# Session Hijacking

## Module 11

# Session Hijacking

*Session hijacking is when an attacker takes over either a valid TCP communication session between two computers or a valid user session in a web application.*

ICON KEY

- Valuable information
- Test your knowledge
- Web exercise
- Workbook review

## Lab Scenario

A session hijacking attack refers to the exploitation of a session token-generation mechanism or token security controls that enables an attacker to establish an unauthorized connection with a target server. The attacker guesses or steals a valid session ID (which identifies authenticated users) and uses it to establish a session with the server.

As an ethical hacker or penetration tester, you should understand different session hijacking concepts, how attackers perform application- and network-level session hijacking, and the various tools used to launch this kind of attack. You should also be able to implement security measures at both the application and network levels to protect your network from session hijacking. Application-level hijacking involves gaining control over the Hypertext Transfer Protocol (HTTP) user session by obtaining the session IDs. Network-level hijacking is prevented by packet encryption, which can be achieved with protocols such as IPsec, SSL, and SSH.

## Lab Objectives

The objective of the lab is to perform session hijacking and other tasks that include, but are not limited to:

- Hijack a session by intercepting traffic between server and client
- Steal a user session ID by intercepting traffic
- Detect session hijacking attacks

**Tools demonstrated in this lab are available in E:\CEH-Tools\CEHv11 Module 11 Session Hijacking**

## Lab Environment

To carry out this lab, you need:

- Windows 10 virtual machine
- Windows Server 2019 virtual machine
- Parrot Security virtual machine
- Web browsers with an Internet connection
- Administrator privileges to run the tools

## Lab Duration

Time: 40 Minutes

# Overview of Session Hijacking

Session hijacking can be either active or passive, depending on the degree of involvement of the attacker:

- **Active session hijacking**: An attacker finds an active session and takes it over
- **Passive session hijacking**: An attacker hijacks a session, and, instead of taking over, monitors and records all the traffic in that session

# Lab Tasks

Ethical hackers or penetration testers use numerous tools and techniques to perform session hijacking on the target systems. Recommended labs that will assist you in learning various session hijacking techniques include:

| Lab No. | Lab Exercise Name | Core* | Self-study** | iLabs *** |
|---|---|---|---|---|
| 1 | Perform Session Hijacking | √ | √ | √ |
| | 1.1 Hijack a Session using Zed Attack Proxy (ZAP) | √ | | √ |
| | 1.2 Intercept HTTP Traffic using bettercap | | √ | √ |
| 2 | Detect Session Hijacking | √ | | √ |
| | 2.1 Detect Session Hijacking using Wireshark | √ | | √ |

**Remark**

EC-Council has prepared a considered amount of lab exercises for student to practice during the 5-day class and at their free time to enhance their knowledge and skill.

***Core** - Lab exercise(s) marked under Core are recommended by EC-Council to be practised during the 5-day class.

****Self-study** - Lab exercise(s) marked under self-study is for students to practise at their free time. Steps to access the additional lab exercises can be found in the first page of CEHv11 volume 1 book.

*****iLabs** - Lab exercise(s) marked under iLabs are available in our iLabs solution. iLabs is a cloud-based virtual lab environment preconfigured with vulnerabilities, exploits, tools and scripts, and can be accessed from anywhere with an Internet connection. If you are interested to learn more about our iLabs solution, please contact your training center or visit https://ilabs.eccouncil.org.

# Lab Analysis

Analyze and document the results related to the lab exercise. Give your opinion on your target's security posture and exposure.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.

Lab 1

# Perform Session Hijacking

*In a session hijacking attack, an attacker takes over (hijacks) a victim's valid user session in order to establish an unauthorized connection with a target server.*

ICON KEY

- Valuable information
- Test your knowledge
- Web exercise
- Workbook review

## Lab Scenario

Session hijacking allows an attacker to take over an active session by bypassing the authentication process. It involves stealing or guessing a victim's valid session ID, which the server uses to identify authenticated users, and using it to establish a connection with the server. The server responds to the attacker's requests as though it were communicating with an authenticated user, after which the attacker is able to perform any action on that system.

Attackers can use session hijacking to launch various kinds of attacks such as man-in-the-middle (MITM) and Denial-of-Service (DoS) attacks. A MITM attack occurs when an attacker places himself/herself between the authorized client and the server to intercept information flowing in either direction. A DoS attack happens when attackers sniff sensitive information and use it to make host or network resource unavailable to users, usually by flooding the target with requests until the system is overloaded.

As a professional ethical hacker or penetration tester, you must possess the required knowledge to hijack sessions in order to test the systems in the target network.

The labs in this exercise demonstrate how to hijack an active session between two endpoints.

**Tools demonstrated in this lab are available in E:\CEH-Tools\CEHv11 Module 11 Session Hijacking**

## Lab Objectives

- Hijack a session using Zed Attack Proxy (ZAP)
- Intercept HTTP traffic using bettercap

## Lab Environment

To carry out this lab, you need:

- Windows 10 virtual machine
- Windows Server 2019 virtual machine

- Parrot Security virtual machine
- Web browsers with an Internet connection
- Administrator privileges to run the tools
- OWASP ZAP located at **E:\CEH-Tools\CEHv11 Module 11 Session Hijacking\OWASP ZAP**
- You may also download the latest version of **OWASP ZAP** from the official website. If you do so, the screenshots shown in the lab might differ.

## Lab Duration

Time: 30 Minutes

## Overview of Session Hijacking

Session hijacking can be divided into three broad phases:

- **Tracking the Connection**: The attacker uses a network sniffer to track a victim and host, or uses a tool such as Nmap to scan the network for a target with a TCP sequence that is easy to predict
- **Desynchronizing the Connection**: A desynchronized state occurs when a connection between the target and host has been established, or is stable with no data transmission, or when the server's sequence number is not equal to the client's acknowledgment number (or vice versa)
- **Injecting the Attacker's Packet**: Once the attacker has interrupted the connection between the server and target, they can either inject data into the network or actively participate as the man-in-the-middle, passing data between the target and server, while reading and injecting data at will

## Lab Tasks

**TASK 1**

### Hijack a Session using Zed Attack Proxy (ZAP)

Here, we will hijack a session using ZAP. You will learn how to intercept the traffic of victims' machines with a proxy and how to view all the requests and responses from them.

**Note**: Before starting this task, we need to configure the proxy settings in the victim's machine, which in this lab will be the **Windows 10** virtual machine.

1. Turn on the **Windows 10** and **Windows Server 2019** virtual machines.

**TASK 1.1**

**Set Up a Proxy**

2. In the **Windows 10** virtual machine, log in with the credentials **Admin** and **Pa$$w0rd** and open any web browser (in this example, we are using **Google Chrome**).
3. In **Google Chrome**, click the **Customize and control Google Chrome** icon (  ), and select **Settings** from the context menu.

Zed Attack Proxy (ZAP) is an integrated penetration testing tool for finding vulnerabilities in web applications. It offers automated scanners as well as a set of tools that allow you to find security vulnerabilities manually. It is designed to be used by people with a wide range of security experience, and as such is ideal for developers and functional testers who are new to penetration testing

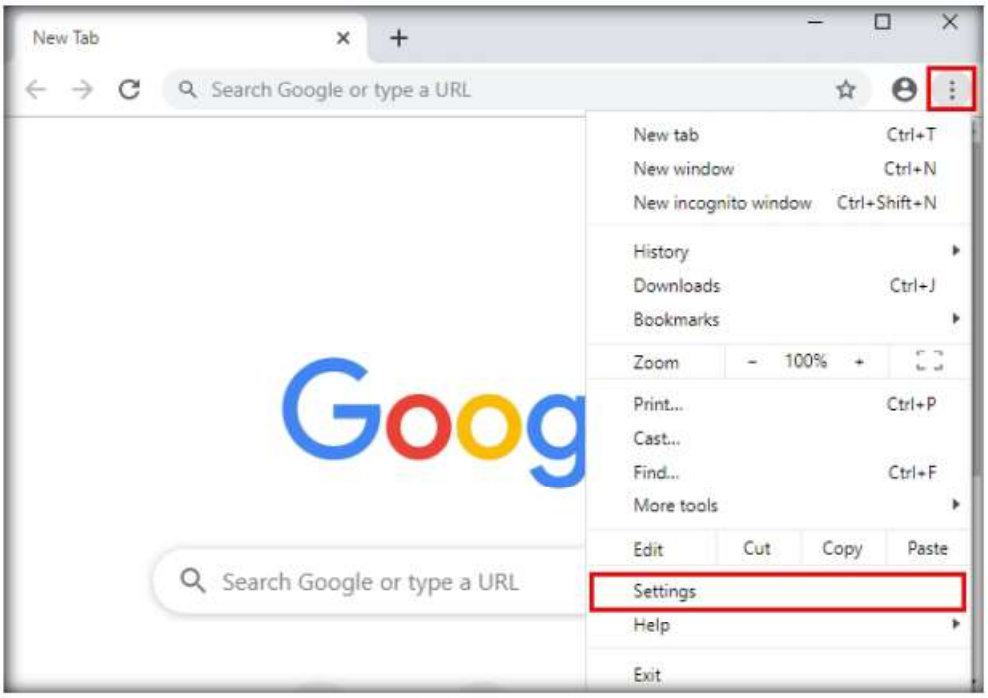


Figure 1.1.1: Google Chrome Settings

4. On the **Settings** page, scroll down and click the **Advanced** option in the browser.

ZAP allows you to see all the requests you make to a web app and all the responses you receive from it. Among other things, it allows you to see AJAX calls that may not otherwise be outright visible. You can also set breakpoints, which allow you to change the requests and responses in real-time.

