SESSION HIJACKING

Module-11

Aniket Sunil Pagare.

📚 Table of Contents

- 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

---

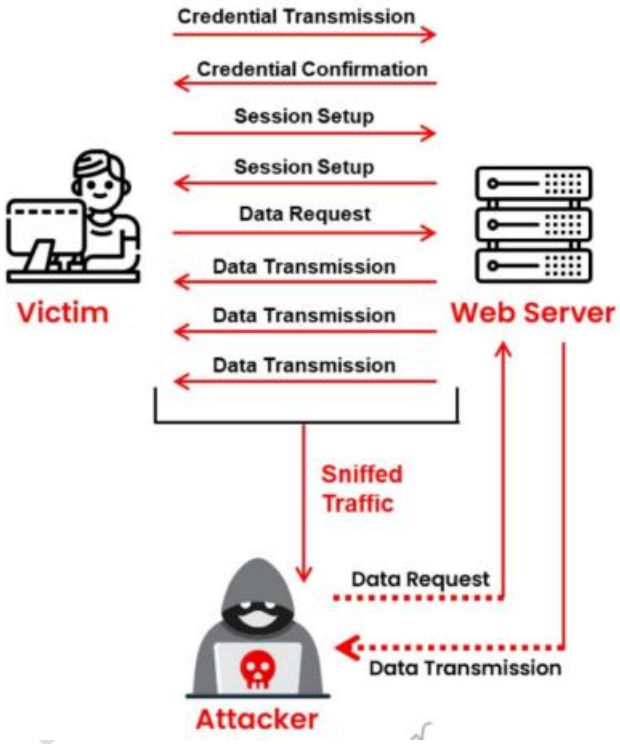
3. How to Prevent Session Hijacking

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:

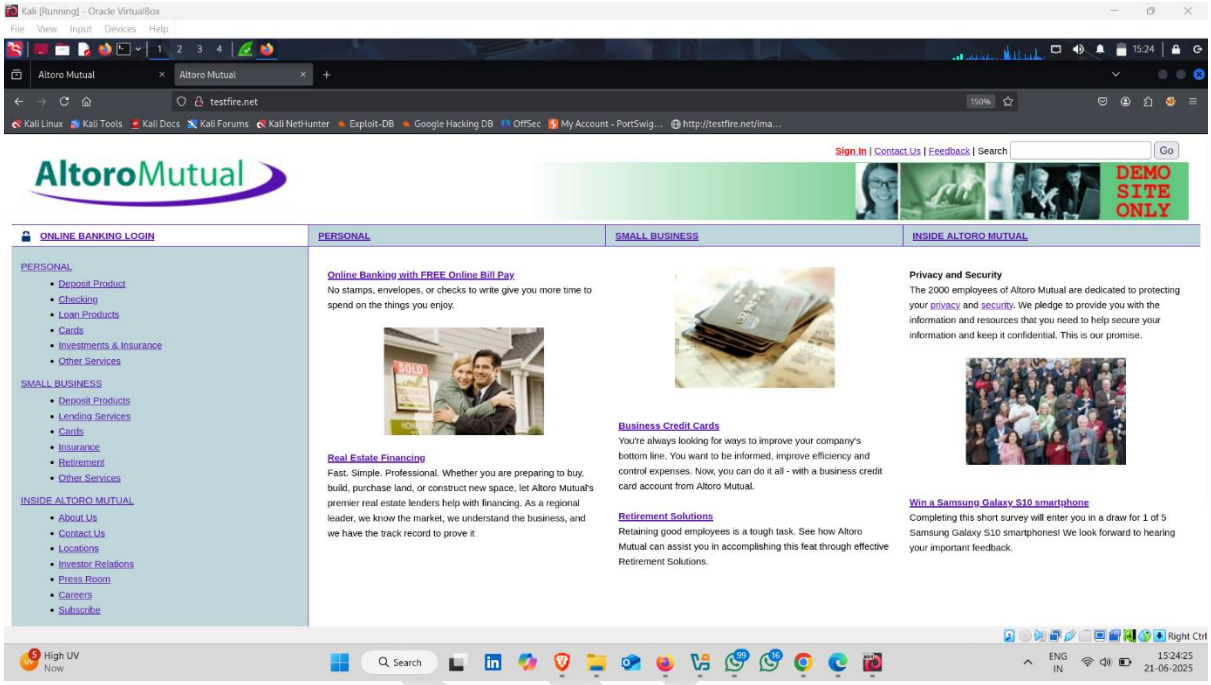
- Show how session hijacking can occur when sensitive session data, such as cookies and login tokens, is transmitted over unencrypted HTTP protocols.
- Highlight the risks associated with insecure web application communications.
- Demonstrate the practical execution of **Man-in-the-Middle (MITM)** attacks using tools like **Bettercap, 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.

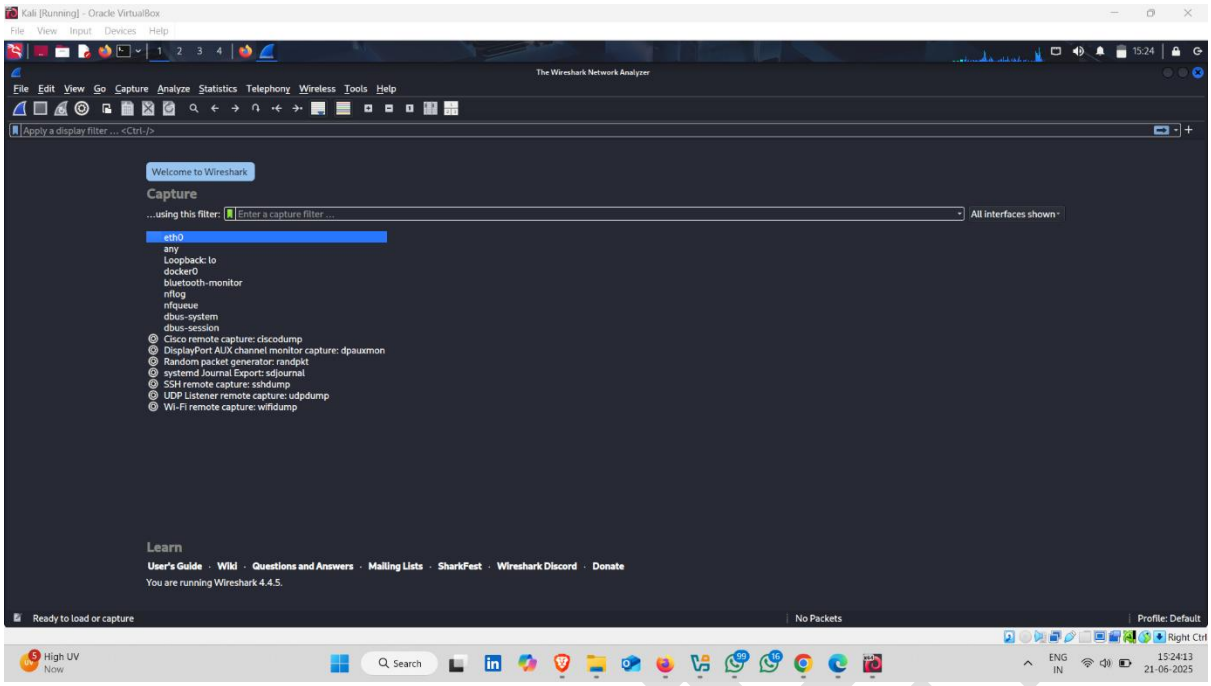
1.Session Hijacking Using Wireshark

How to do it -:

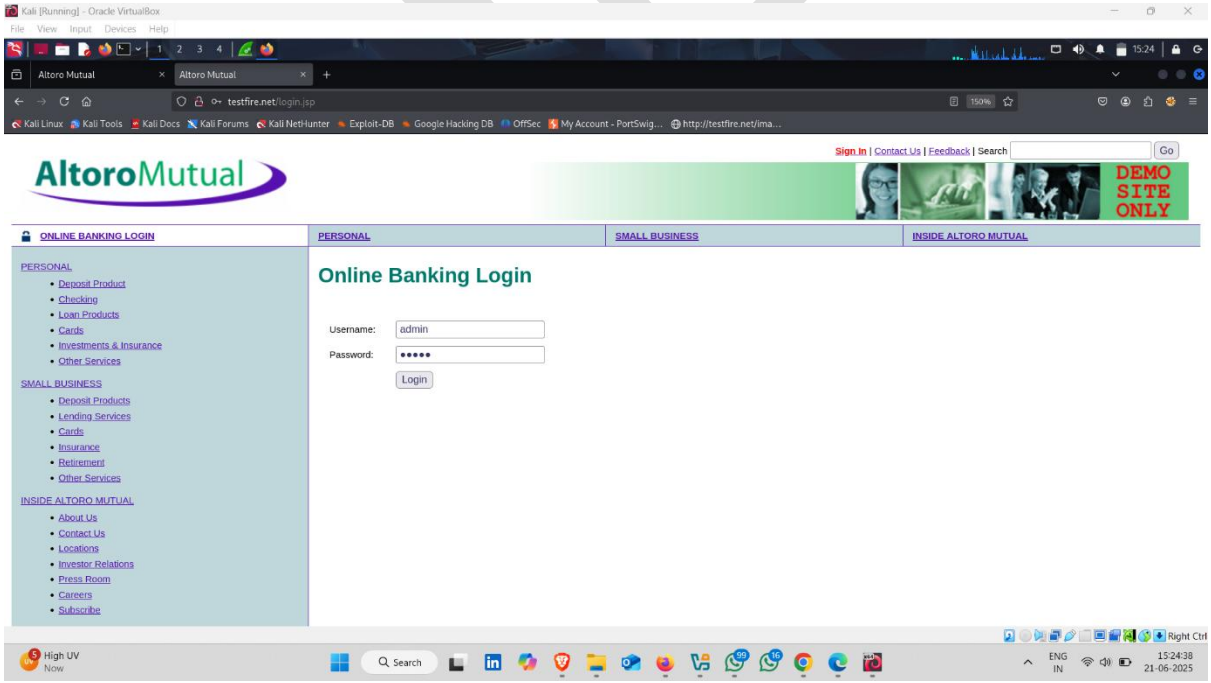
- Target Website 👇



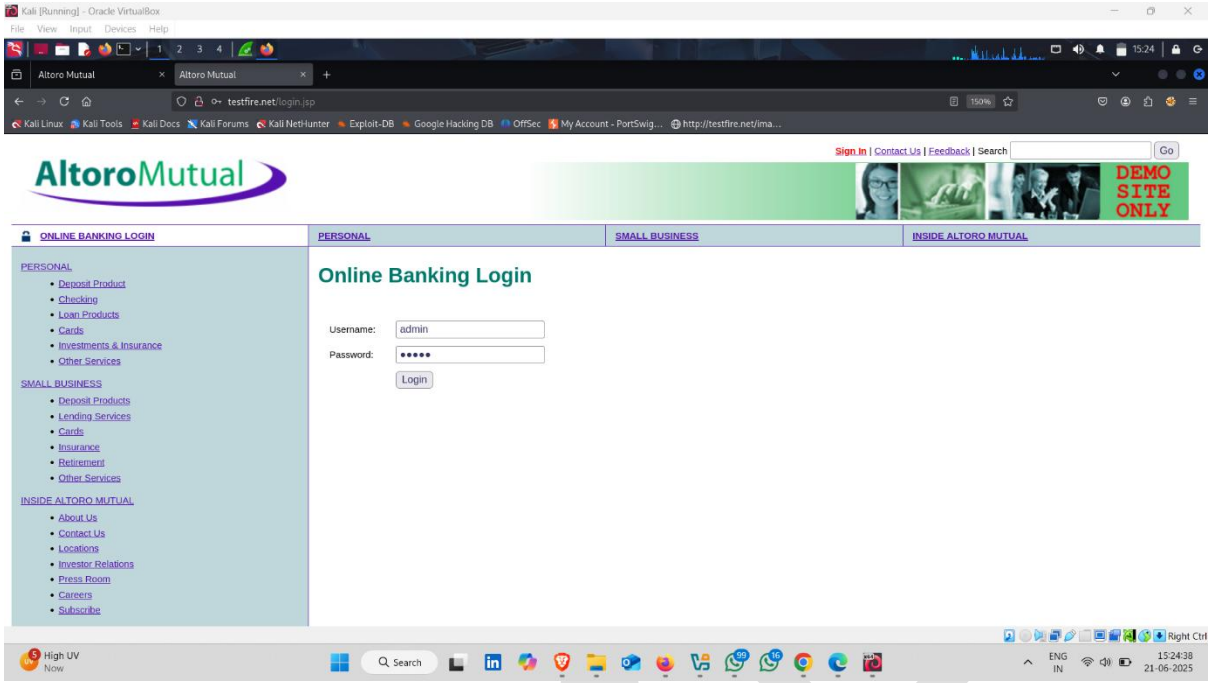
- Start Wireshark 👇



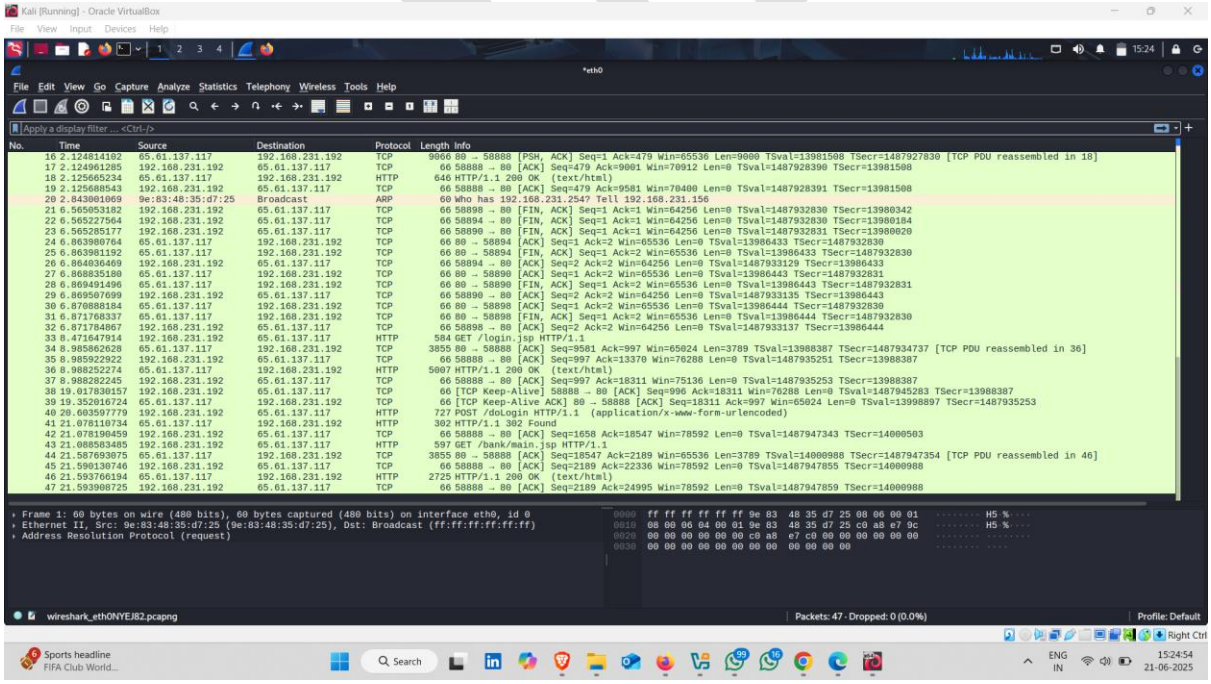
- Click on **Sing In** option and enter username and password 👇



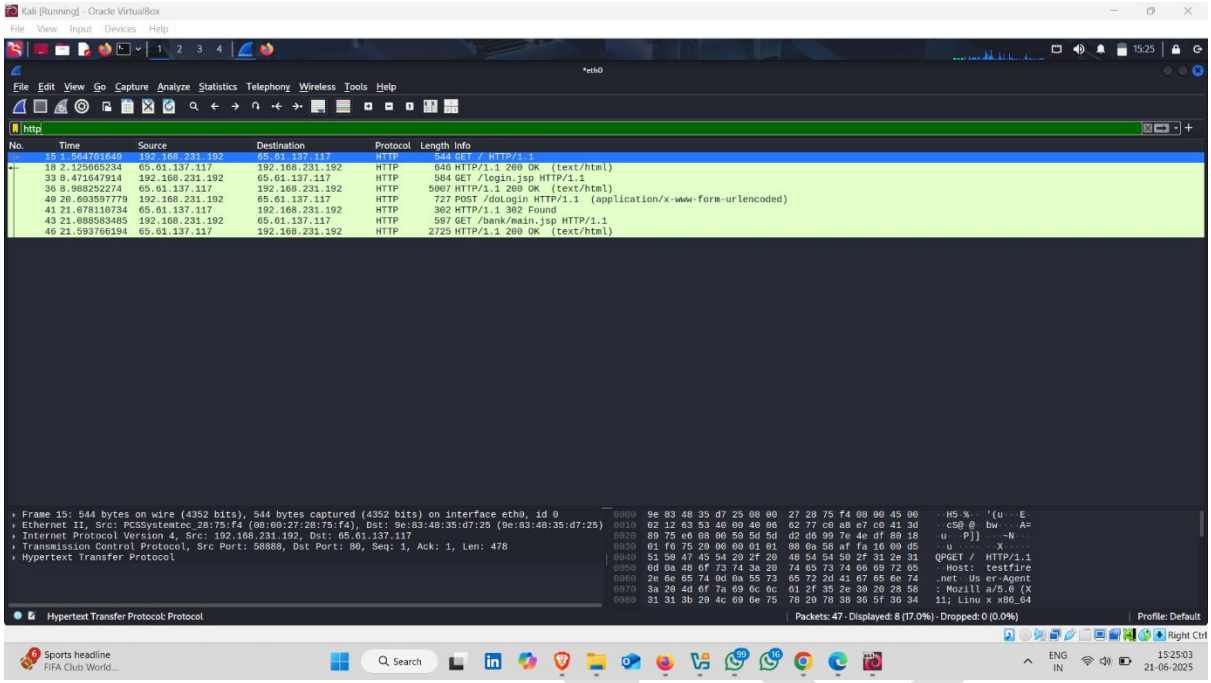
- Click on **login**



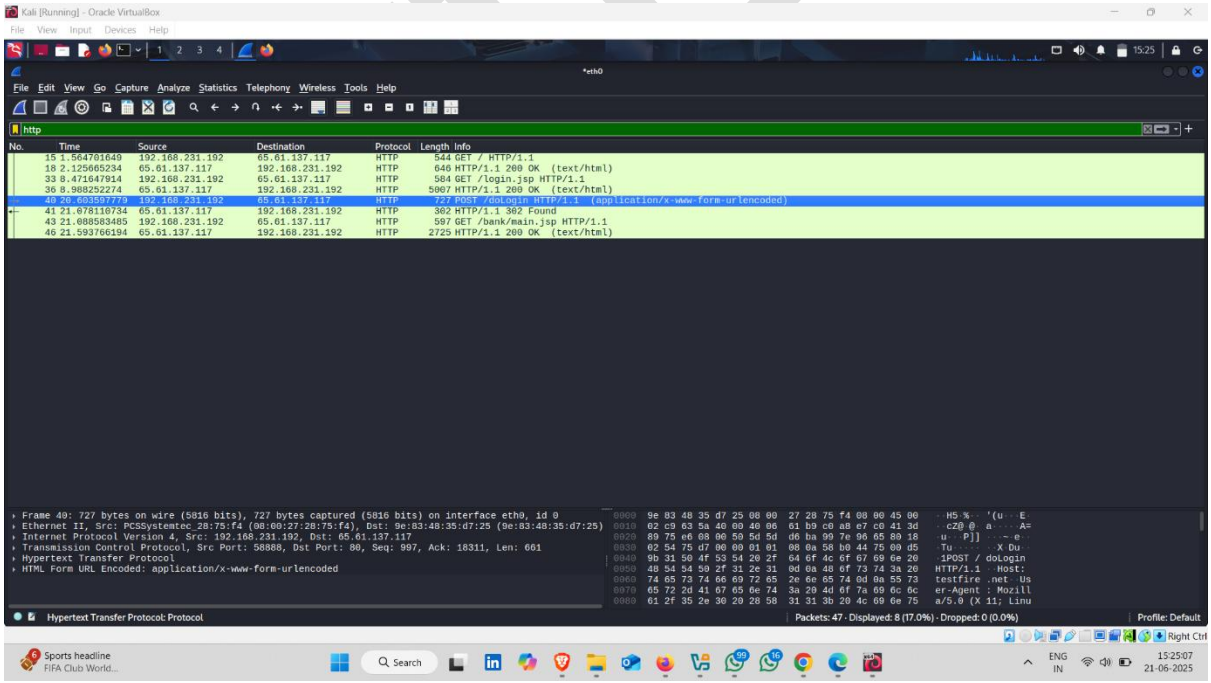
- Open wireshark and **find http Post Request**



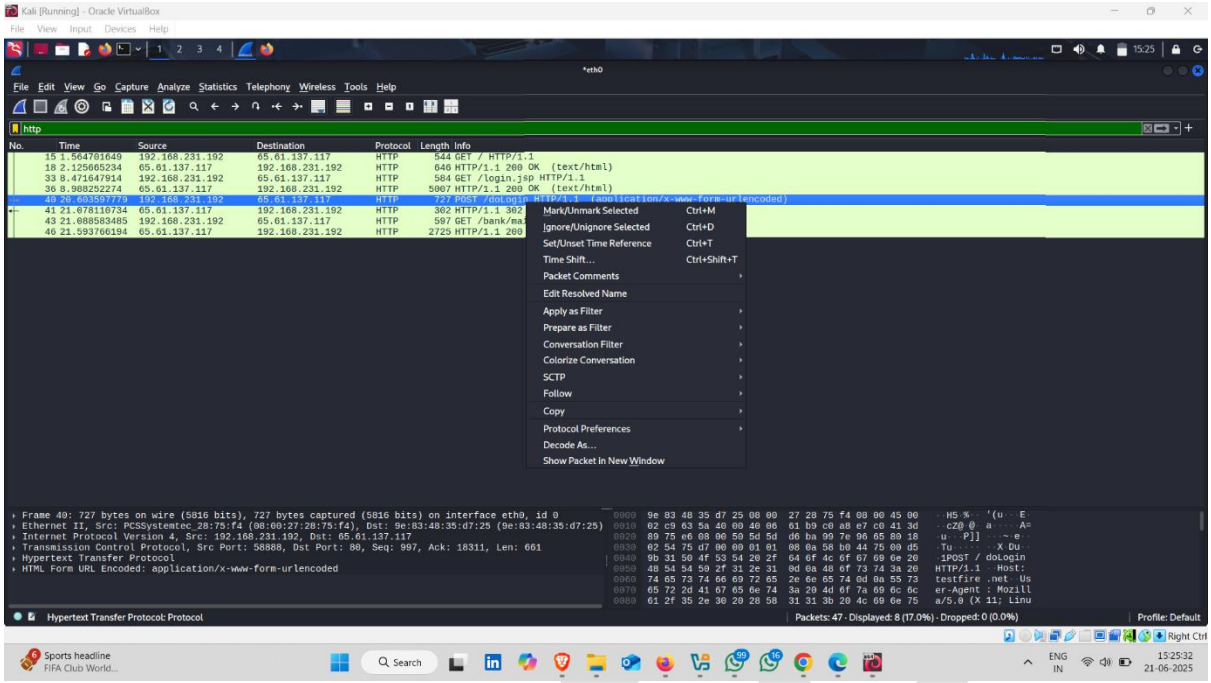
- Or Search **Http** 👇



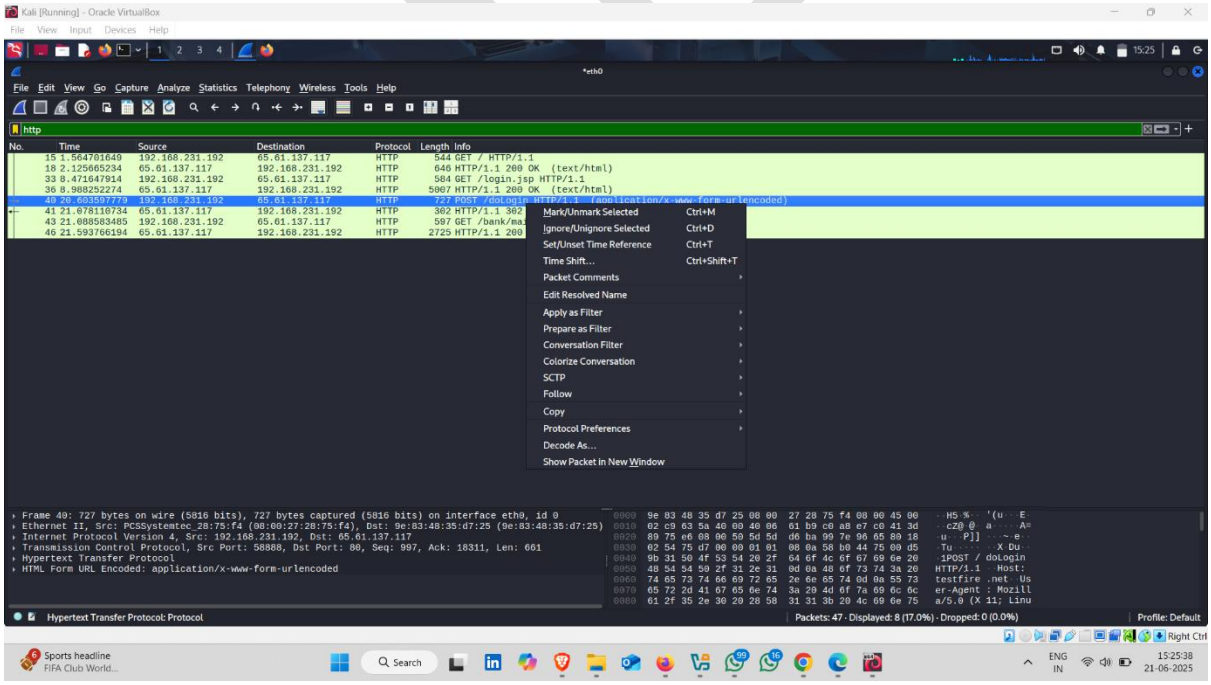
- Here , post request 👇✅



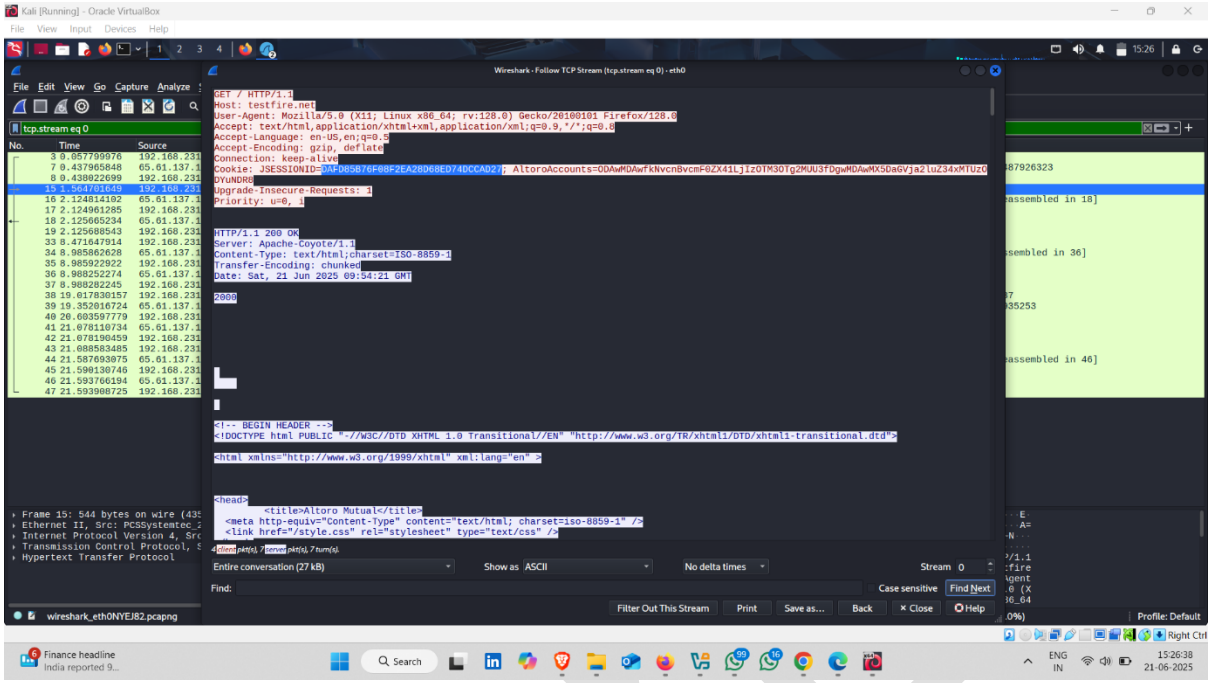
- Right click on **POST Request**



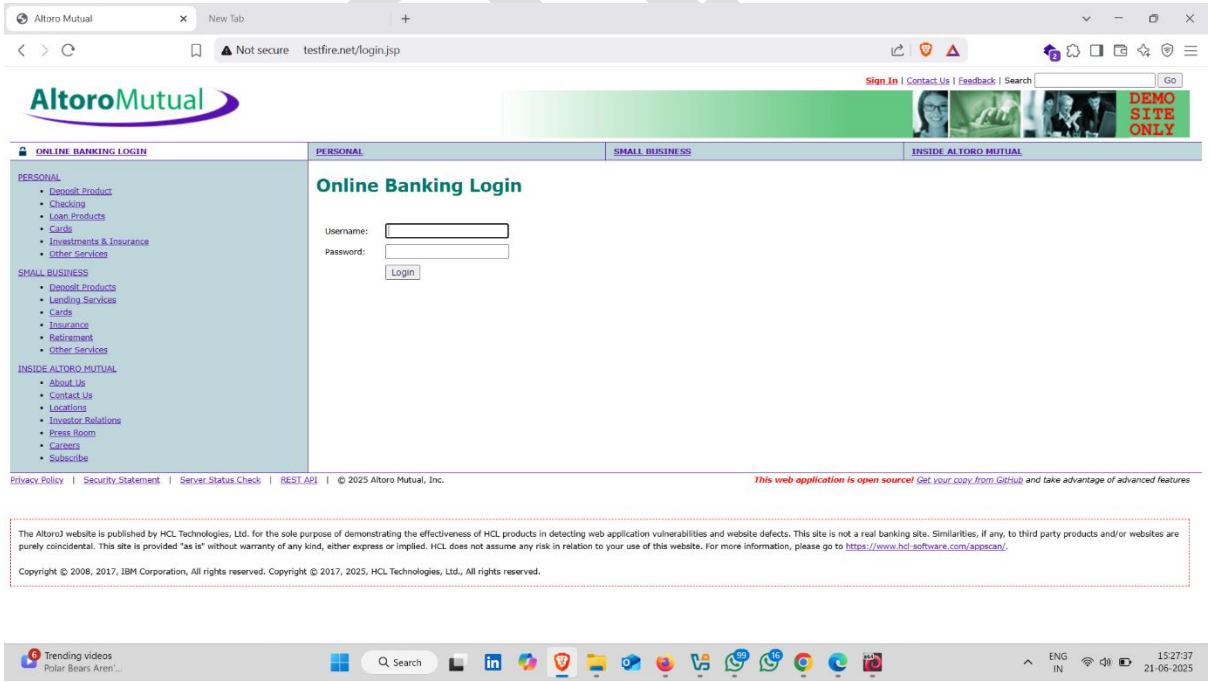
- Click on **Follow** and then click on **TCP View** ✅



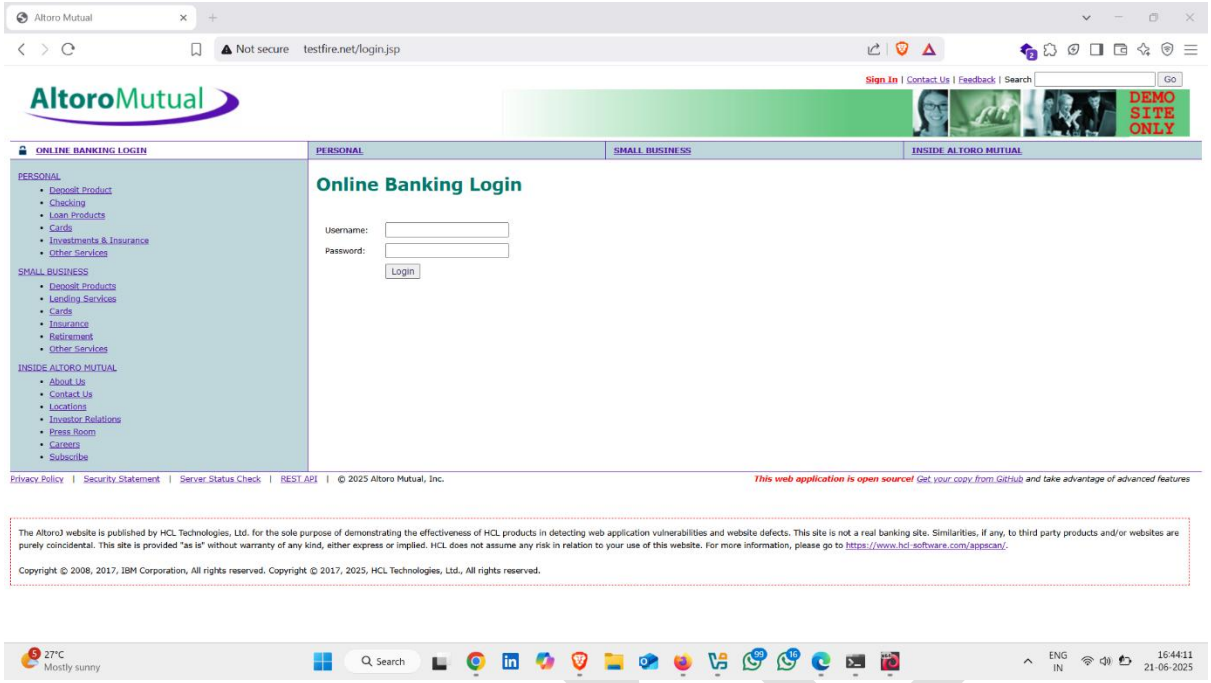
- Now , copy **JSESSIONID** ✅ 👇



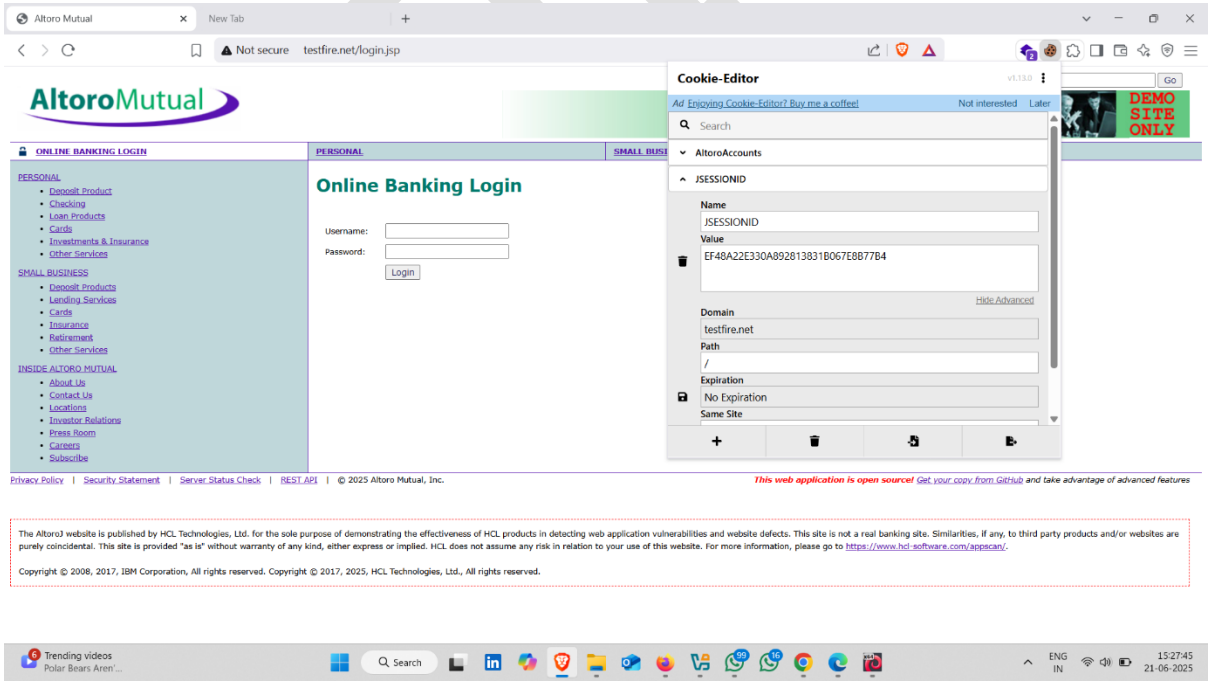
- Now , go to another browser and open target website , sign in Option



