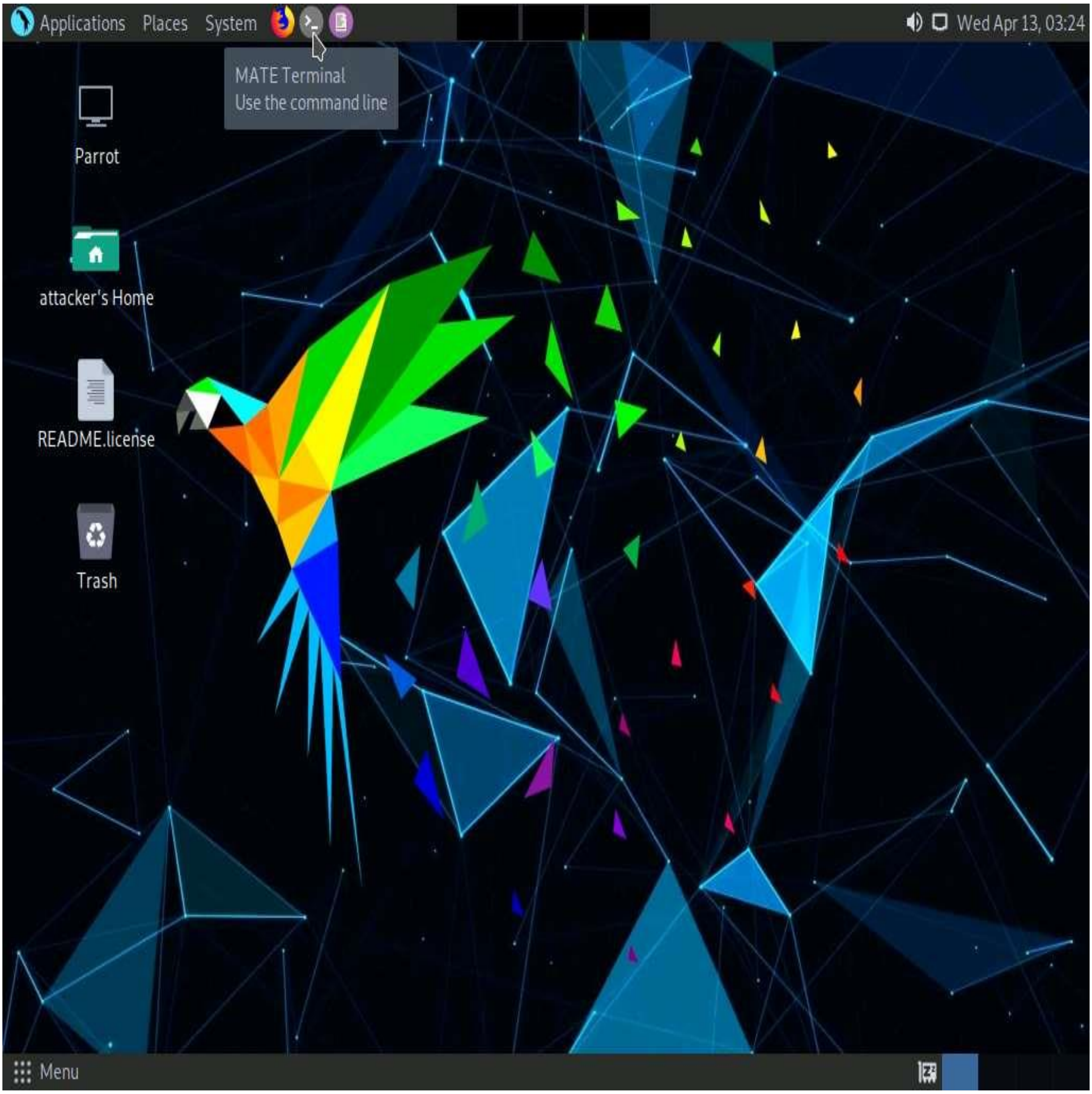

The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Welcome to Wireshark
Capture
...using this filter:
Enter a capture filter ...
All interfaces shown
Ethernet
Adapter for loopback traffic capture
Learn
User's Guide · Wiki · Questions and Answers · Mailing Lists
Ready to load or capture
No Packets
Profile: Default
4:23 AM
4/13/2022

You are running Wireshark 3.6.3 (v3.6.3-0-g6d348e4611e2). You receive automatic updates.