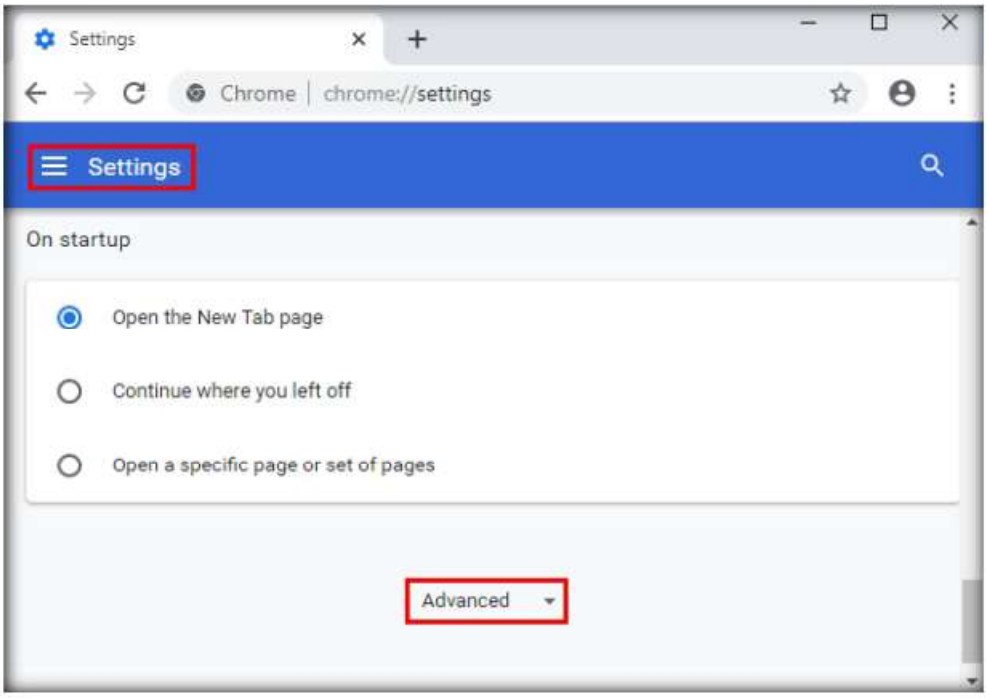


Figure 1.1.2: Google Chrome: Show Advanced settings

5. Scroll down to the **System** section and click **Open your computer's proxy settings** to configure a proxy.

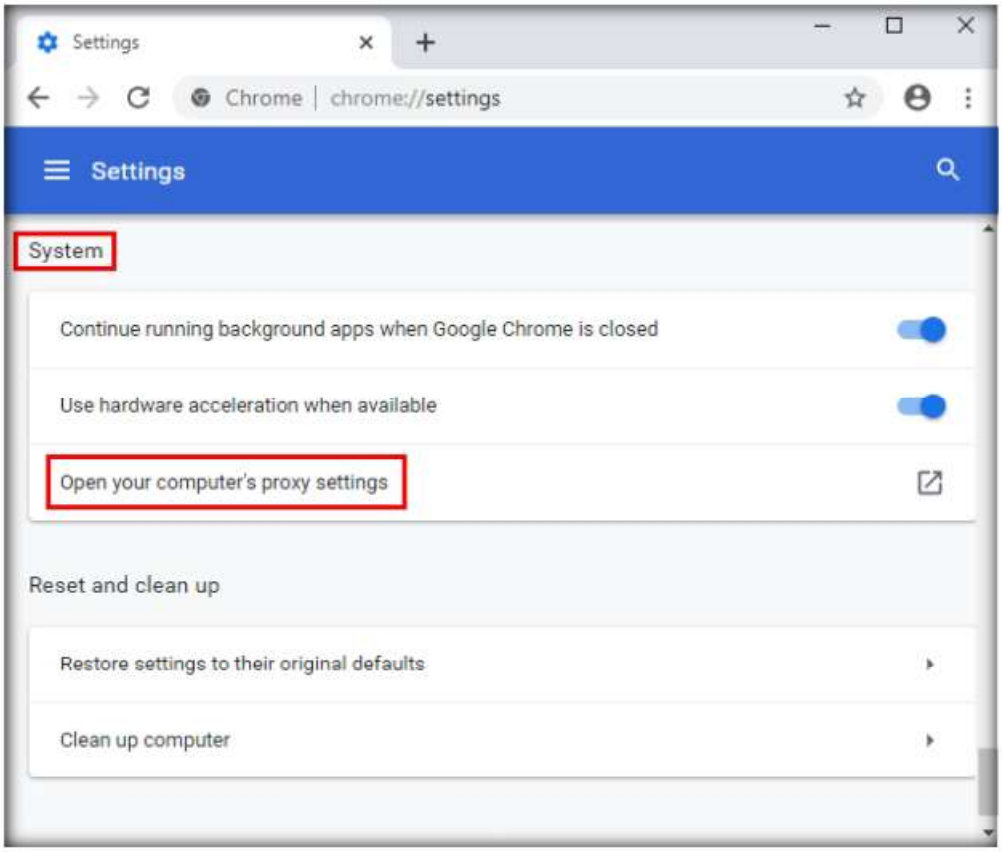


Figure 1.1.3: Google Chrome: Change proxy settings

6. A Windows 10 **Settings** window opens, with the **Proxy** settings in the right pane
7. Under the **Manual proxy setup** section, make the following changes:
   - Under the **Use a proxy server** option, click the **Off** button to switch it **On**.
   - In the **Address** field, type **10.10.10.19** (the IP address of the attacker's machine).
   - In the **Port** field, type **8080**.
   - Click **Save**.

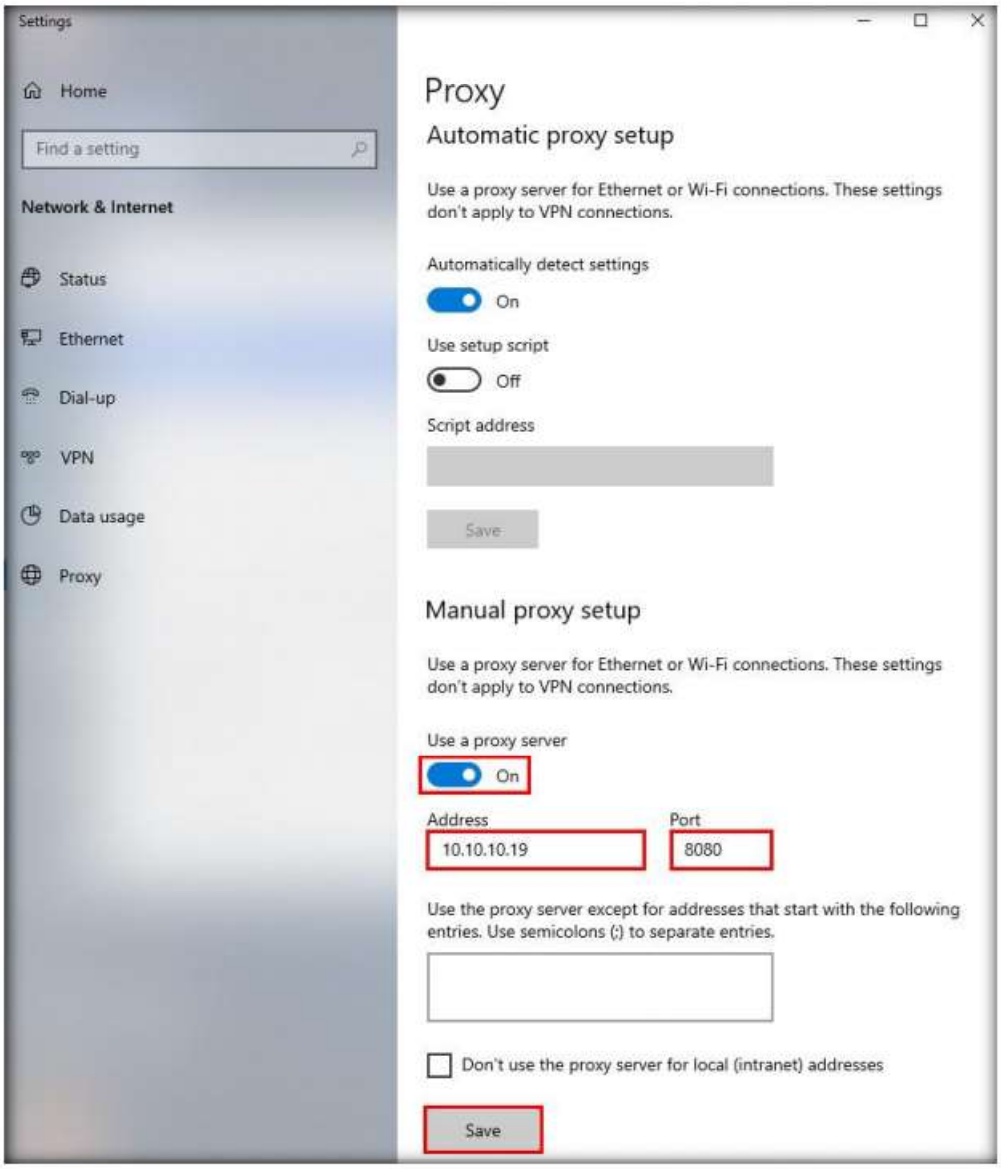


Figure 1.1.4: Settings window: Proxy setup

8. After saving, close the **Settings** and browser windows. You have now configured the proxy settings of the victim's machine.
9. Switch to the **Windows Server 2019** virtual machine, which in this lab will be the attacker's machine; log in with the credentials **Administrator** and **Pa$$w0rd**.

**TASK 1.2**

**Install & Configure OWASP ZAP**

10. To install **OWASP ZAP**, navigate to **Z:\CEHv11 Module 11 Session Hijacking\OWASP ZAP**, double-click **ZAP_2_8_0_windows.exe**, and follow the installation steps.