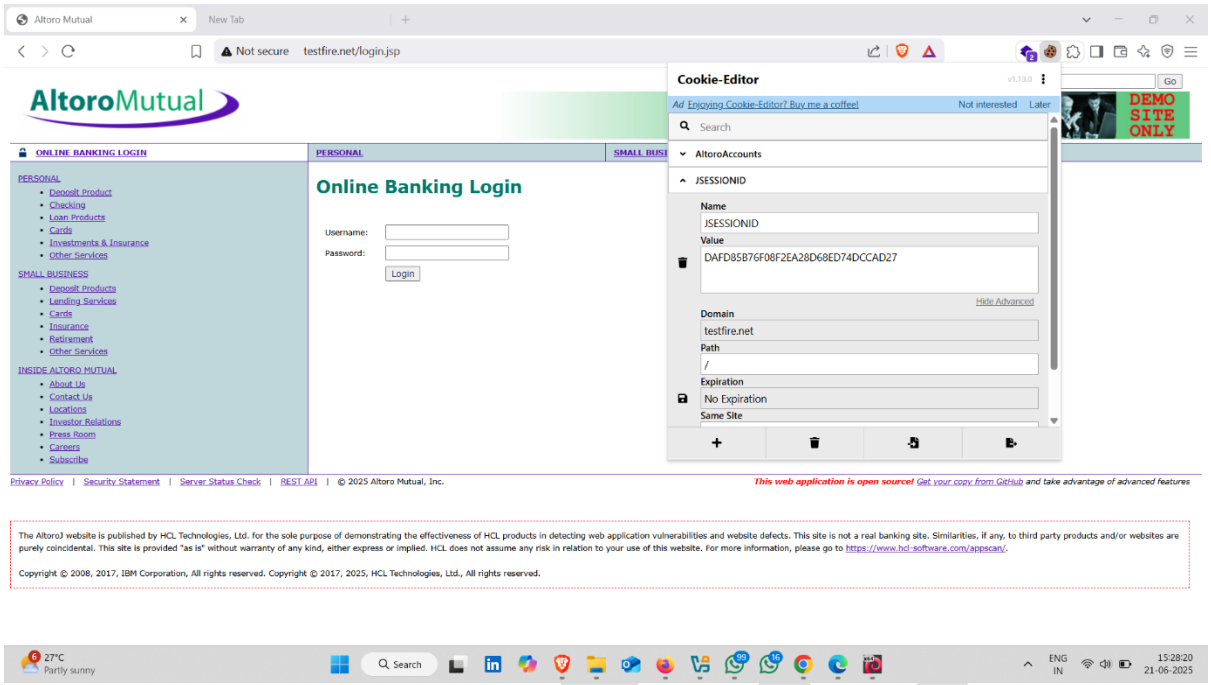
- Download Extension Cookies Editor extention 👇



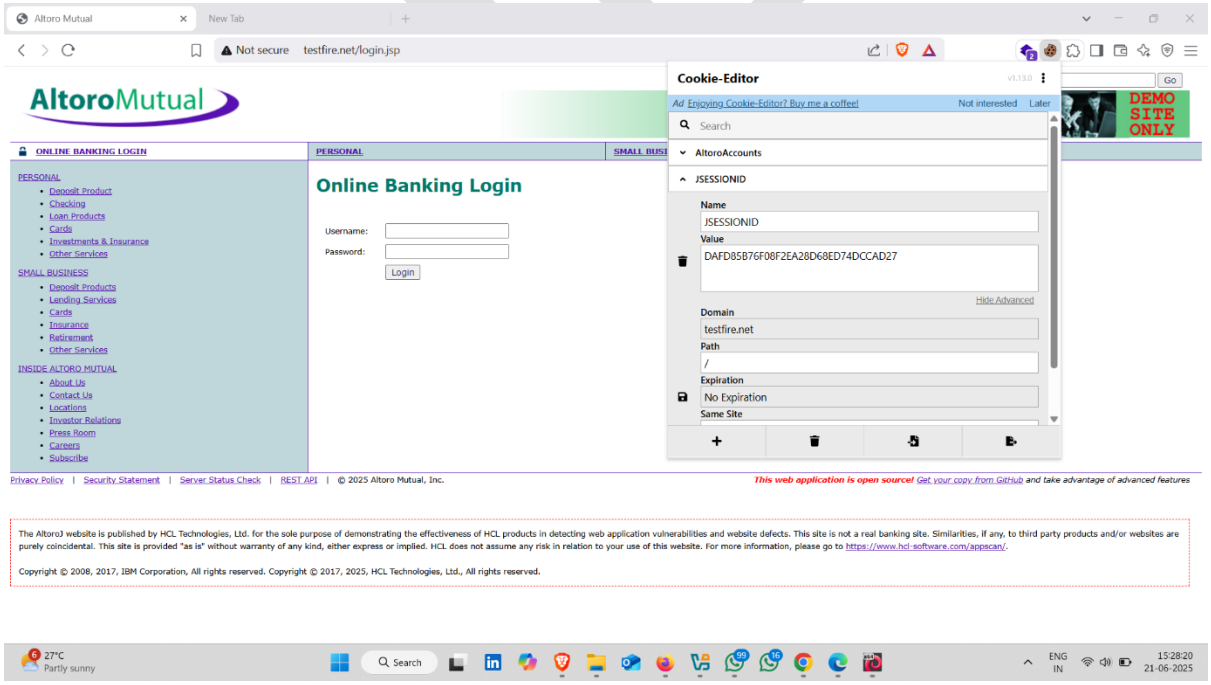
- Now open cookies editor extention and replace this **JSESSIONID**



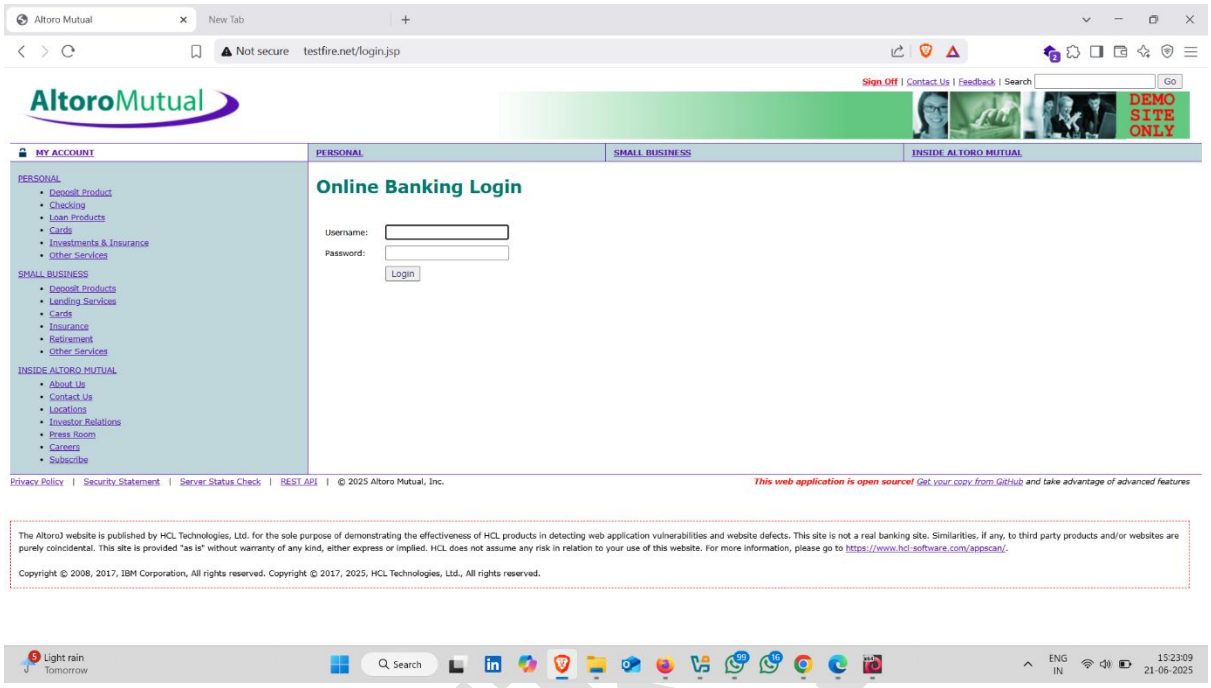
- Paste session Id that you copy



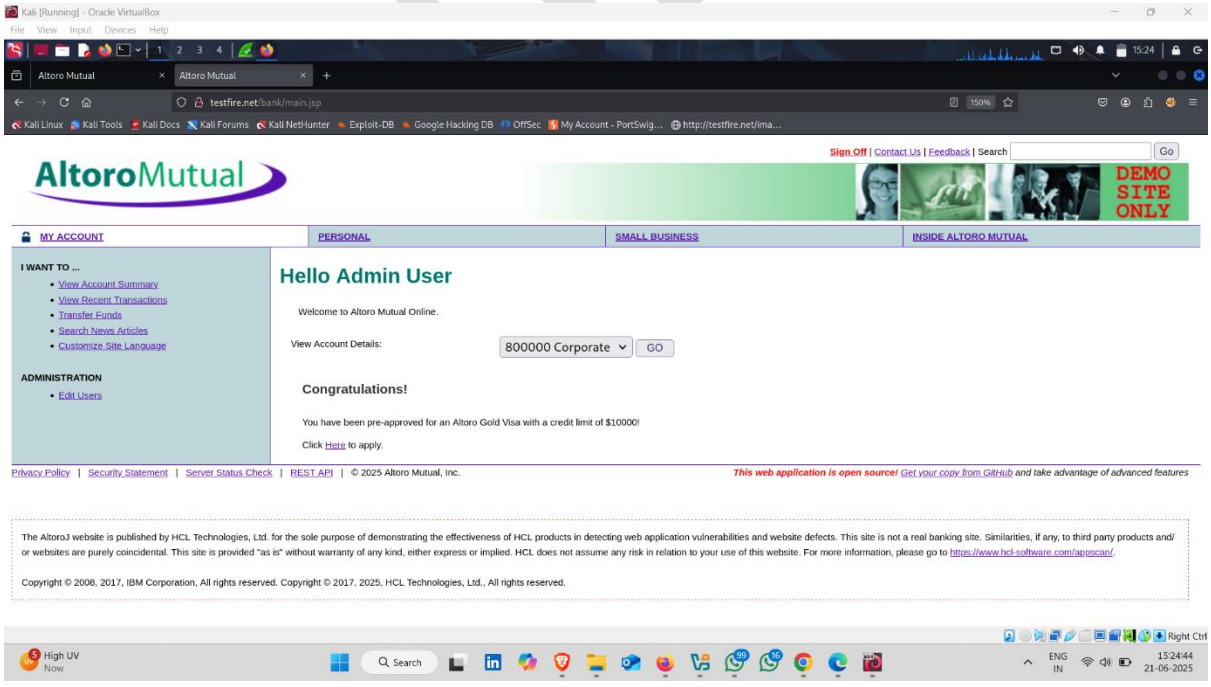
- Click on Save 👇



- 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 ✅ 👇



- Login Successfully 👇 ✅



2.Testing Session ID Predictability with Burp Suite Sequencer

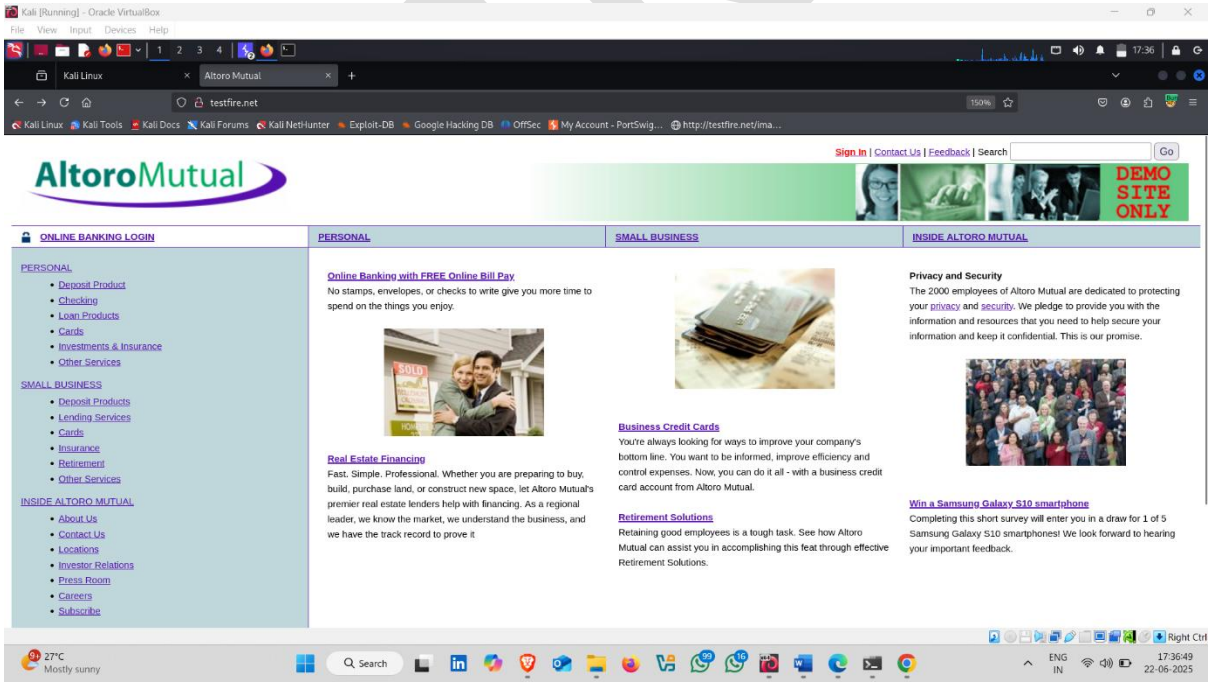
The Burp Suite Sequencer tool is used to assess the strength and unpredictability of session tokens generated by web applications. By capturing multiple session IDs, Sequencer analyzes the randomness and entropy to determine if the tokens can be predicted by an attacker.

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.

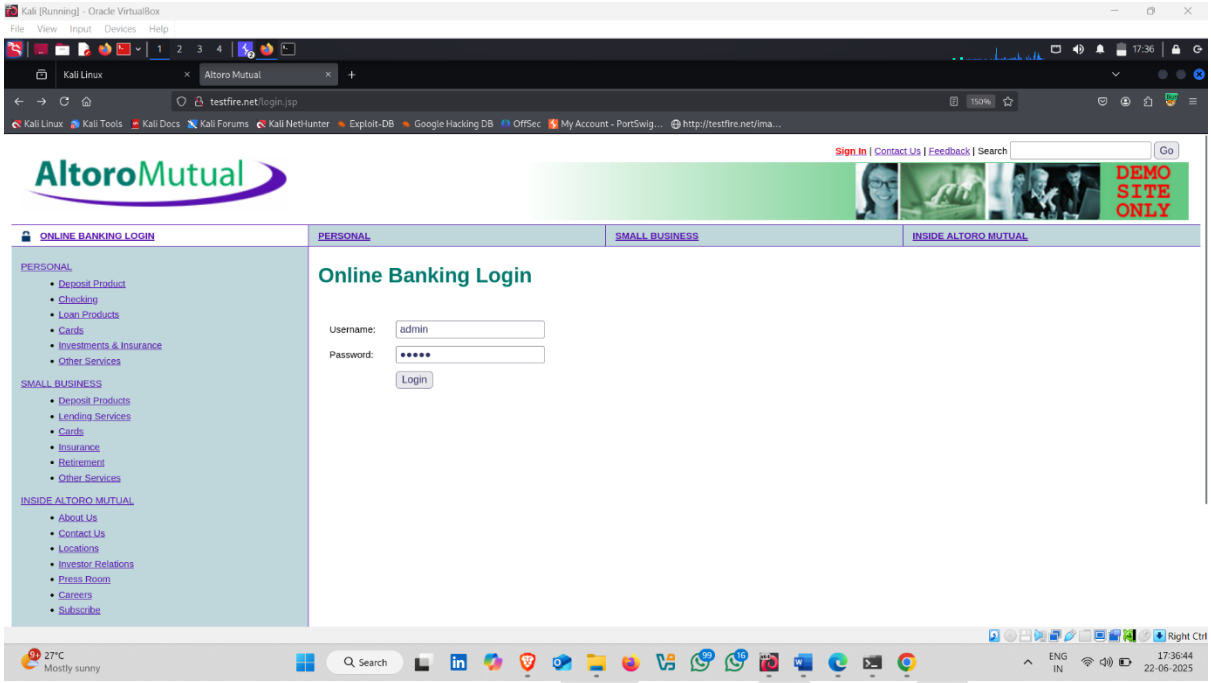
This analysis helps ensure that the session management mechanism is secure and resistant to token prediction attacks.

How to use it -:

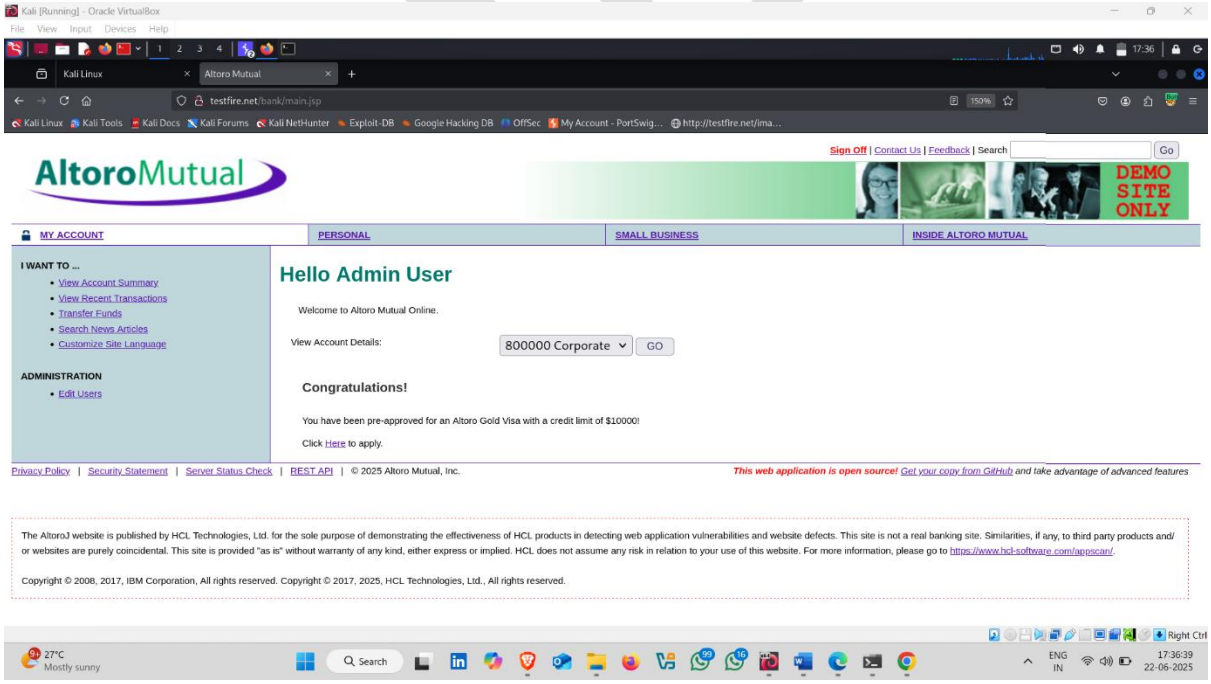
- Target Website



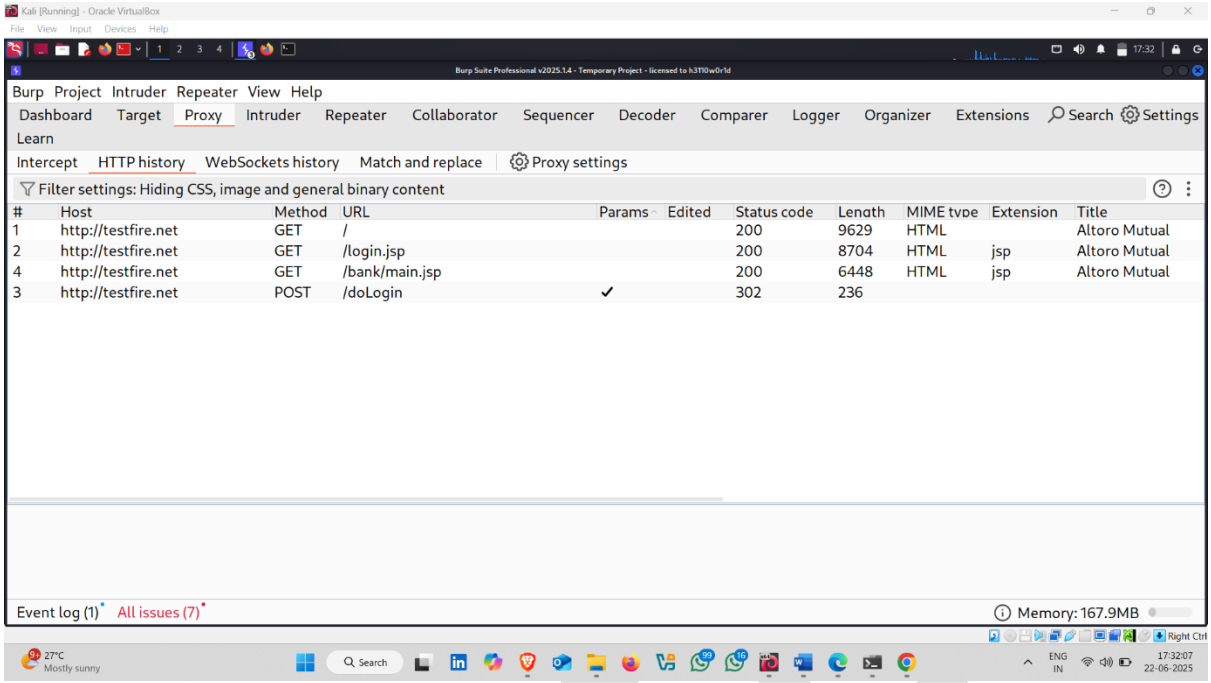
- Enter username and password and click on login



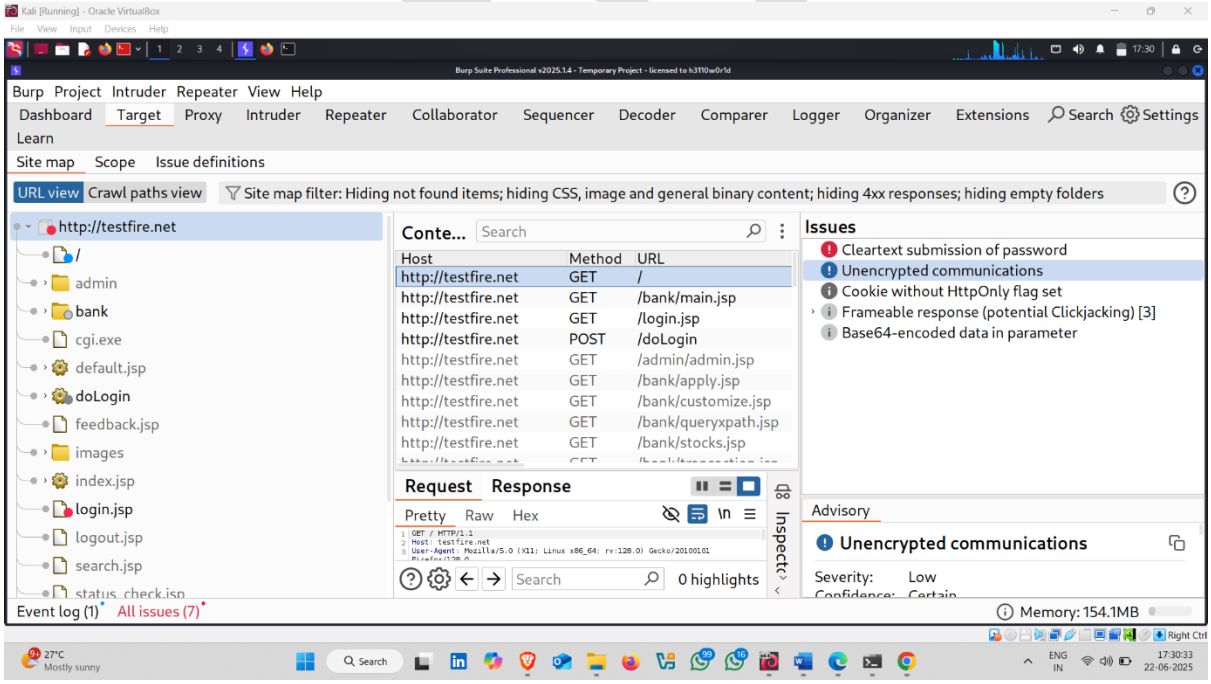
- Account Login ✅ 👇



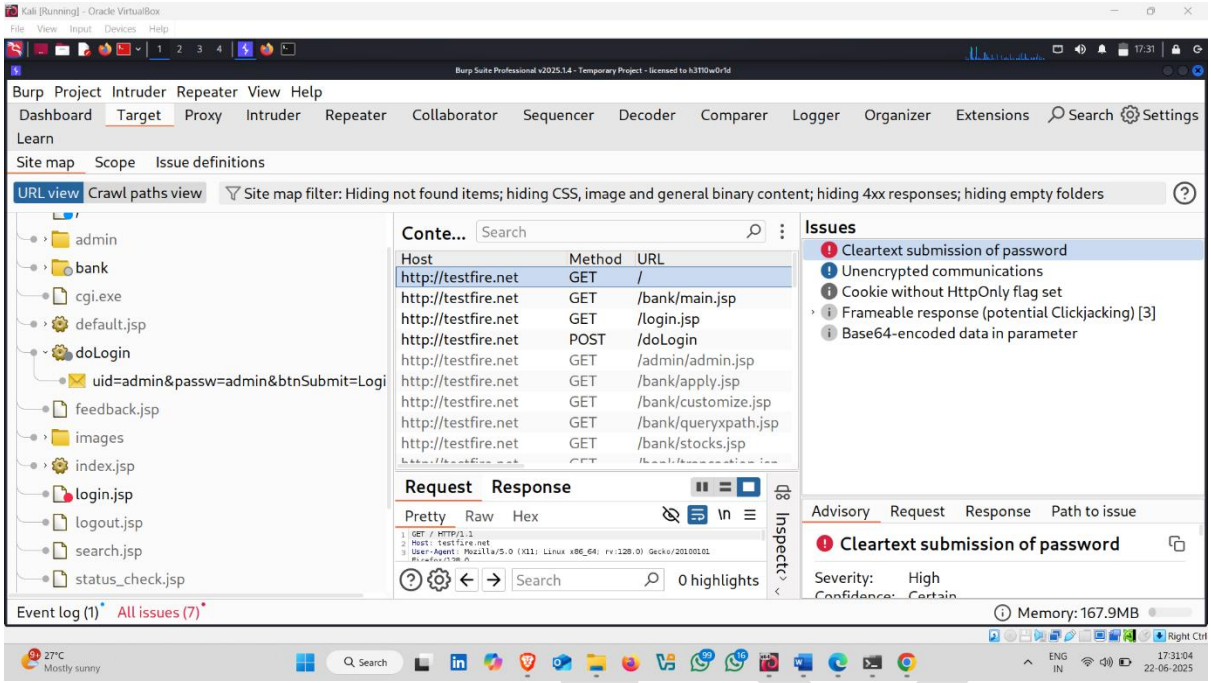
- Open **burp suite** and click on **target** option



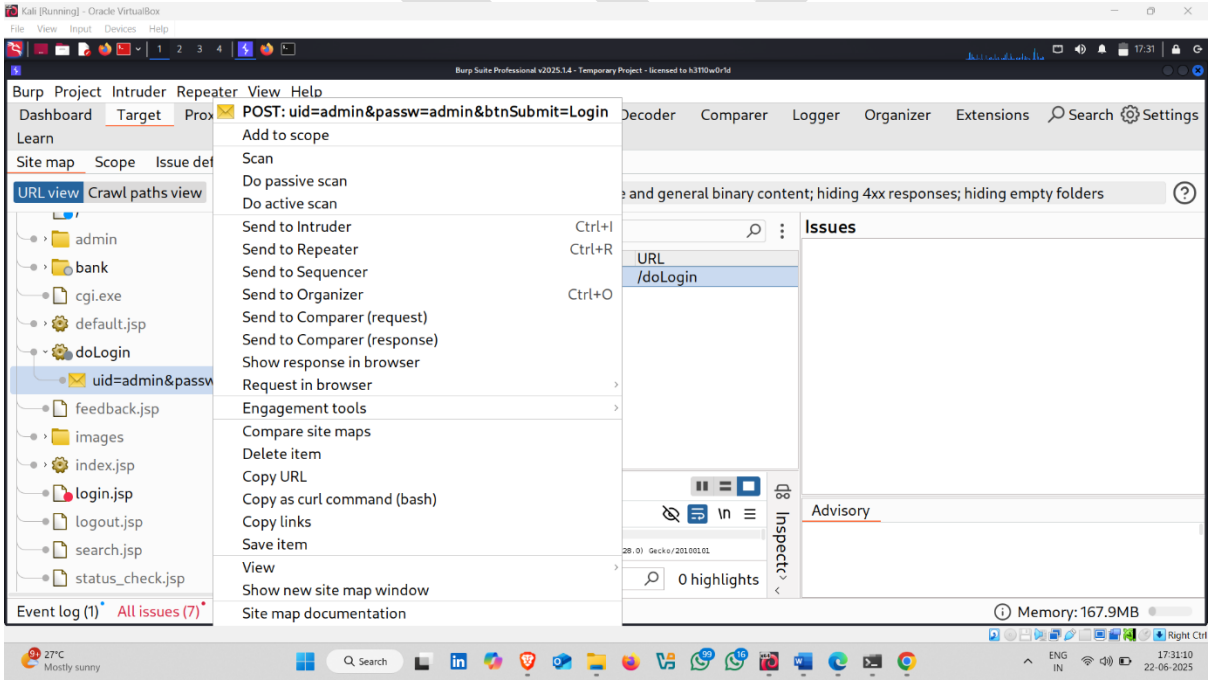
- Now click on **doLogin** Option



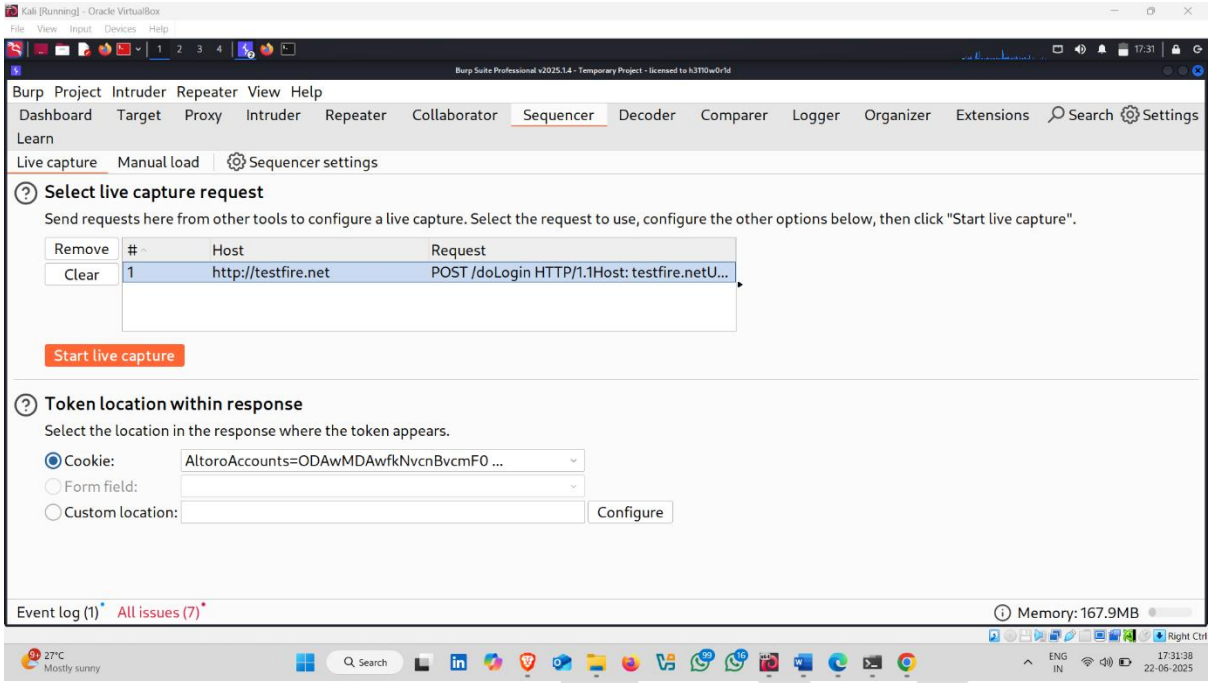
- Now , **right click on uid and passw section**