4. □ **Wireshark** starts capturing network traffic. Leave it running.
5. □ Now, we shall launch a session hijacking attack on the target machine (**Windows 11**) using **bettercap**.

To do so, you may either follow Steps **8-15** below, or refer to Task 2 (Intercept HTTP Traffic using bettercap) in Lab 1.

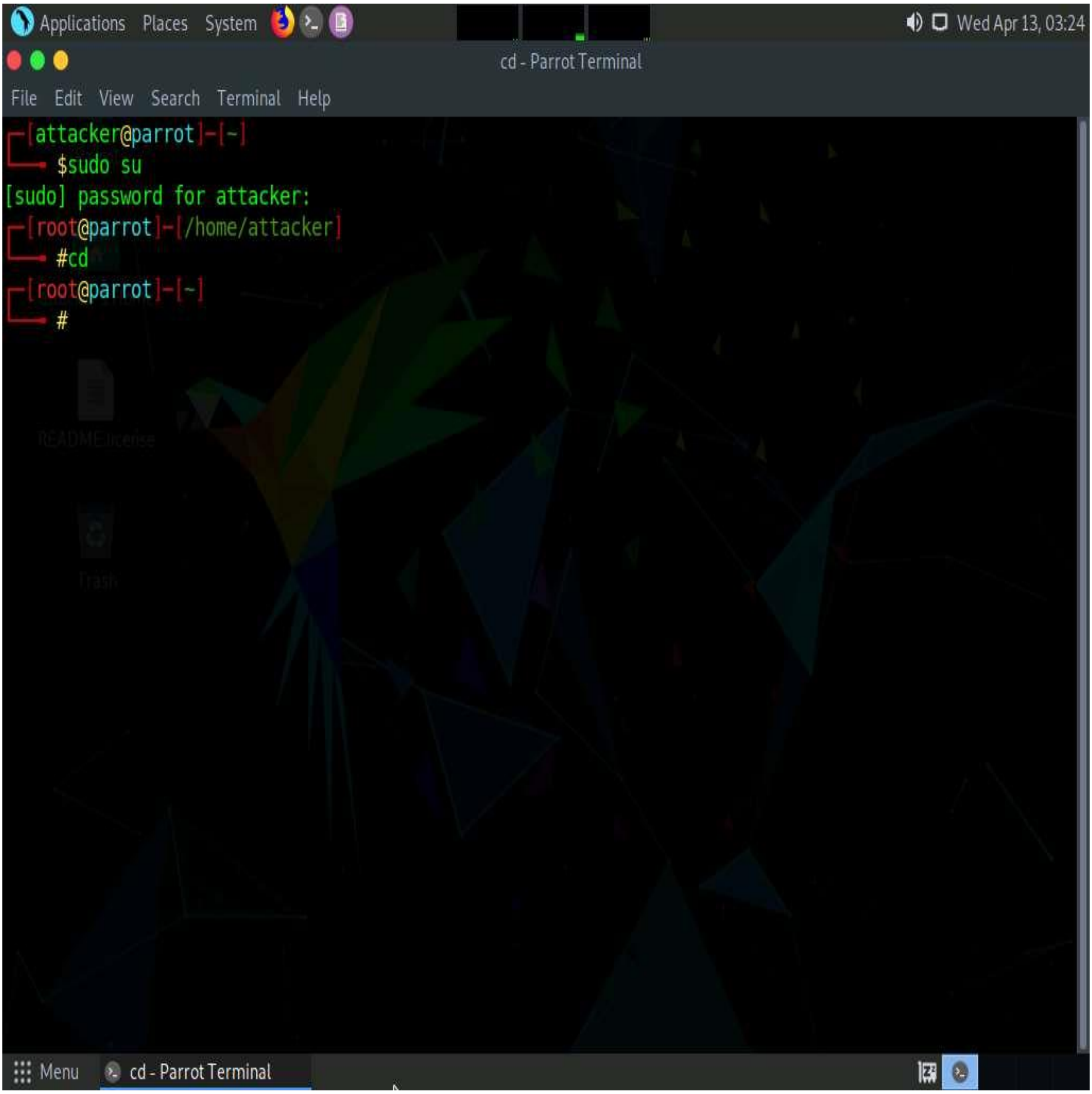
6. □ Click Parrot Security to switch to the **Parrot Security** machine.
7. □ Click the **MATE Terminal** icon at the top of the **Desktop** window to open a **Terminal** window.