11. The **Setup - OWASP Zed Attack Proxy** window appears; click **Next**.

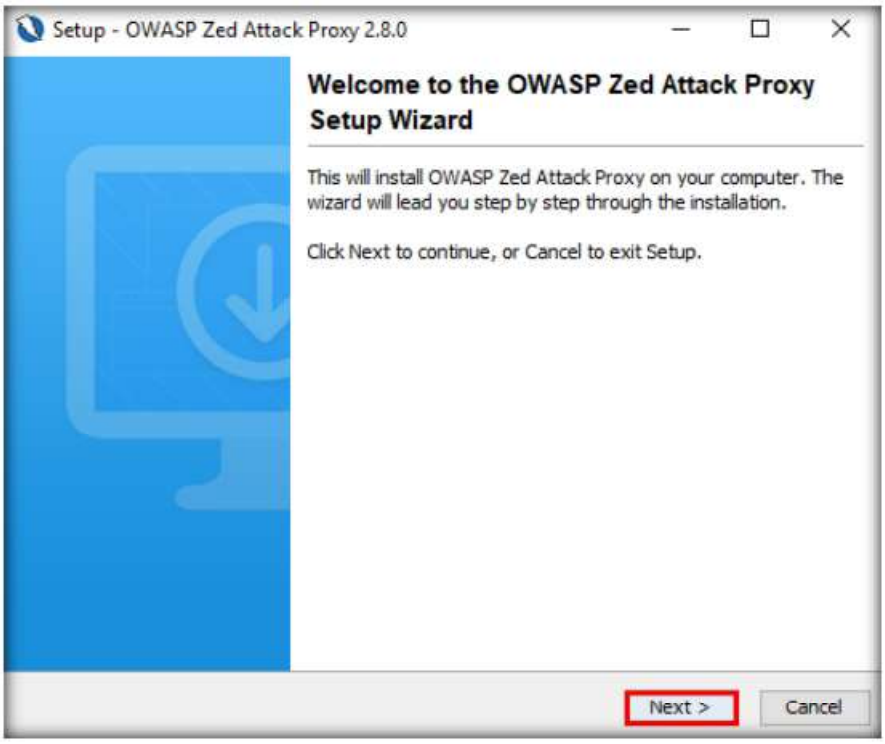


Figure 1.1.5: Setup - OWASP Zed Attack Proxy

12. In the **Select Installation Type** wizard, ensure that the **Standard installation** radio button is selected and click **Next**.
13. Follow the steps to install **OWASP ZAP** using the default settings.
14. After the installation completes, the **Completing the OWASP Zed Attack Proxy Setup Wizard** appears; click **Finish**.
15. Double-click the **OWASP ZAP** shortcut on **Desktop** to launch the application.
16. A prompt that reads **Do you want to persist the ZAP Session?** appears. Select the **No, I do not want to persist this session at this moment in time** radio button and click **Start**.


Figure 1.1.6: OWASP ZAP: Do you want to persist the ZAP Session?

17. The **OWASP ZAP** main window appears. Click on the "**+**" icon in the right pane and select **Break** from the options.

**Note**: The **Break** tab allows you to modify a response or request when ZAP has caught it. It also allows you to modify certain elements that you cannot modify through your browser, including:

a) The header

b) Hidden fields

c) Disabled fields

d) Fields that use JavaScript to filter out illegal characters


Figure 1.1.7: OWASP ZAP adding the Break tab

18. The **Break** tab is added to your **OWASP ZAP** window.

19. To configure ZAP as a proxy, click the **Settings** icon ( ) from the toolbar.


Figure 1.1.8: OWASP ZAP Break tab

20. In the **Options** window, click **Local Proxies** in the left pane. In the right pane, under the **Local Proxy** section, type **10.10.10.19** (the IP address of the **Windows Server 2019** virtual machine) in the **Address** field and set the **Port** value to the default, **8080**; click **OK**.

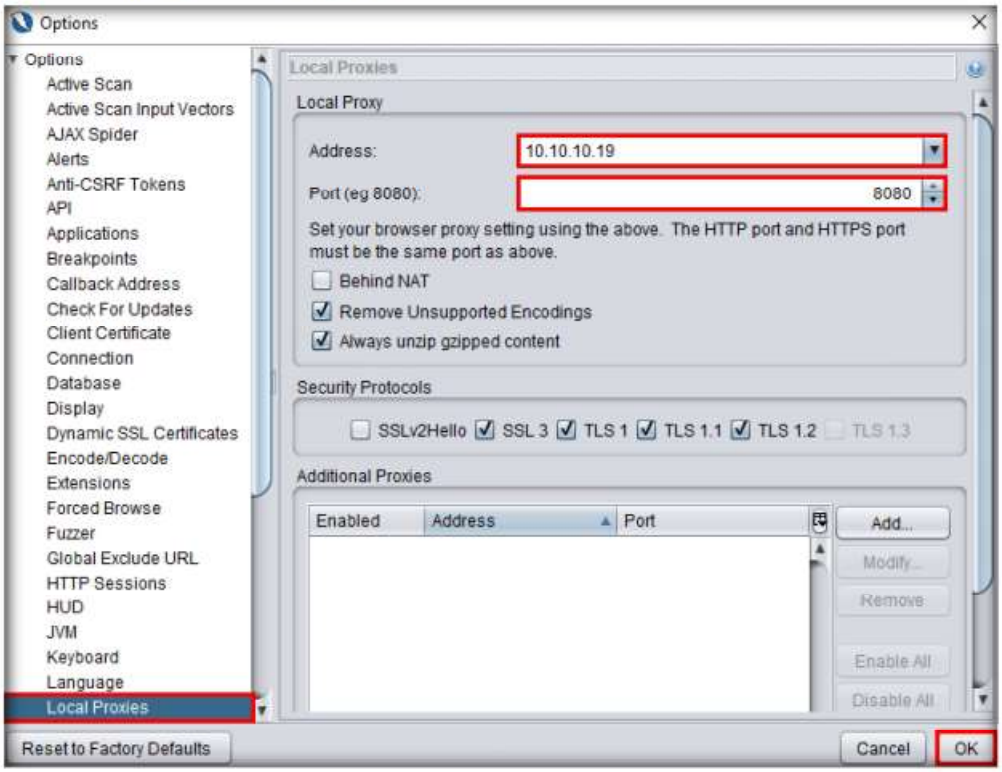


Figure 1.1.9: OWASP ZAP Options window

21. Click the **Set break on all requests and responses** icon (  ) on the main ZAP toolbar. This button sets and unsets a global breakpoint that will trap and display the next response or request from the victim's machine in the **Break** tab.

**Note**: The **Set break on all requests and responses** icon turns automatically from green to red.


Figure 1.1.10: OWASP ZAP: Setting a breakpoint

**TASK 1.3**

**Browse Website as a Victim**

22. Now, switch back to the victim's machine (**Windows 10**) and launch the same browser in which you configured the proxy settings. In this lab, we have configured the **Google Chrome** browser.
23. Type **www.moviescope.com** in the address bar and press **Enter**.
24. A message appears, stating that **Your connection is not private**. Click the **Advanced** button.

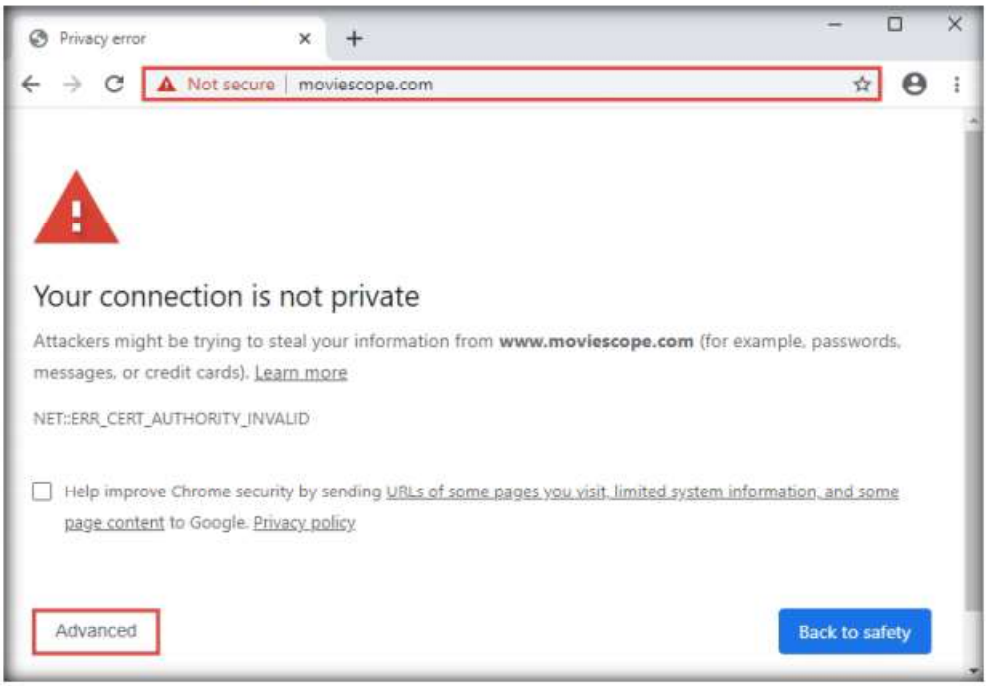


Figure 1.1.11: Your connection is not private message in the browser

25. On the next page, click **Proceed to www.moviescope.com (unsafe)** to open the website.

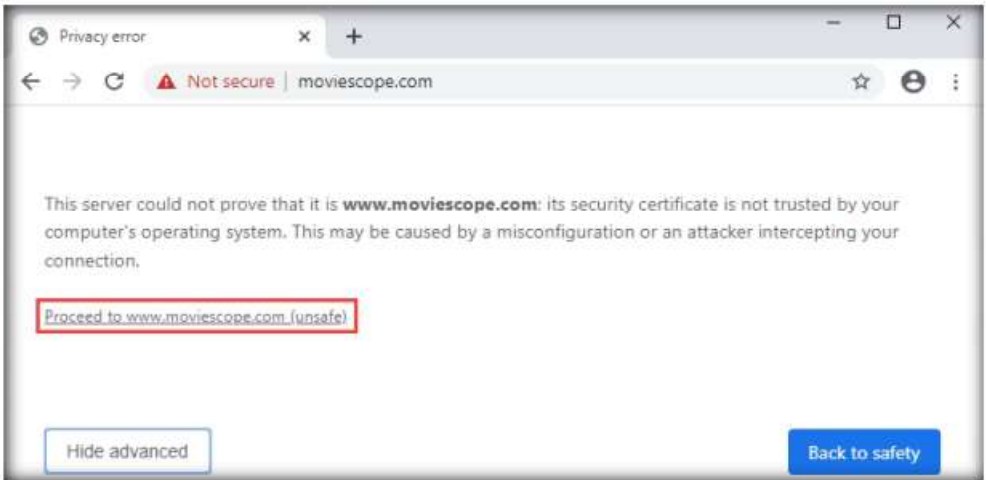


Figure 1.1.12: Proceed to the website

26. Now, switch back to the attacker machine (**Windows Server 2019**) and observe that **OWASP ZAP** has begun to capture the requests of the victim's machine.

