

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 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.

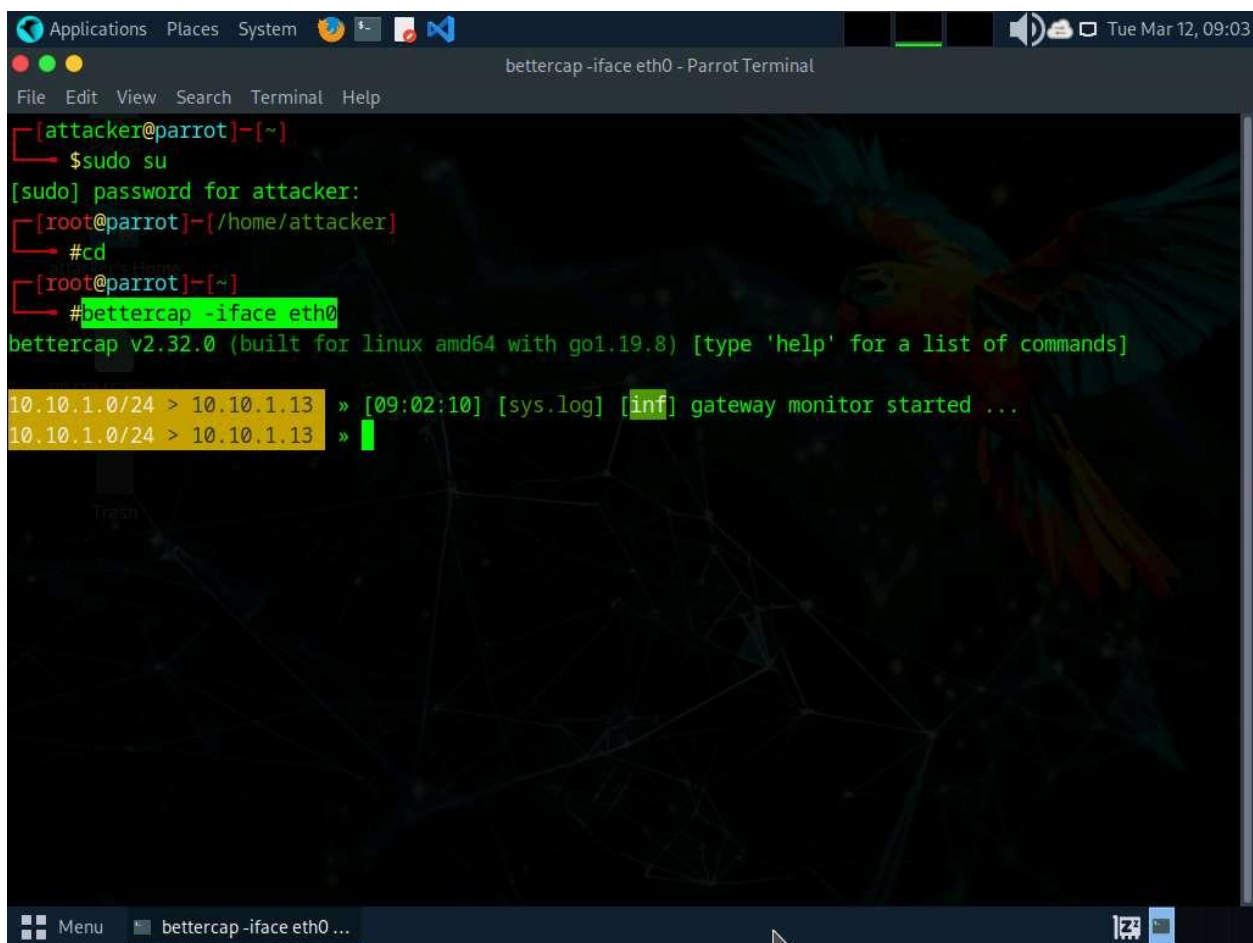
1. Click [Windows 11](#) to switch to the **Windows 11** machine.
2. Click the windows **Search** icon on the **Desktop**, search for **Wireshark** in the search bar and launch it.
3. **The Wireshark Network Analyzer** window appears, start capturing the network traffic on the primary network interface (here, **Ethernet**).