8. □ A **Parrot Terminal** window appears. In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user.
9. □ In the **[sudo] password for attacker** field, type **toor** as a password and press **Enter**.

   The password that you type will not be visible.

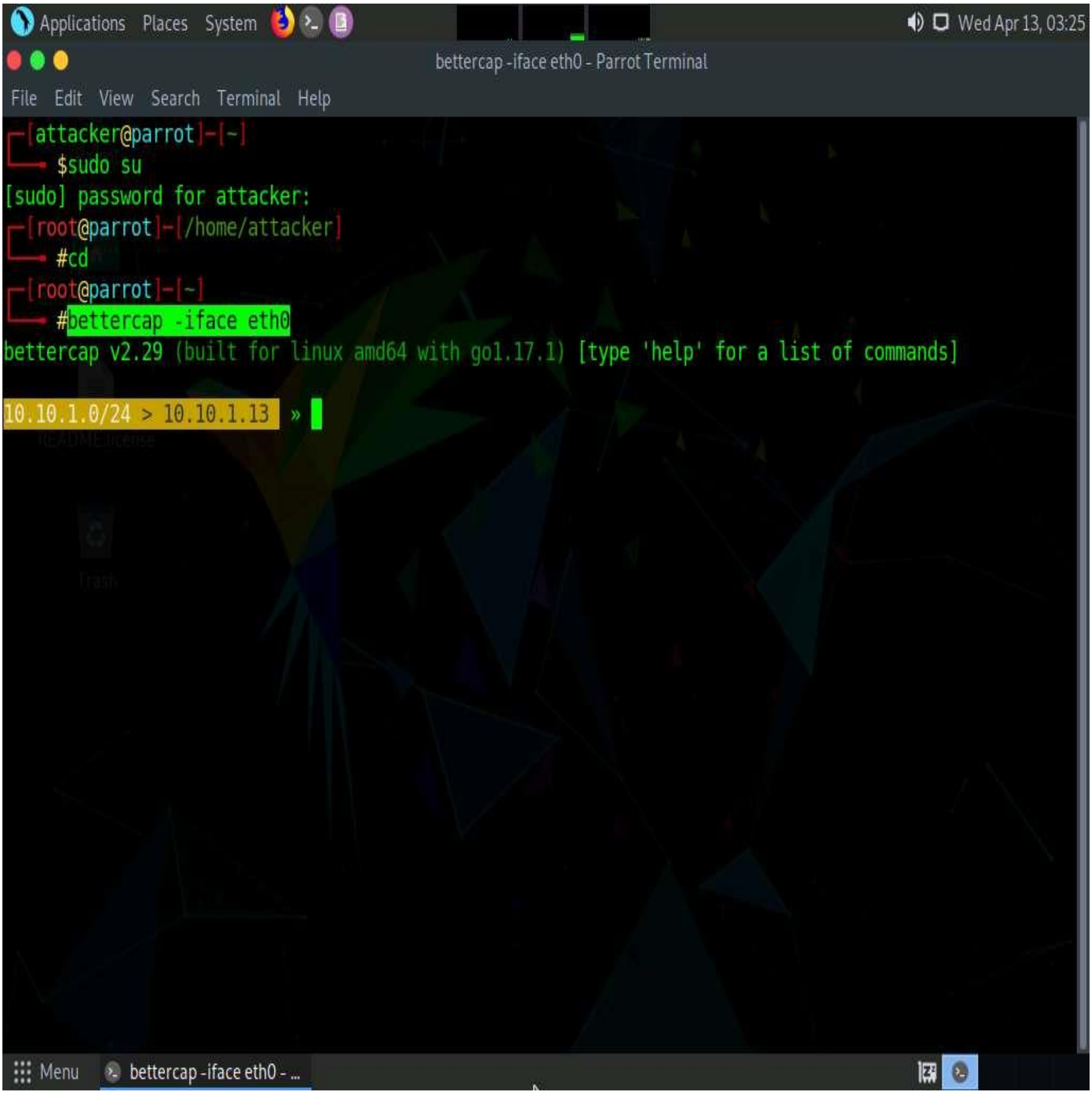
10. □ Now, type **cd** and press **Enter** to jump to the root directory.



```
┌─[attacker@parrot]─[~]
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #cd
┌─[root@parrot]─[~]
└──╼ #
```

11. □ In the terminal window, type **bettercap -iface eth0** and press **Enter** to set the network interface.