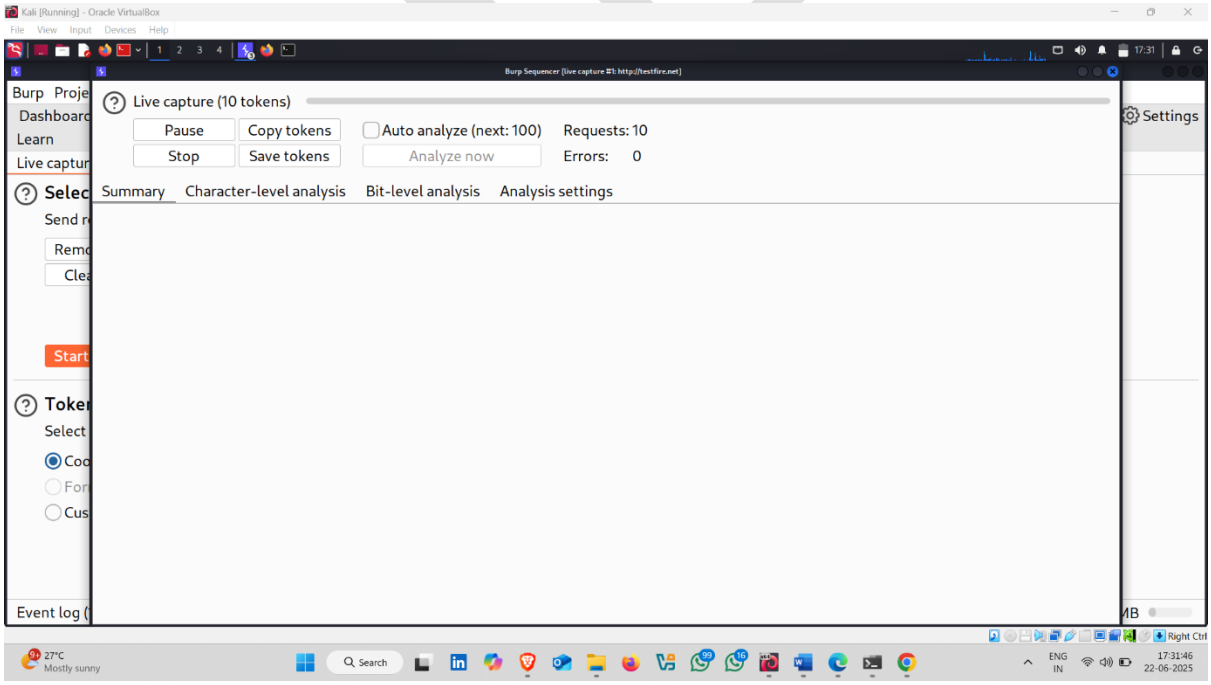
- Send to **Sequencer**



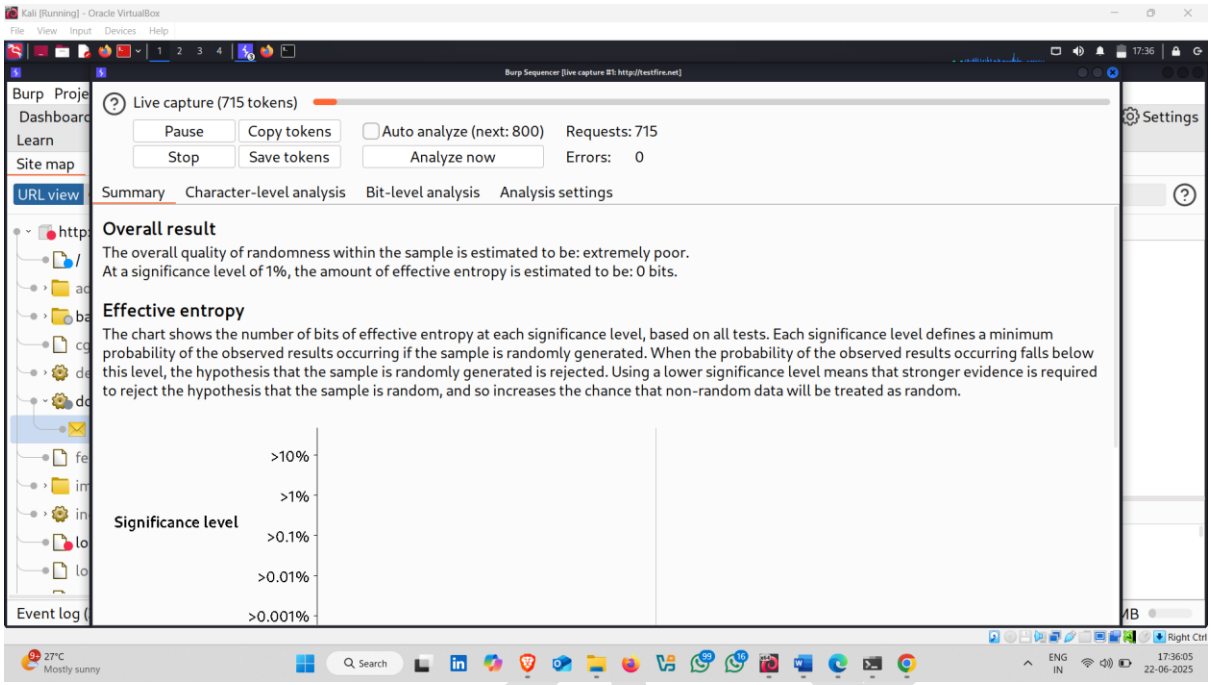
- Click on **Start live capture**



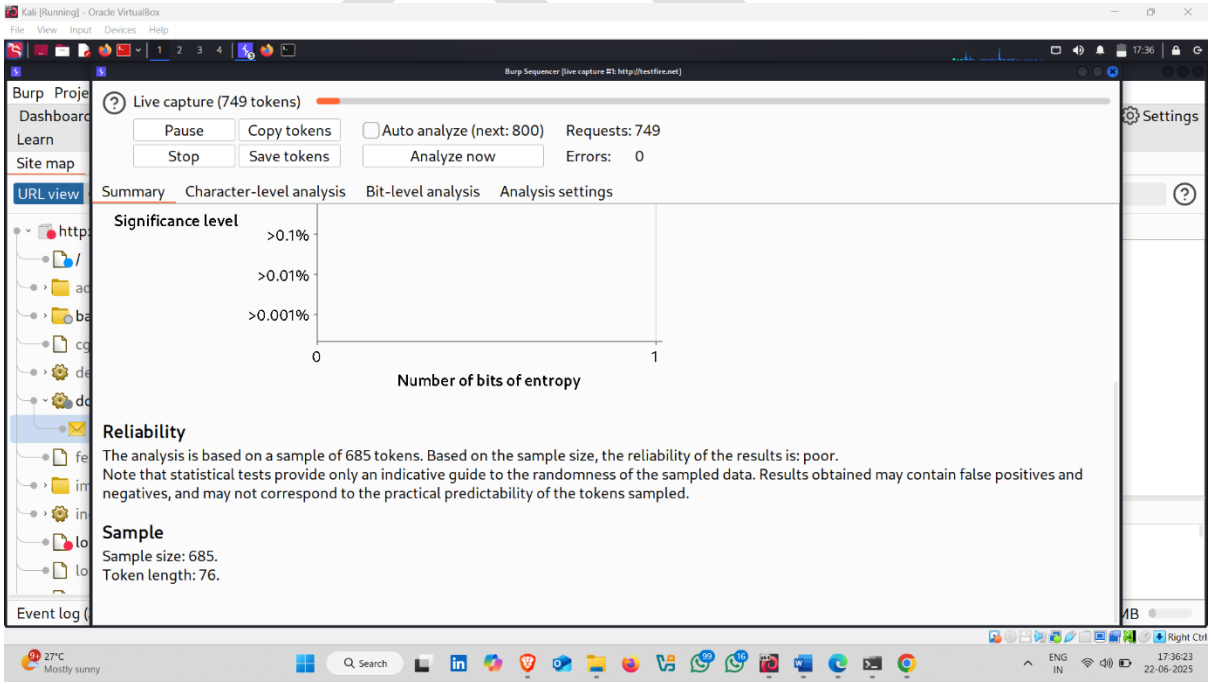
- Started 👇 ✅



- 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:**

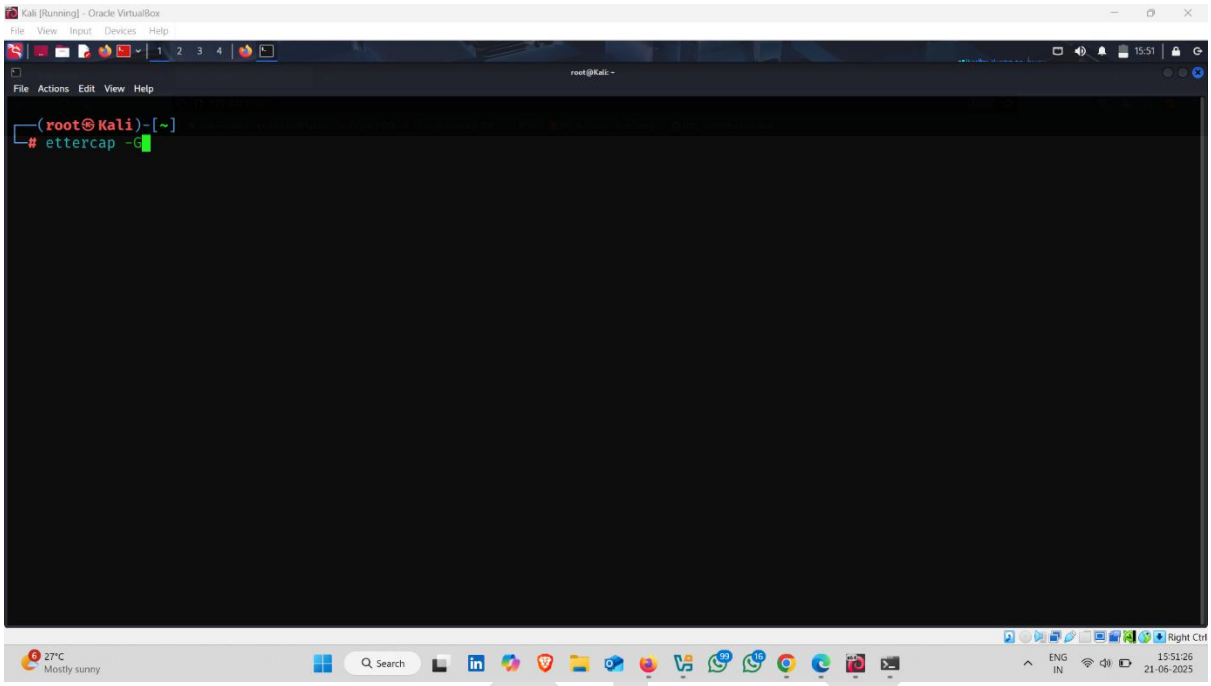
- **ARP Spoofing:** Redirects network traffic through the attacker's system.
- **Packet Sniffing:** Captures data flowing across the network.
- **Session Hijacking:** Takes over active sessions by stealing session cookies.
- **Protocol Dissection:** Understands and displays protocol-level details.
-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

✅ **Ettercap in Session Hijacking:**

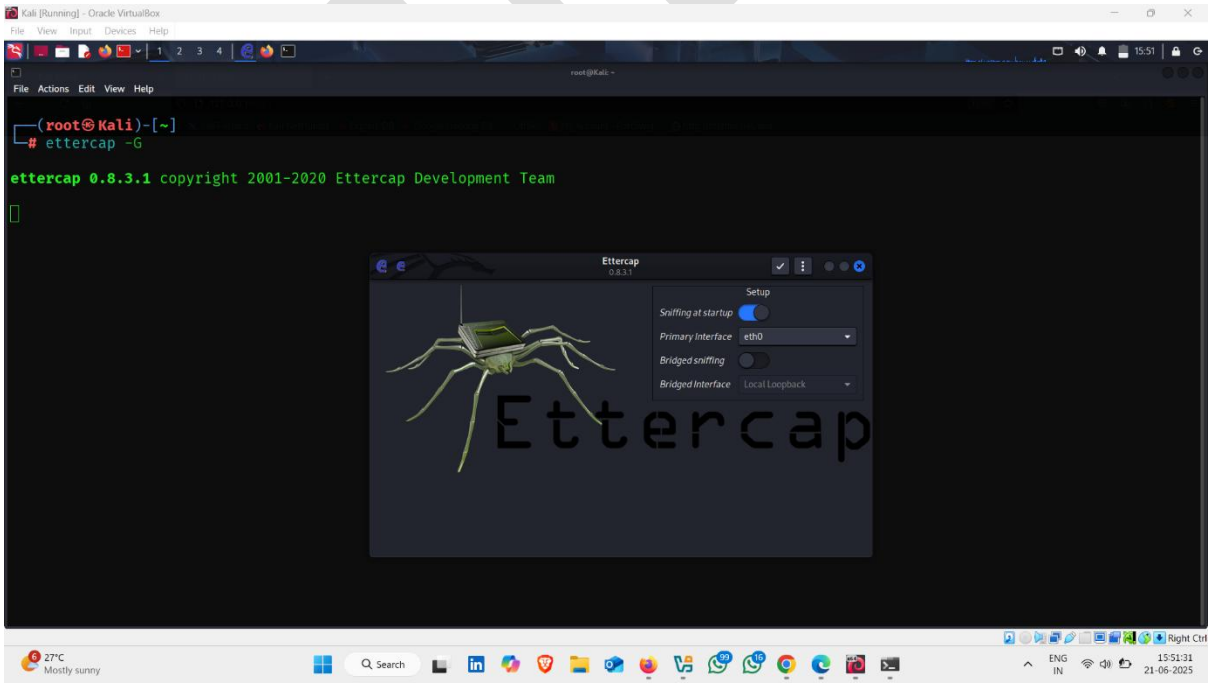
✔ **Step 1: Attacker uses ARP spoofing to become the Man-in-the-Middle.**
✔ **Step 2: Victim browses an HTTP/HTTPS website and authenticates.**
✔ **Step 3: Ettercap captures the session cookies or authentication tokens.**
✔ **Step 4: Attacker** uses the captured cookies to hijack the victim's active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

How to use it -:

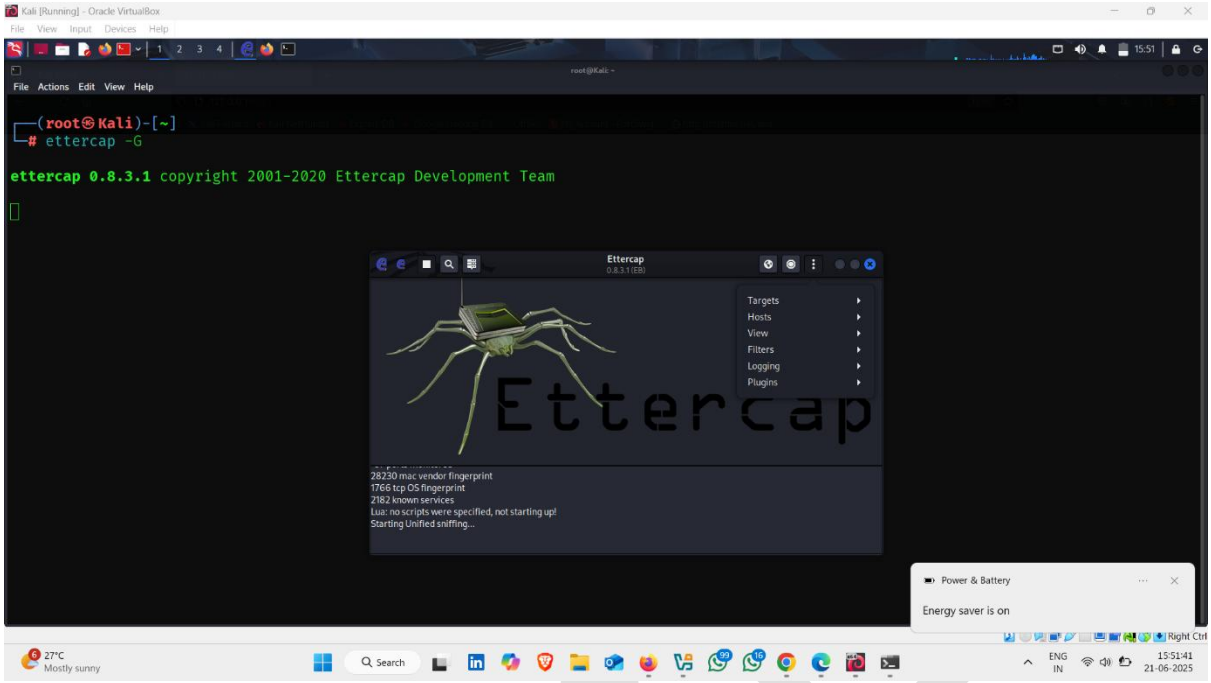
- Open Ettercap GUI using this command



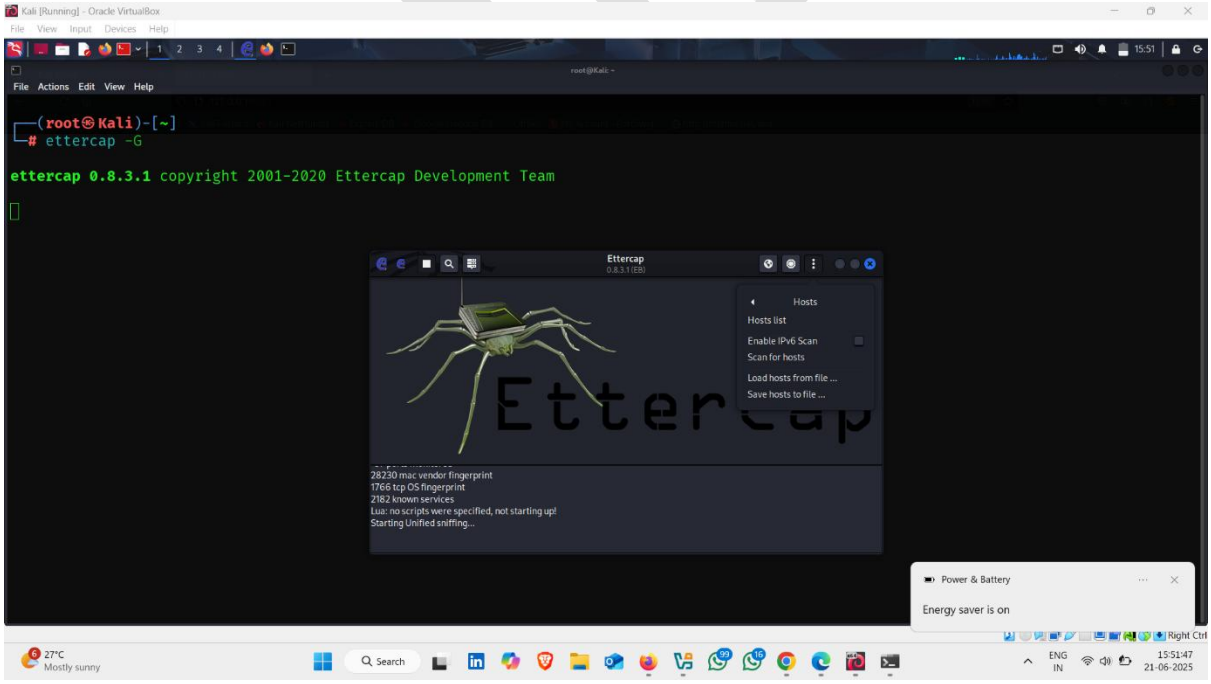
- **Click on ✅ this icon**



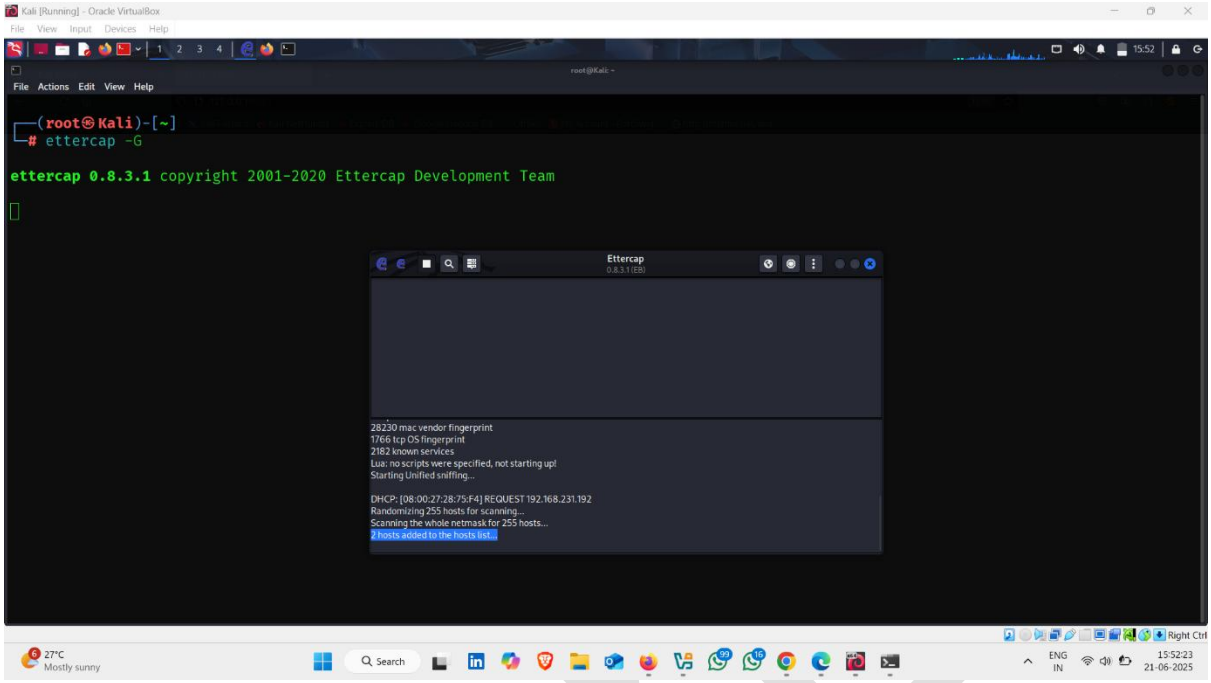
- Now , click on **three dot** and then click on **Hosts**



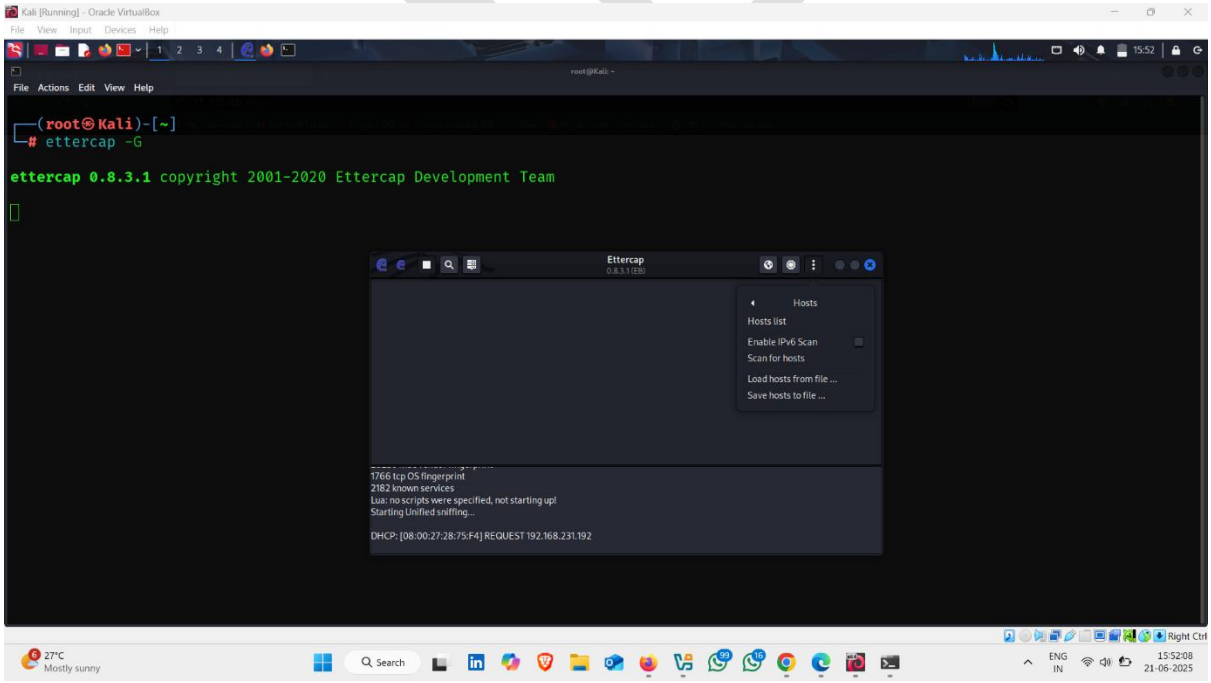
- Click on **Scan For Hosts**



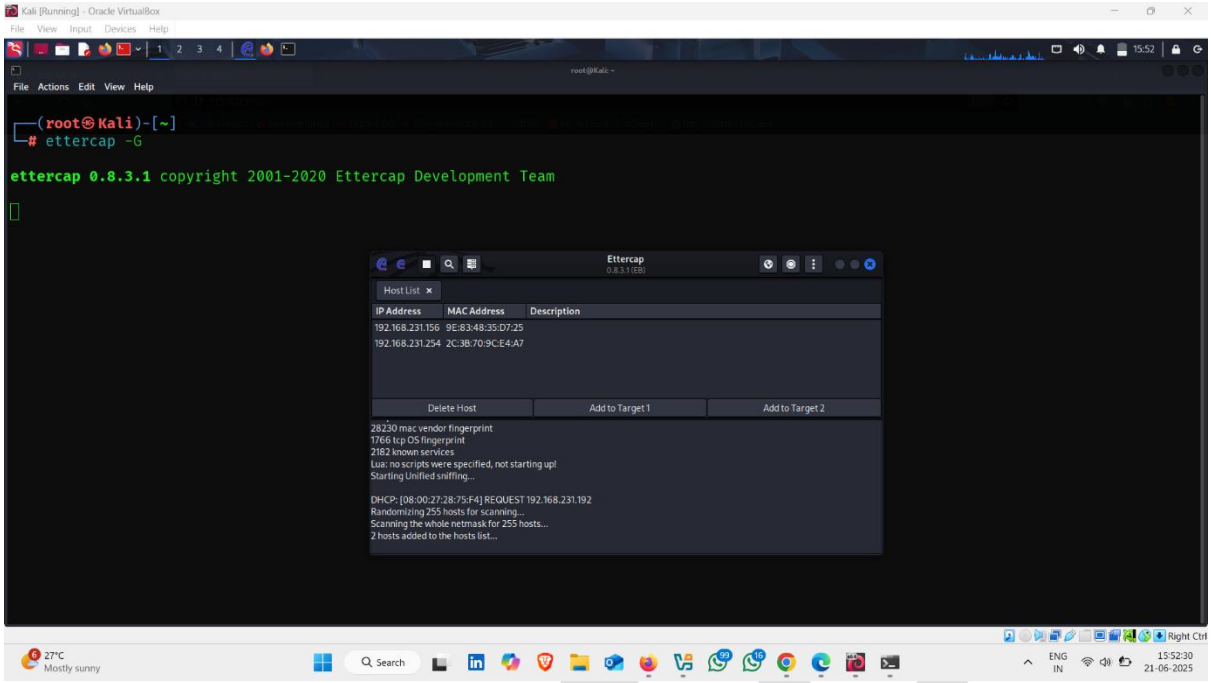
- Two hosts scan ✅ 👇



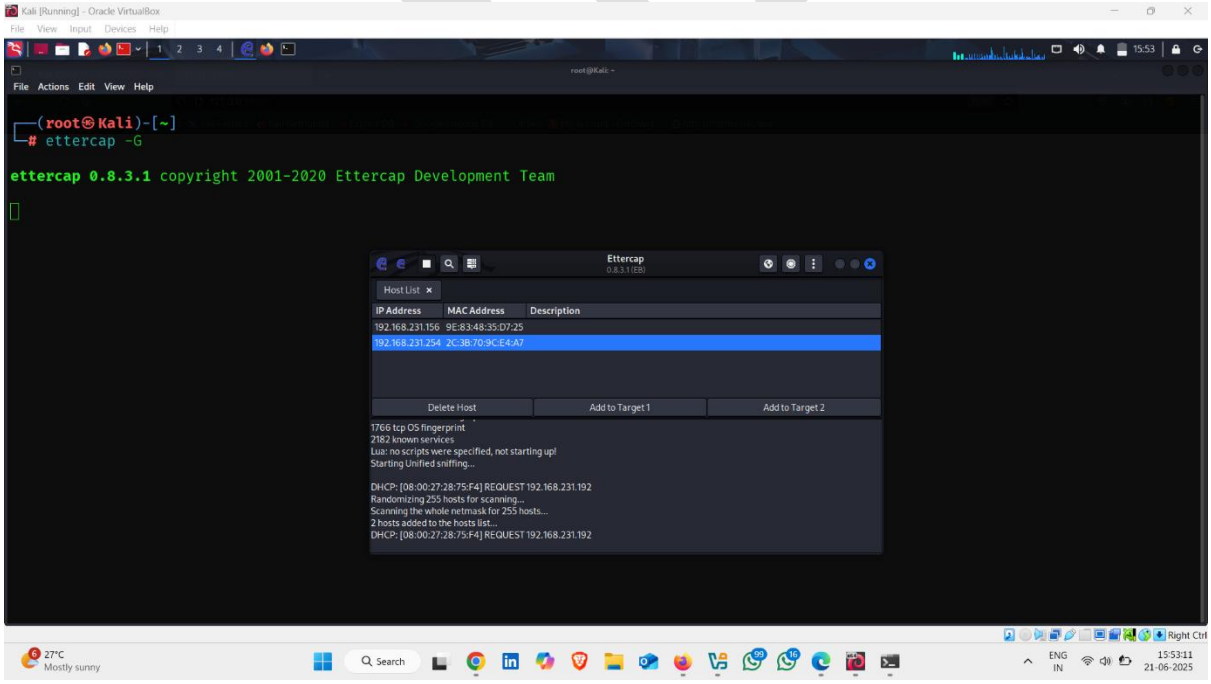
- Once Again click on **Three Dot** and click on **Host List**



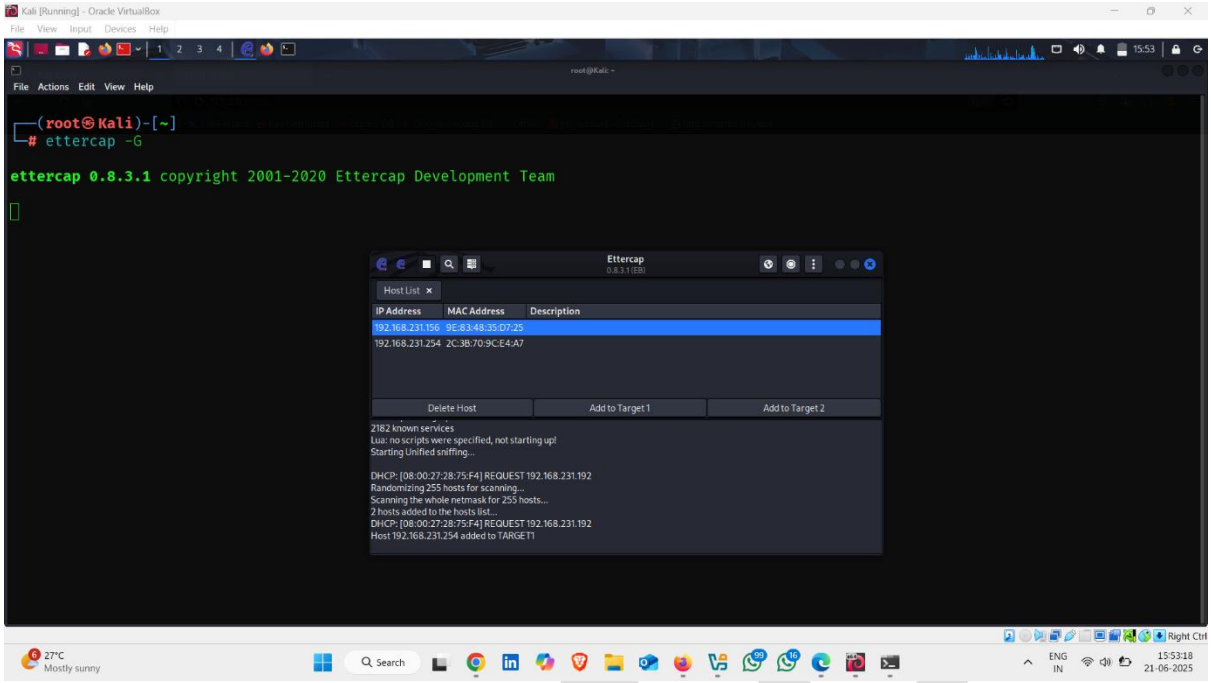
- **Host list appears**



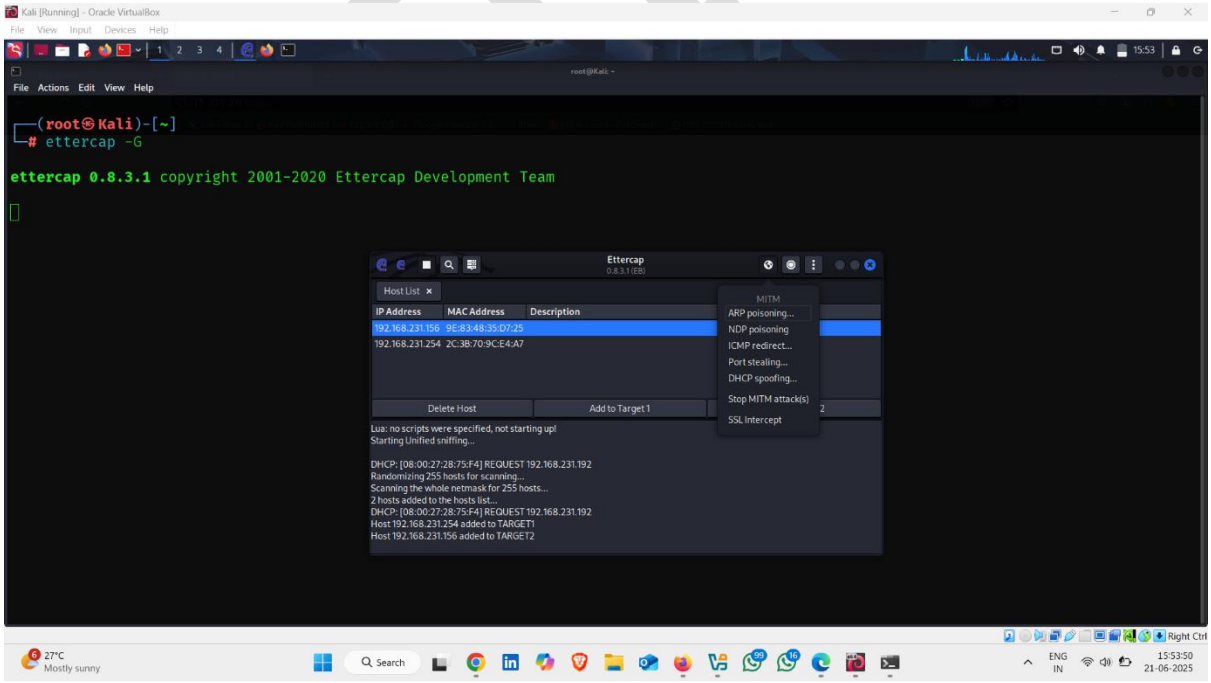
- Click on **target ip address** and **add this ip as target 1**



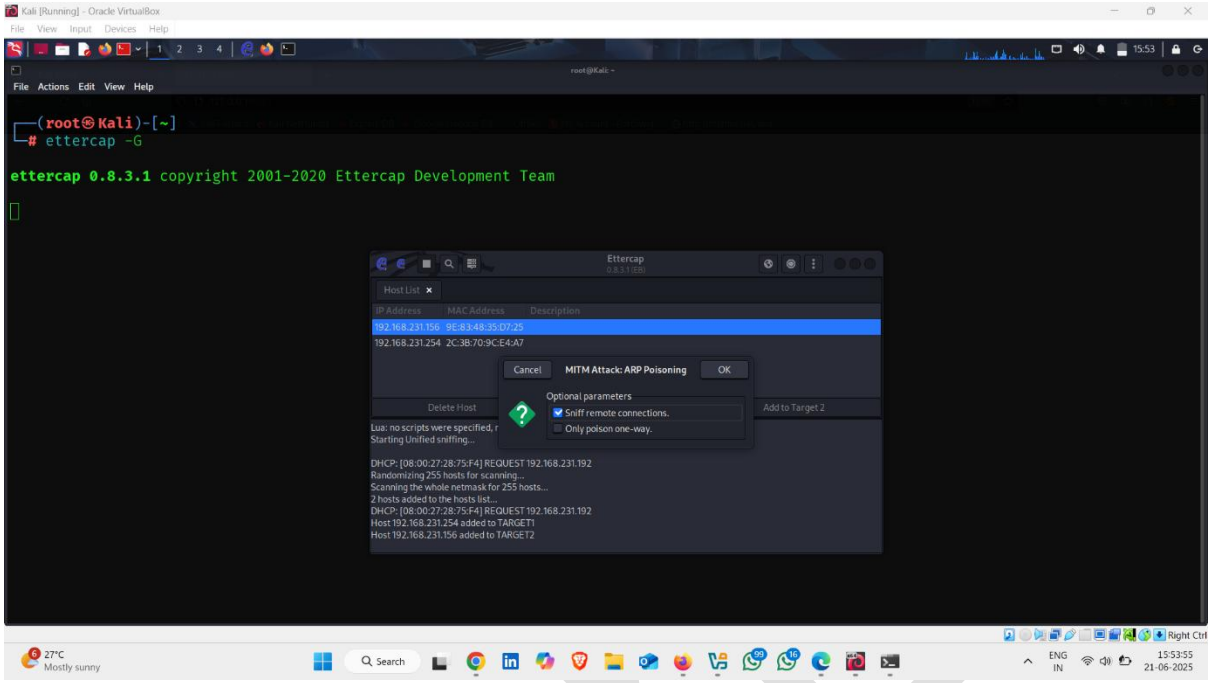
- Now click on **gateway Ip address** and **add this as target 2**



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

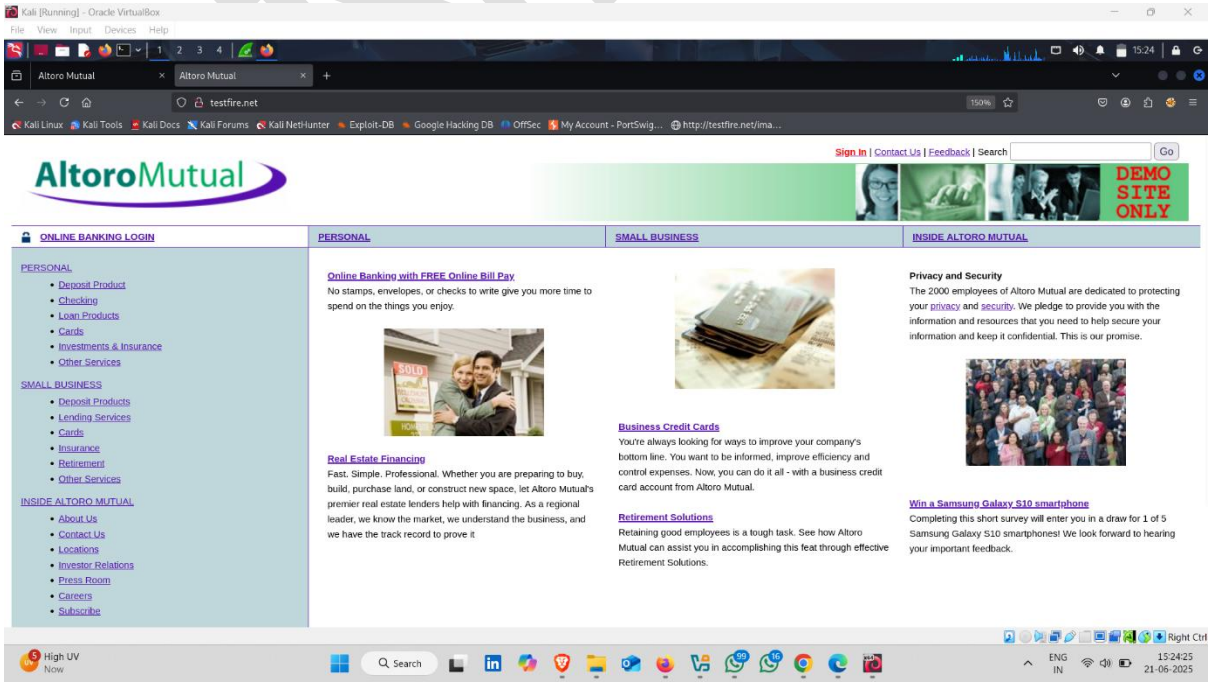


- Check ✅ -- **Sniff Remote Connections** and click on **OK**

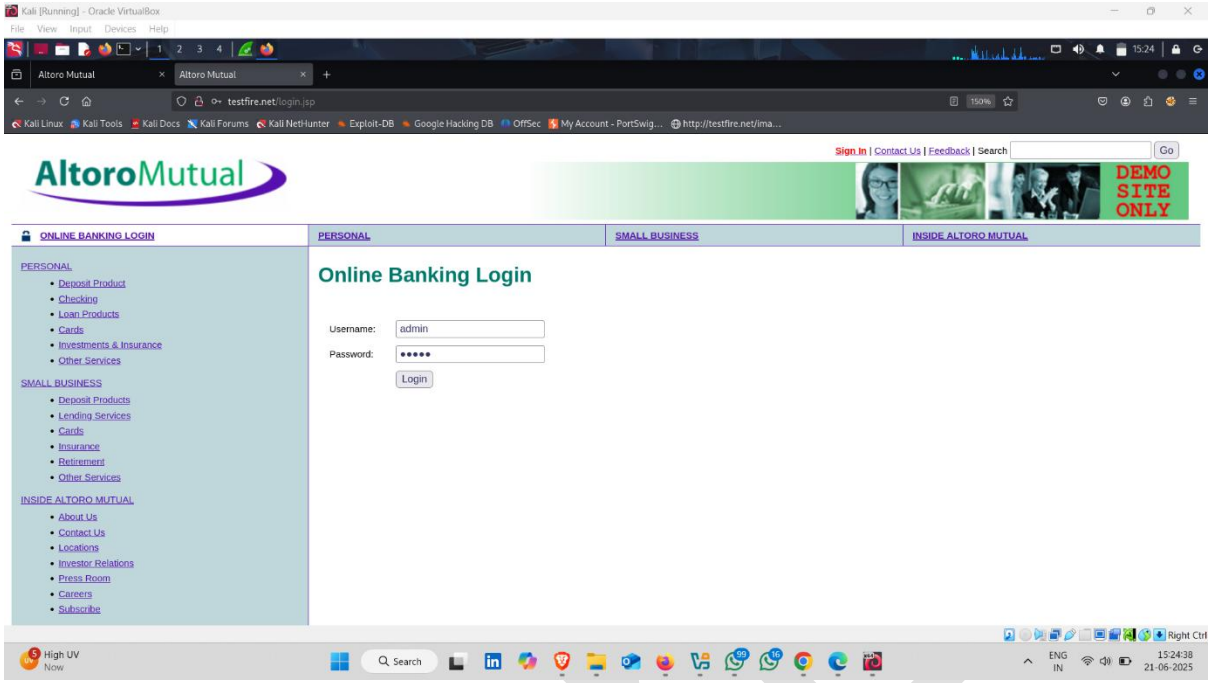


Now , Assume That Your Target will sign in some website and you want to hijack their session and cookies .