- Now, we shall launch a session hijacking attack on the target machine (**Windows 11**) using **bettercap**.

To do so, you may either follow Steps **8-11** below, or refer to Task 2 (Intercept HTTP Traffic using bettercap) in Lab 1.

- Click [Parrot Security](#) to switch to the **Parrot Security** machine.
- Open a **Terminal** window and execute **sudo su** to run the programs as a root user (When prompted, enter the password **toor**). Run **cd** to jump to the root directory.
- Run **bettercap -iface eth0** to set the network interface.

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



```
Applications  Places  System  bettercap -iface eth0 - Parrot Terminal
File Edit View Search Terminal Help

[attacker@parrot]~$ sudo su
[sudo] password for attacker:
[root@parrot]~/home/attacker$ #cd
[root@parrot]~$ #bettercap -iface eth0
bettercap v2.32.0 (built for linux amd64 with go1.19.8) [type 'help' for a list of commands]

10.10.1.0/24 > 10.10.1.13 » [09:02:10] [sys.log] [inf] gateway monitor started ...
10.10.1.0/24 > 10.10.1.13 »
```

- 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.
- 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.

10. Type **net.sniff on** and press **Enter**. This module is responsible for performing sniffing on the network.
11. You can observe that bettercap starts sniffing network traffic on different machines in the network, as shown in the screenshot.

The screenshot shows a terminal window in Parrot OS with the title "bettercap -iface eth0 - Parrot Terminal". The user is logged in as root at the parrot machine. The terminal shows the following commands and output:

```
[root@parrot]~/home/attacker
#cd
[root@parrot]~
#bettercap -iface eth0
bettercap v2.32.0 (built for linux amd64 with go1.19.8) [type 'help' for a list of commands]

10.10.1.0/24 > 10.10.1.13 » [09:02:10] [sys.log] [inf] gateway monitor started ...
10.10.1.0/24 > 10.10.1.13 » net.probe on
[09:04:08] [sys.log] [inf] net.probe starting net.recon as a requirement for net.probe
10.10.1.0/24 > 10.10.1.13 » [09:04:08] [sys.log] [inf] net.probe probing 256 addresses on 10.10.1.0/24
10.10.1.0/24 > 10.10.1.13 » [09:04:08] [endpoint.new] endpoint 10.10.1.9 detected as 02:15:5d:45:41:2f.
10.10.1.0/24 > 10.10.1.13 » [09:04:08] [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 » [09:04:09] [endpoint.new] endpoint 10.10.1.19 (SERVER2019) detected as 02:15:5d:45:41:2e.
10.10.1.0/24 > 10.10.1.13 » [09:04:09] [endpoint.new] endpoint 10.10.1.22 (SERVER2022) detected as 00:15:5d: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 » [09:04:21] [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 » [09:04:46] [net.sniff.mdns] mdns fe80::15:5dff:fe45:4130 : Android.local
is fe80::15:5dff:fe45:4130
10.10.1.0/24 > 10.10.1.13 »
```