**TASK 1.4**

**Modify GET Request Content**

27. In **Steps 23-25**, we visited **www.moviescope.com** in the victim's browser. Look in the **Break** tab and click the **Submit and step to next request and response** icon ( ) on the toolbar to capture the **www.moviescope.com** request.

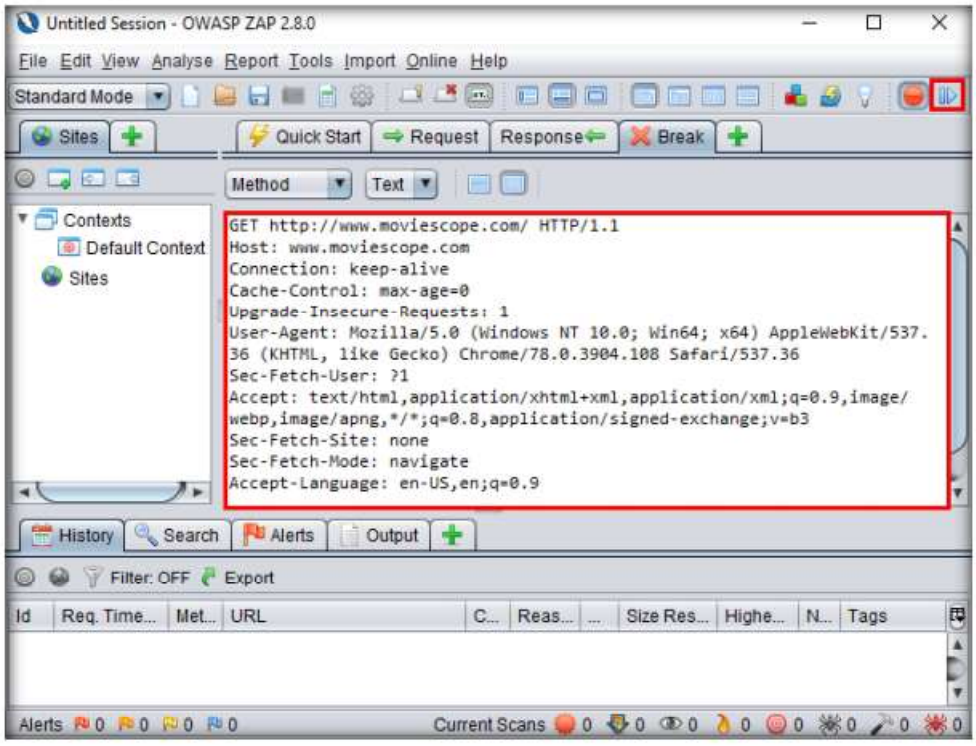


Figure 1.1.13: OWASP ZAP: Capturing a request

28. A **HTTP response** appears; click the icon ( ) on the toolbar.

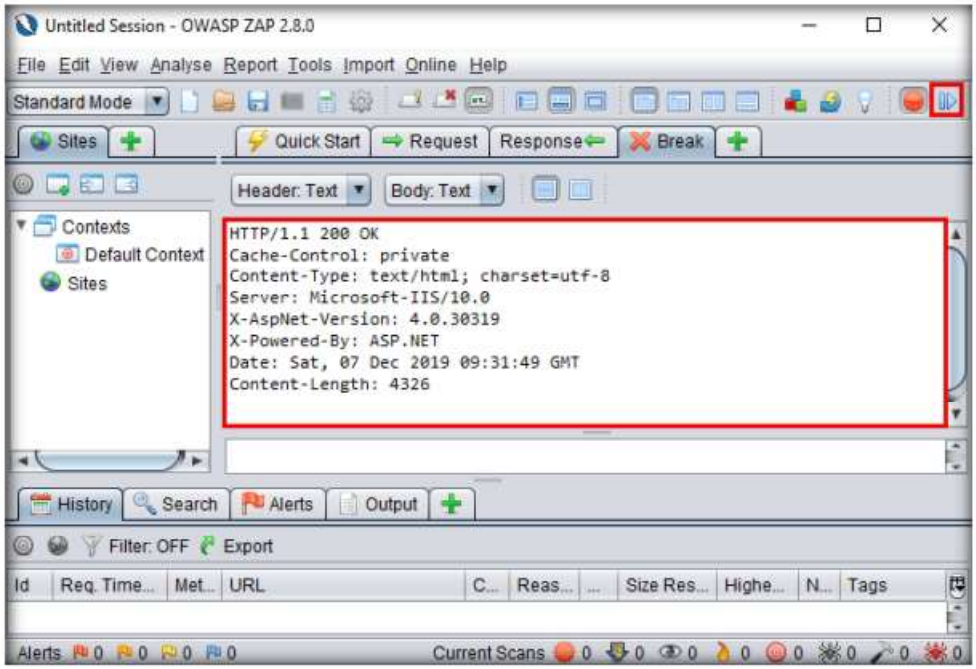


Figure 1.1.14: OWASP ZAP: Capturing an HTTP response

29. Now, in the **Break** tab, modify **www.moviescope.com** to **www.goodshopping.com** in all the captured GET requests.

**Note**: If you find any URL starting with **https**, modify it to **http**.

30. Once you have modified the GET requests, click the ( ) icon on the toolbar to forward the traffic to the victim's machine.

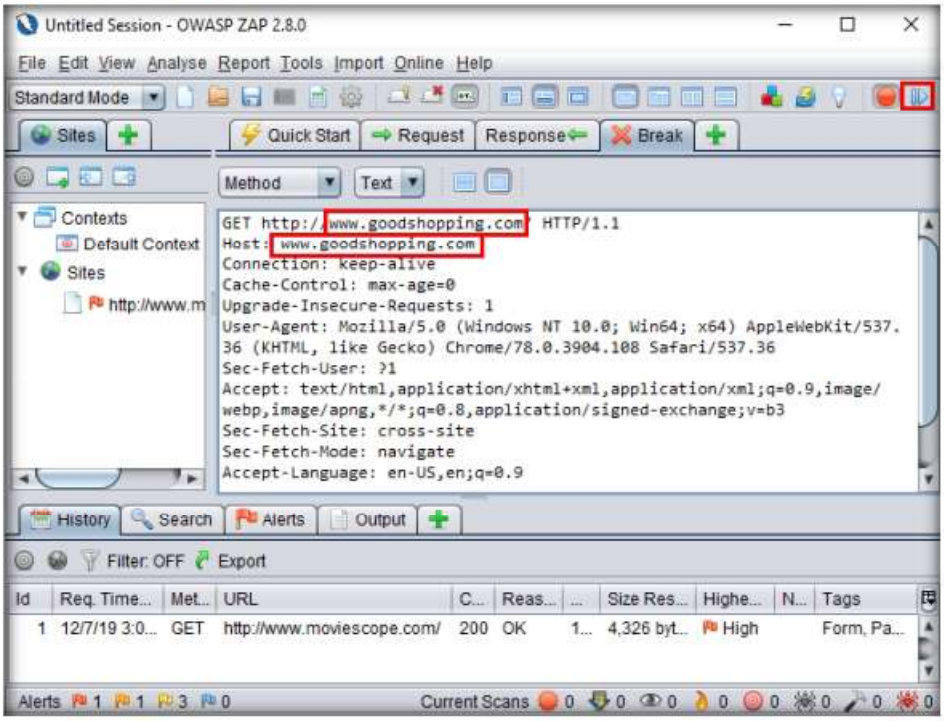


Figure 1.1.15: OWASP ZAP: Modifying the GET requests

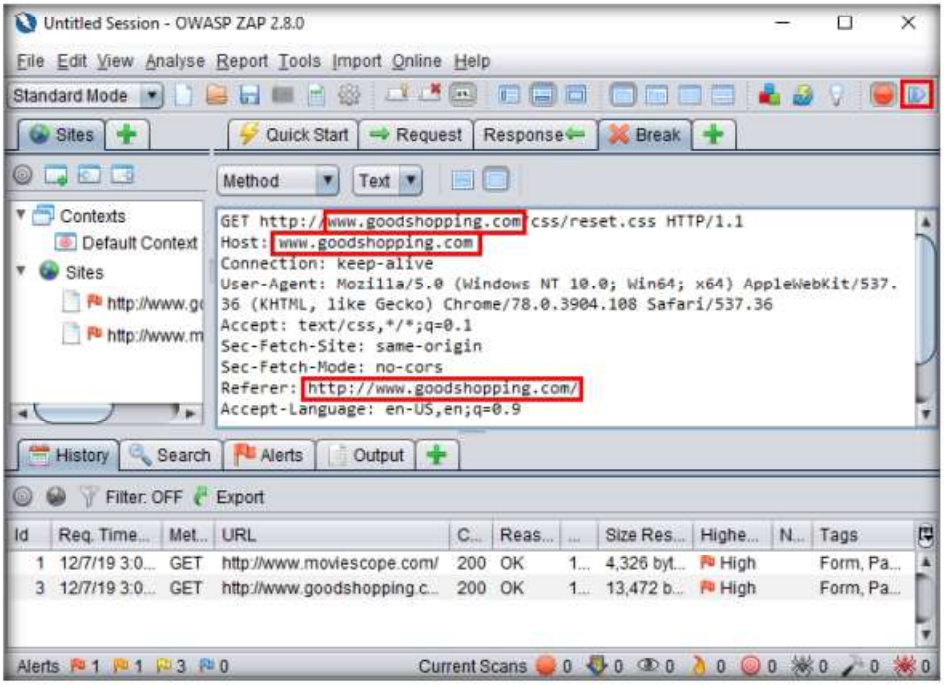


Figure 1.1.16: OWASP ZAP: Modifying the GET requests

31. Modify every **HTTP** request captured by **OWASP ZAP** until you see the **www.goodshopping.com** page in the victim's machine.