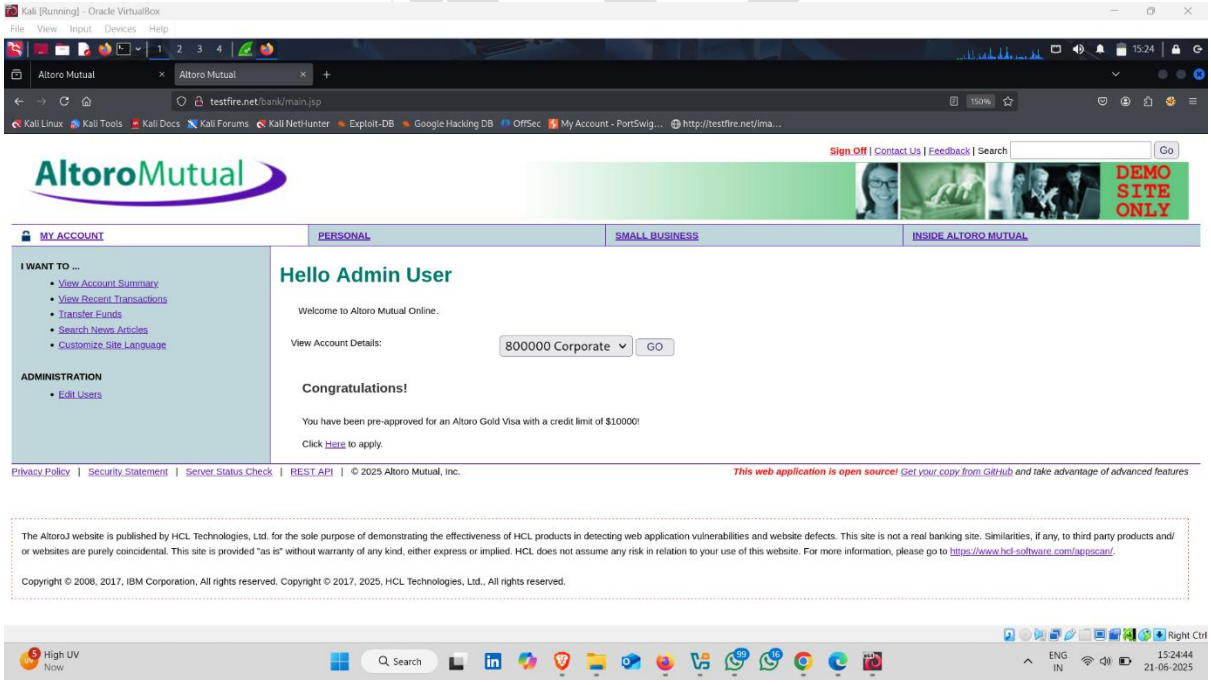
- **Target Website** 👇



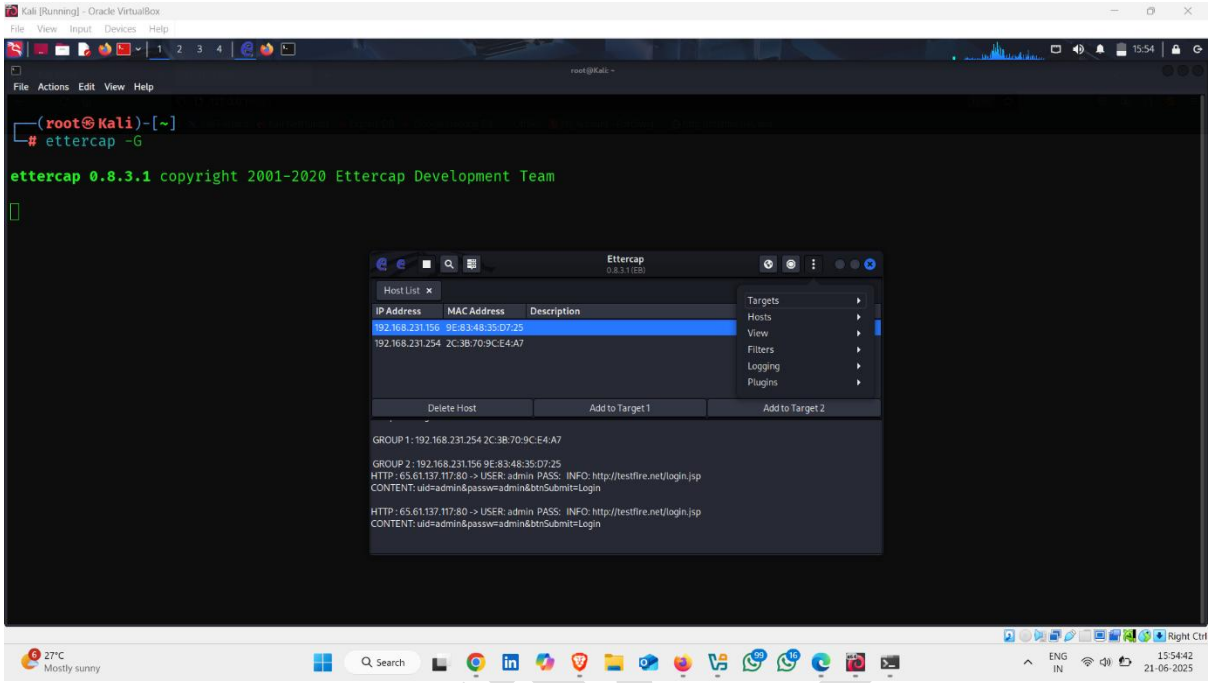
- Login Process 👇



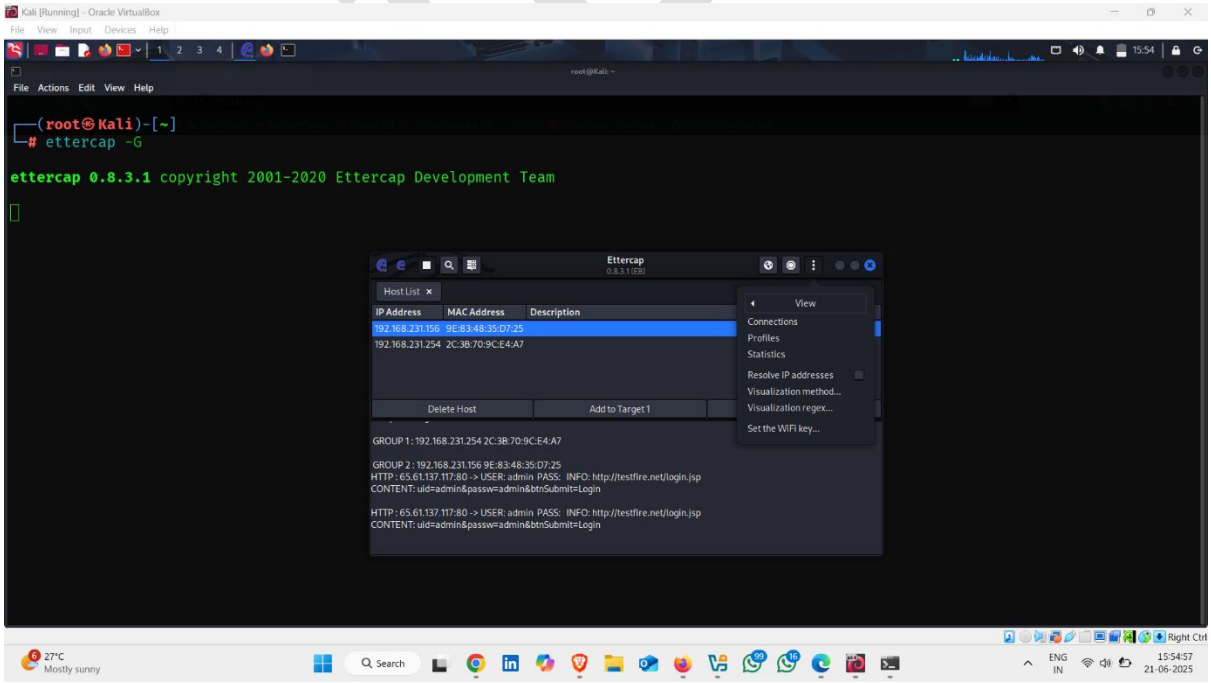
- **Login Successful** 👇



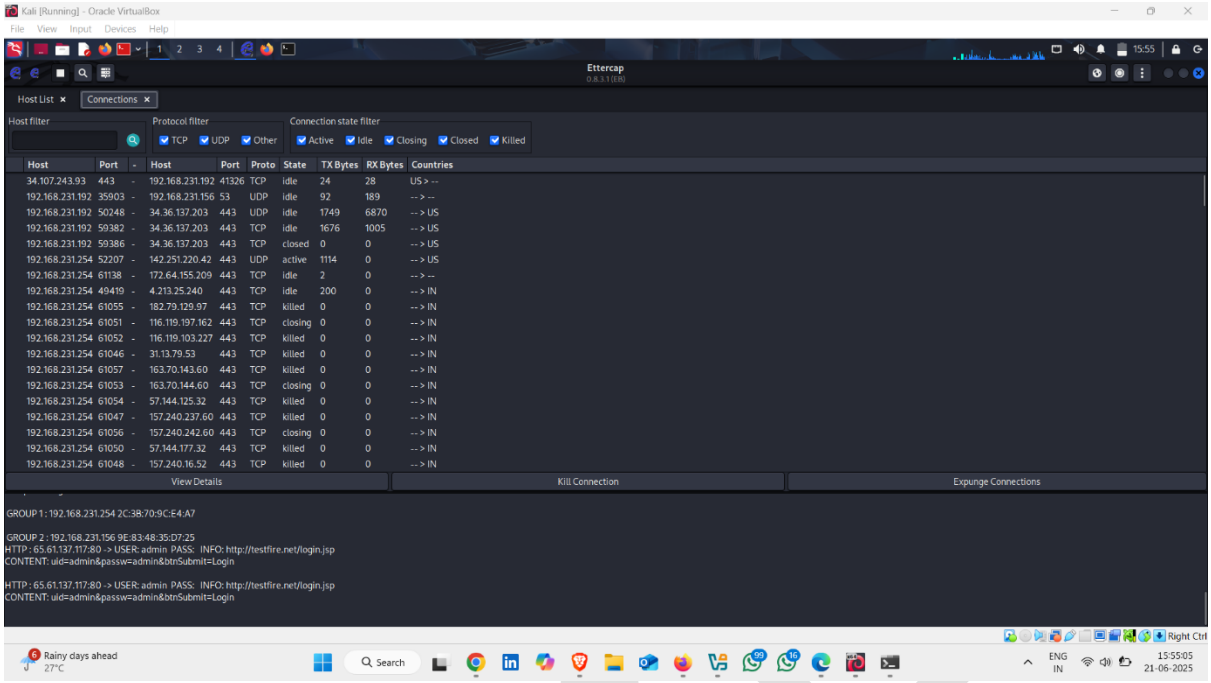
- **Now go to Ettercap interface and check thet capture the request or not**
- **Now ,** Click on Three dot and then click on View



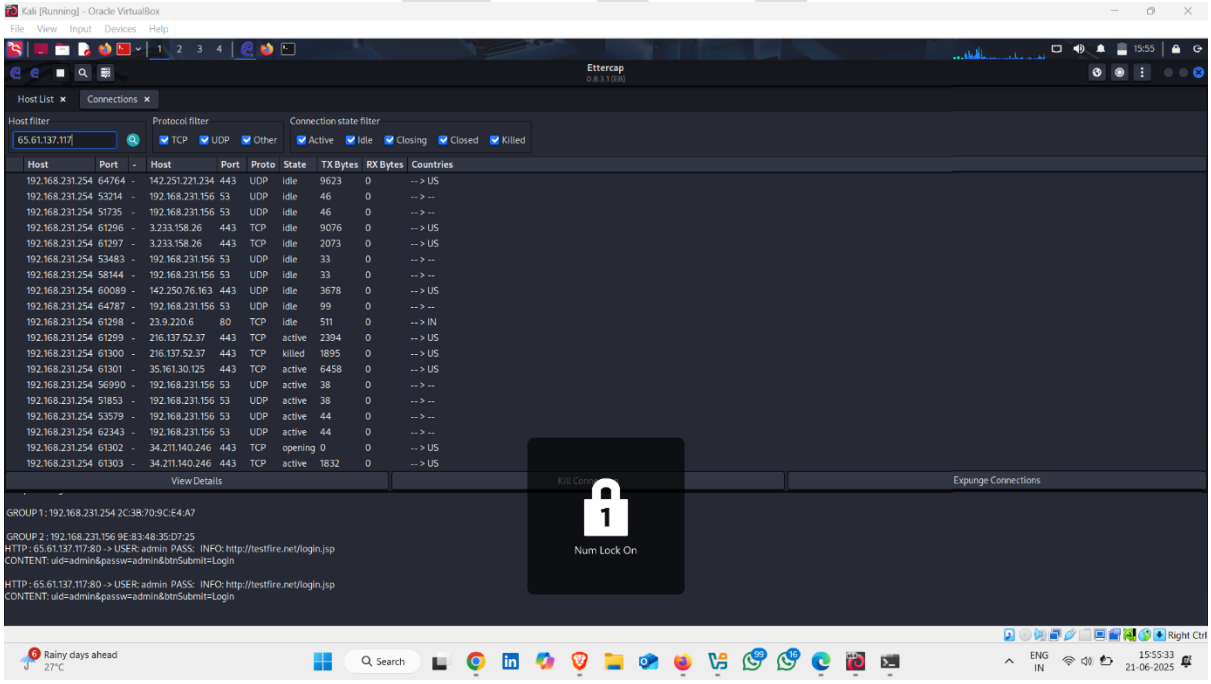
- Click on **Connections**



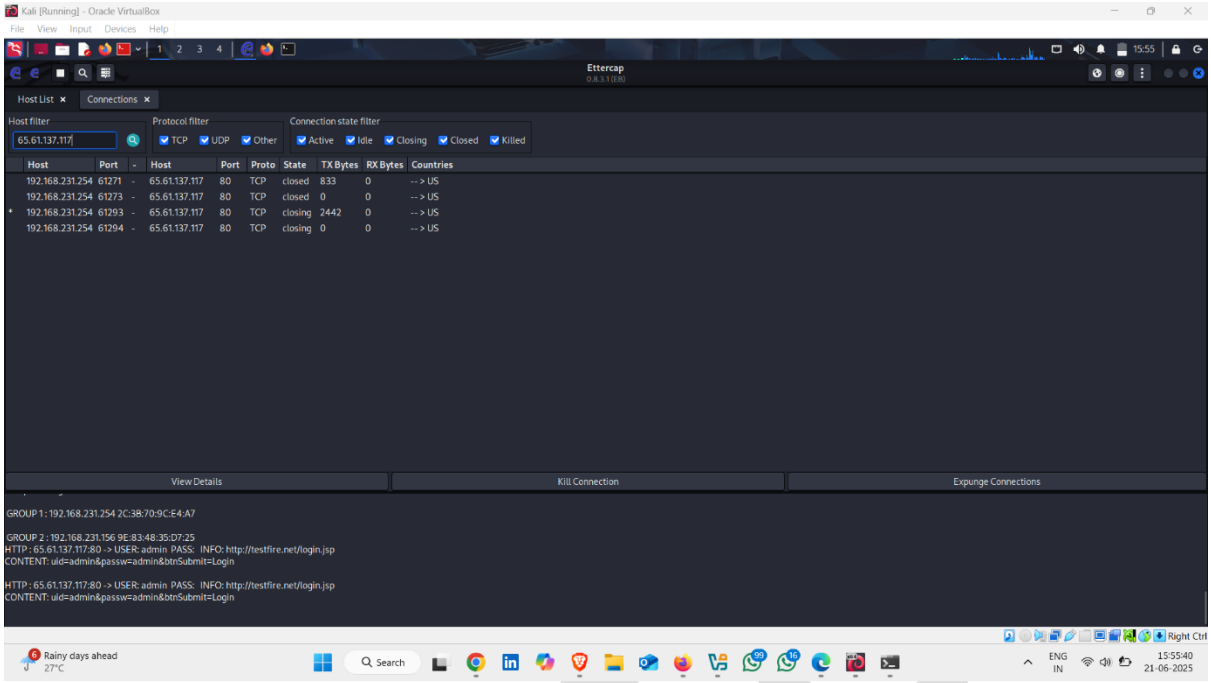
- Here , it capture Many packets



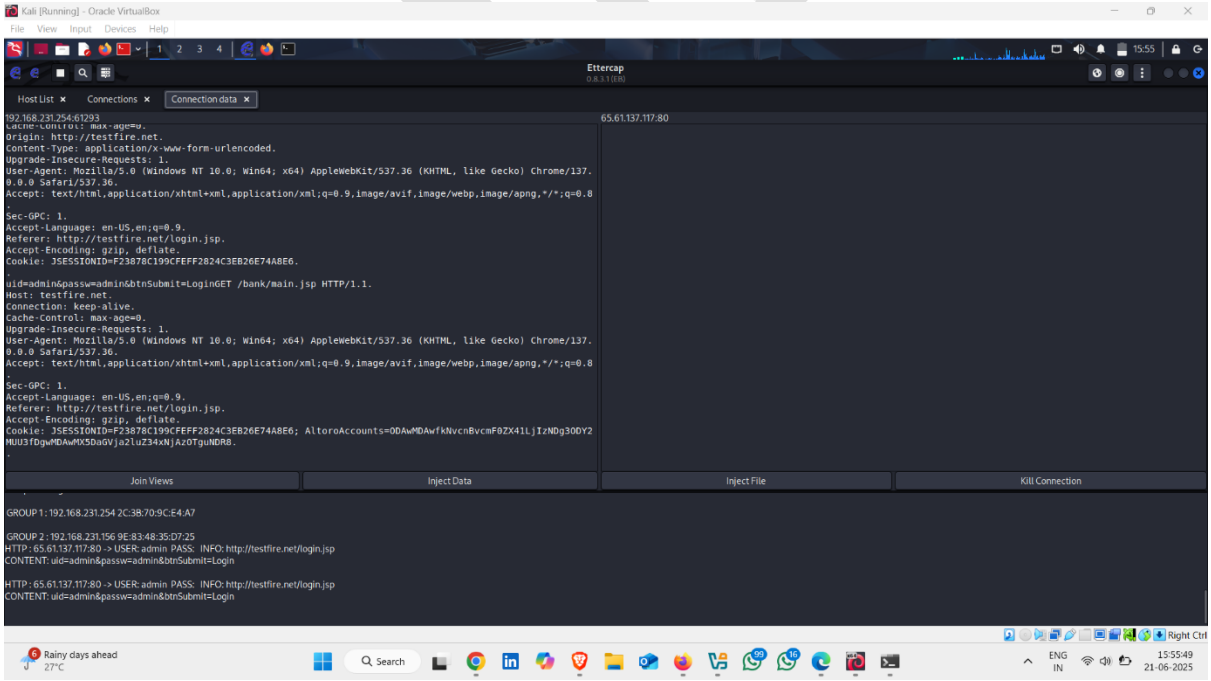
- Search Using Target Ip address



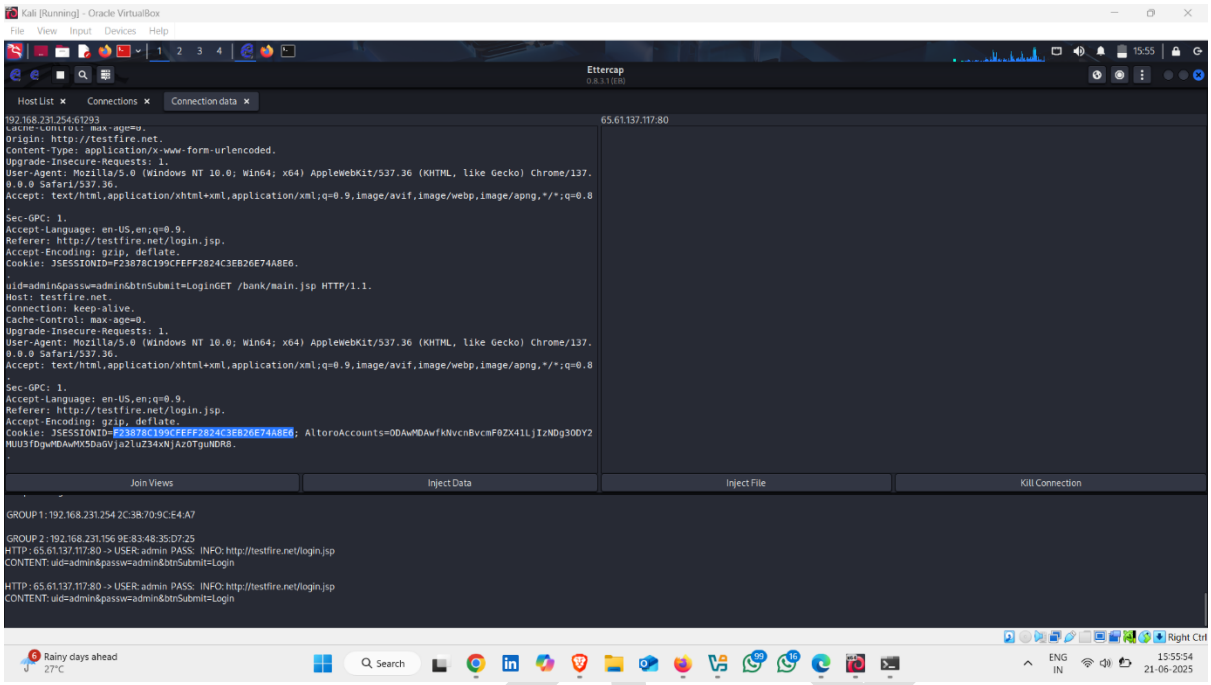
- **Now ,** Open **Port number 80** packet



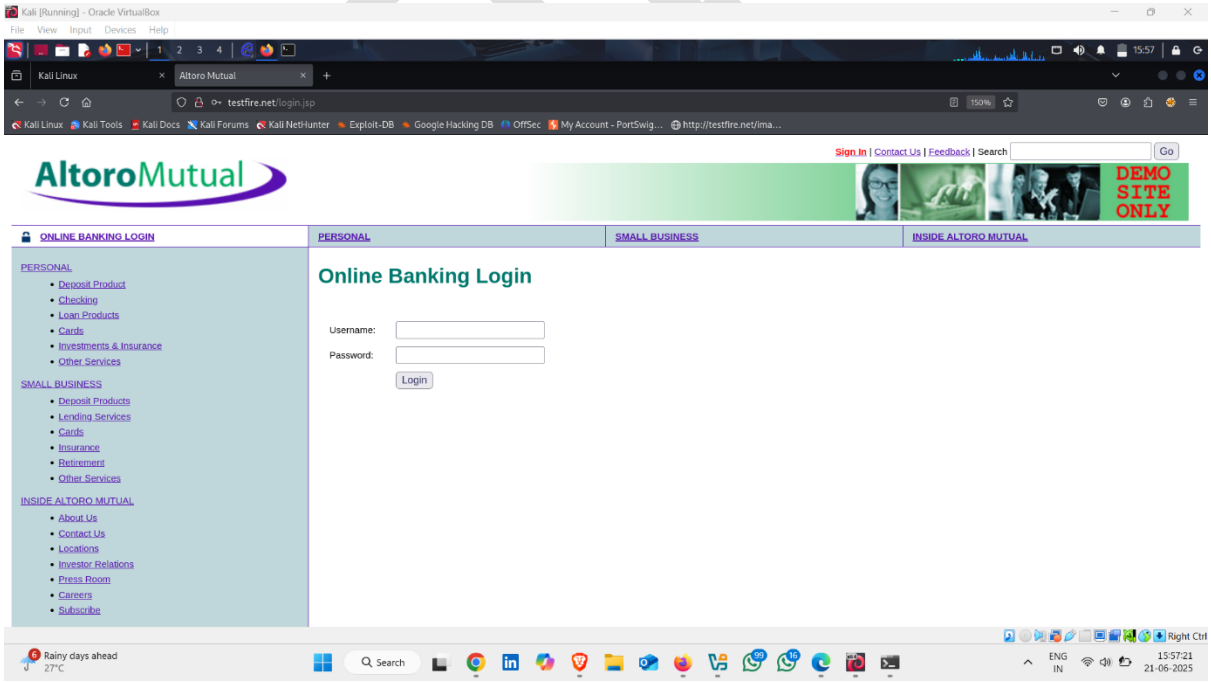
- here , it capture session id and all like website and other things



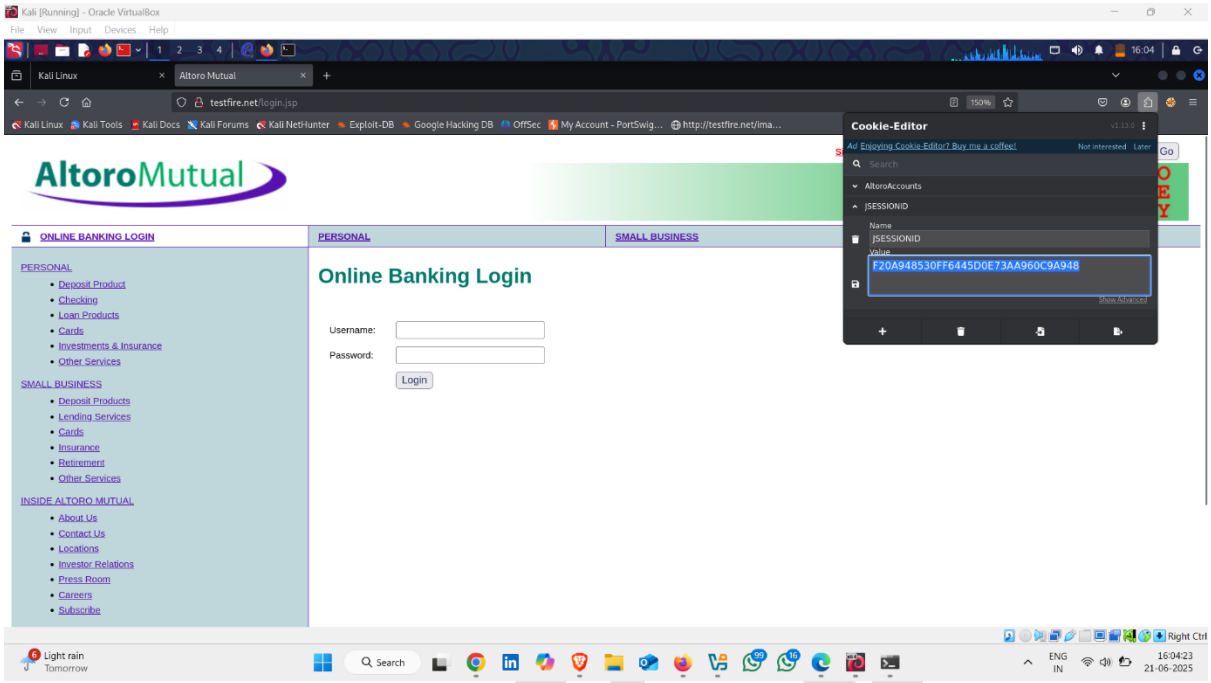
- **Session Id 👇**
- **Now , copy this Session Id**



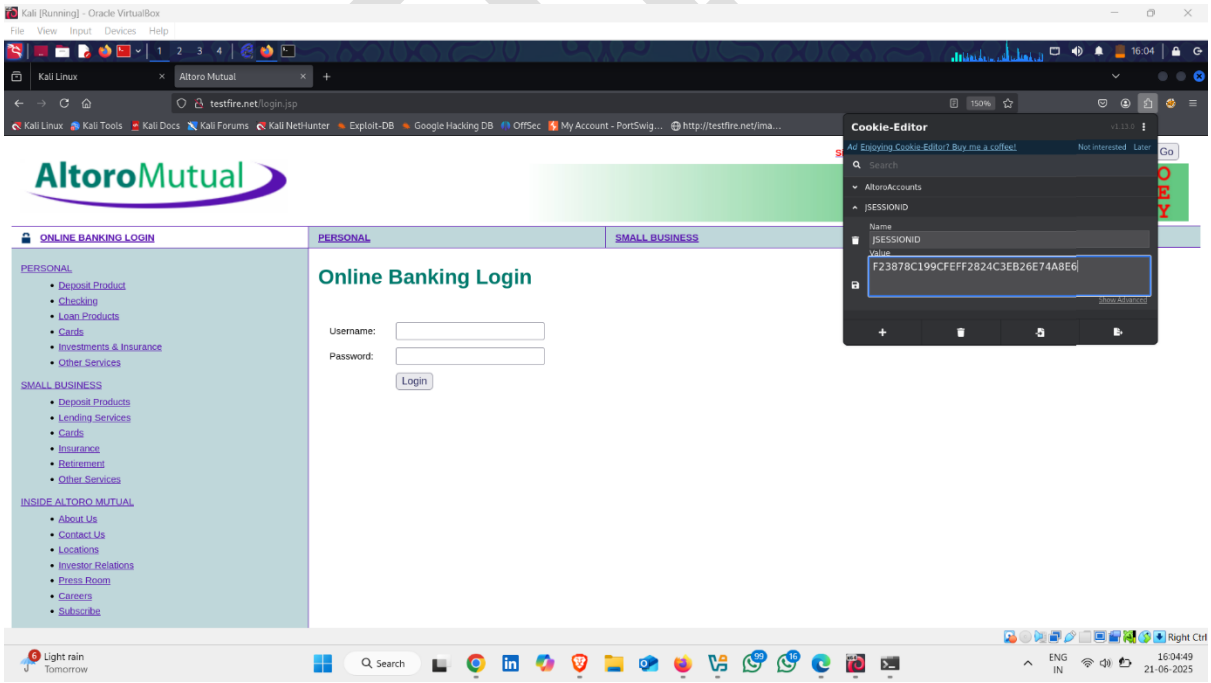
- Open target Website



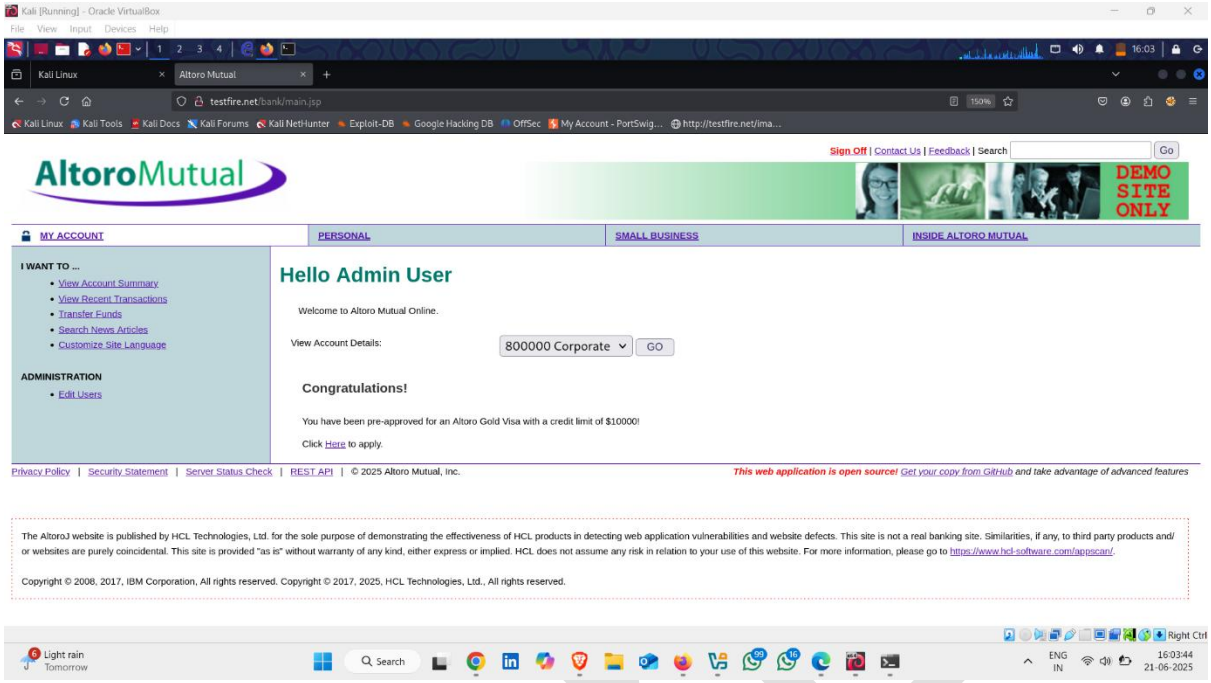
- Go to Extensions and open Cookies Editor
- Replace this session id



- To this , that you copy from Ettercap 👇



- Login 👇



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:**

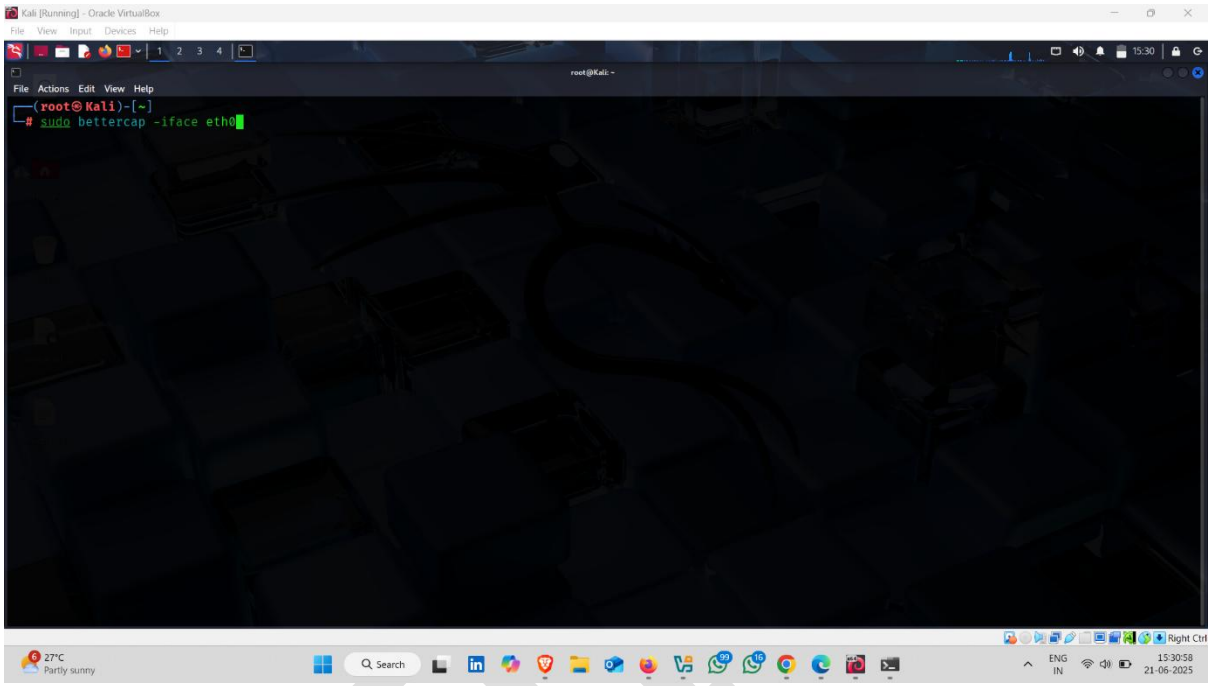
✔ **Step 1:** ARP spoofing to position attacker in MITM.
✔ **Step 2:** Start live traffic sniffing.
✔ **Step 3:** Use SSLStrip to force HTTP traffic (if required).
✔ **Step 4:** Capture session cookies or login tokens.
✔ **Step 5:** Use the cookies to hijack active sessions.