-iface: specifies the interface to bind to (here, **eth0**).



Applications Places System
Wed Apr 13, 03:25
bettercap -iface eth0 - Parrot Terminal
File Edit View Search Terminal Help
Menu
bettercap -iface eth0 - ...

```
┌─[attacker@parrot]─[~]
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #cd
┌─[root@parrot]─[~]
└──╼ #bettercap -iface eth0
bettercap v2.29 (built for linux amd64 with go1.17.1) [type 'help' for a list of commands]

10.10.1.0/24 > 10.10.1.13  »
```

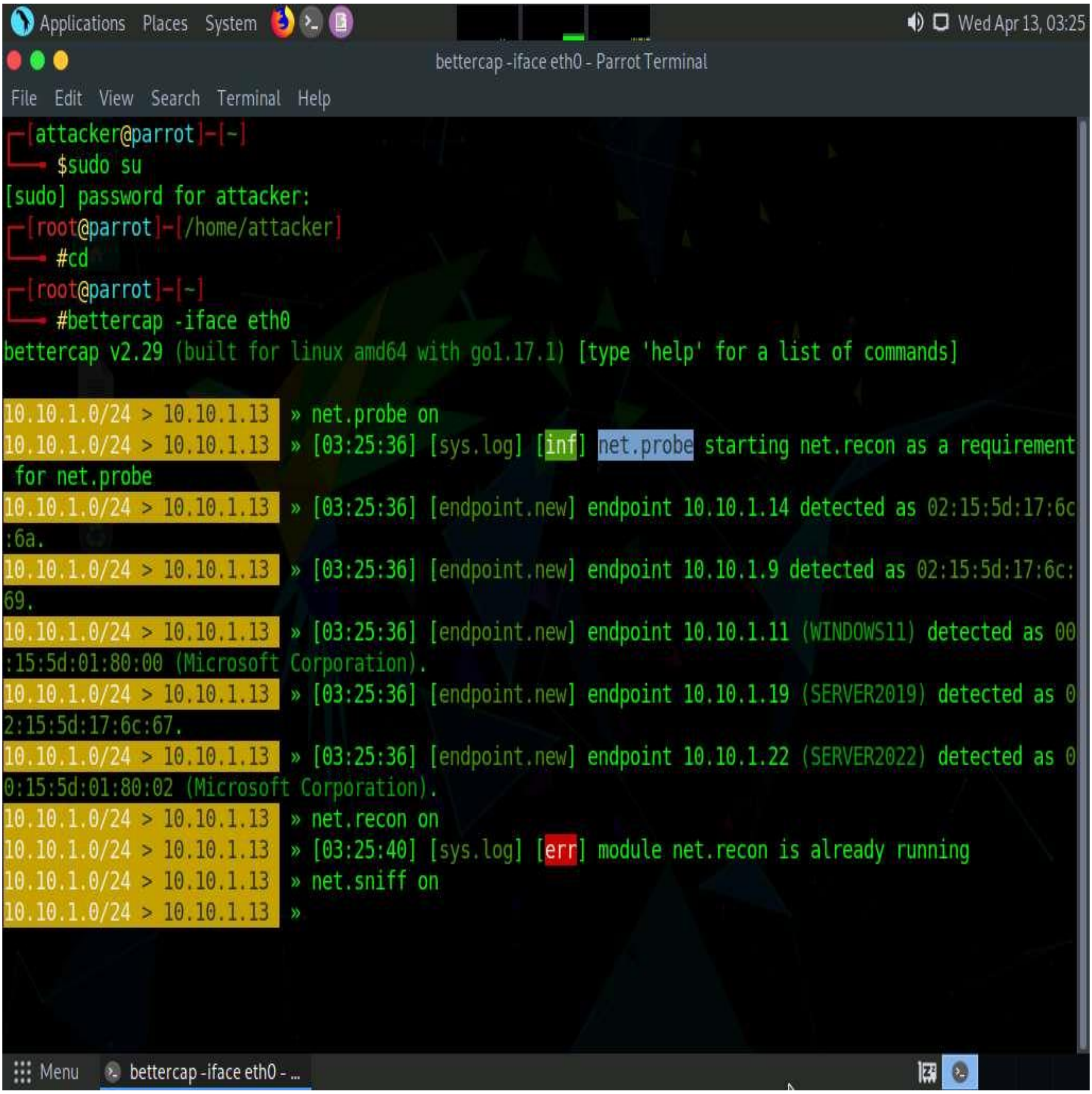
12. □ Type **net.probe on** and press **Enter**. This module will send different types of probe packets to each IP in the current subnet for the **net.recon** module to detect them.