**Note**: You will need to switch back and forth from the victim's machine to see the browser status while you do this.

32. Now, switch to the victim's machine (**Windows 10**); the browser displays the website that the attacker wants the victim's machine to see (in this example, **www.goodshopping.com**).

**Note**: It takes multiple iterations to open the Good Shopping site in the victim's machine.

33. The victim has navigated to **www.moviescope.com**, but now sees **www.goodshopping.com**; while the address bar displays **www. moviescope.com**, the window displays **www. goodshopping.com**.

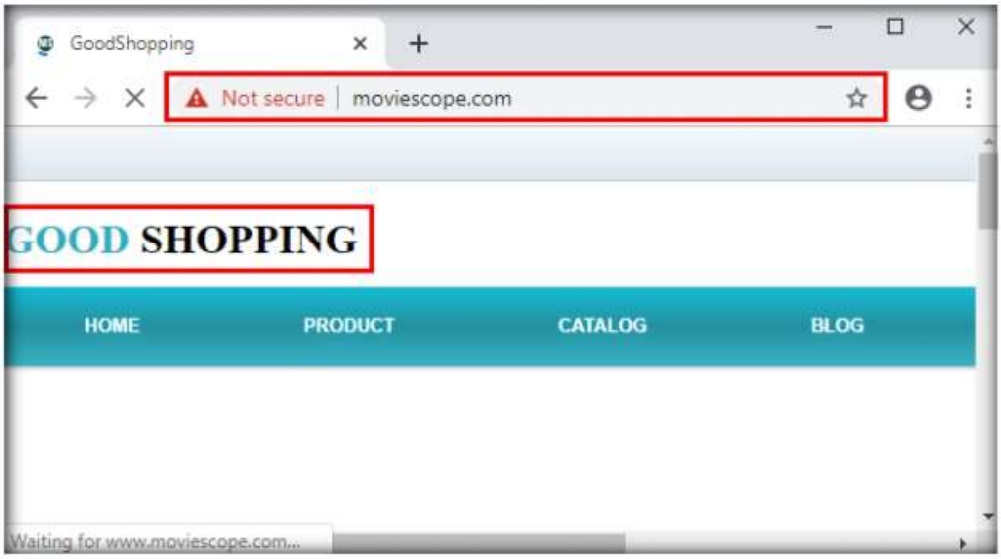


Figure 1.1.17: The right address, the wrong page

**TASK 1.5**

**Change the Proxy Settings Back to Default**

34. Now, we shall change the proxy settings back to the default settings. To do so, perform **Steps 3-5** again.
35. In the **Settings** window, under the **Manual proxy setup** section in the right pane, click the **On** button to toggle it back to **Off**, as shown in the screenshot.

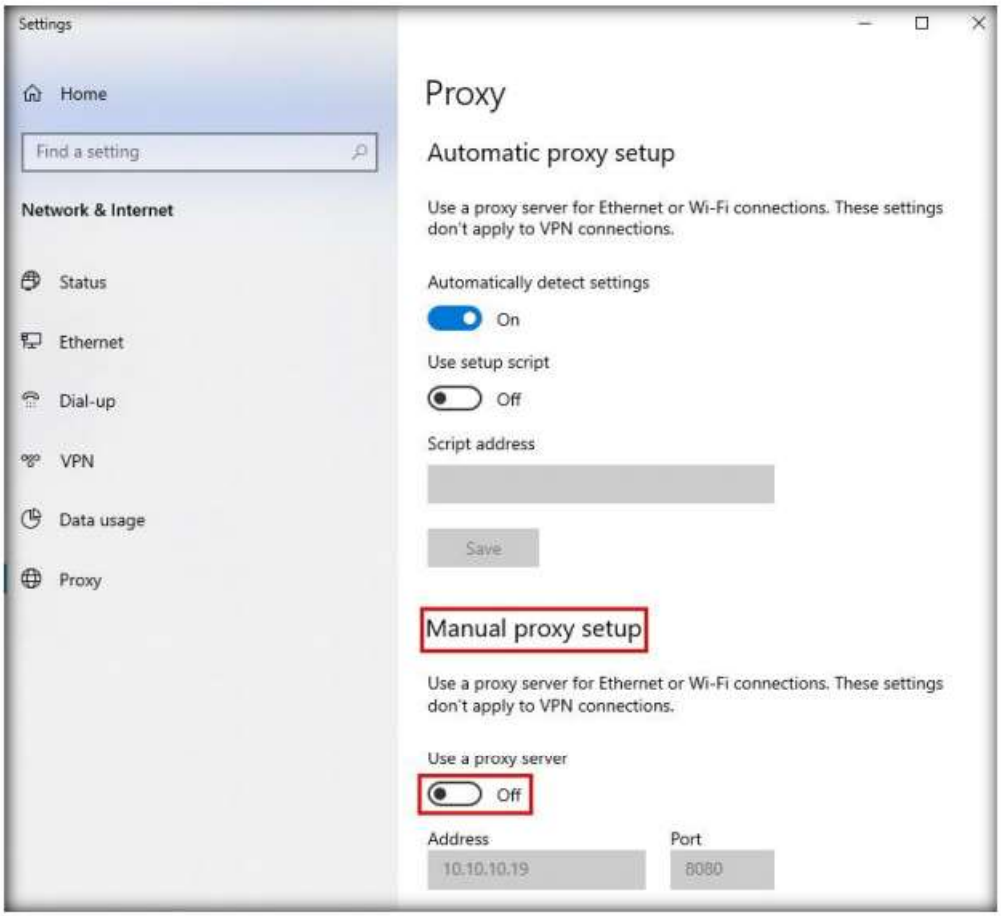


Figure 1.1.18: Settings window: Proxy setup

36. This concludes the demonstration of performing session hijacking using ZAP.
37. Close all open windows and document all the acquired information.

TASK 2

## Intercept HTTP Traffic using bettercap

Attackers can use session hijacking to launch various kinds of attacks such as man-in-the middle (MITM) attacks. In an MITM attack, the attacker places himself/herself between the authorized client and the webserver so that all information traveling in either direction passes through them.

An ethical hacker or a penetration tester, you must know how MITM attacks work, so that you can protect your organization's sensitive information from them.

Here, we will use the bettercap tool to intercept HTTP traffic on the target system.

**Note**: Ensure that the **Windows 10** and **Windows Server 2019** virtual machines are running.

1. Turn on the **Parrot Security** virtual machine.

2. In the login page, the **attacker** username will be selected by default. Enter password as **toor** in the **Password** field and press **Enter** to log in to the machine.

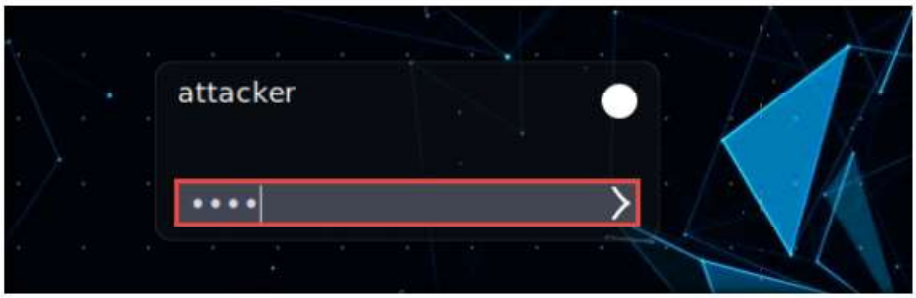


Figure 1.2.1: Parrot Security login page

**Note**:

- If a **Parrot Updater** pop-up appears at the top-right corner of **Desktop**, ignore and close it.
- If a **Question** pop-up window appears asking you to update the machine, click **No** to close the window.

