SESSION HIJACKING

-ATHARVA KATKAR

TABLE OF CONTENTS

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 Concepts:

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

1. **Active Hijacking**
    - Attacker takes over an active session.
    - User may get logged out or see unusual behavior.
2. **Passive Hijacking**
    - Attacker only monitors the session.
    - Used mainly to steal sensitive data.

3. **Session Fixation**
    - Attacker forces a known session ID on the user.
    - After login, attacker reuses the same session ID.
4. **Man-in-the-Middle (MITM)**
    - Attacker intercepts traffic between user and server.
    - Common on unsecured Wi-Fi networks.
5. **Cross-Site Scripting (XSS) Based Hijacking**
    - Malicious script steals cookies/session IDs.

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

Session Hijacking Using Wireshark

Target Website (testfire.net)

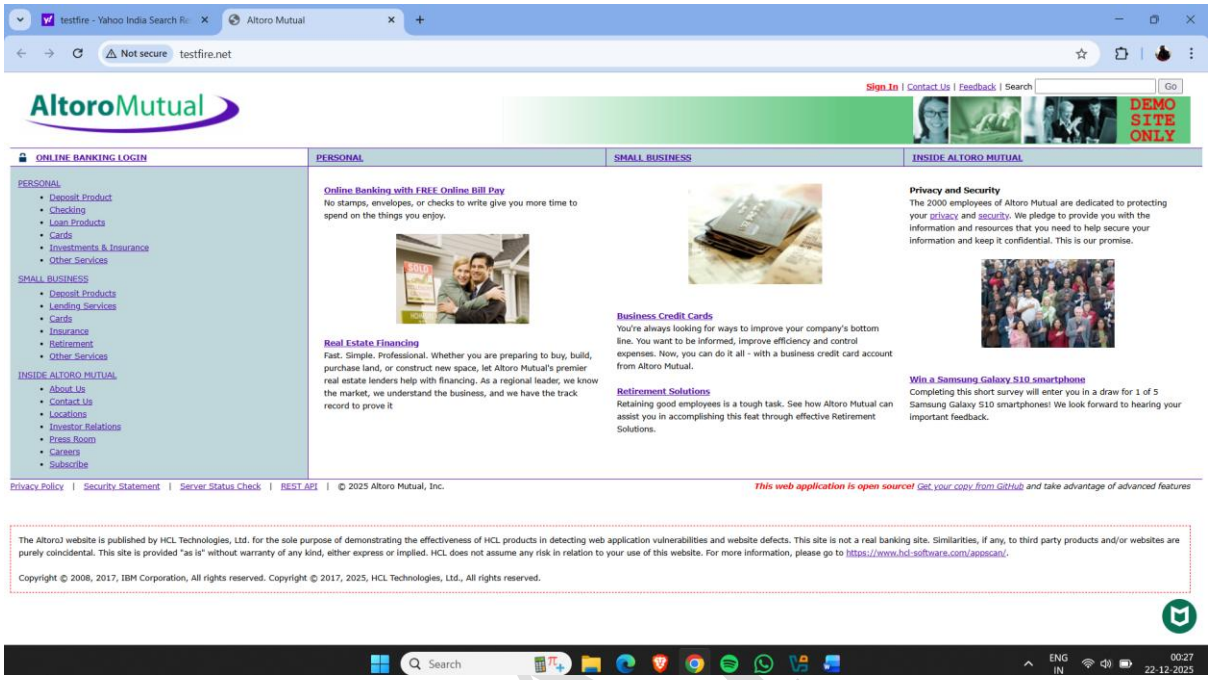


Figure 1

- Click on Sing in option and enter username and password

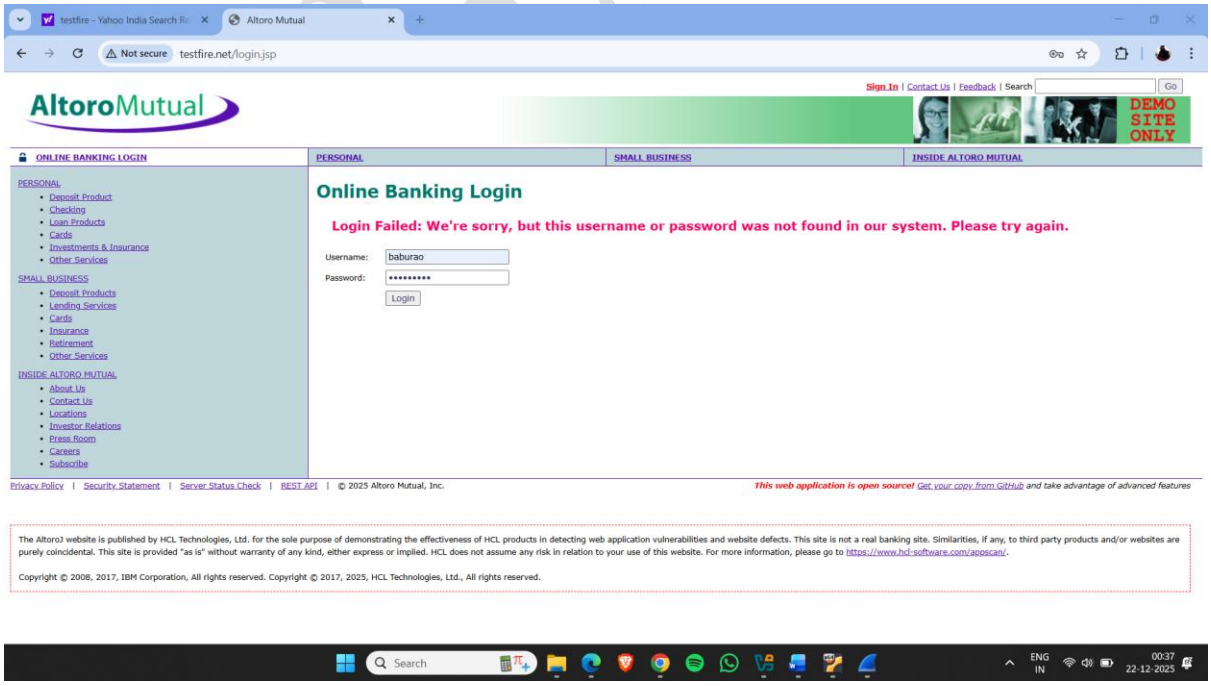


Figure 2

- Open wireshark to capture the packet
- find HTTP Post Request or search HTTP

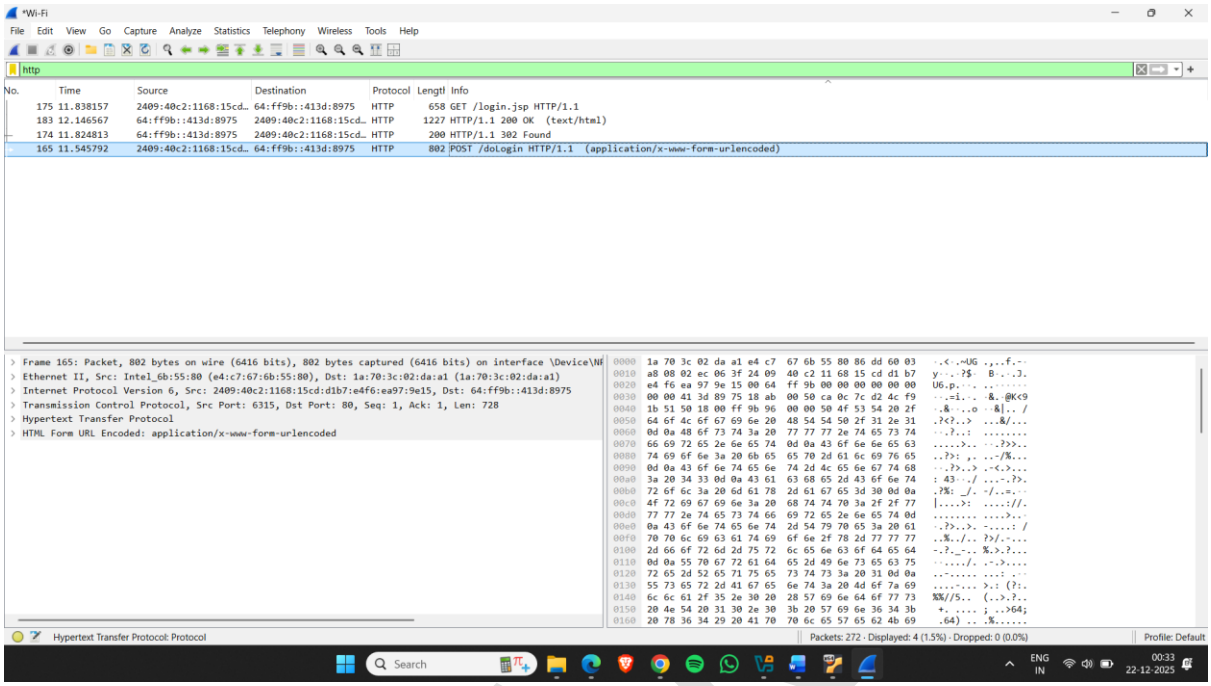


Figure 3

- HTTP post request found.

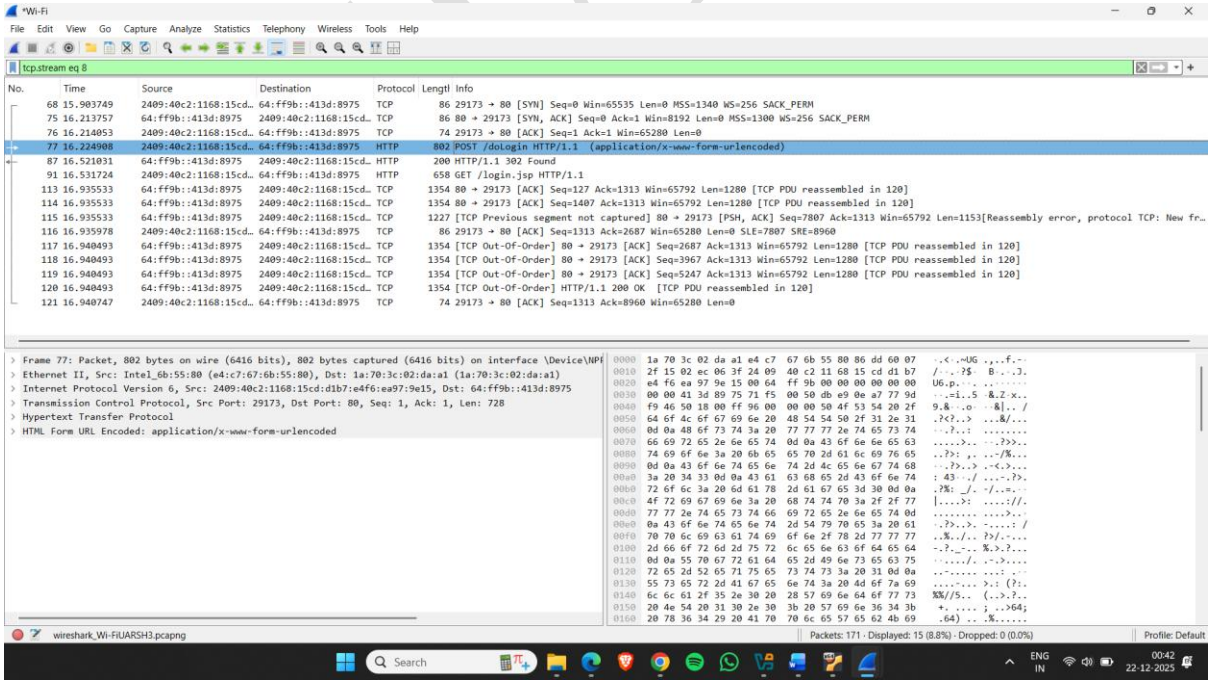


Figure 4

- Right click on POST Request
- Click on Follow and then click on TCP View
- Now copy **JSESSIONID.**