How To Use It -:

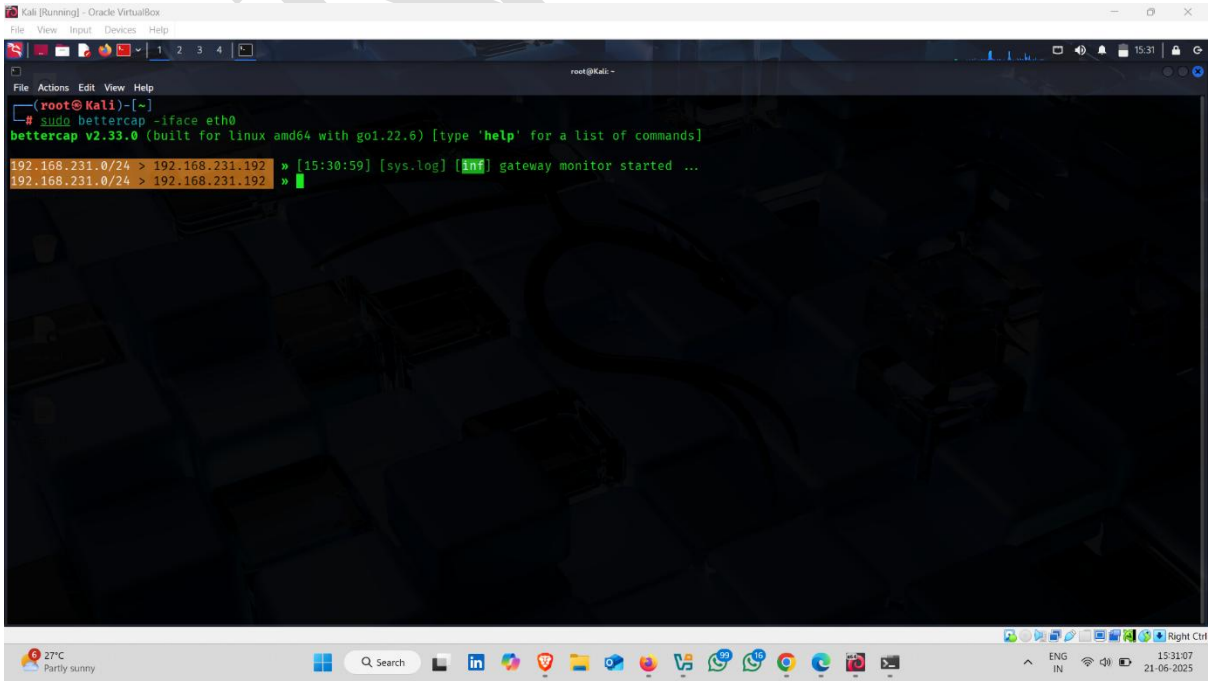
- Start bettercap Using This Command

Command – sudo bettercap -iface eth0

Explanation -: Start Bettercap on the specified interface

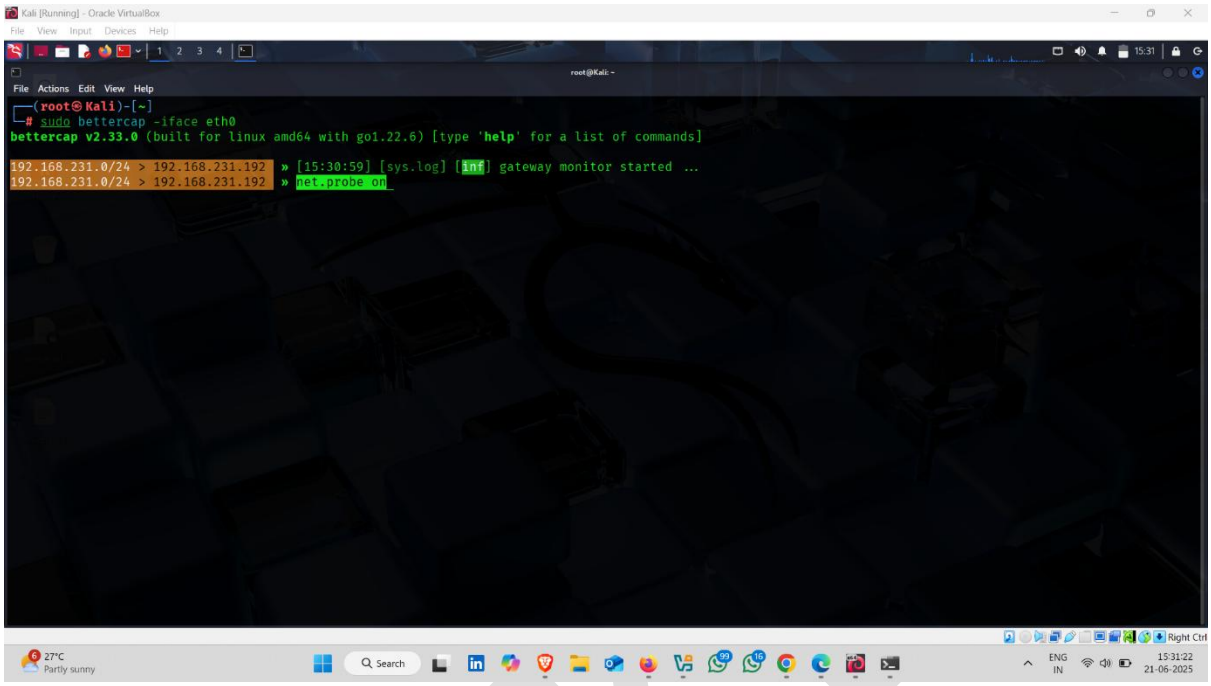


- Bettercap Started 👇

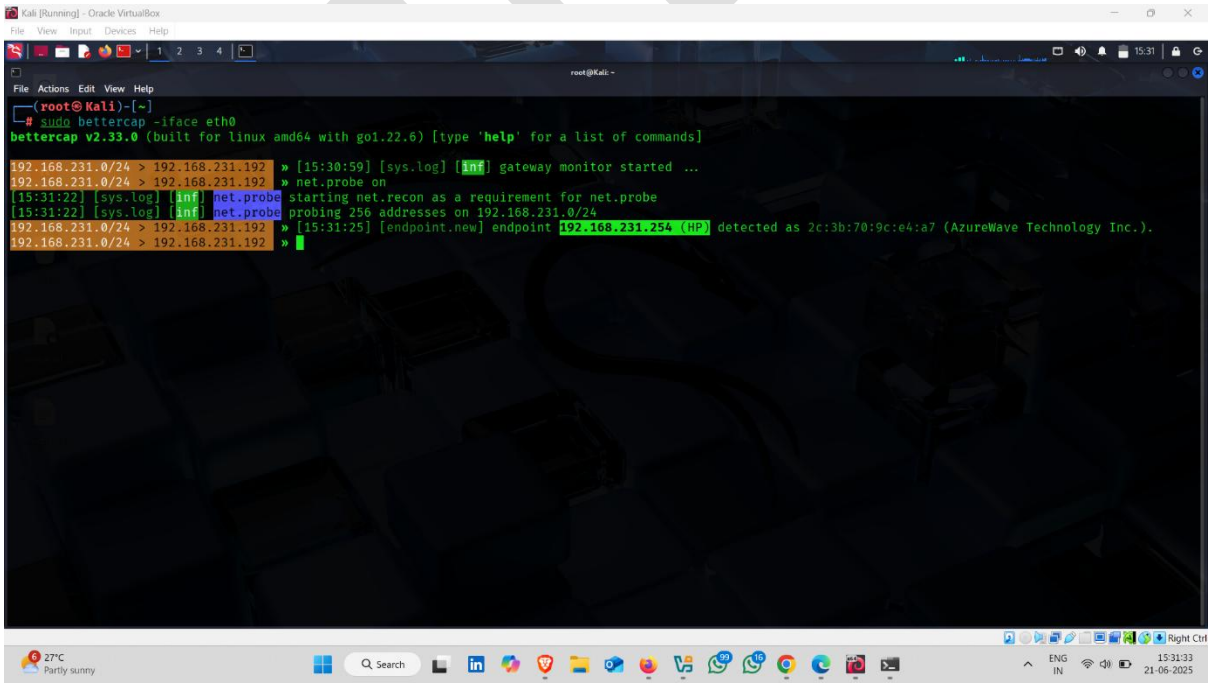


- Use Next Command

Command -: **net.probe on**



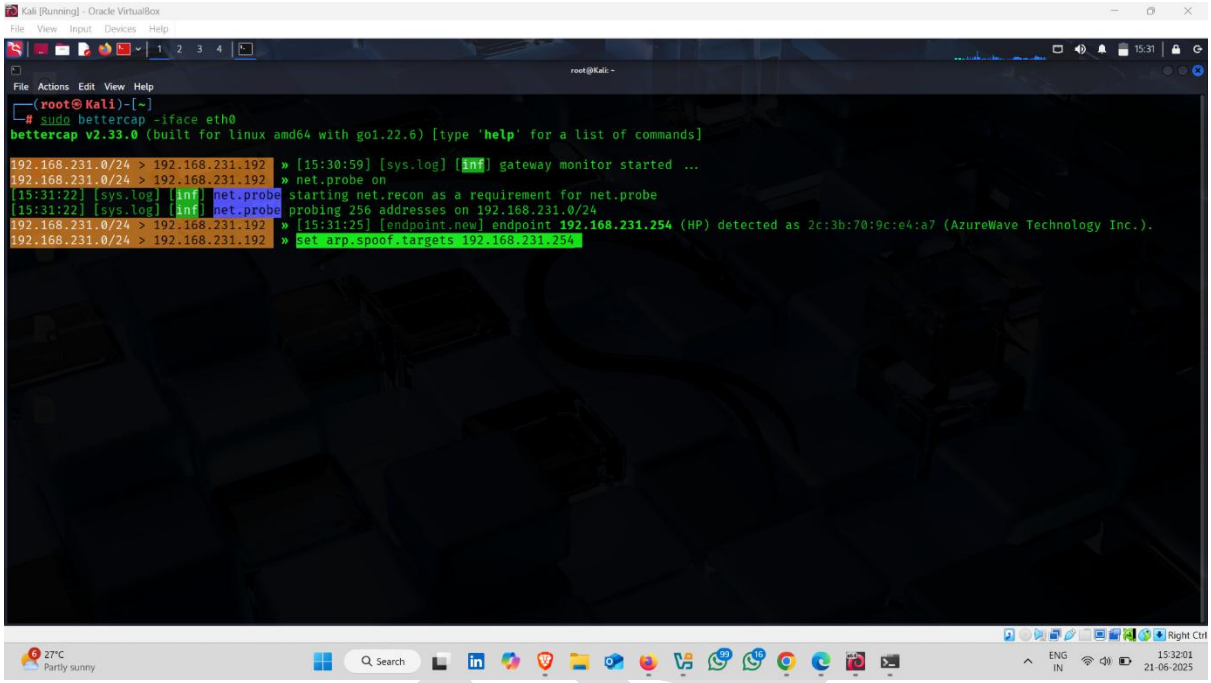
- Capture devices in network 👇



- Use next Command 👇

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

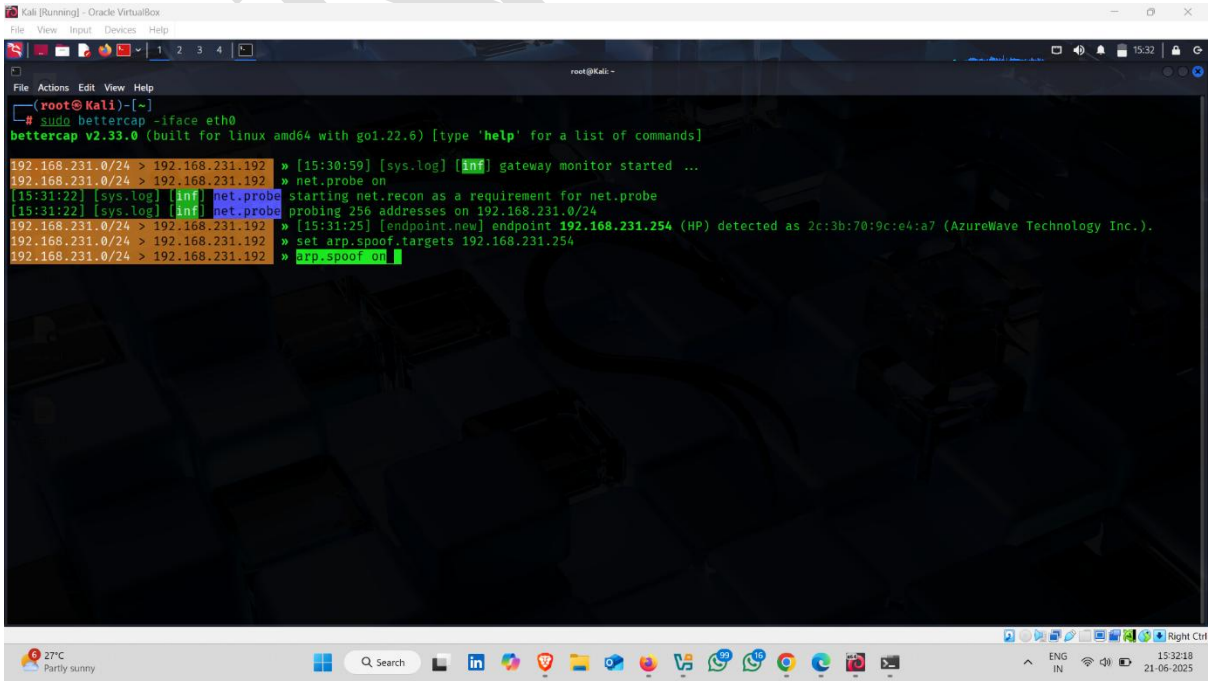
Explanation -: Set the victim's IP address



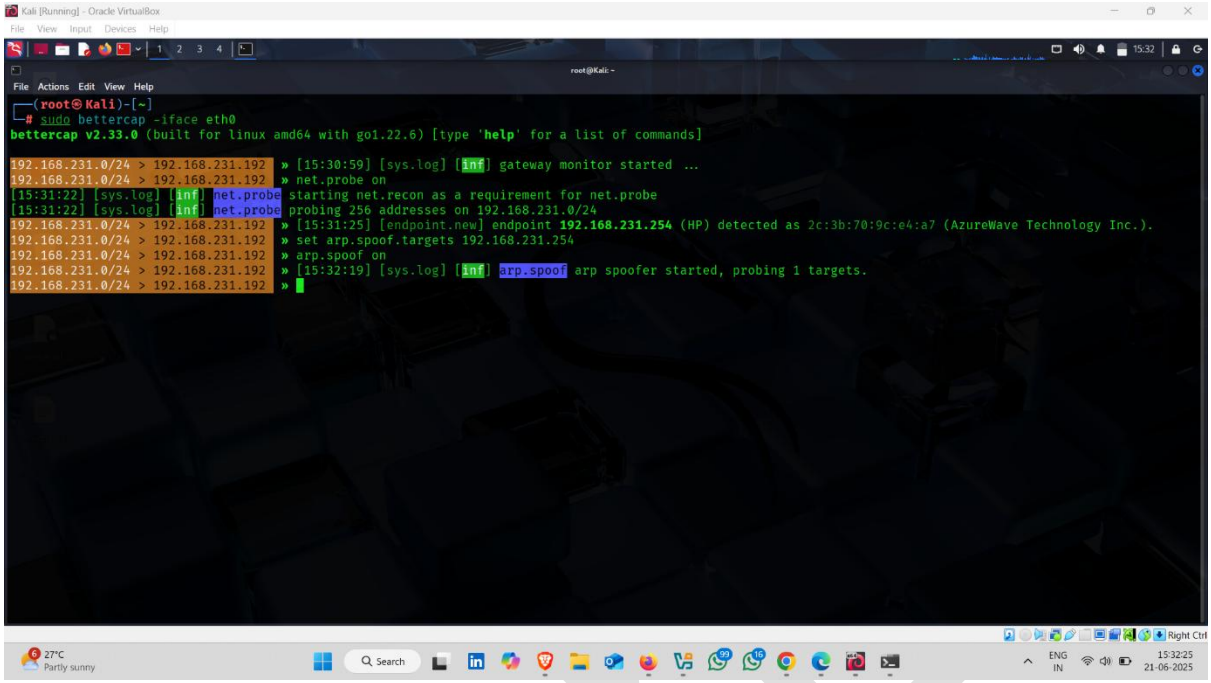
- Next Command

Command-: arp.spoof on

Explanation -: Start ARP spoofing



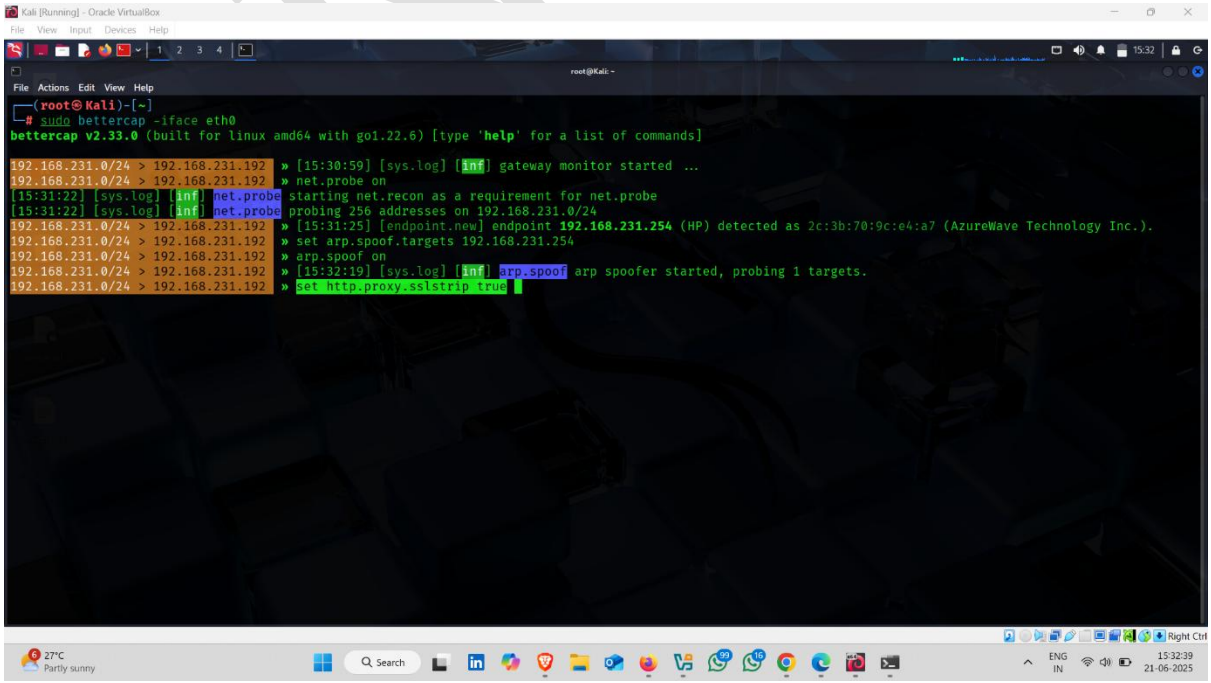
- Arp spoofing stared 👇



- Use next command

Command -: set http.proxy.sslstrip true

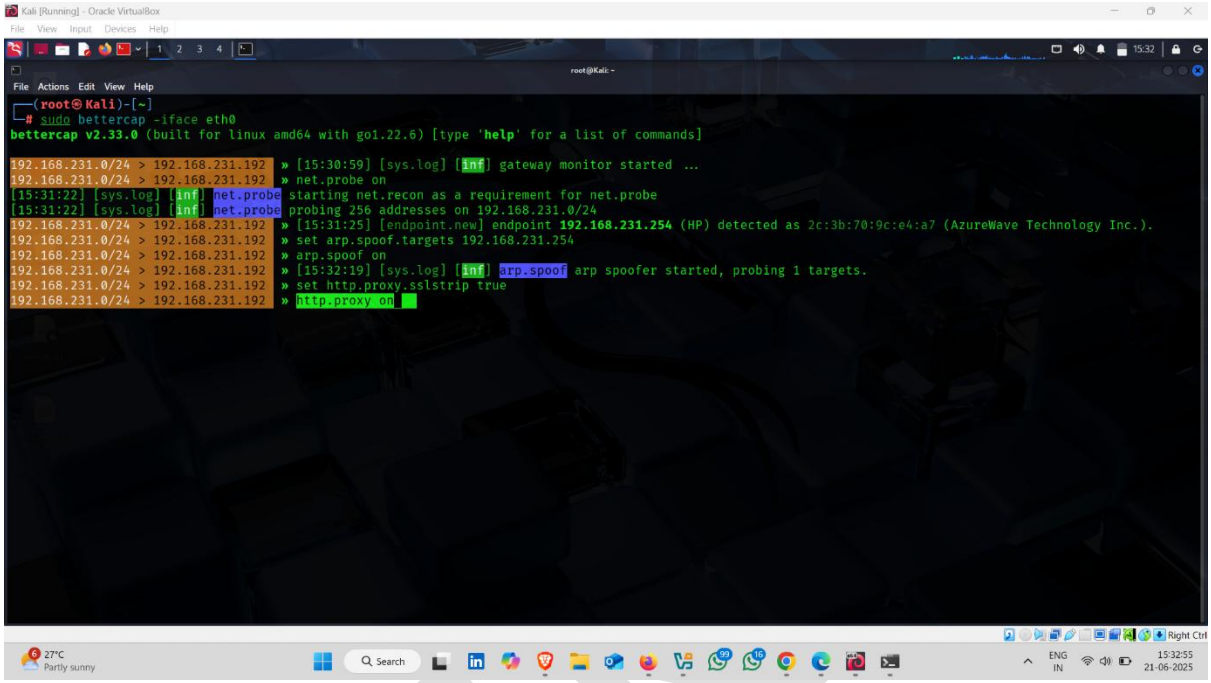
Explanation -: enable SSL stripping in Bettercap.



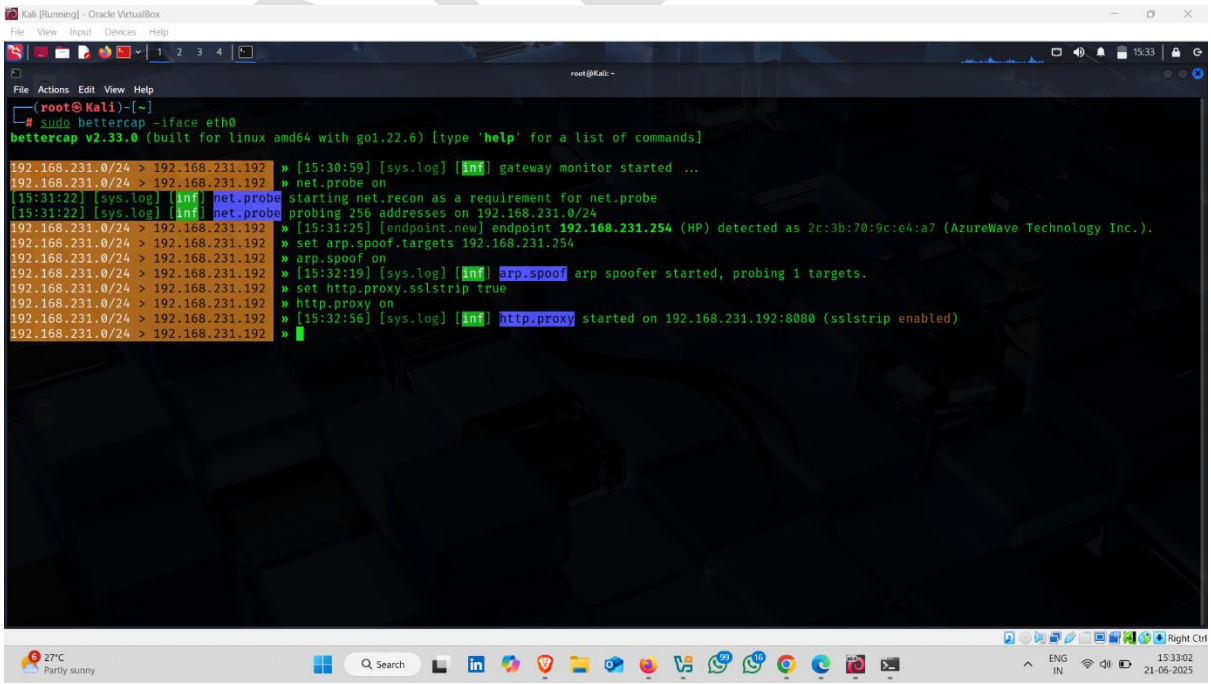
- Next Command 👇

Command -: http.proxy on

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



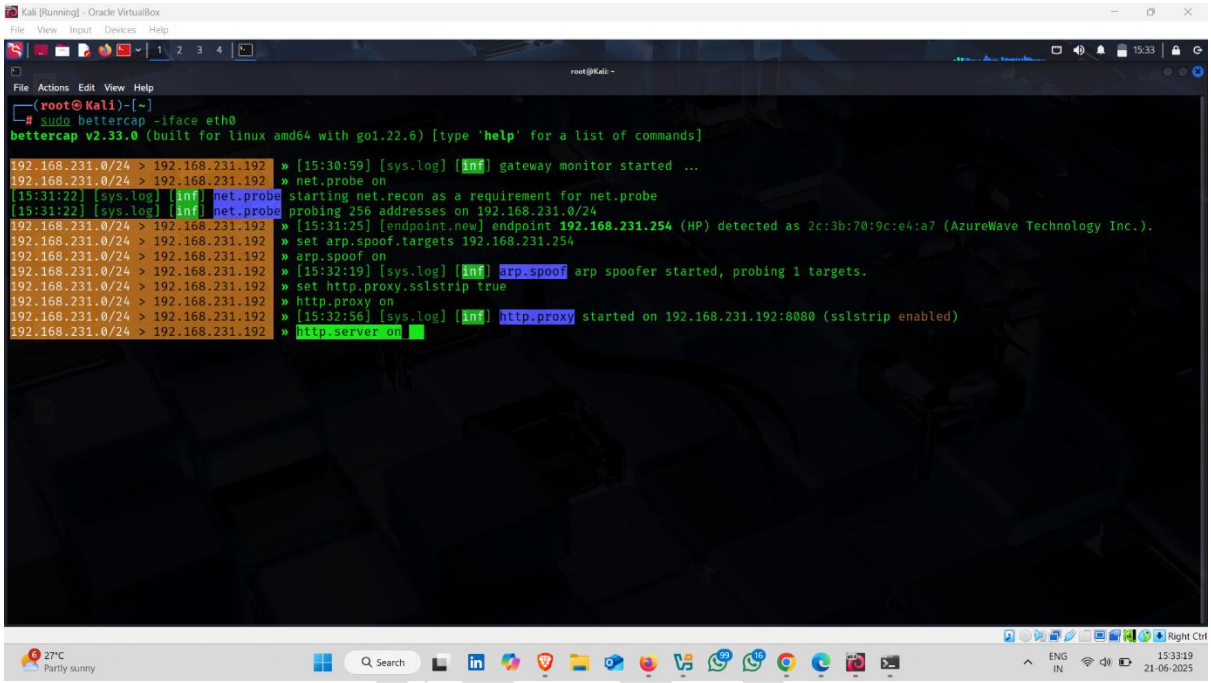
- Sslstrip enable 👇



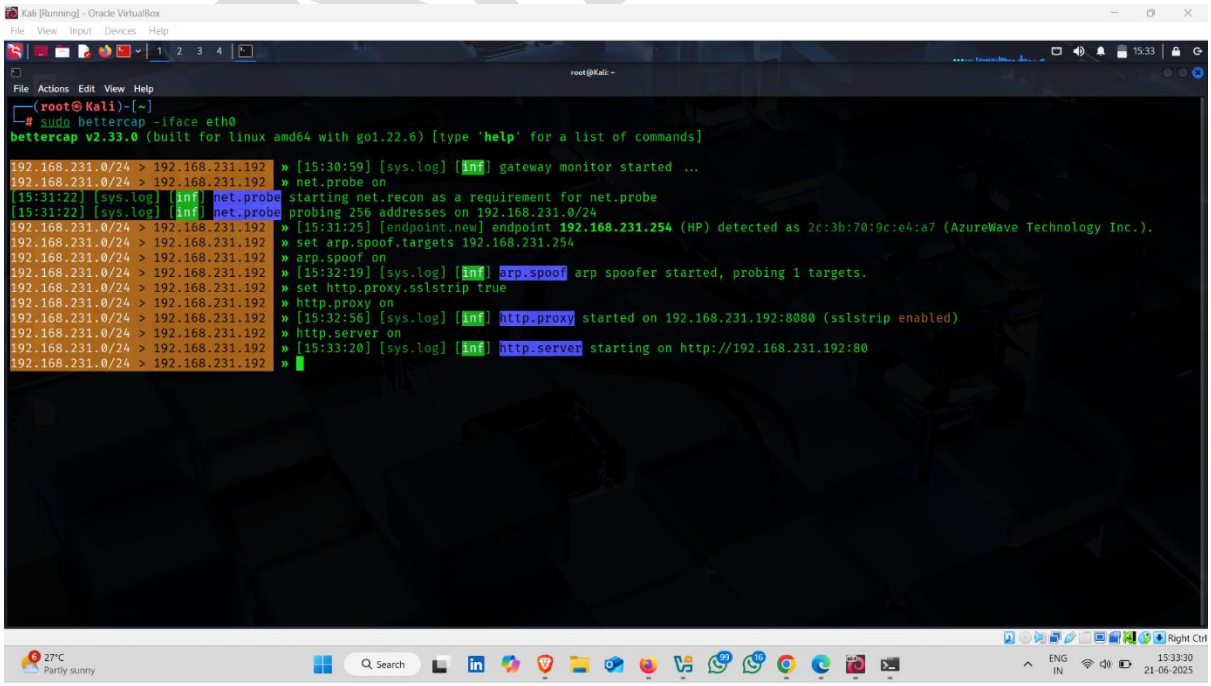
- Now use next Command

Command -: http.server on

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



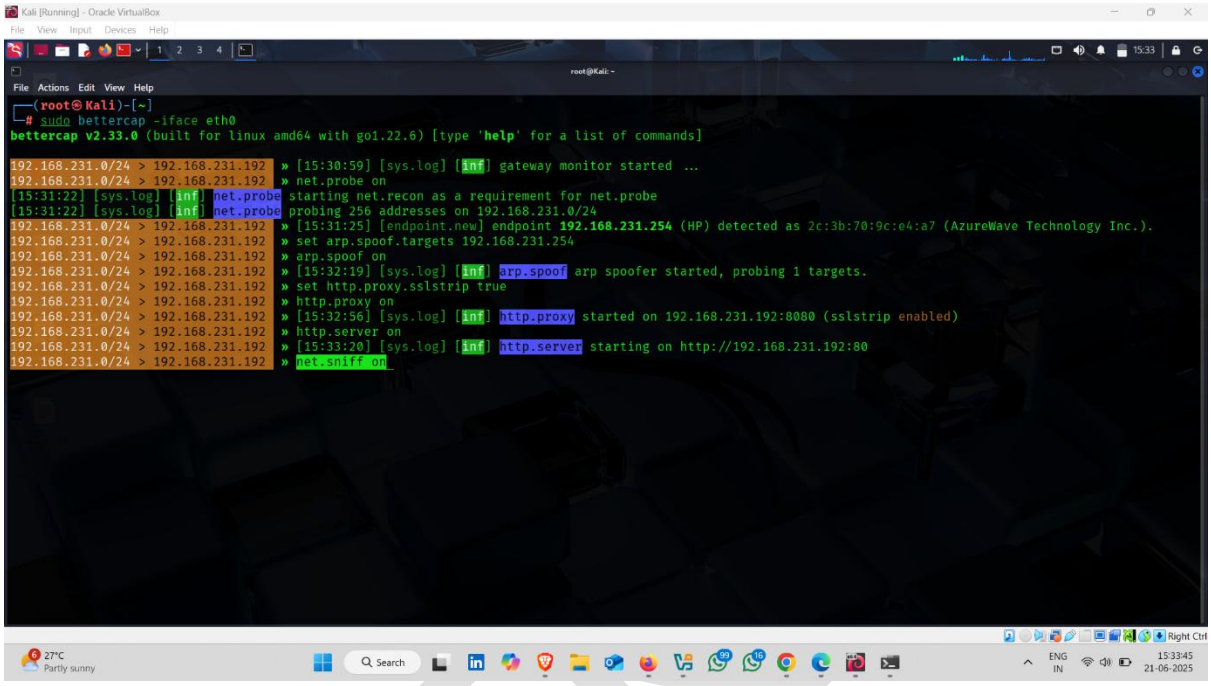
- Traffic redirect of your device started



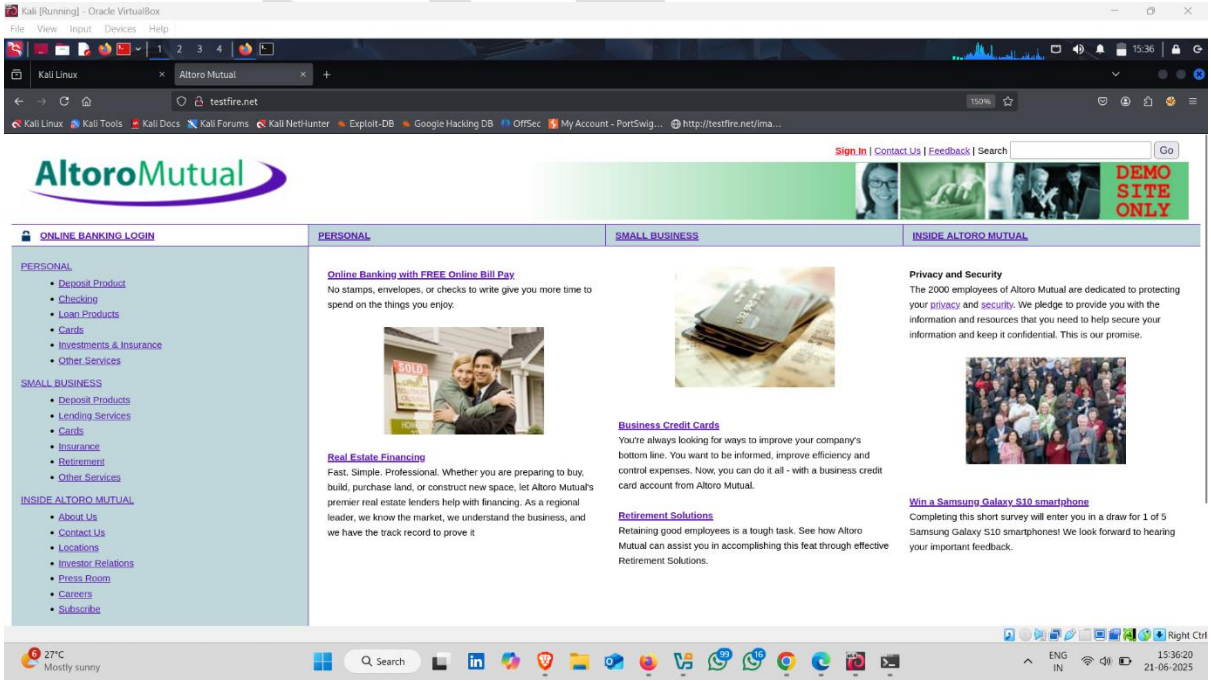
- Next command

Command -: net.sniff on

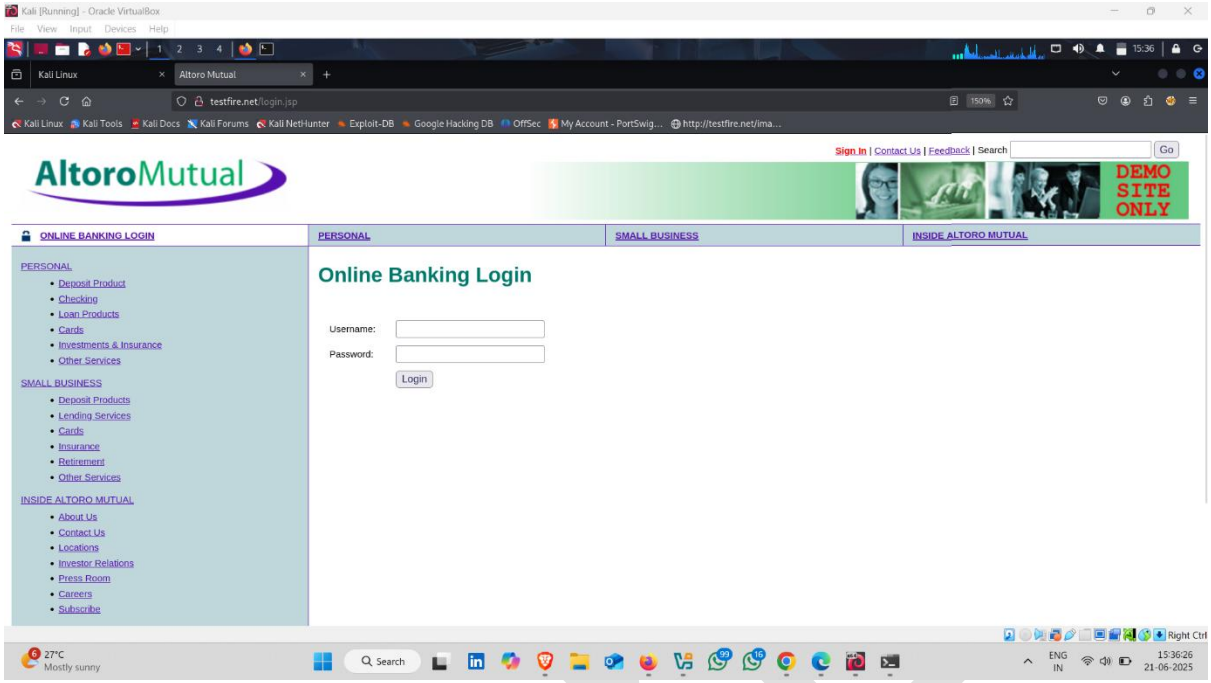
Explanation -: start real-time packet sniffing on the network.