3. Click the **MATE Terminal** icon at the top of the **Desktop** window to open a **Terminal** window.

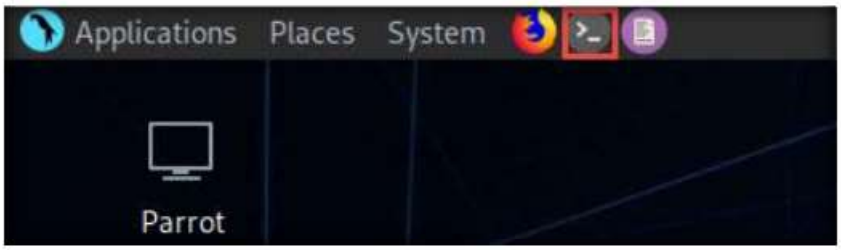


Figure 1.2.2: MATE Terminal Icon

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

**Note**: The password that you type will not be visible.

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

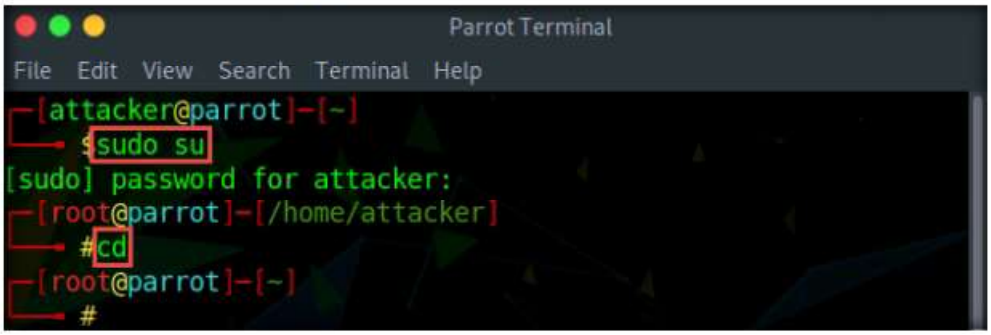


Figure 1.2.3: Running the programs as a root user

**TASK 2.1**

**Launch & Configure bettercap**

7. In the **Parrot Terminal** window, type **bettercap -h** and press **Enter**

**Note**: In this command, **-h:** requests a list of the available options.

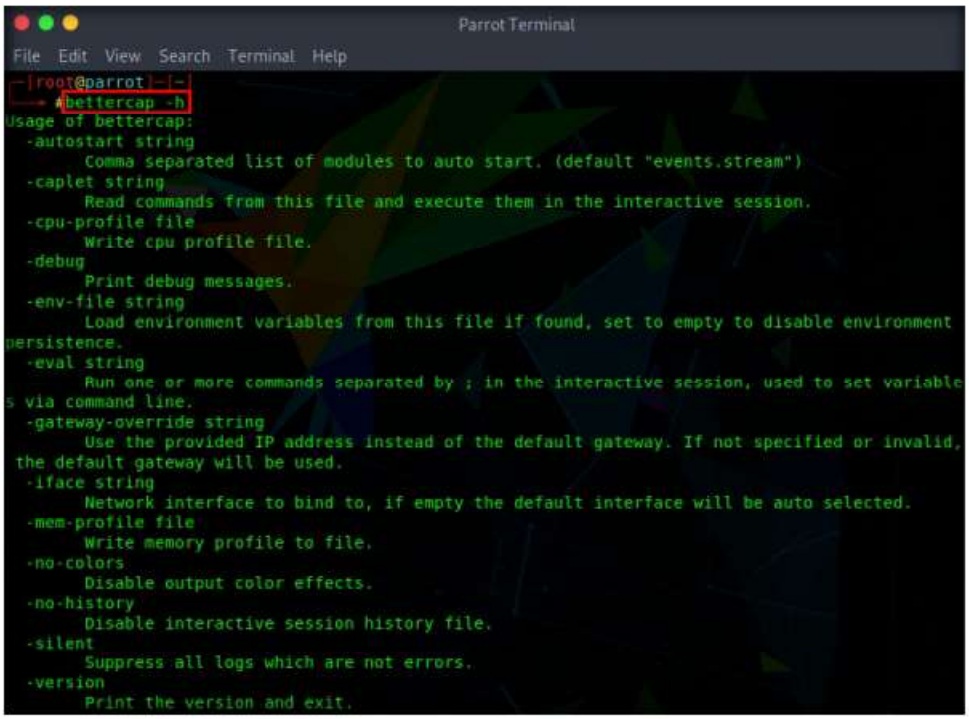


Figure 1.2.4: bettercap help

bettercap is a powerful, flexible, and portable tool created to perform various types of MITM attacks against a network; manipulate HTTP, HTTPS, and TCP traffic in real-time; sniff for credentials; etc.

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

**Note**: **-iface:** specifies the interface to bind to (in this example, **eth0**).

**Note**: If the bettercap version in your lab environment is old, run the following commands:

- **sudo apt remove bettercap**
- **sudo rm /usr/local/bin/bettercap**
- **ln -s /usr/lib/x86_64-linux-gnu/libpcap.so.1.8.1 /usr/lib/x86_64-linux-gnu/libpcap.so.1**
- **wget "https://github.com`curl -s https://github.com/bettercap/bettercap/releases | grep -E -o '/bettercap/bettercap/releases/download/v[0-9.]+/bettercap_linux_amd64_[0-9.]+zip' | head -n 1`"**


Figure 1.2.5: bettercap network interface binding

9. Type **help** and press **Enter** to view the list of available modules in bettercap.

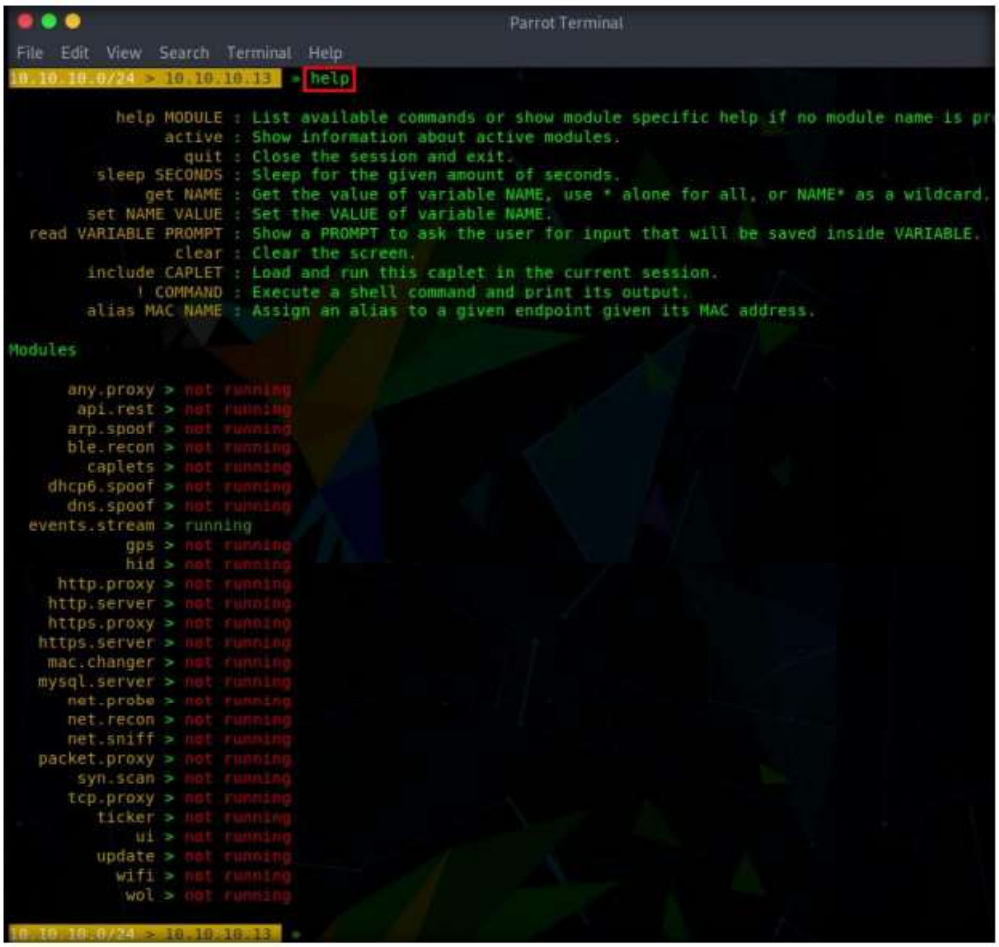


Figure 1.2.6: bettercap modules

10. 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.

11. 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.

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

12. Type **set net.sniff.regexp '.*password=.+'** and press **Enter**. This module will only consider the packets sent with a payload matching the given regular expression (in this case, **'.*password=.+'**).

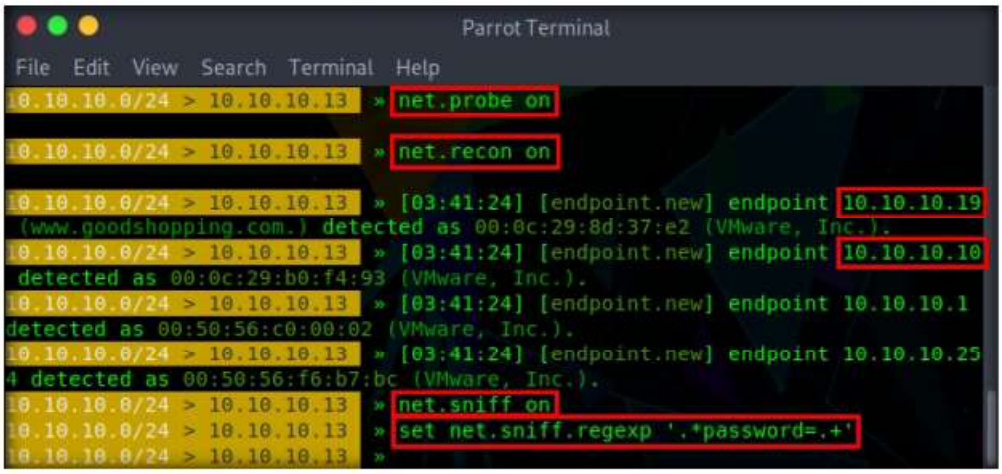


Figure 1.2.7: Initializing the required bettercap modules

13. You can observe that bettercap starts sniffing network traffic on target machine **Windows 10**, as shown in the screenshot.