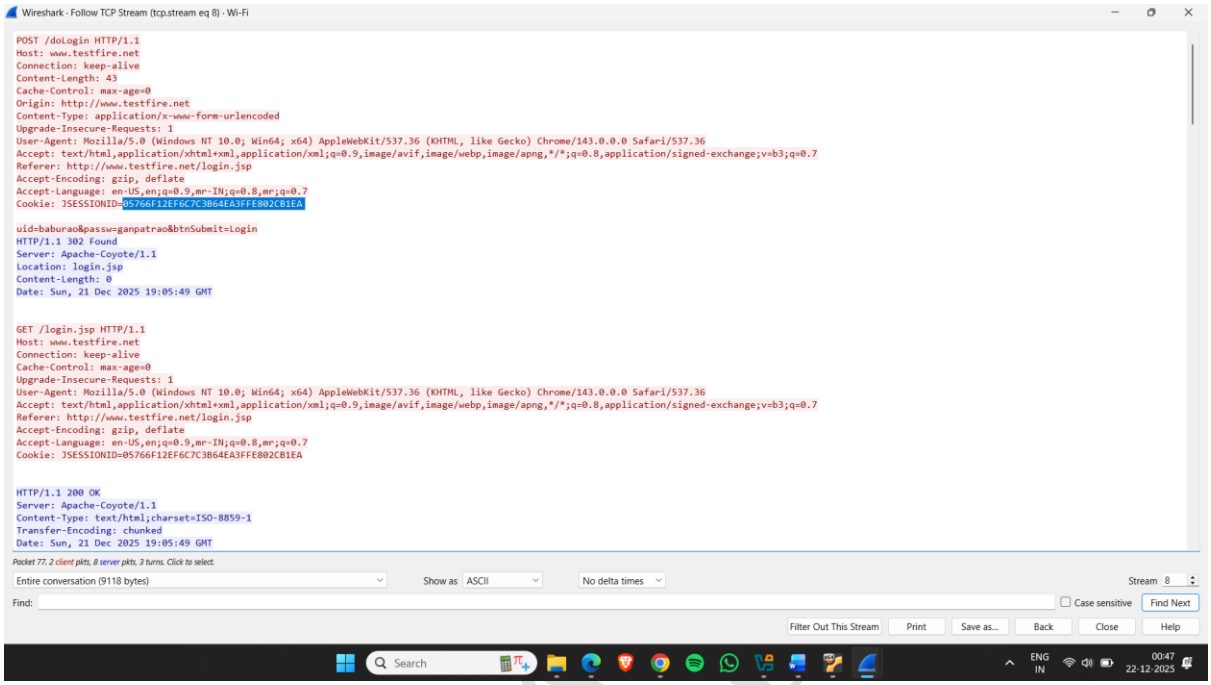


Figure 5

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

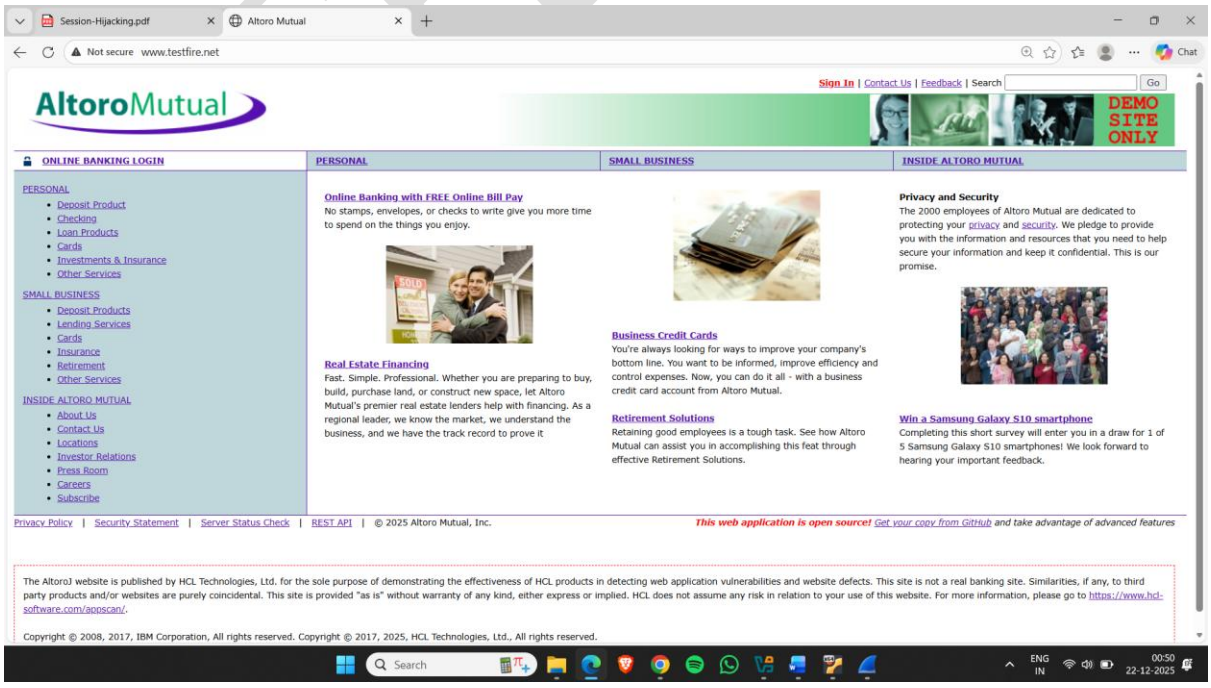


Figure 6