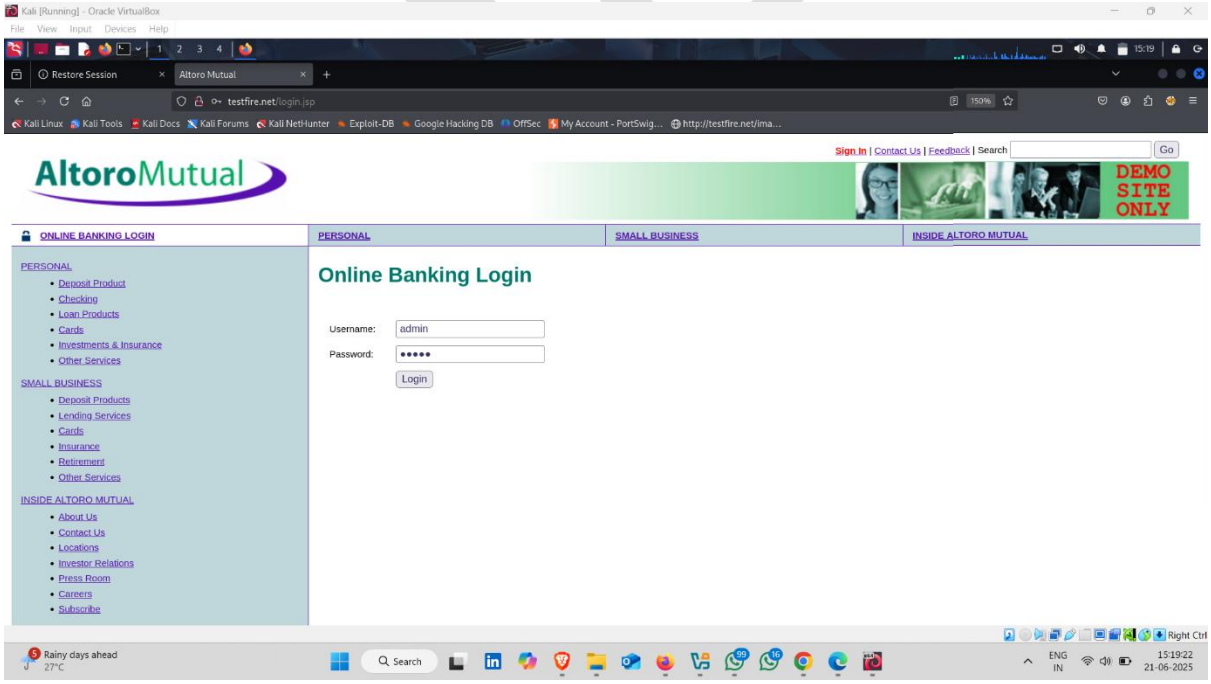
- Now open victim's Browser



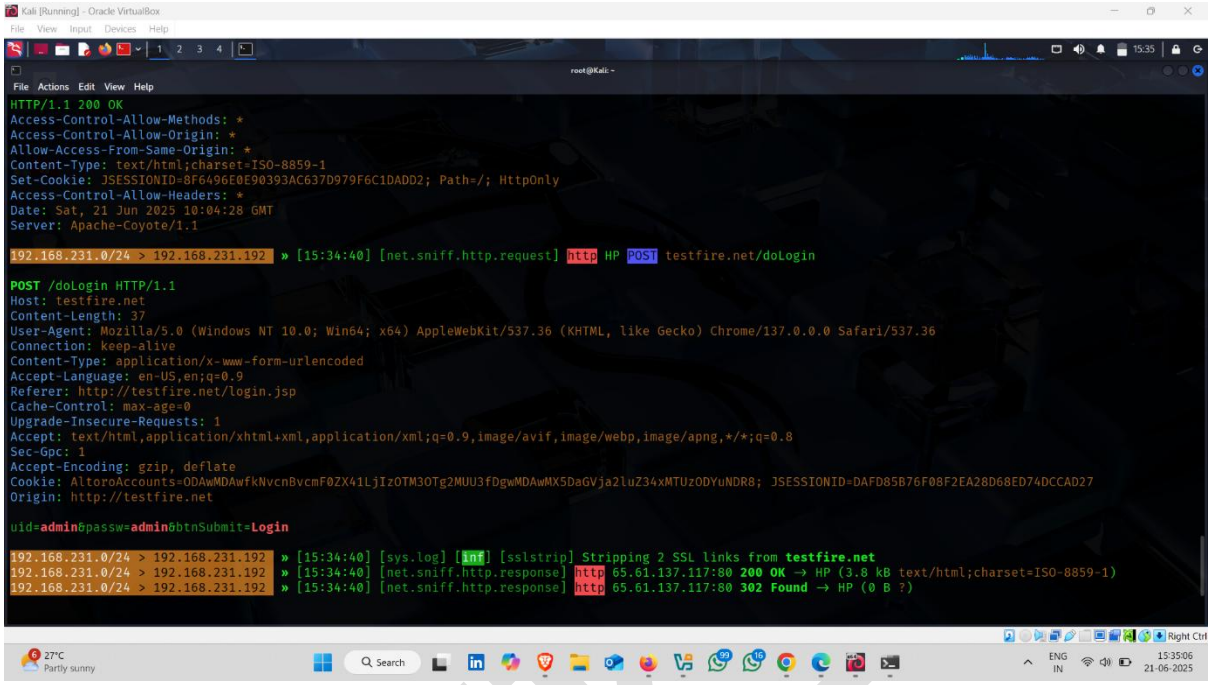
- Enter username and password 👇



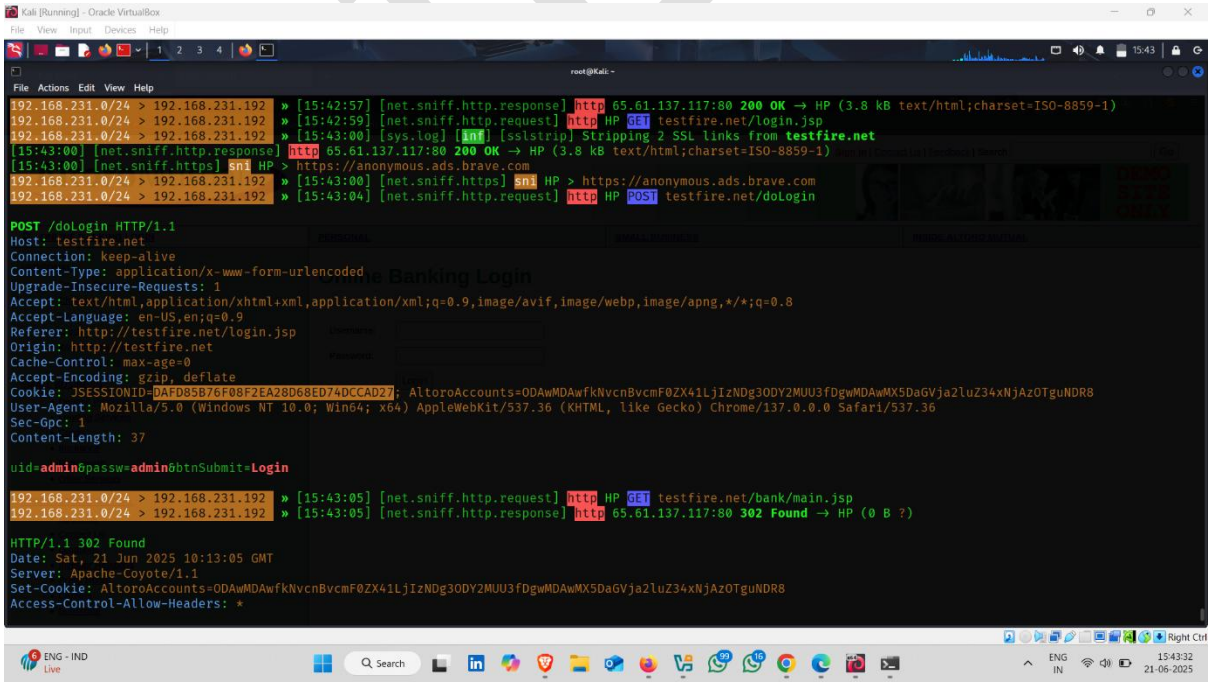
- Click on Login



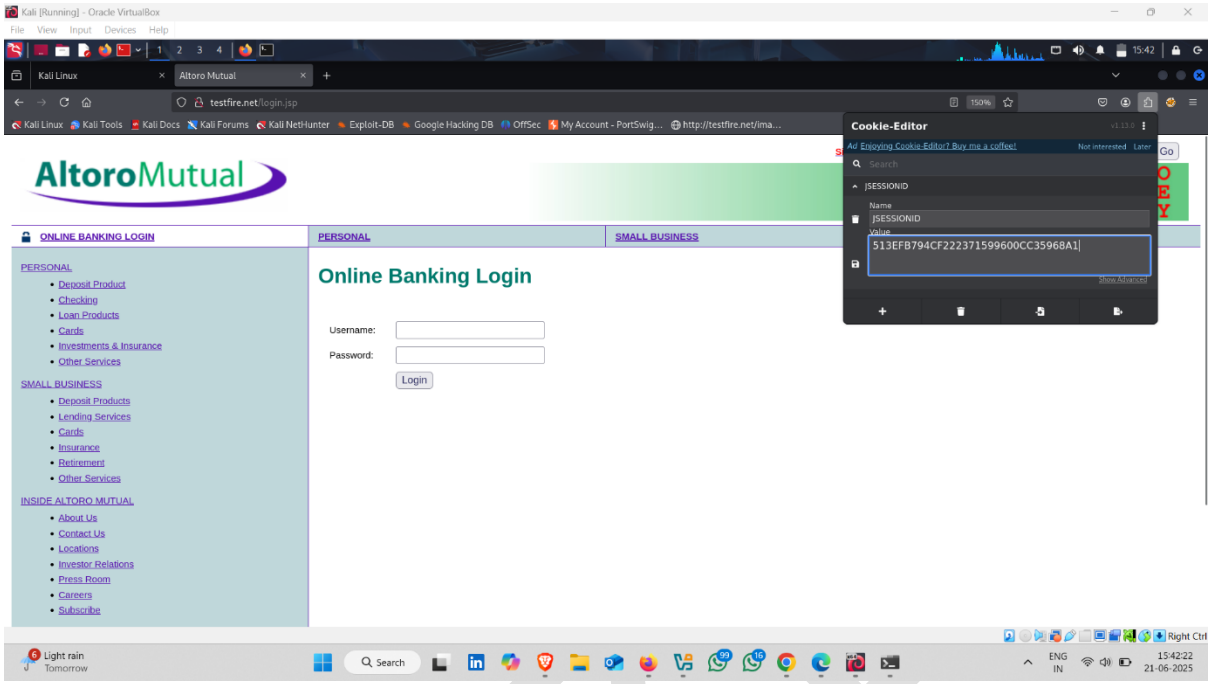
- Now back to the kali terminal and check bettercap capture anything or not
- Here , bettercap captured the request



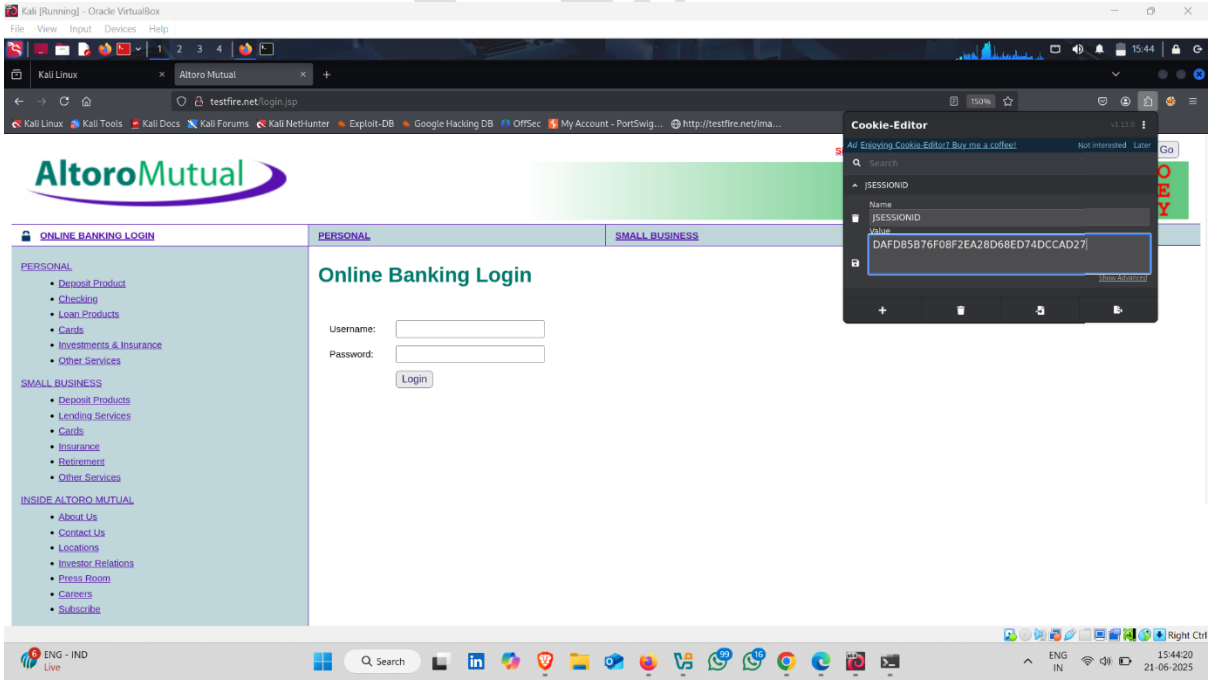
- Now copy this **session Id**



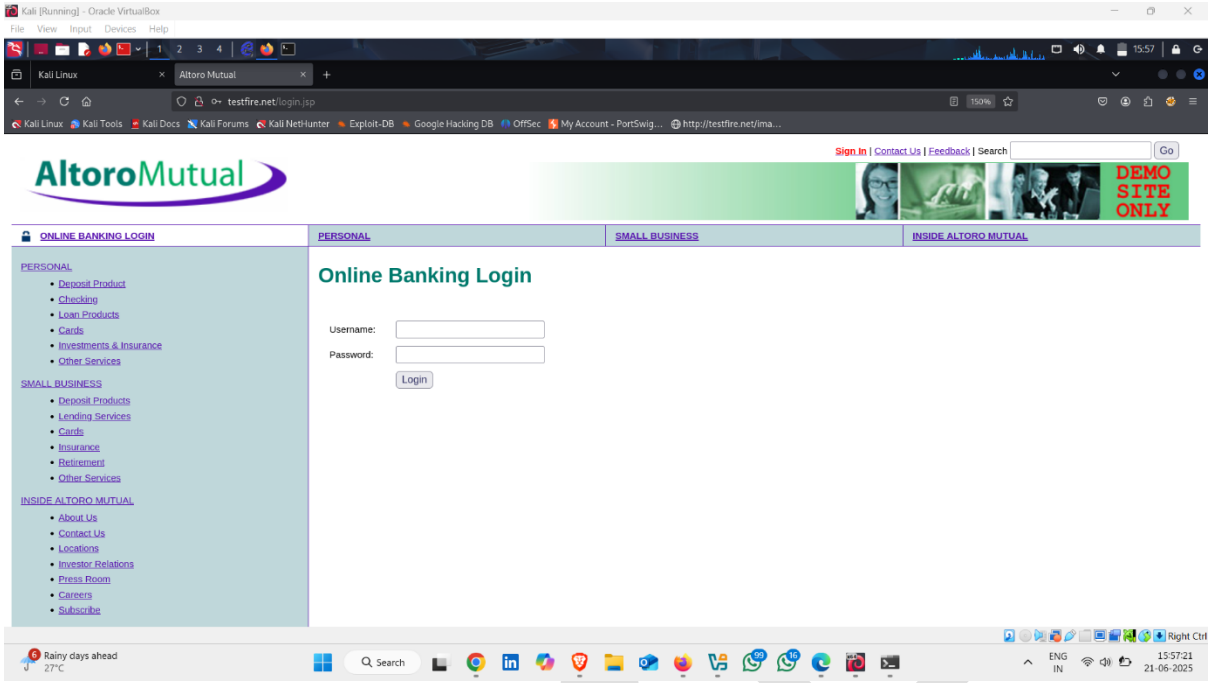
- **Open target website and open cookies editor**
- **Now replace this session id**



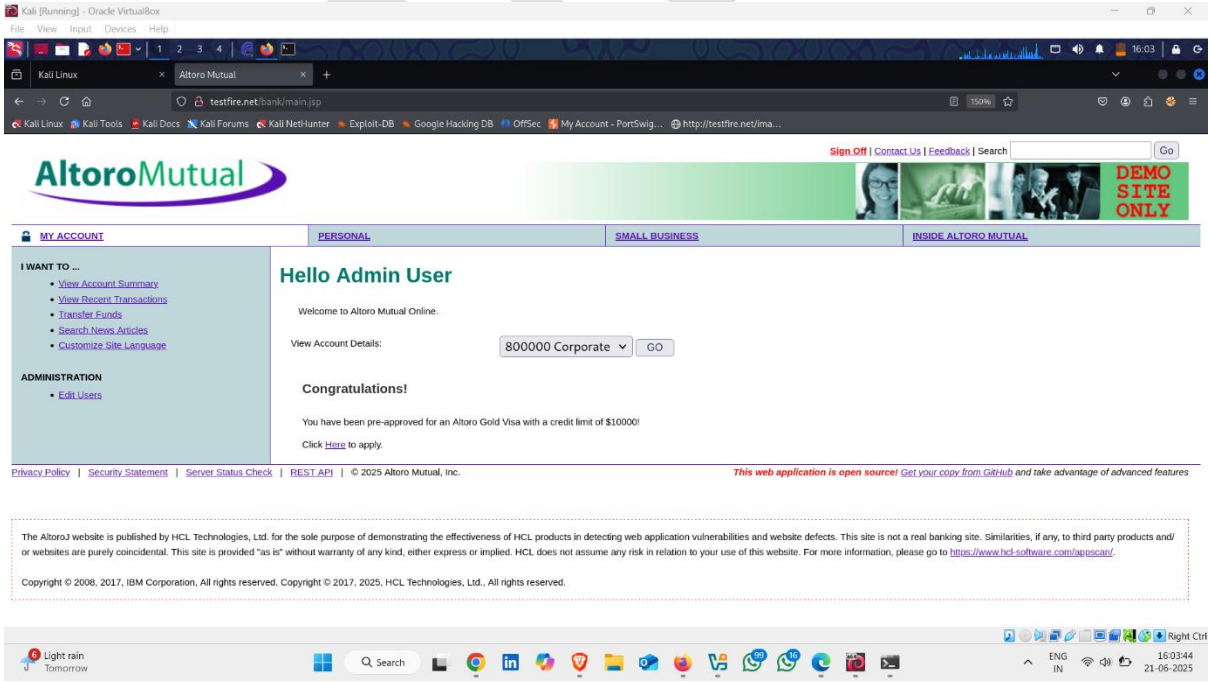
- With This 👇 and click on **save**



- 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:

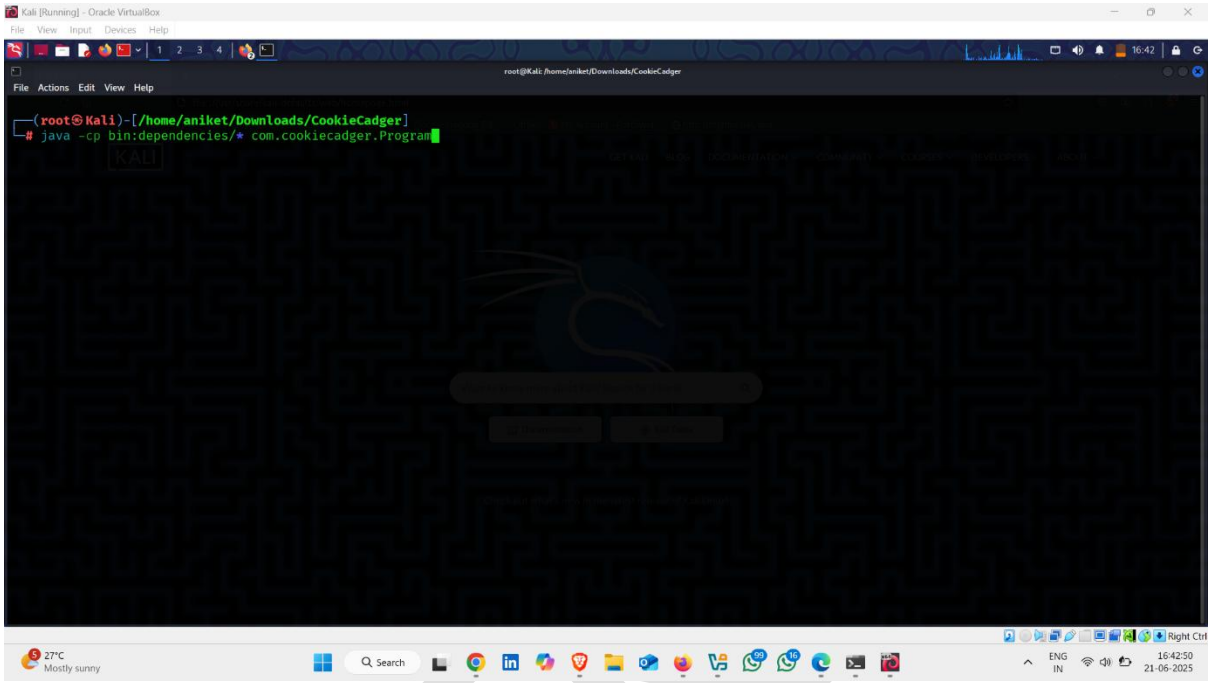
✔ To **demonstrate and practice session hijacking** using stolen HTTP cookies from users on the same network.

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

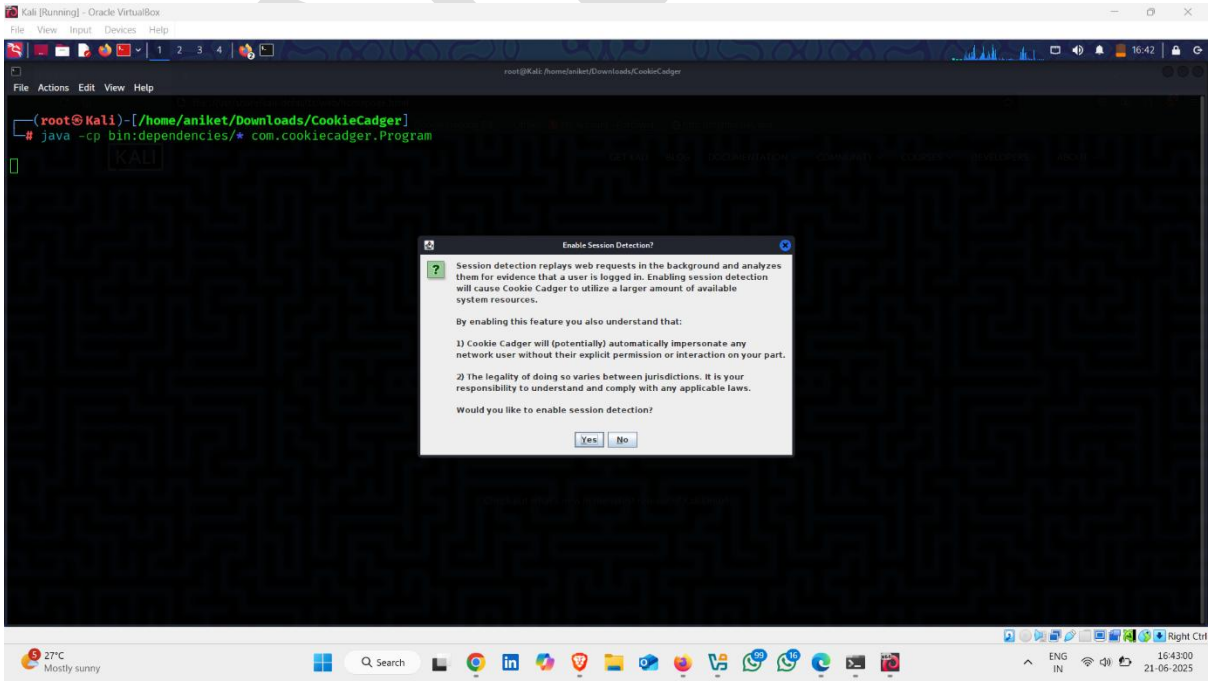
How To Use It -:

- Run cookie Cadger Using This command

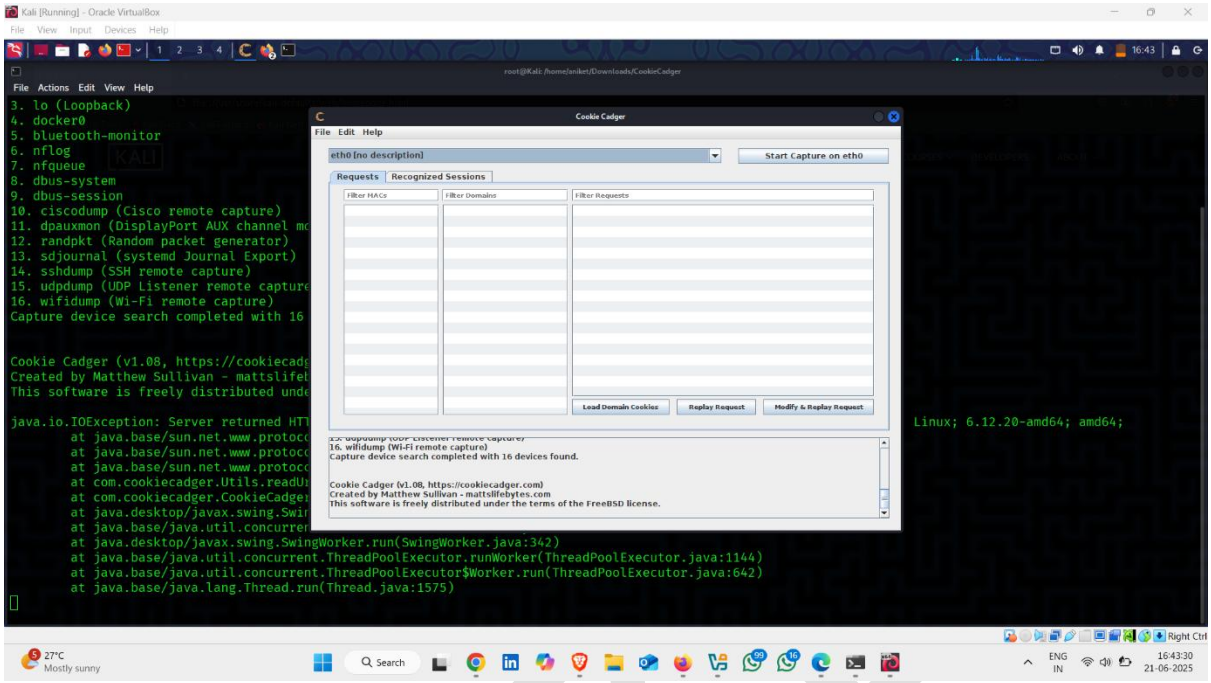
Command -: -cp bin:dependencies/* com.cookiecadger.program



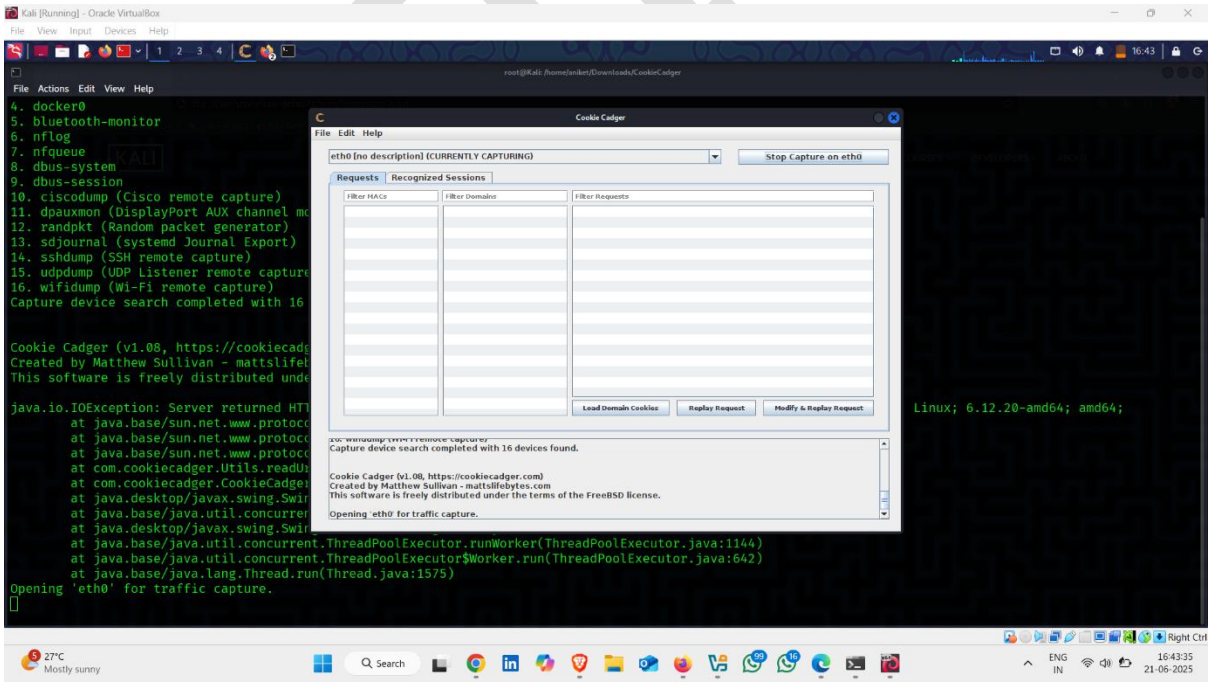
- Click on Yes ✅



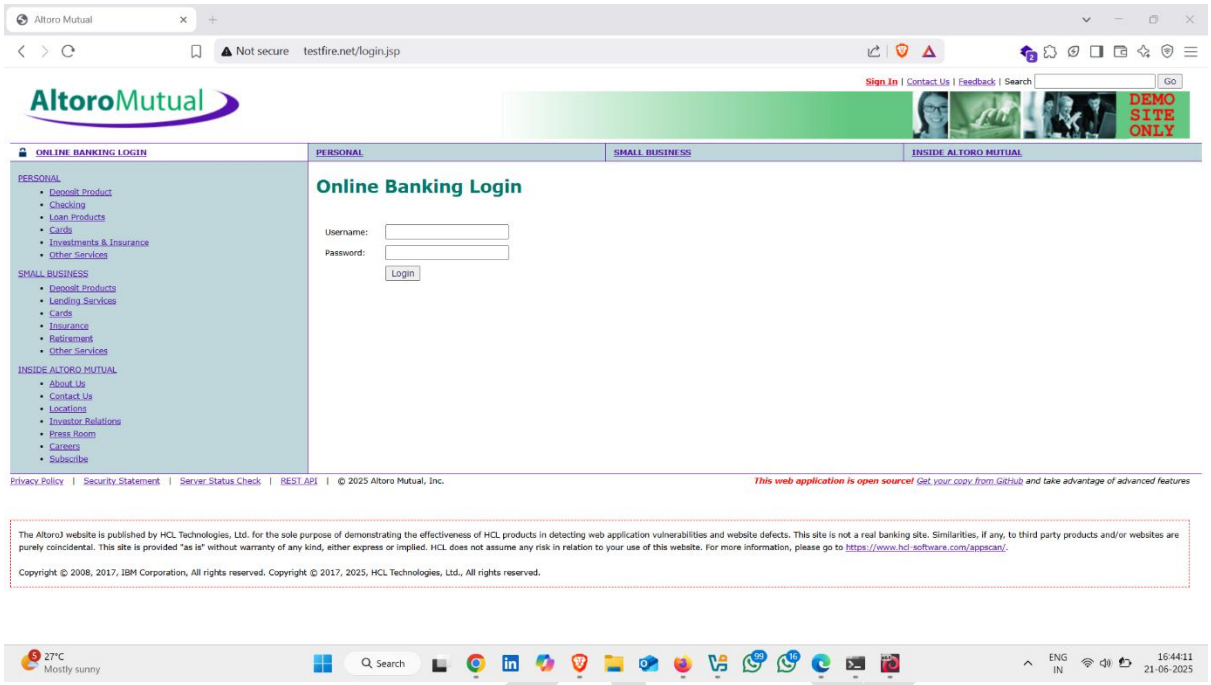
- Select the network Interface and then click on **Start Capture on Eth0**



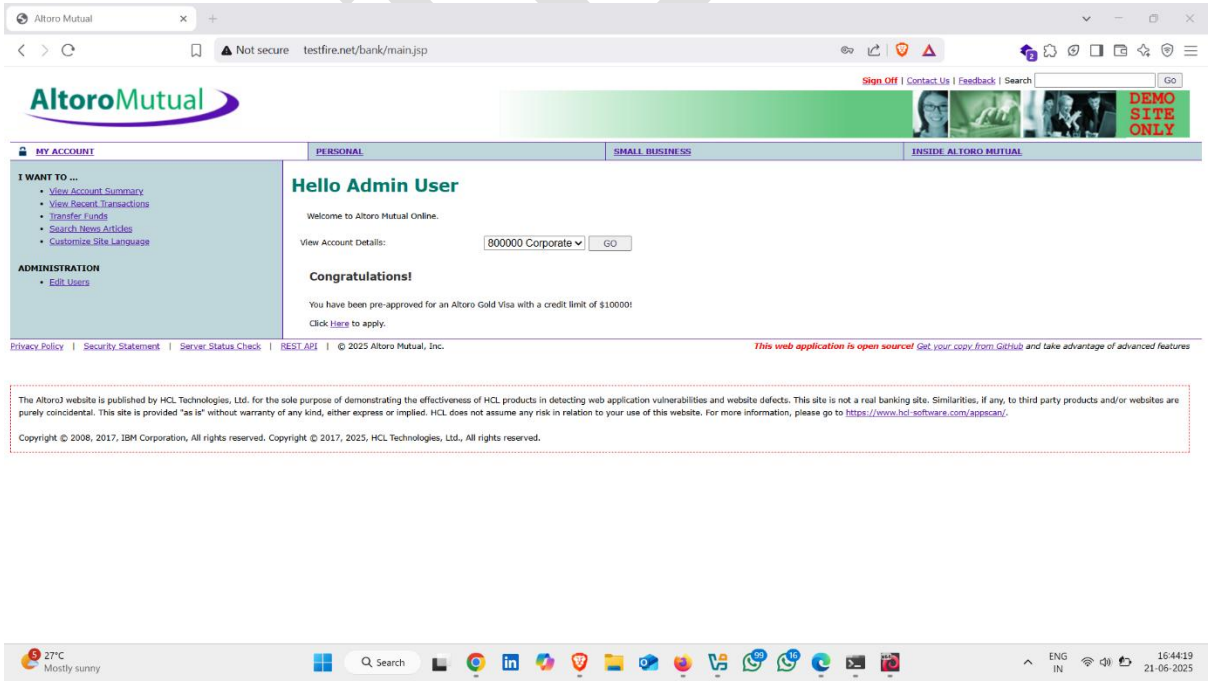
- It started capturing



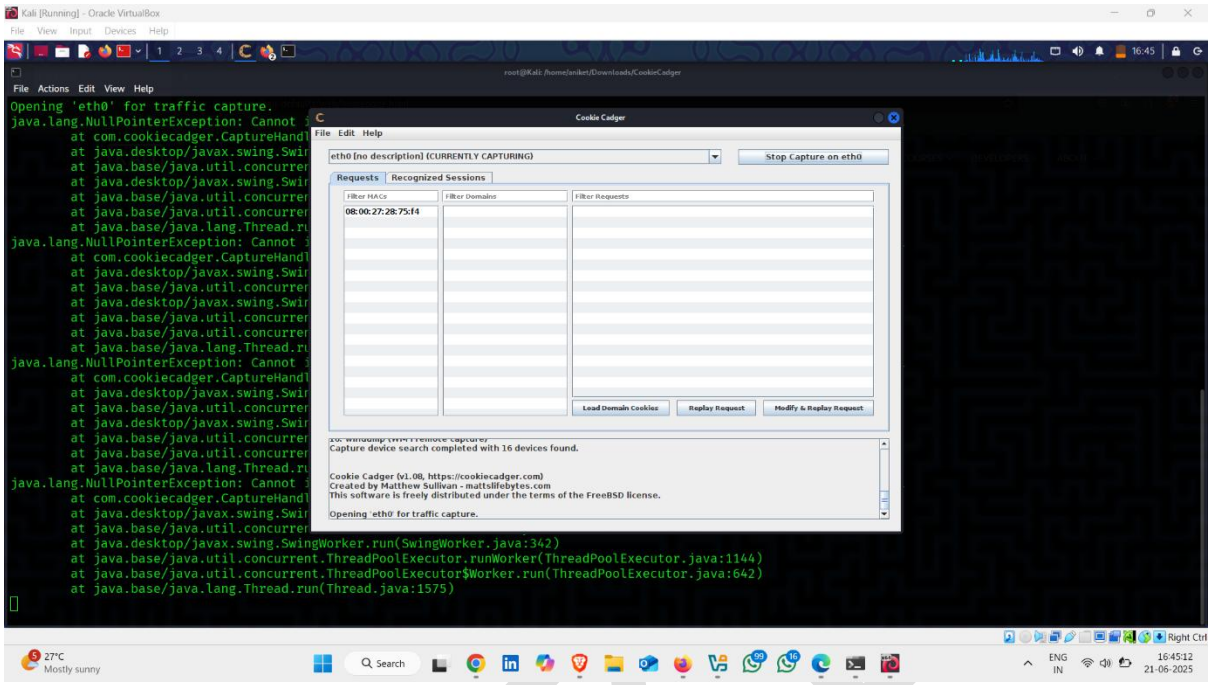
- Open website and enter username and password and click on login ✅ 👇