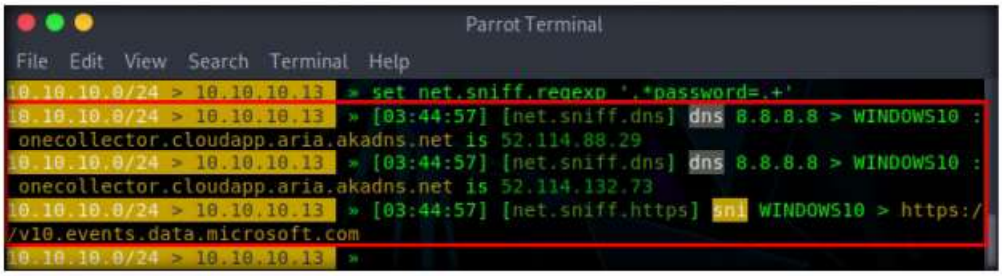


Figure 1.2.8: bettercap sniffing traffic

**TASK 2.2**

**Log in to a Website as a Victim**

14. Now, switch to the **Windows 10** virtual machine. Open any web browser (in this case, **Mozilla Firefox**), type **www.moviescope.com** in the address bar, and press **Enter**.

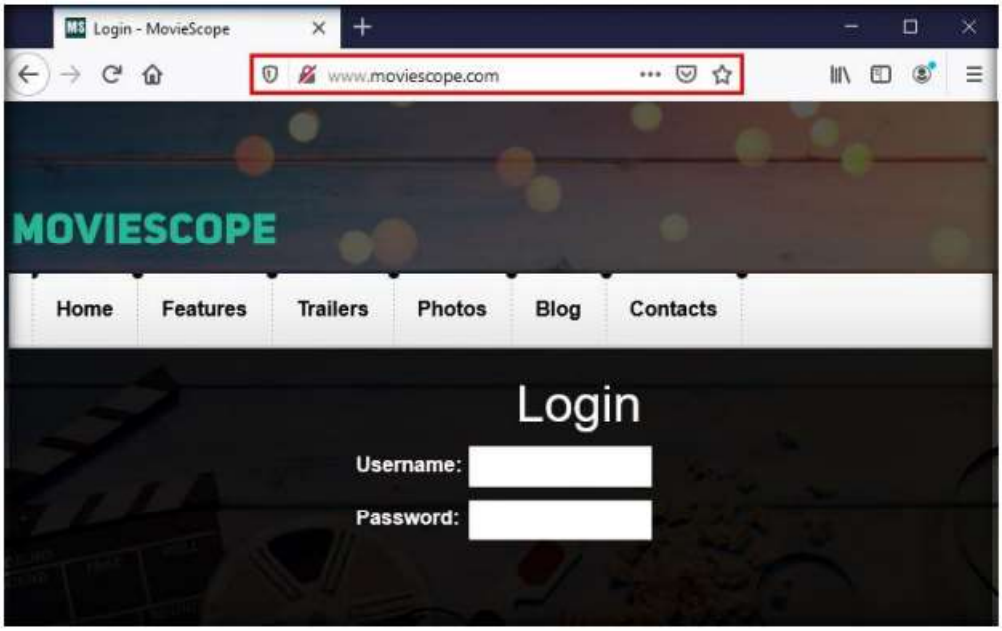


Figure 1.2.9: Navigate to a website as the victim

15. Switch back to the **Parrot Security** virtual machine. You can observe that bettercap has sniffed the website browsed by the victim on the target system, as shown in the screenshot.

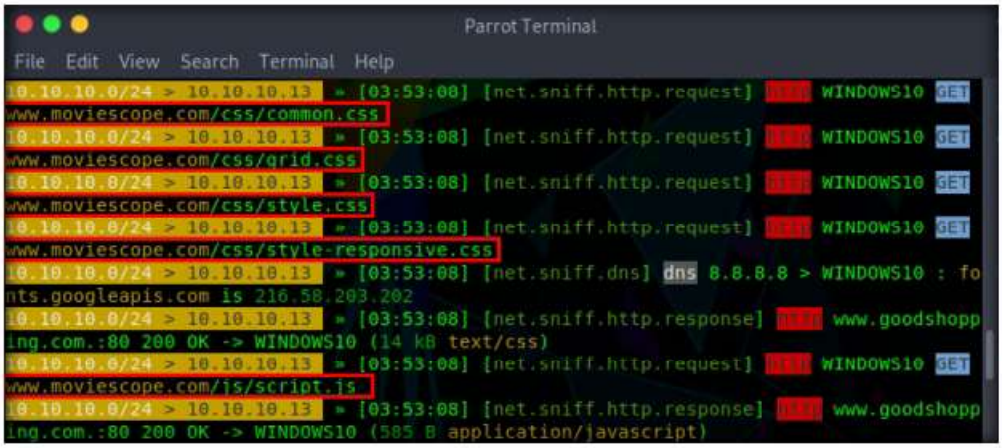


Figure 1.2.10: bettercap sniffs the browsed website

16. Now, switch to the **Windows 10** virtual machine again. On the **MovieScope** website, enter any credentials (in this example, **sam**/**test**) and press **Enter** to log in.

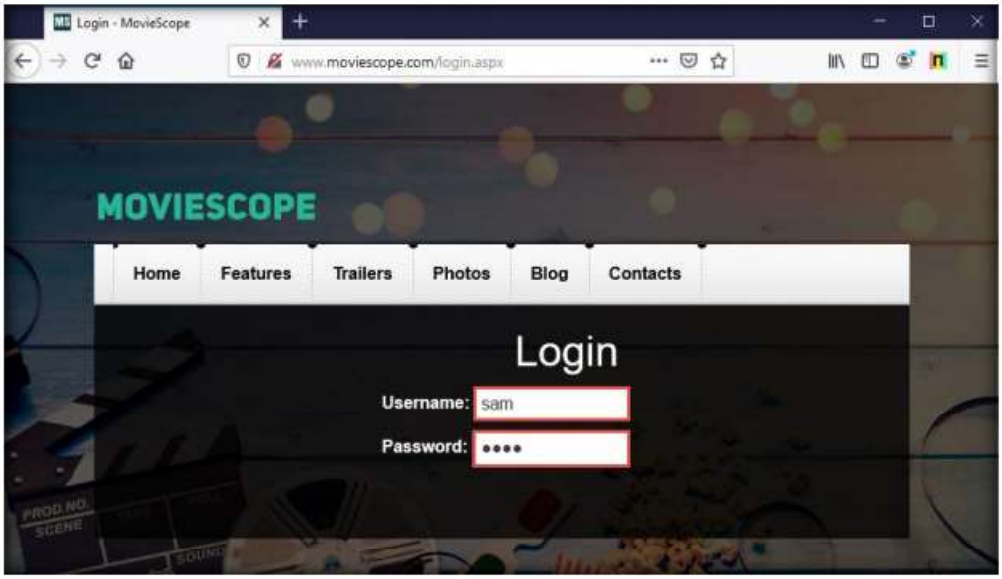


Figure 1.2.11: Log in to the MovieScope website as the victim

**TASK 2.3**

**Observe Captured Credentials**

17. Switch to the **Parrot Security** virtual machine. You can observe the details of both the browsed website and the credentials obtained in plain text, as shown in the screenshot.

**Note**: bettercap collects all http logins used by routers, servers, and websites that do not have SSL enabled. In this task, we are using **www.moviescope.com** for demonstration purposes, as it is http-based. To use bettercap to sniff network traffic from https-based websites, you must enable the SSL strip module by issuing the command **set http.proxy.sslstrip true**.

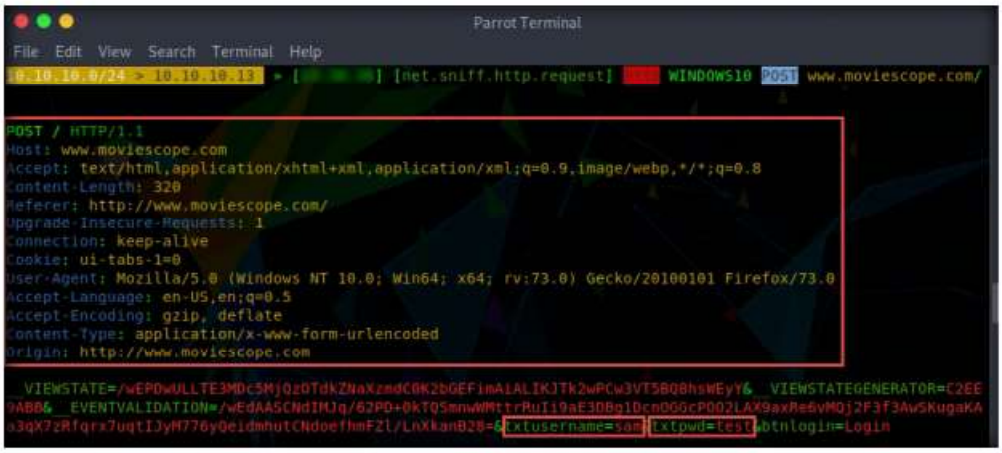


Figure 1.2.12: bettercap sniffs the password

18. After obtaining the credentials, press **Ctrl+C** to terminate bettercap. The credentials can be used to log in to the target user's account and obtain further sensitive information.
19. When the **Are you sure you want to quit this session?** message appears, press **y**, and then **Enter**.


Figure 1.2.13: Terminate bettercap

20. This concludes the demonstration of how to intercept HTTP traffic using bettercap.
21. Close all open windows and document all the acquired information.
22. Turn off the **Windows 10**, **Windows Server 2019** and **Parrot Security** virtual machines.