13. □ Type **net.recon on** and press **Enter**. This module is responsible for periodically reading the system ARP table to detect new hosts on the network.

The net.recon module displays the detected active IP addresses in the network. In real-time, this module will start sniffing network packets.

14. □ Type **net.sniff on** and press **Enter**. This module is responsible for performing sniffing on the network.

15. □ You can observe that bettercap starts sniffing network traffic on different machines in the network, as shown in the screenshot.

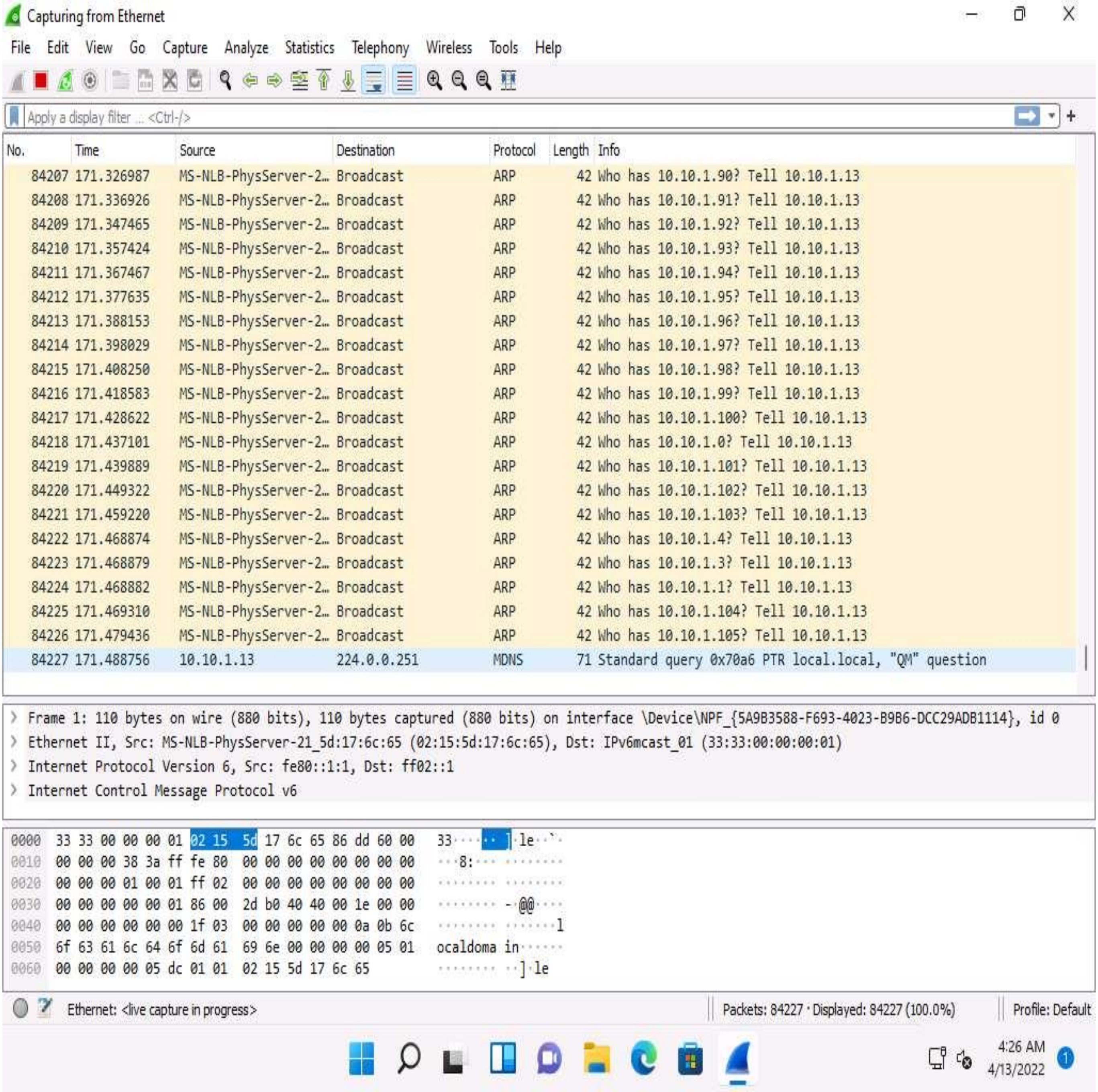

Applications Places System
Wed Apr 13, 03:25
bettercap -iface eth0 - Parrot Terminal
File Edit View Search Terminal Help
Menu
bettercap -iface eth0 - ...

```
┌─[attacker@parrot]─[~]
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #cd
┌─[root@parrot]─[~]
└──╼ #bettercap -iface eth0
bettercap v2.29 (built for linux amd64 with go1.17.1) [type 'help' for a list of commands]

10.10.1.0/24 > 10.10.1.13  » net.probe on
10.10.1.0/24 > 10.10.1.13  » [03:25:36] [sys.log] [inf] net.probe starting net.recon as a requirement
 for net.probe
10.10.1.0/24 > 10.10.1.13  » [03:25:36] [endpoint.new] endpoint 10.10.1.14 detected as 02:15:5d:17:6c
:6a.
10.10.1.0/24 > 10.10.1.13  » [03:25:36] [endpoint.new] endpoint 10.10.1.9 detected as 02:15:5d:17:6c:
69.