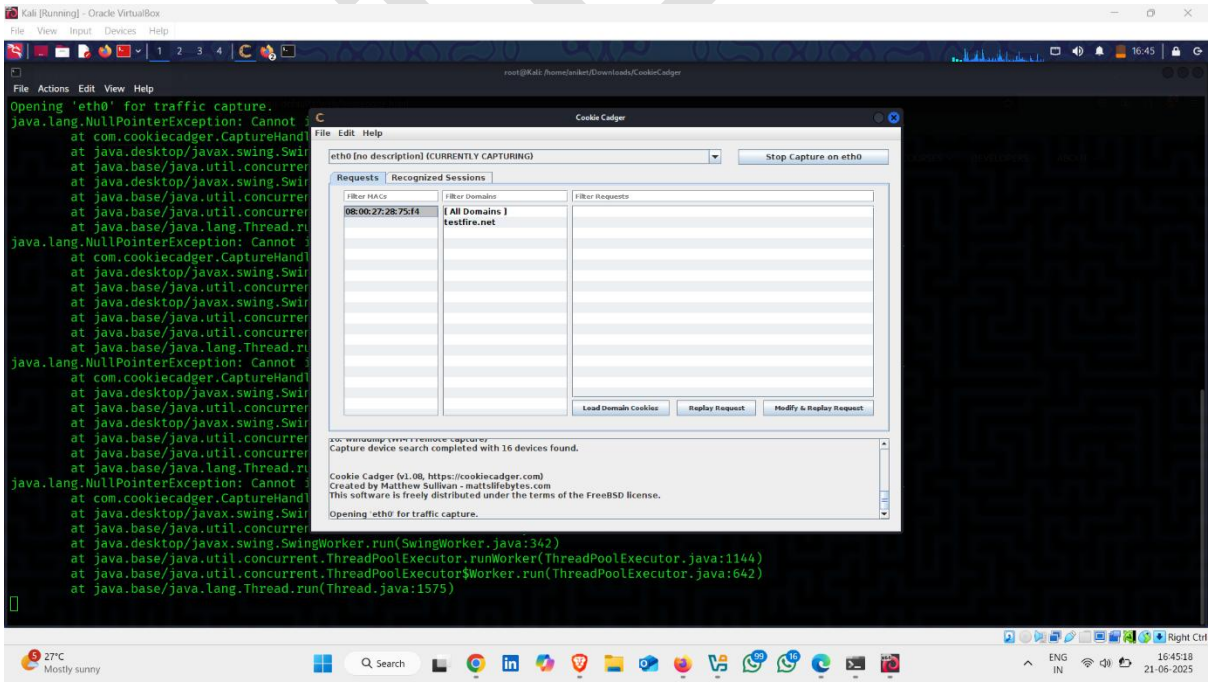
- Login ✅



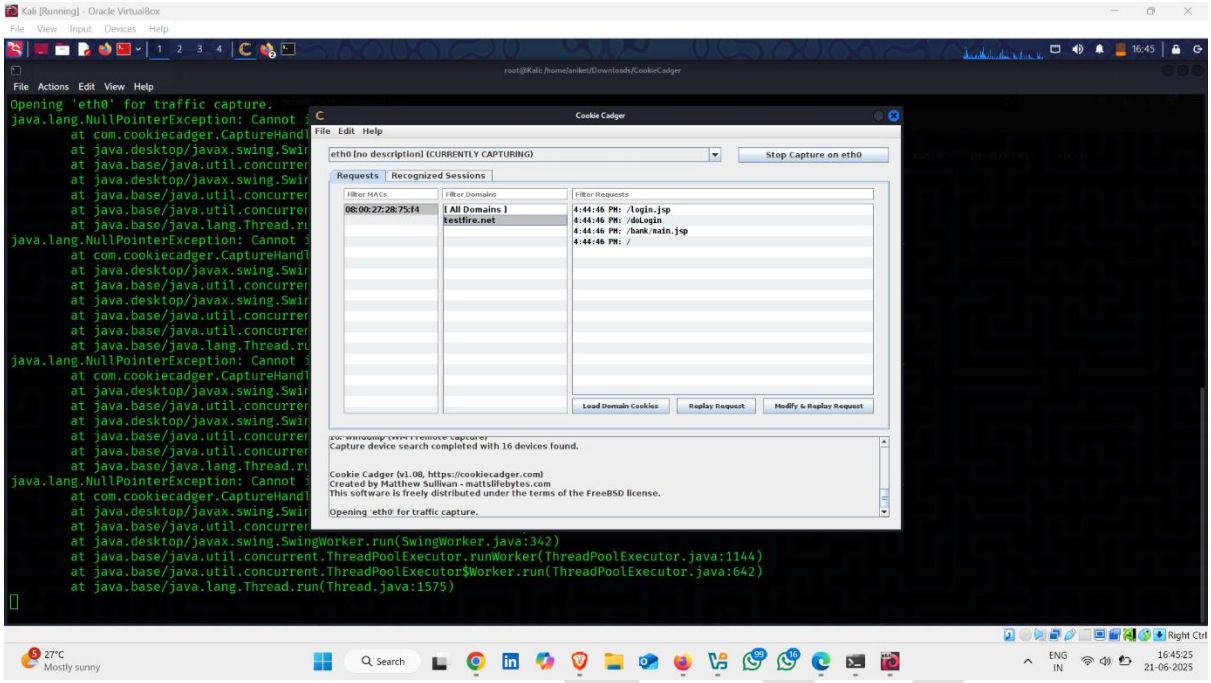
- Here , it capture
- Click on Ip address ✅ 👇



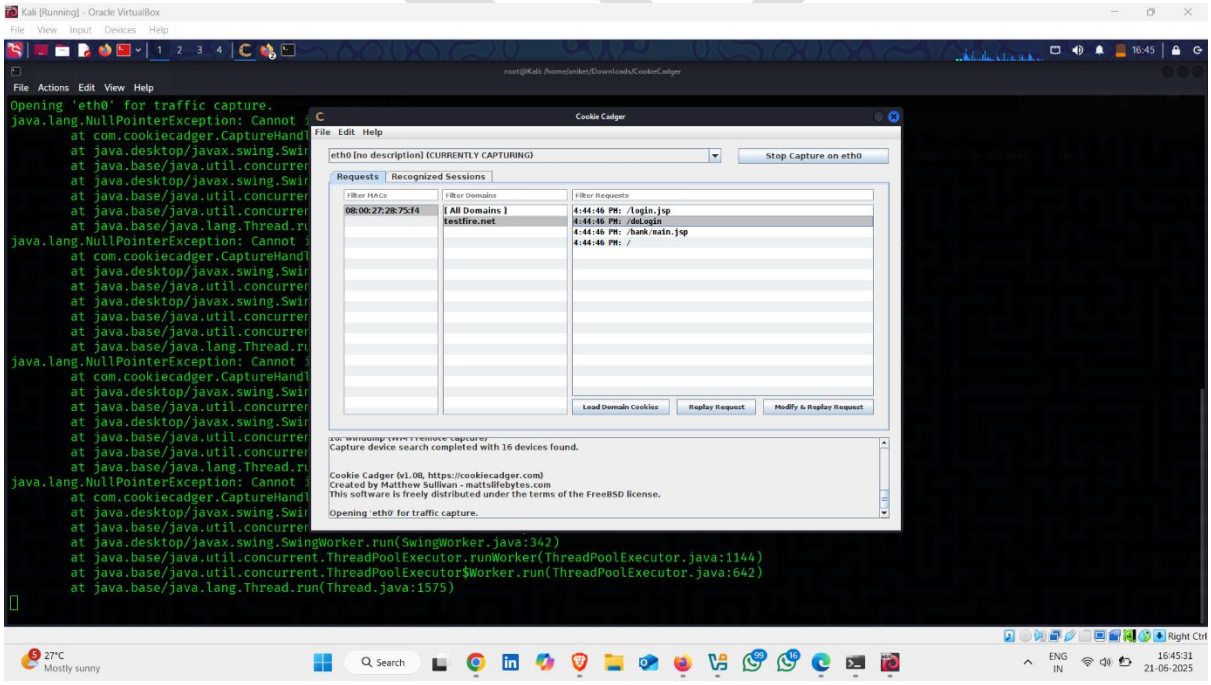
- Then click on website



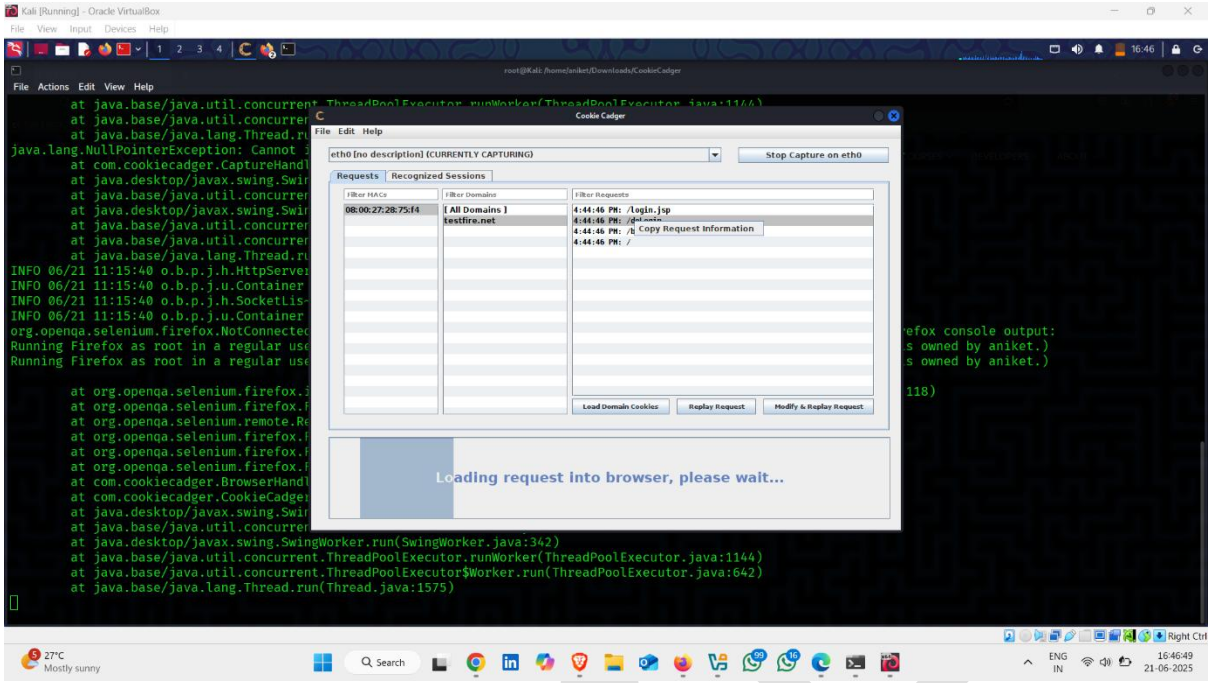
- Here , login request 👇



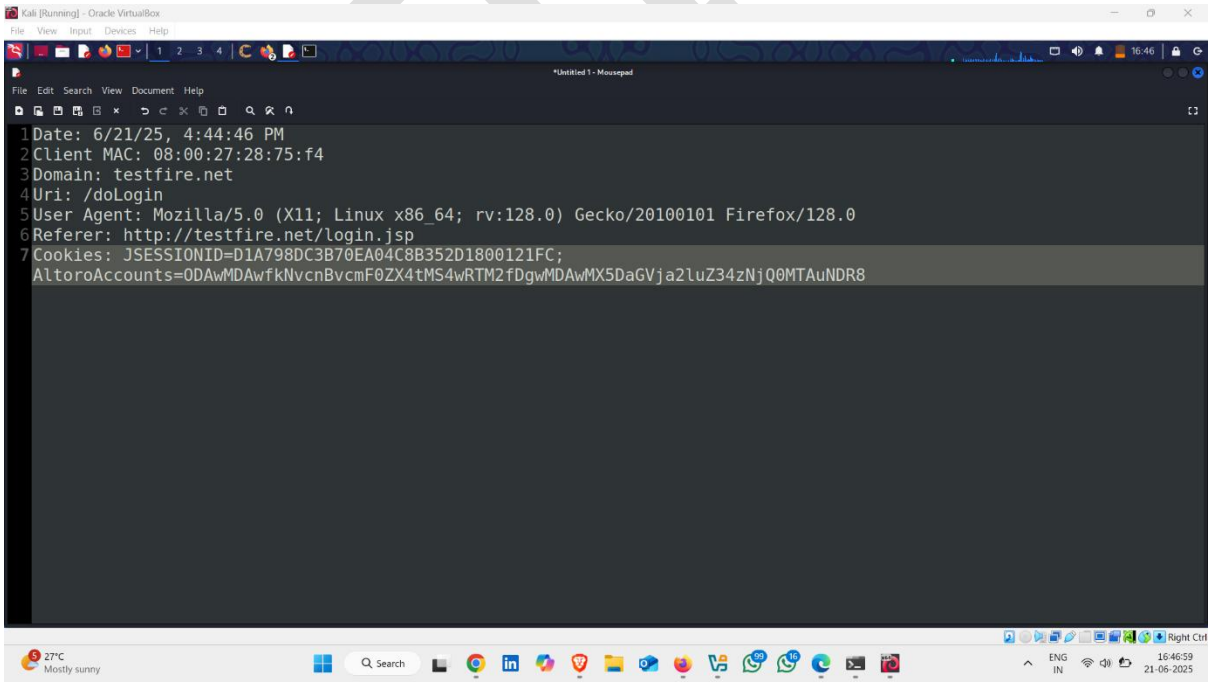
- Now , right click on **do login**



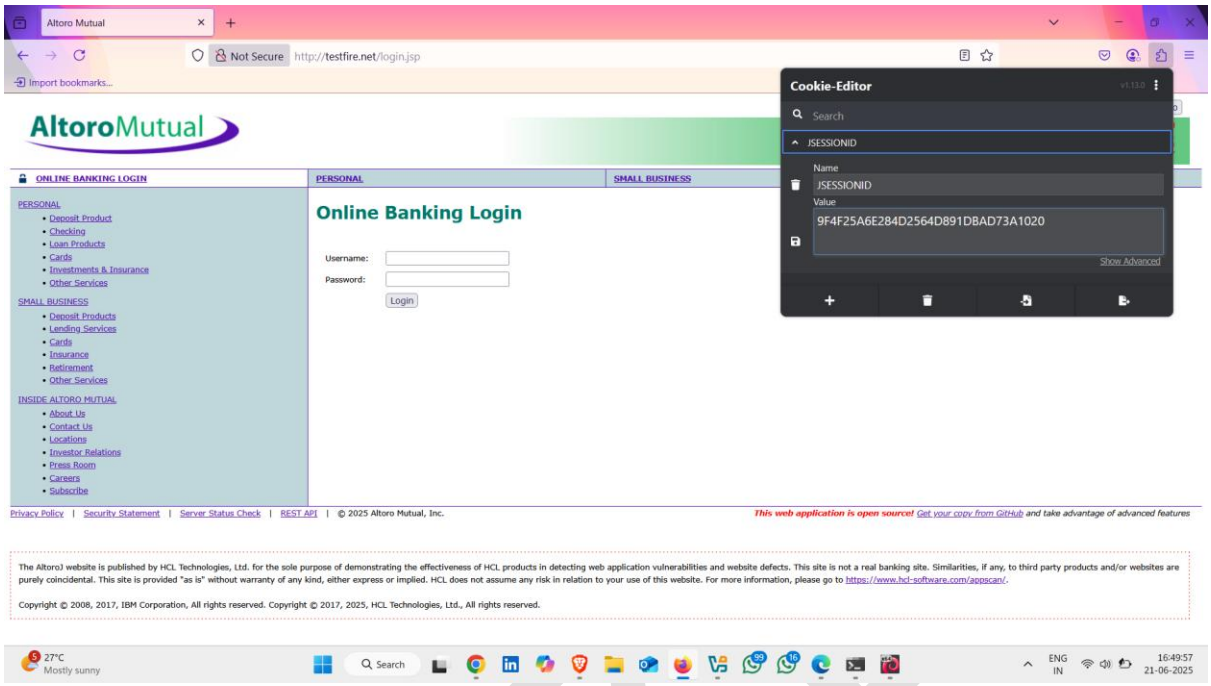
- And click on **Copy Request Information**



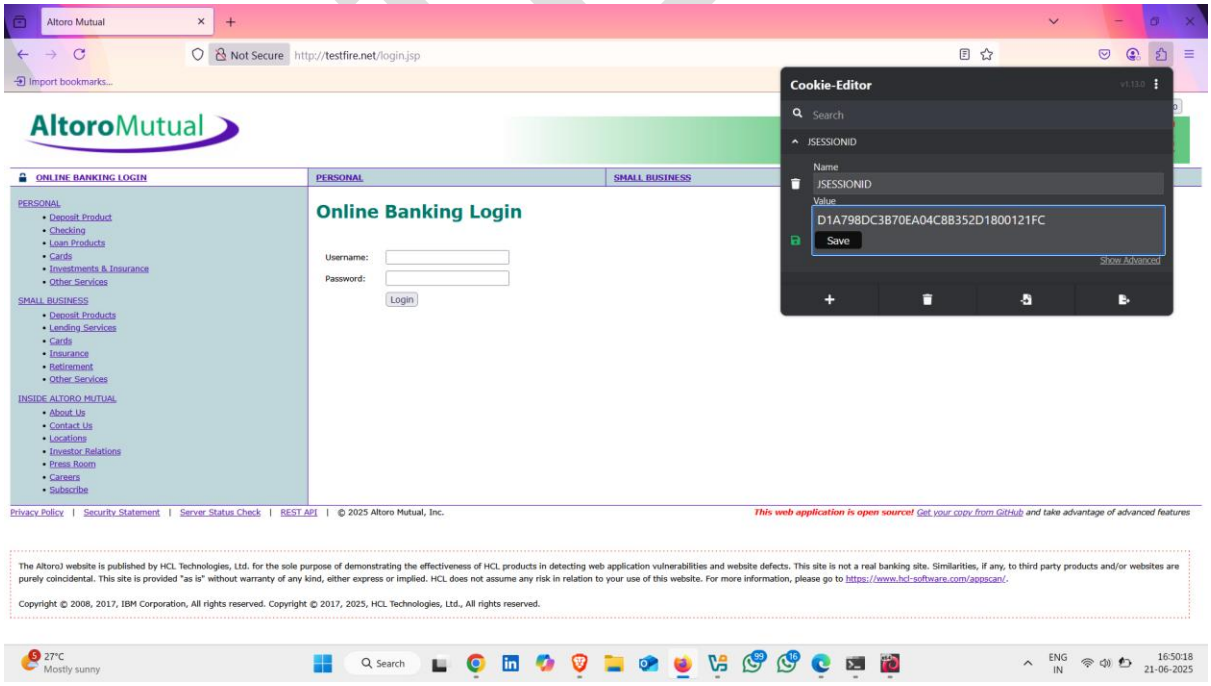
- Open text Editor and paste here
- Here it **domain , website name , session and other things**



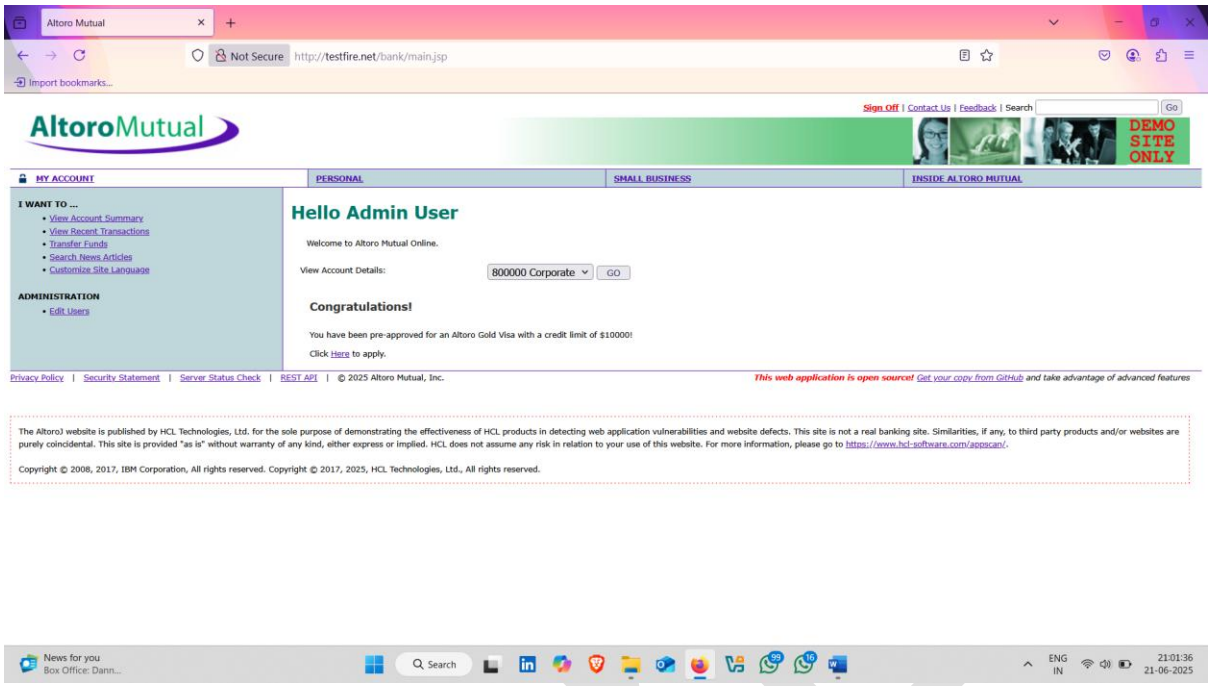
- Now open another browser and open cookies extension
- And replace this session id



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



- Login Successful ✅ 👇



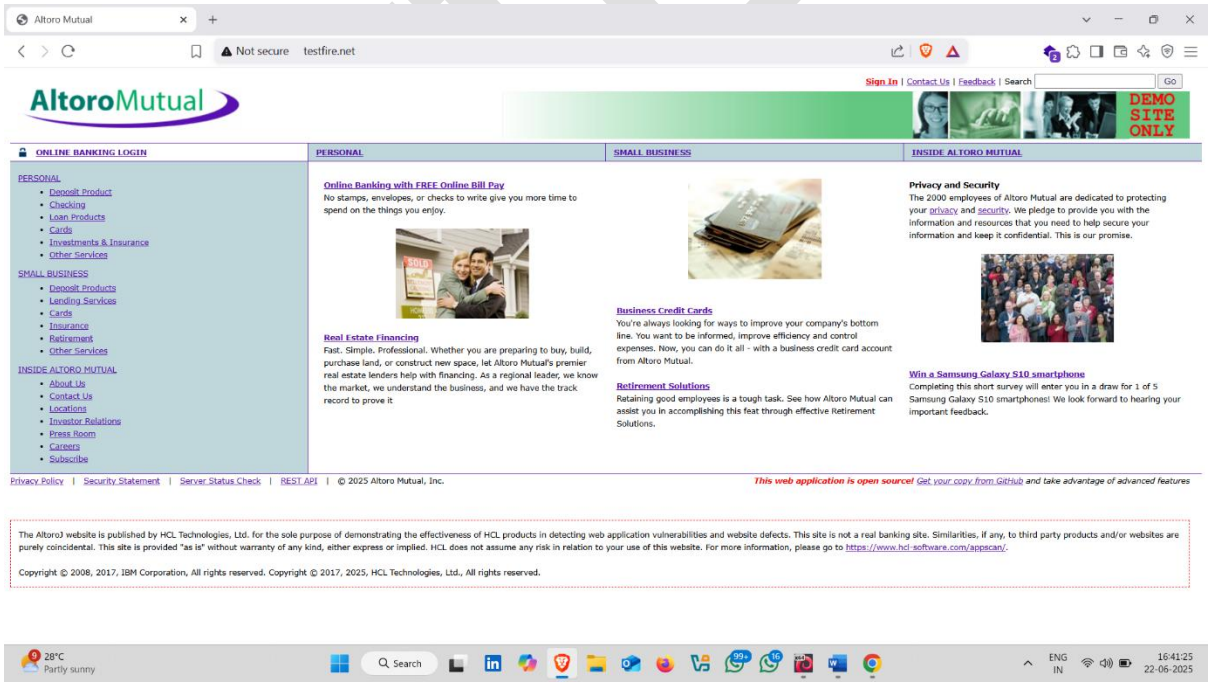
4. Session Hijacking through Manual Cookie Theft.

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.

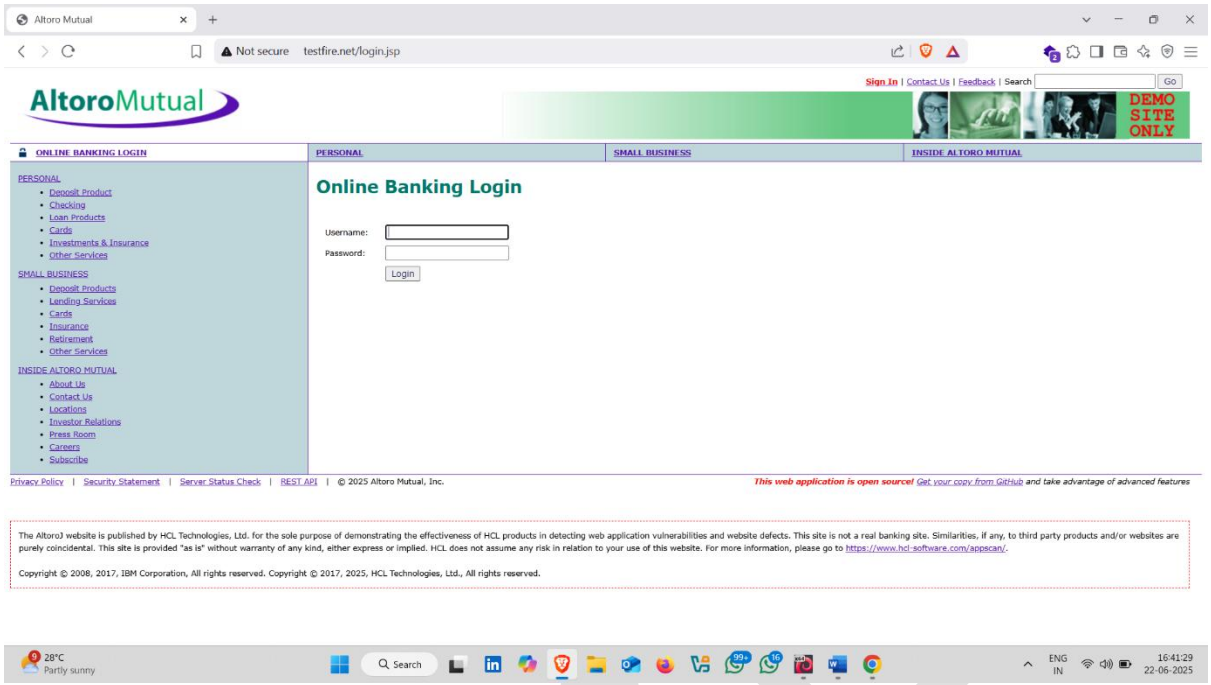
In this scenario, if a user logs into a website and leaves the session unattended—especially in a shared or public environment—a trusted person or an attacker nearby can manually inspect the browser, copy the active session cookie, and reuse it to hijack the session.

How to do it -:

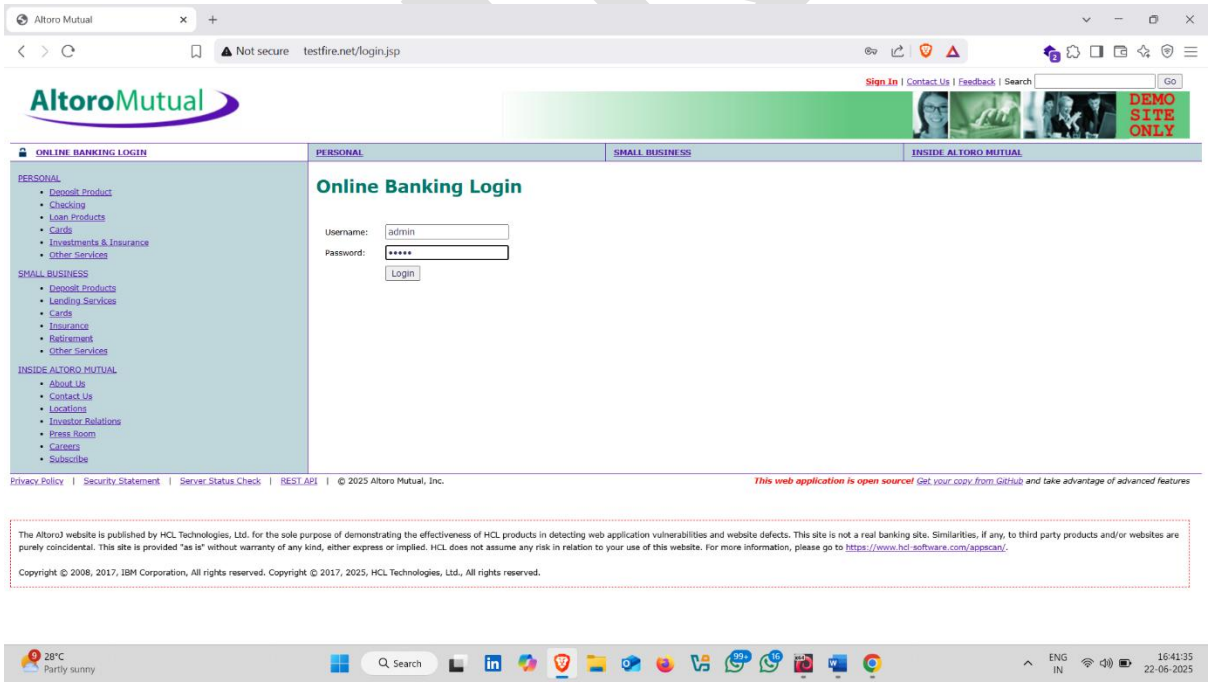
- Target Website



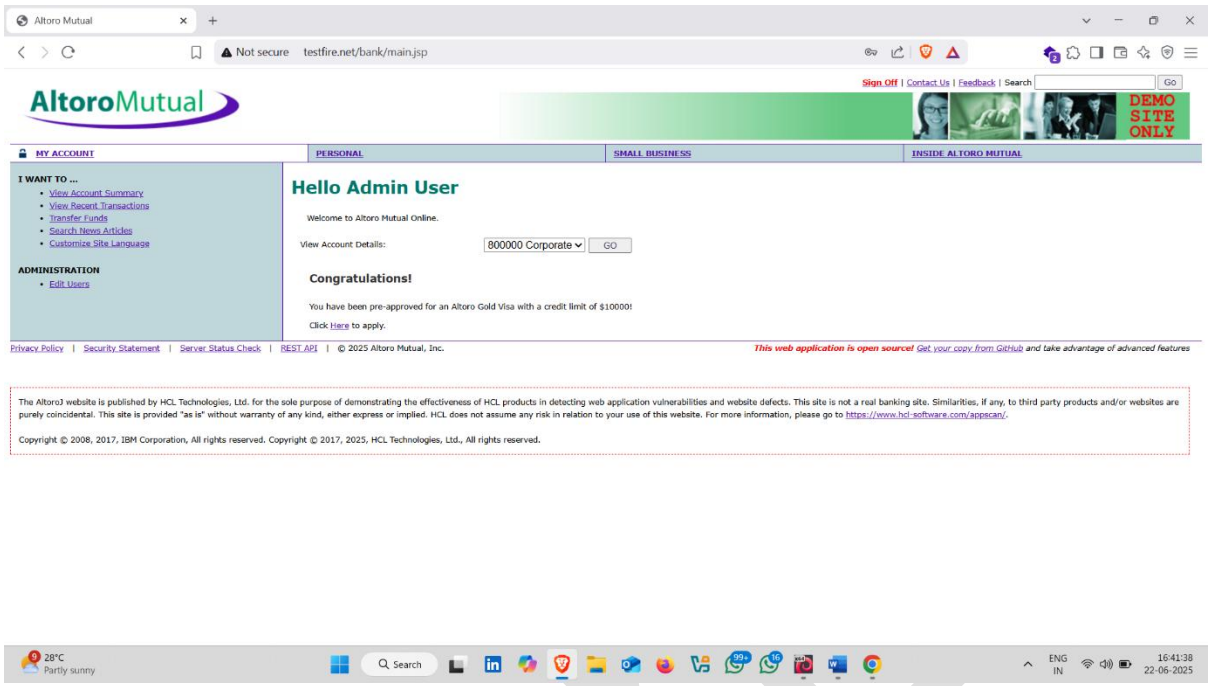
- Enter username and password



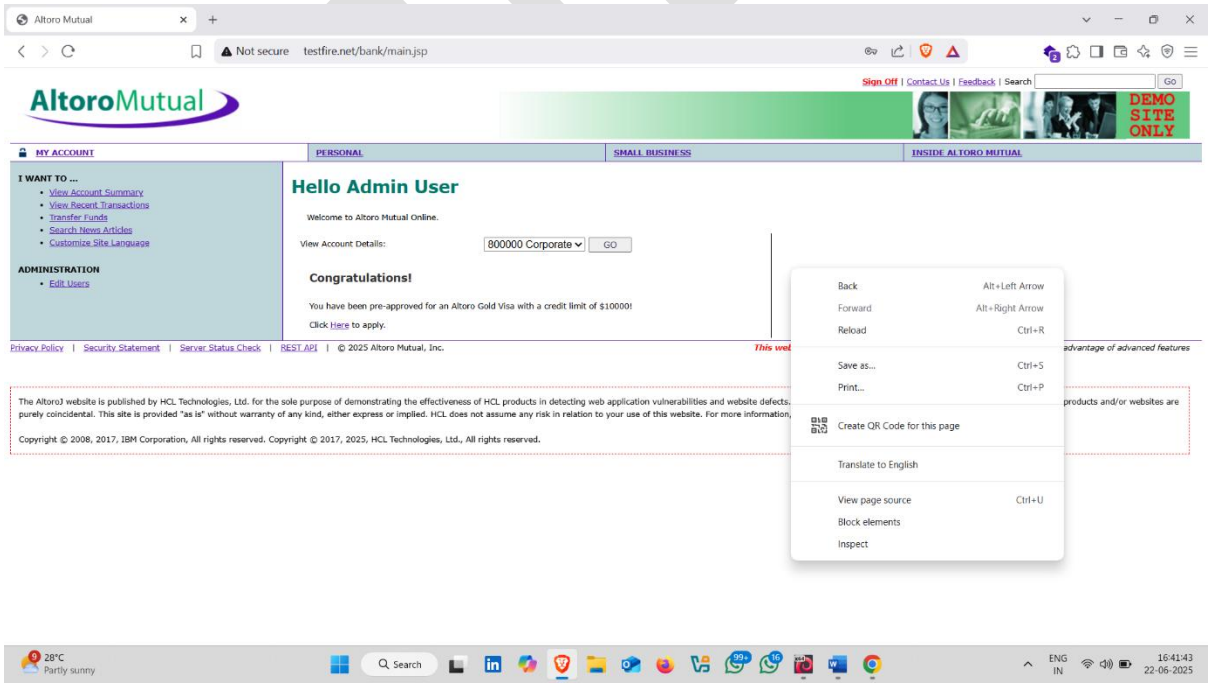
- Click on Login



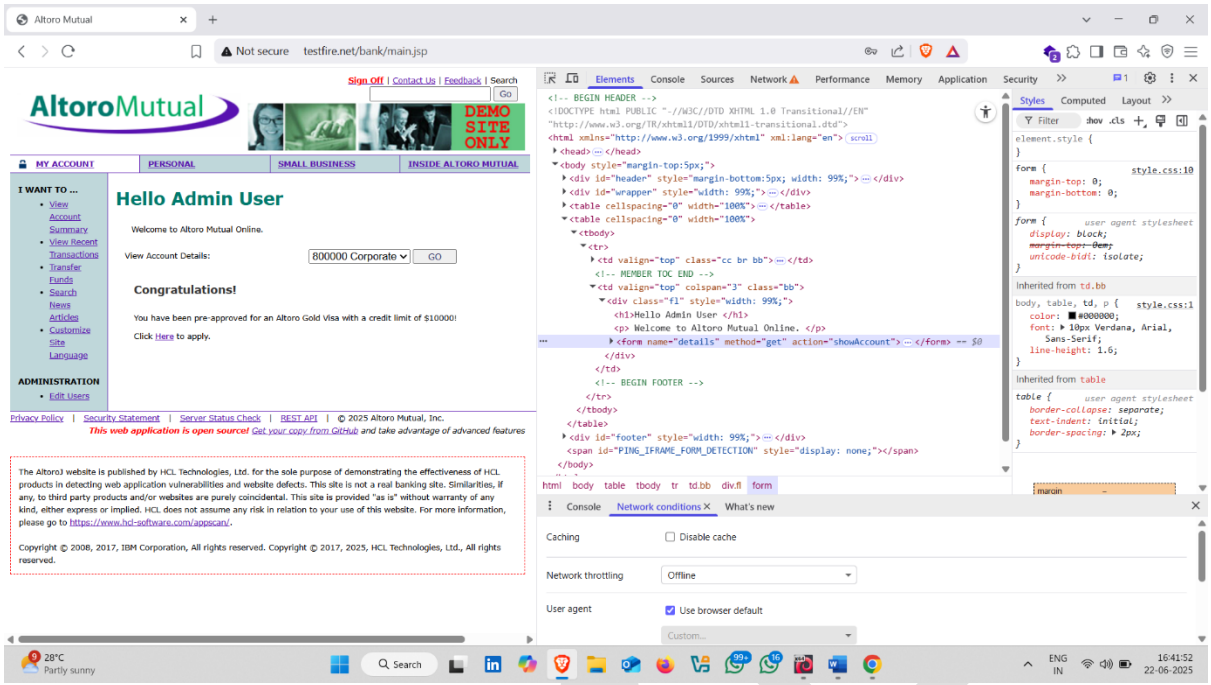
- **Login Successful 👇**



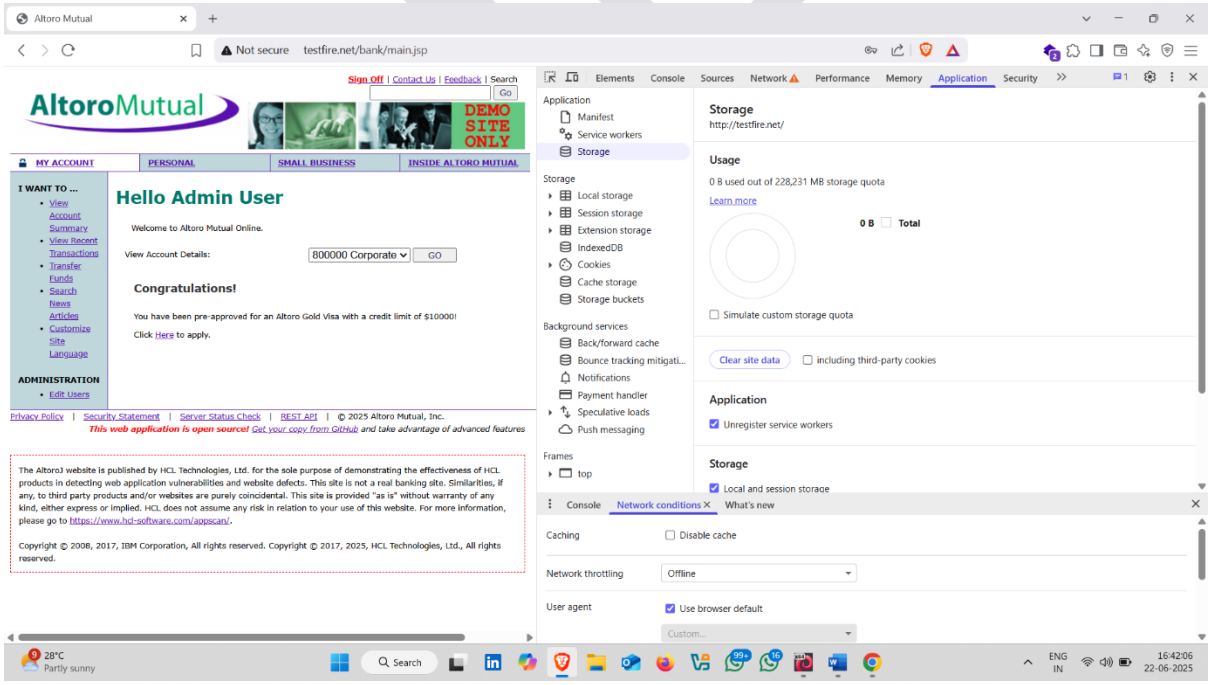
- Right click on page and then click on inspect option