## Lab Analysis

Analyze and document all the results discovered in the lab exercise.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS ABOUT THIS LAB.

| Internet Connection Required | |
|---|---|
| ☐ Yes | ☑ No |
| **Platform Supported** | |
| ☑ Classroom | ☑ iLabs |

Lab 2

# Detect Session Hijacking

*Ethical hackers and penetration testers have various tools and techniques at their disposal for detecting session hijacking attacks, which make the detection process an easy task.*

ICON KEY

- Valuable Information
- Test Your Knowledge
- Web Exercise
- Workbook Review

## 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

## Lab Environment

To carry out this lab, you need:

- Windows 10 virtual machine
- Parrot Security virtual machine
- Web browsers with an Internet connection
- Administrator privileges to run the tools

## Lab Duration

Time: 10 Minutes

# 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

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

**Note**: We will use the **Parrot Security** (**10.10.10.13**) virtual machine to carry out a session hijacking attack on the **Windows 10** (**10.10.10.10**) virtual machine.

1. Turn on the **Parrot Security** and **Windows 10** virtual machines.
2. In the **Windows 10** virtual machine, log in with the credentials **Admin** and **Pa$$w0rd**.

**TASK 1.1**

**Launch Wireshark**

3. In the **Type here to search** field at the bottom of **Desktop**, type **wireshark**. Click **Wireshark** from the results.
4. **The Wireshark Network Analyzer** window opens. Double-click the primary network interface (in this case, **Ethernet0**) to start capturing network traffic.

**Note**: The network interface might differ in your lab environment.

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.

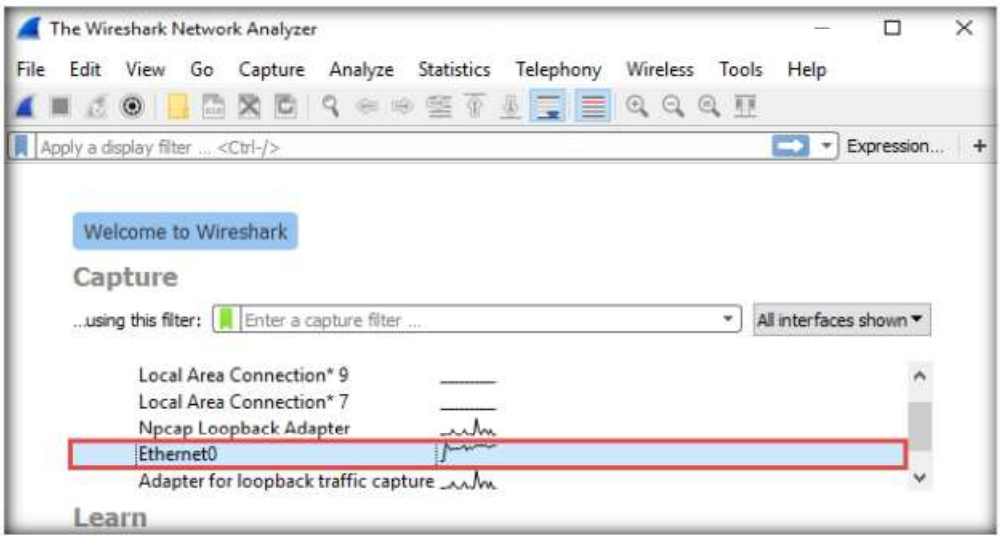


Figure 2.1.1: Capturing Traffic with Wireshark

Wireshark 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.

5. **Wireshark** starts capturing network traffic. Leave it running.
6. Now, we shall launch a session hijacking attack on the target machine (**Windows 10**) using **bettercap**.

**Note**: To do so, you may either follow **Steps 7-15** below, or refer to **Task 2** (**Intercept HTTP Traffic using bettercap**) in **Lab 1**.

7. Switch to the **Parrot Security** virtual machine. In the login page, the **attacker** username will be selected by default. Enter password as **toor** in the **Password** field and press **Enter** to log in to the machine.

Note:

- If a **Parrot Updater** pop-up appears at the top-right corner of **Desktop**, ignore and close it.
- If a **Question** pop-up window appears asking you to update the machine, click **No** to close the window.

8. Click the **MATE Terminal** icon at the top of the **Desktop** window to open a **Terminal** window.
9. A **Parrot Terminal** window appears. In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user.
10. In the **[sudo] password for attacker** field, type **toor** as a password and press **Enter**.

**Note**: The password that you type will not be visible.

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

**TASK 1.2**

**Launch Session Hijacking Attack**

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

**Note**: In this command, **-iface:** specifies the interface to bind to (in this case, **eth0**). The network interface might differ in your lab environment.

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.
14. Type **net.recon on** and press **Enter**. This module periodically reads the system ARP table to detect new hosts on the network.

**Note**: The net.recon module displays the detected active IP addresses in the network.

15. Type **net.sniff on** and press **Enter**. This module will start sniffing network packets.

16. You can observe that bettercap starts sniffing network traffic on the **Windows 10** machine, as shown in the screenshot.

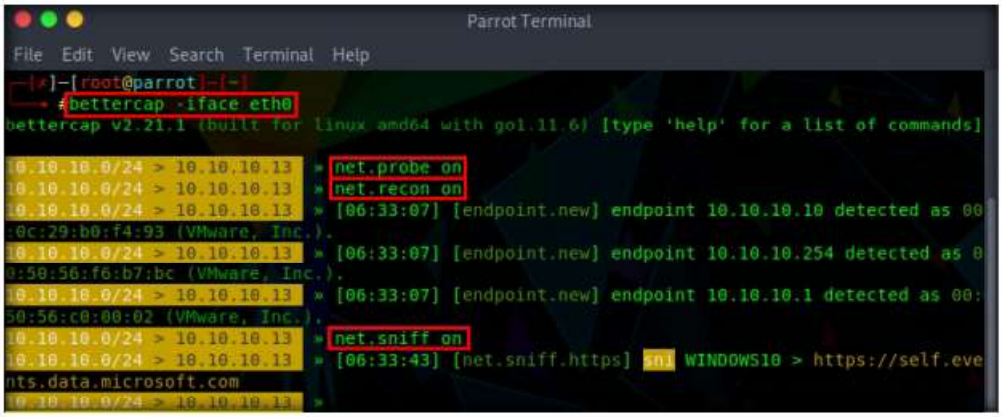


Figure 2.1.2: bettercap starts sniffing

**TASK 1.3**

**Analyze Captured Packets**

17. Switch back to the **Windows 10** virtual machine and observe the huge number of **ARP packets** captured by the **Wireshark**, as shown in the screenshot.

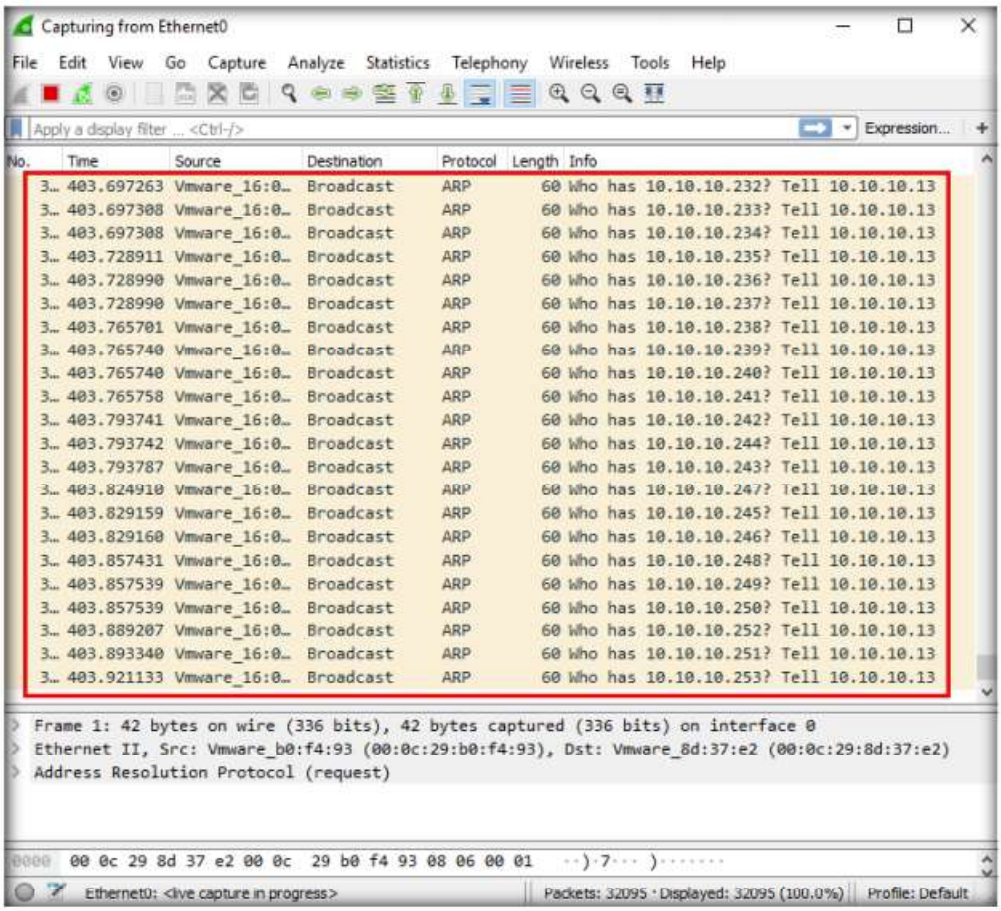


Figure 2.1.3: Wireshark captures ARP requests

**Note**: bettercap sends several ARP broadcast requests to the hosts (or potentially active hosts). A high number of ARP requests indicates that the system at **10.10.10.13** (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, **10.10.10.10**) will first go to the host system (**10.10.10.13**), 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.

18. This concludes the demonstration of how to detect a session hijacking attack using Wireshark.
19. Close all open windows and document all the acquired information.
20. Turn off the **Windows 10** and **Parrot Security** virtual machines.

## Lab Analysis

Analyze and document all the results discovered in the lab exercise.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS ABOUT THIS LAB.

| Internet Connection Required | |
|---|---|
| ☐ Yes | ☑ No |
| **Platform Supported** | |
| ☑ Classroom | ☑ iLabs |