10.10.1.0/24 > 10.10.1.13  » [03:25:36] [endpoint.new] endpoint 10.10.1.11 (WINDOWS11) detected as 00
:15:5d:01:80:00 (Microsoft Corporation).
10.10.1.0/24 > 10.10.1.13  » [03:25:36] [endpoint.new] endpoint 10.10.1.19 (SERVER2019) detected as 0
2:15:5d:17:6c:67.
10.10.1.0/24 > 10.10.1.13  » [03:25:36] [endpoint.new] endpoint 10.10.1.22 (SERVER2022) detected as 0
0:15:5d:01:80:02 (Microsoft Corporation).
10.10.1.0/24 > 10.10.1.13  » net.recon on
10.10.1.0/24 > 10.10.1.13  » [03:25:40] [sys.log] [err] module net.recon is already running
10.10.1.0/24 > 10.10.1.13  » net.sniff on
10.10.1.0/24 > 10.10.1.13  »
```

16. □ Click Windows 11 to switch back to the **Windows 11** machine and observe the huge number of **ARP packets** captured by the **Wireshark**, as shown in the screenshot.

> bettercap sends several ARP broadcast requests to the hosts (or potentially active hosts). A high number of ARP requests indicates that the system at **10.10.1.13** (the attacker's system in this task) is acting as a client for all the IP addresses in the subnet, which means that all the packets from the victim node (in this case, **10.10.1.11**) will first go to the host system (**10.10.1.13**), and then the gateway. Similarly, any packet destined for the victim node is first forwarded from the gateway to the host system, and then from the host system to the victim node.
>
> more...