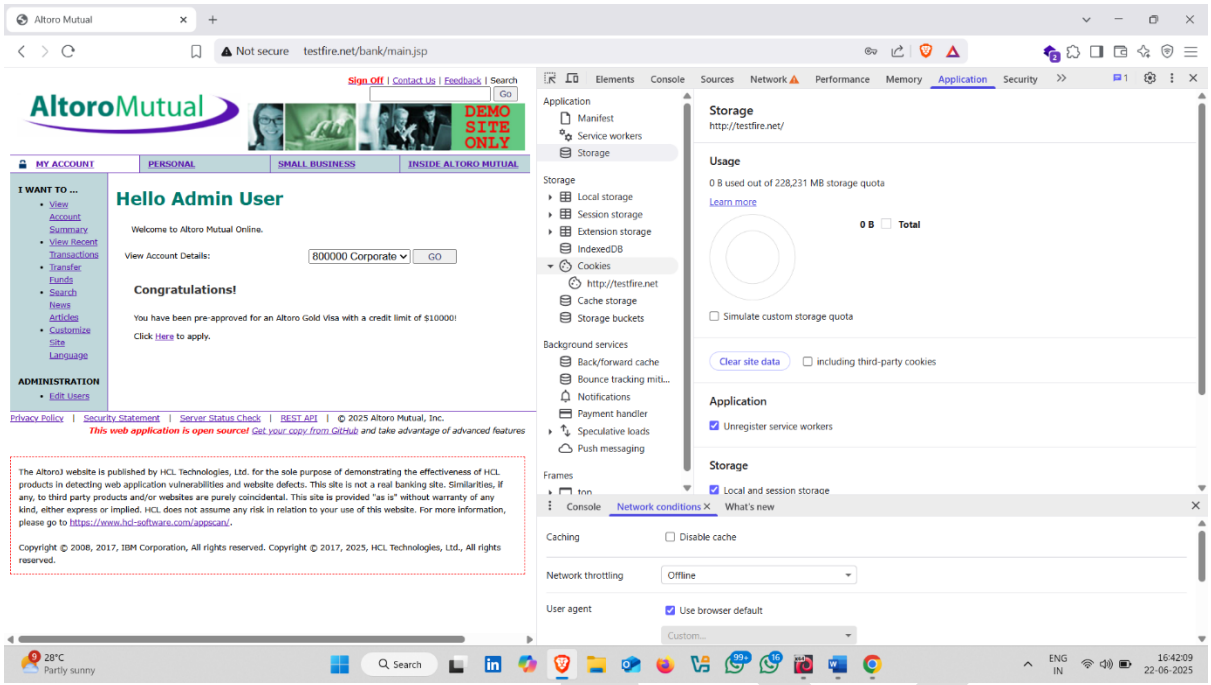
- Now click on **Application** option



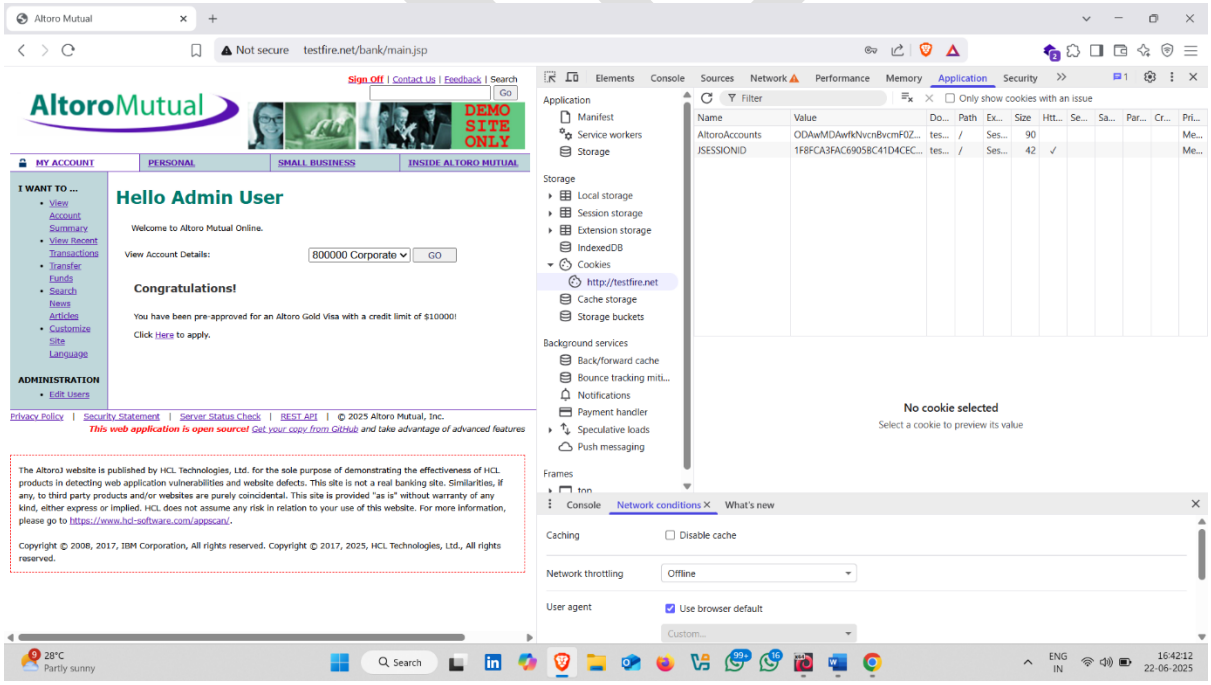
- Now , Click on **Cookies** Section 👇



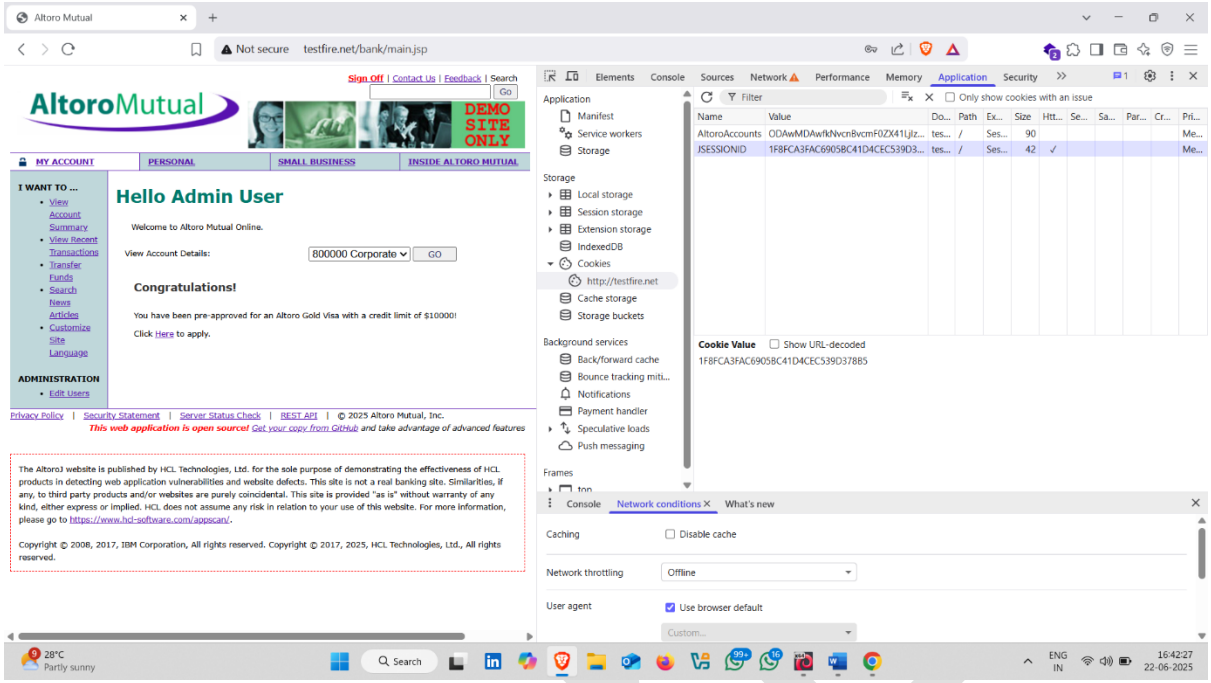
- Click on **Website**



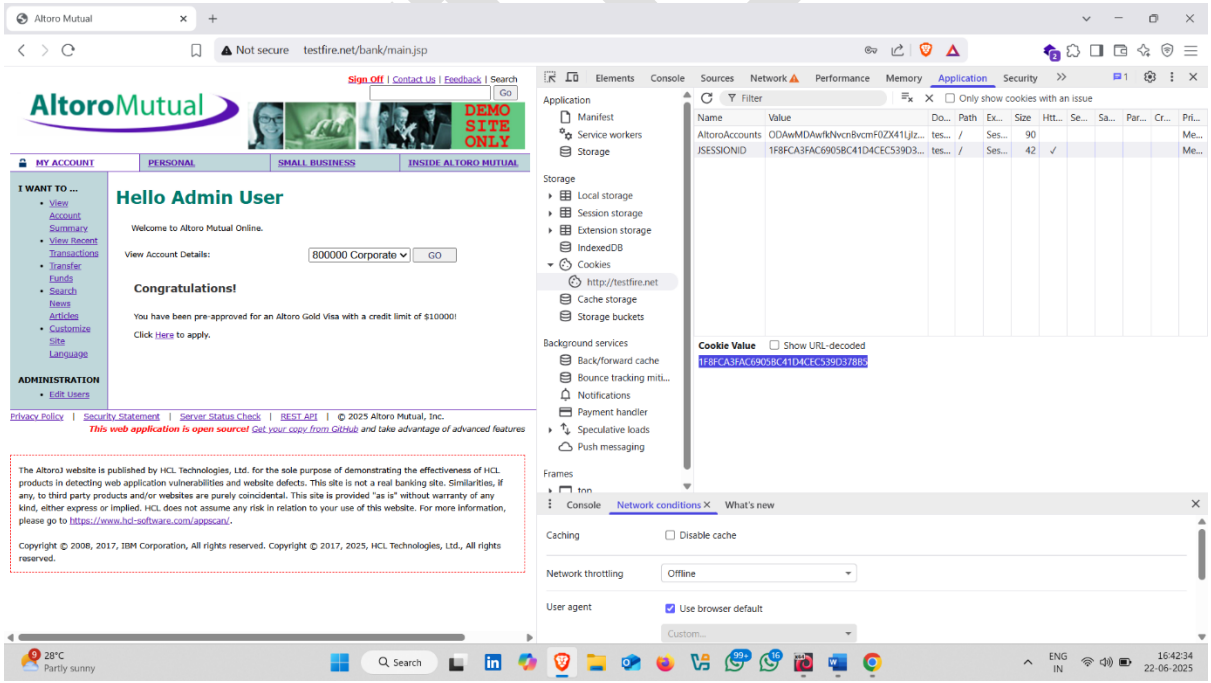
- Click on **JSESSIONID** 👇 ✅



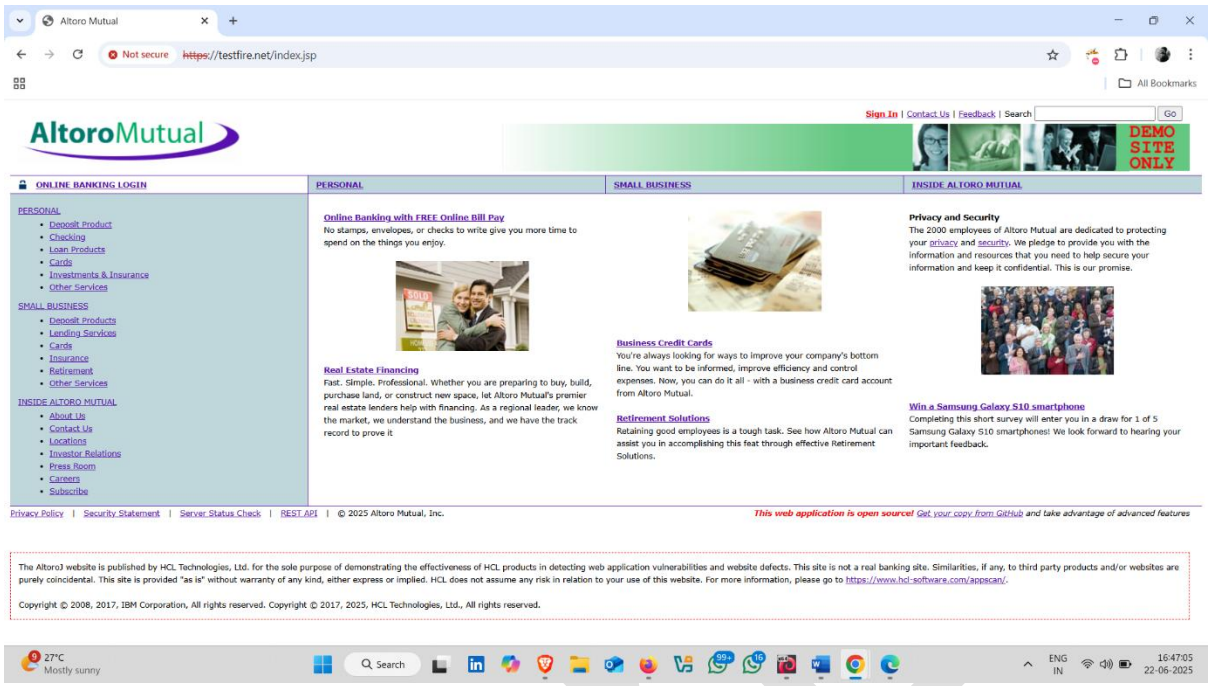
- Session Id ✅



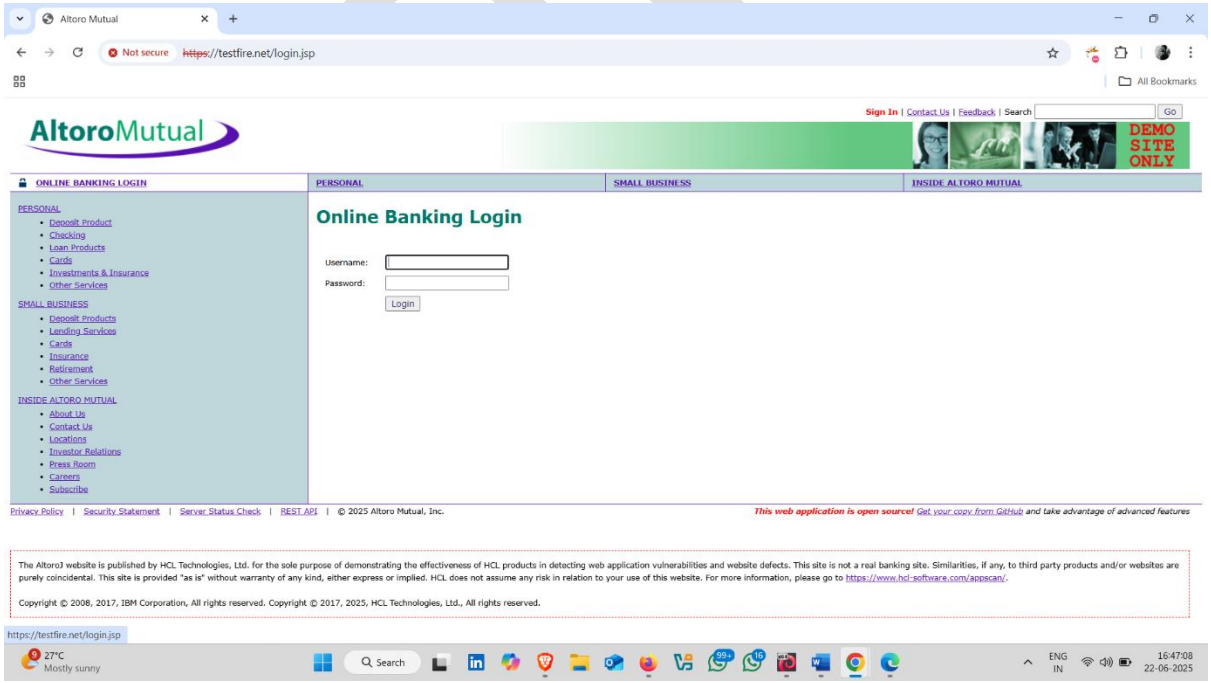
- Now, copy this **session Id** ✅ 👇



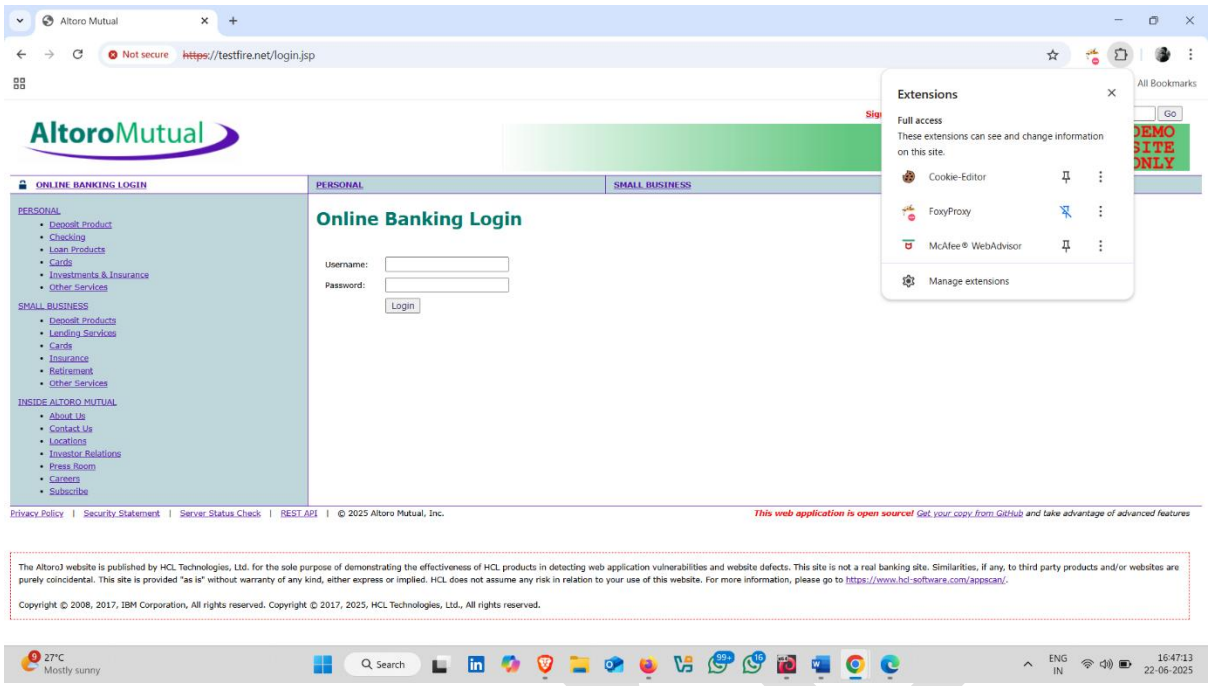
- Now Switch to another browser and open same website 👇 ✅



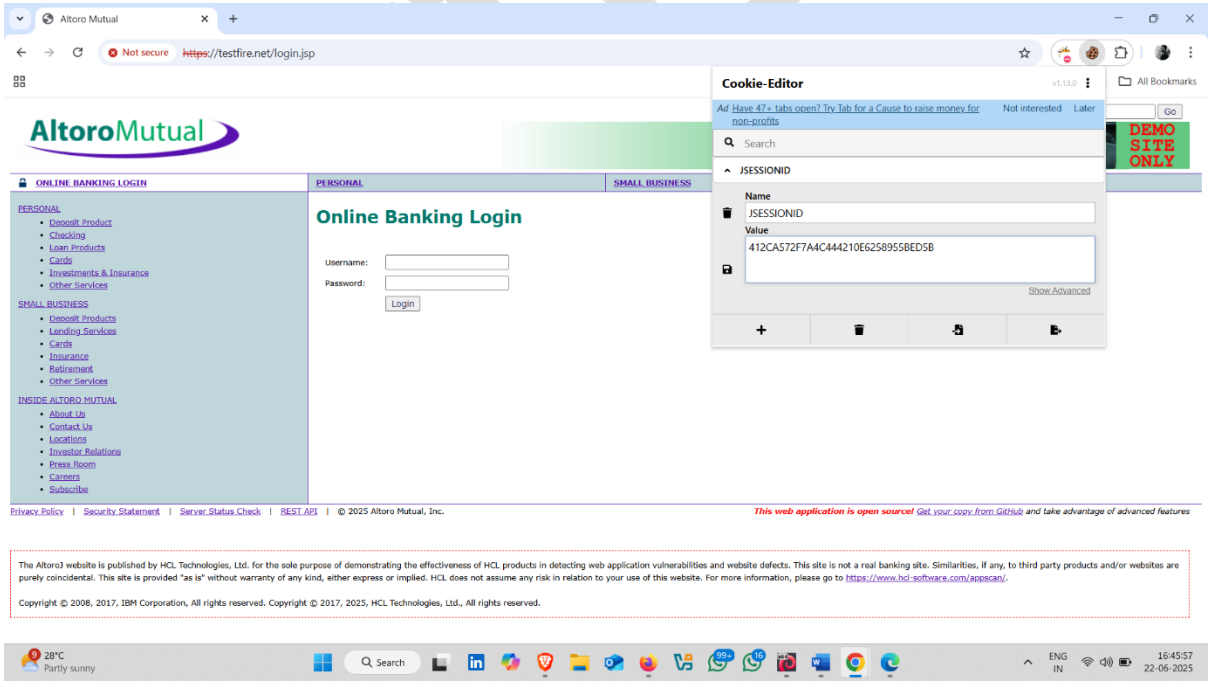
- Sign in Option 👇 , **But assume that we don't have any username and password but we have a session Id**



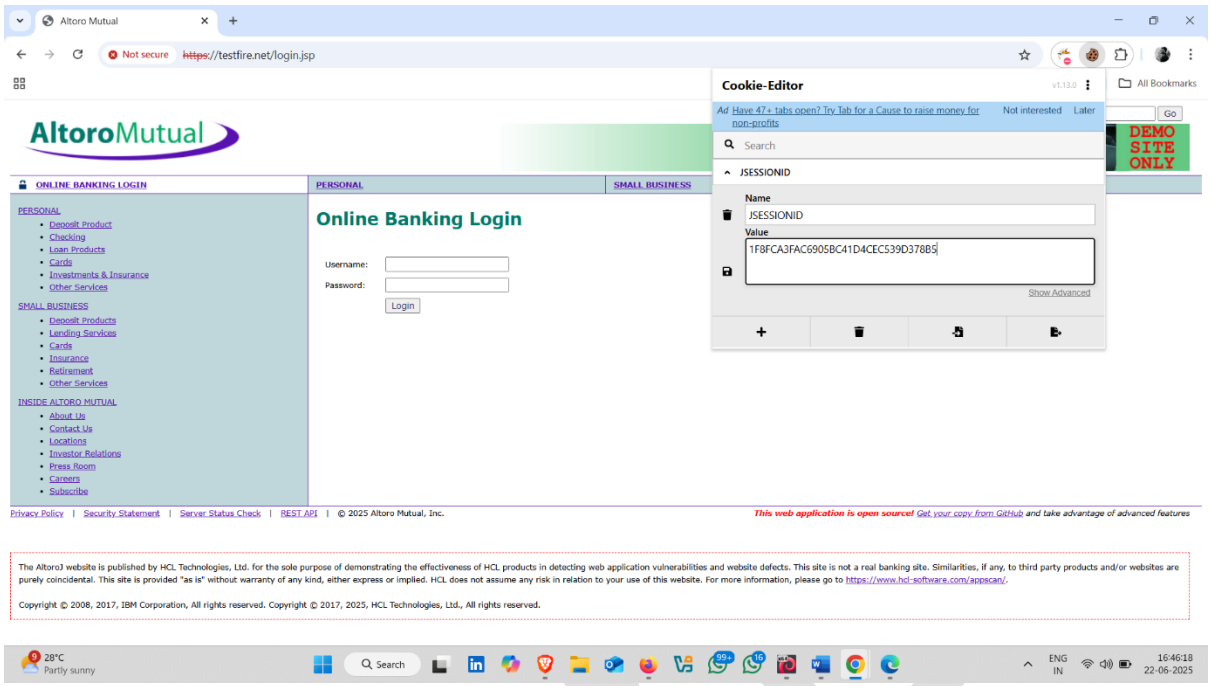
- Open cookies Editor Extension 👇 ✅



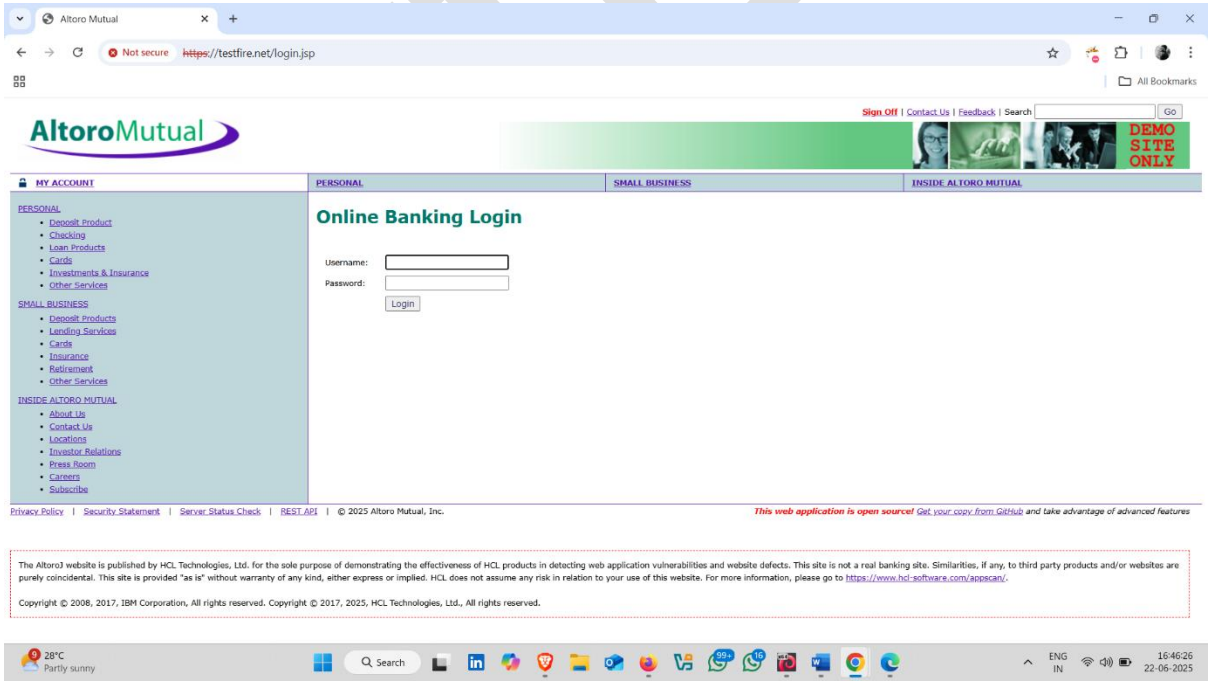
- Replace this Session id 👇



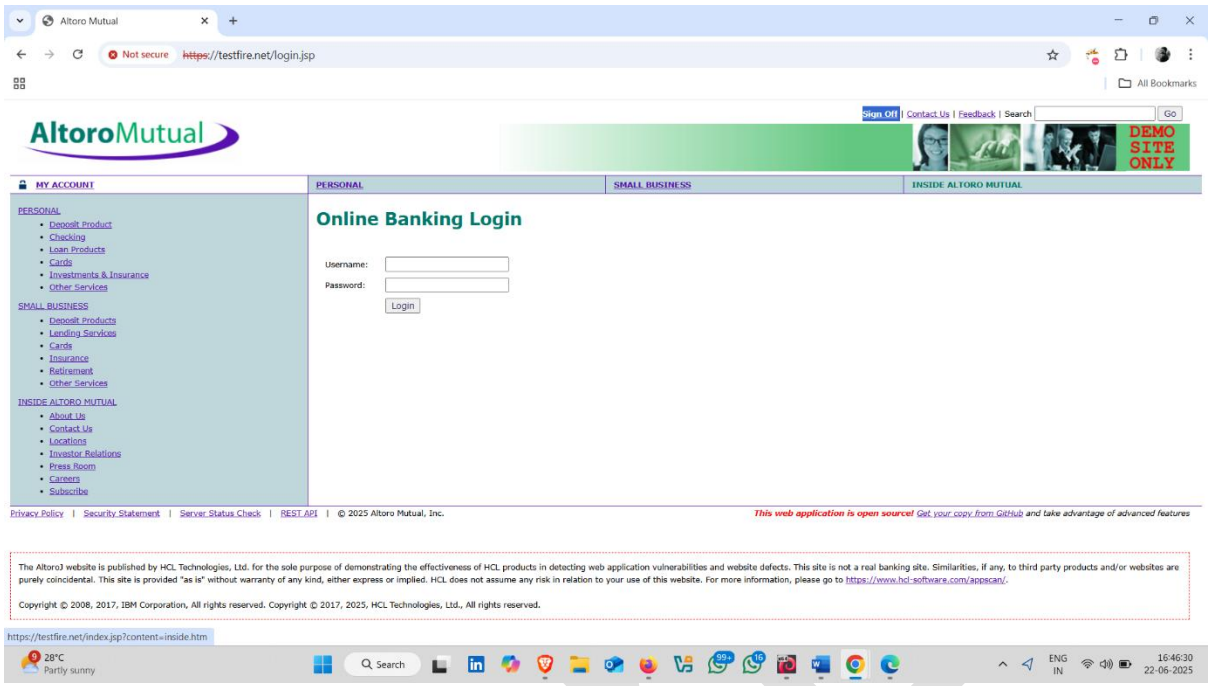
- With this 👇 ✅, And click on Save ✅.



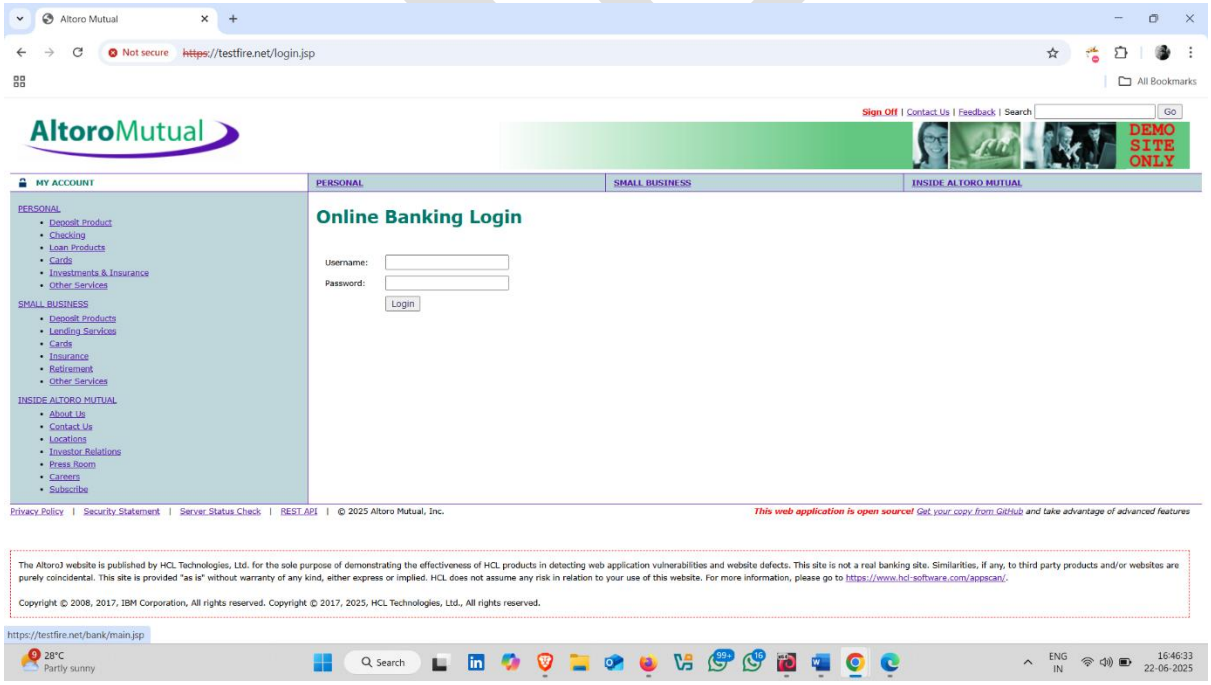
- Login Successful ✅



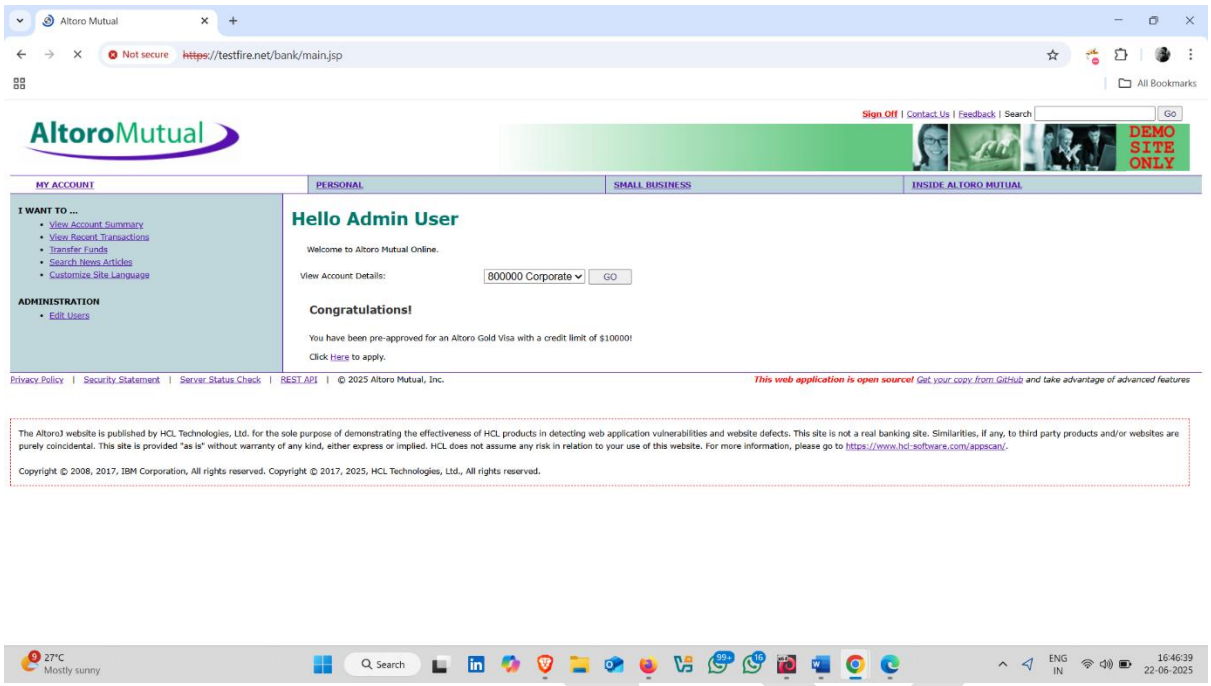
- Sign off option ✅ 👇 .. **Without username and password**



- Click on my account



- **Admin User account 👇✅**



🎯 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.