- 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
3360	443.659266	MS-NLB-PhysServer-2	Broadcast	ARP	42	Who has 10.10.1.172? Tell 10.10.1.13
3360	443.662624	MS-NLB-PhysServer-2	Broadcast	ARP	42	Who has 10.10.1.74? Tell 10.10.1.13
3360	443.662659	MS-NLB-PhysServer-2	Broadcast	ARP	42	Who has 10.10.1.76? Tell 10.10.1.13
3360	443.665750	MS-NLB-PhysServer-2	Broadcast	ARP	42	Who has 10.10.1.75? Tell 10.10.1.13
3360	443.669357	MS-NLB-PhysServer-2	Broadcast	ARP	42	Who has 10.10.1.173? Tell 10.10.1.13
3360	443.679878	MS-NLB-PhysServer-2	Broadcast	ARP	42	Who has 10.10.1.174? Tell 10.10.1.13
3360	443.689116	MS-NLB-PhysServer-2	Broadcast	ARP	42	Who has 10.10.1.79? Tell 10.10.1.13
3360	443.689136	MS-NLB-PhysServer-2	Broadcast	ARP	42	Who has 10.10.1.77? Tell 10.10.1.13
3360	443.689157	MS-NLB-PhysServer-2	Broadcast	ARP	42	Who has 10.10.1.78? Tell 10.10.1.13
3360	443.690431	MS-NLB-PhysServer-2	Broadcast	ARP	42	Who has 10.10.1.175? Tell 10.10.1.13
3360	443.701047	MS-NLB-PhysServer-2	Broadcast	ARP	42	Who has 10.10.1.176? Tell 10.10.1.13
3360	443.711444	MS-NLB-PhysServer-2	Broadcast	ARP	42	Who has 10.10.1.177? Tell 10.10.1.13
3360	443.715597	MS-NLB-PhysServer-2	Broadcast	ARP	42	Who has 10.10.1.80? Tell 10.10.1.13
3360	443.715607	MS-NLB-PhysServer-2	Broadcast	ARP	42	Who has 10.10.1.81? Tell 10.10.1.13
3360	443.721650	MS-NLB-PhysServer-2	Broadcast	ARP	42	Who has 10.10.1.178? Tell 10.10.1.13
3360	443.732229	MS-NLB-PhysServer-2	Broadcast	ARP	42	Who has 10.10.1.179? Tell 10.10.1.13
3360	443.742351	MS-NLB-PhysServer-2	Broadcast	ARP	42	Who has 10.10.1.82? Tell 10.10.1.13
3360	443.742376	MS-NLB-PhysServer-2	Broadcast	ARP	42	Who has 10.10.1.83? Tell 10.10.1.13
3360	443.742487	MS-NLB-PhysServer-2	Broadcast	ARP	42	Who has 10.10.1.84? Tell 10.10.1.13
3360	443.742915	MS-NLB-PhysServer-2	Broadcast	ARP	42	Who has 10.10.1.180? Tell 10.10.1.13
3360	443.753101	MS-NLB-PhysServer-2	Broadcast	ARP	42	Who has 10.10.1.181? Tell 10.10.1.13
3360	443.763652	MS-NLB-PhysServer-2	Broadcast	ARP	42	Who has 10.10.1.182? Tell 10.10.1.13
3360	443.769080	MS-NLB-PhysServer-2	Broadcast	ARP	42	Who has 10.10.1.86? Tell 10.10.1.13
3360	443.772378	MS-NLB-PhysServer-2	Broadcast	ARP	42	Who has 10.10.1.85? Tell 10.10.1.13
3360	443.774189	MS-NLB-PhysServer-2	Broadcast	ARP	42	Who has 10.10.1.183? Tell 10.10.1.13
3360	443.784214	MS-NLB-PhysServer-2	Broadcast	ARP	42	Who has 10.10.1.184? Tell 10.10.1.13
3360	443.794809	MS-NLB-PhysServer-2	Broadcast	ARP	42	Who has 10.10.1.185? Tell 10.10.1.13

> Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
 > Ethernet II, Src: Microsoft_01:80:02 (00:15:5d:01:80:02), Dst: Broadcast
 > Address Resolution Protocol (request)

```

0000  ff ff ff ff ff 00 15 5d 01 80 02 08 06 00 01  ....
0010  08 00 06 04 00 01 00 15 5d 01 80 02 0a 0a 01 16  ....
0020  00 00 00 00 00 00 a9 fe a9 fe  ....
  
```

Ethernet: <live capture in progress> | Packets: 355924 - Displayed: 355924 (100.0%) | Profile: Default

6:05 AM 3/12/2024

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

14. Close all open windows and document all the acquired information.

Question 11.2.1.1

Use the bettercap tool (available in the Parrot Security machine) to sniff the traffic on the target system (10.10.1.11). Use the Wireshark tool on the target system (10.10.1.11) to detect the session hijacking attempt. Traffic on which protocol indicates the session hijacking attempt in Wireshark?