Capturing from Ethernet
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
No.   Time        Source               Destination    Protocol  Length Info
84207 171.326987  MS-NLB-PhysServer-2… Broadcast      ARP       42 Who has 10.10.1.90? Tell 10.10.1.13
84208 171.336926  MS-NLB-PhysServer-2… Broadcast      ARP       42 Who has 10.10.1.91? Tell 10.10.1.13
84209 171.347465  MS-NLB-PhysServer-2… Broadcast      ARP       42 Who has 10.10.1.92? Tell 10.10.1.13
84210 171.357424  MS-NLB-PhysServer-2… Broadcast      ARP       42 Who has 10.10.1.93? Tell 10.10.1.13
84211 171.367467  MS-NLB-PhysServer-2… Broadcast      ARP       42 Who has 10.10.1.94? Tell 10.10.1.13
84212 171.377635  MS-NLB-PhysServer-2… Broadcast      ARP       42 Who has 10.10.1.95? Tell 10.10.1.13
84213 171.388153  MS-NLB-PhysServer-2… Broadcast      ARP       42 Who has 10.10.1.96? Tell 10.10.1.13
84214 171.398029  MS-NLB-PhysServer-2… Broadcast      ARP       42 Who has 10.10.1.97? Tell 10.10.1.13
84215 171.408250  MS-NLB-PhysServer-2… Broadcast      ARP       42 Who has 10.10.1.98? Tell 10.10.1.13
84216 171.418583  MS-NLB-PhysServer-2… Broadcast      ARP       42 Who has 10.10.1.99? Tell 10.10.1.13
84217 171.428622  MS-NLB-PhysServer-2… Broadcast      ARP       42 Who has 10.10.1.100? Tell 10.10.1.13
84218 171.437101  MS-NLB-PhysServer-2… Broadcast      ARP       42 Who has 10.10.1.0? Tell 10.10.1.13
84219 171.439889  MS-NLB-PhysServer-2… Broadcast      ARP       42 Who has 10.10.1.101? Tell 10.10.1.13
84220 171.449322  MS-NLB-PhysServer-2… Broadcast      ARP       42 Who has 10.10.1.102? Tell 10.10.1.13
84221 171.459220  MS-NLB-PhysServer-2… Broadcast      ARP       42 Who has 10.10.1.103? Tell 10.10.1.13
84222 171.468874  MS-NLB-PhysServer-2… Broadcast      ARP       42 Who has 10.10.1.4? Tell 10.10.1.13
84223 171.468879  MS-NLB-PhysServer-2… Broadcast      ARP       42 Who has 10.10.1.3? Tell 10.10.1.13
84224 171.468882  MS-NLB-PhysServer-2… Broadcast      ARP       42 Who has 10.10.1.1? Tell 10.10.1.13
84225 171.469310  MS-NLB-PhysServer-2… Broadcast      ARP       42 Who has 10.10.1.104? Tell 10.10.1.13
84226 171.479436  MS-NLB-PhysServer-2… Broadcast      ARP       42 Who has 10.10.1.105? Tell 10.10.1.13
84227 171.488756  10.10.1.13           224.0.0.251    MDNS      71 Standard query 0x70a6 PTR local.local, "QM" question
```

```
> Frame 1: 110 bytes on wire (880 bits), 110 bytes captured (880 bits) on interface \Device\NPF_{5A9B3588-F693-4023-B9B6-DCC29ADB1114}, id 0
> Ethernet II, Src: MS-NLB-PhysServer-21_5d:17:6c:65 (02:15:5d:17:6c:65), Dst: IPv6mcast_01 (33:33:00:00:00:01)
> Internet Protocol Version 6, Src: fe80::1:1, Dst: ff02::1
> Internet Control Message Protocol v6
```

```
0000  33 33 00 00 00 01 02 15  5d 17 6c 65 86 dd 60 00   33······ ]·le··`·
0010  00 00 00 38 3a ff fe 80  00 00 00 00 00 00 00 00   ···8:··· ········
0020  00 00 00 01 00 01 ff 02  00 00 00 00 00 00 00 00   ········ ········
0030  00 00 00 00 00 01 86 00  2d b0 40 40 00 1e 00 00   ········ -·@@····
0040  00 00 00 00 00 00 1f 03  00 00 00 00 00 0a 0b 6c   ········ ·······l
0050  6f 63 61 6c 64 6f 6d 61  69 6e 00 00 00 00 05 01   ocaldoma in······
0060  00 00 00 00 05 dc 01 01  02 15 5d 17 6c 65         ········ ··]·le
```

Ethernet: <live capture in progress>
Packets: 84227 · Displayed: 84227 (100.0%)
Profile: Default
4:26 AM
4/13/2022

17. □ This concludes the demonstration of how to detect a session hijacking attack using Wireshark.

18. □ Close all open windows and document all the acquired information.