- Download Extension Cookies Editor extention

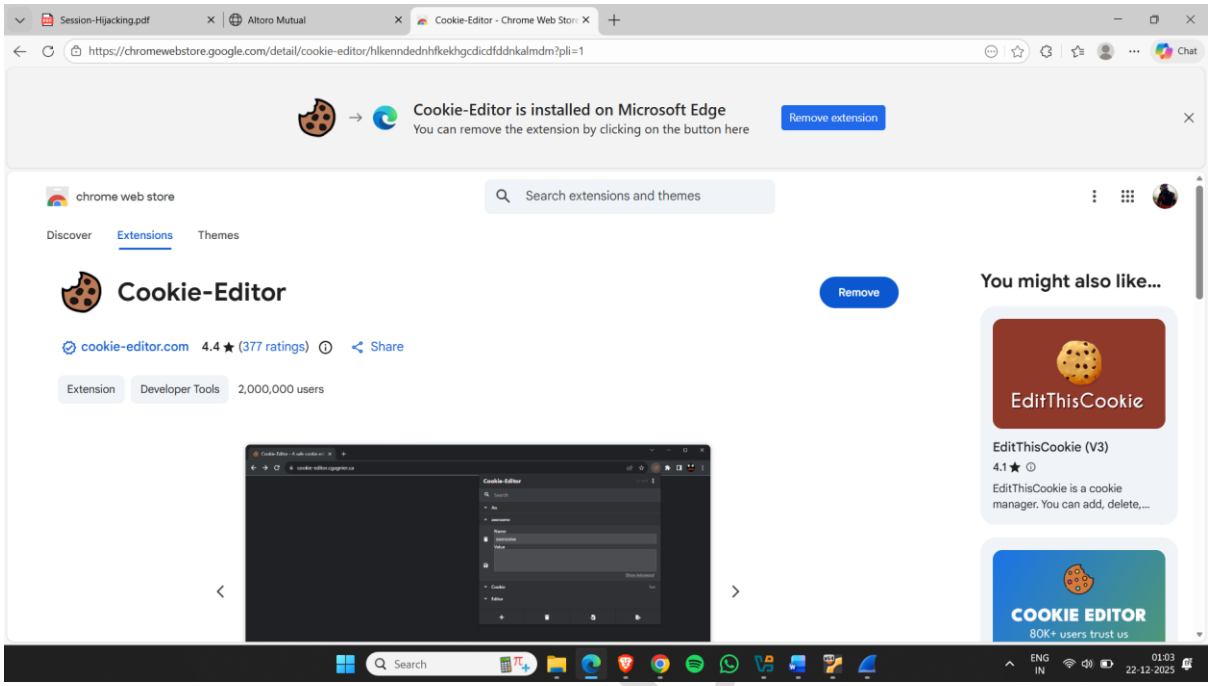


Figure 7

- Now open cookies editor extention and replace this JSESSIONID & Click on Save.

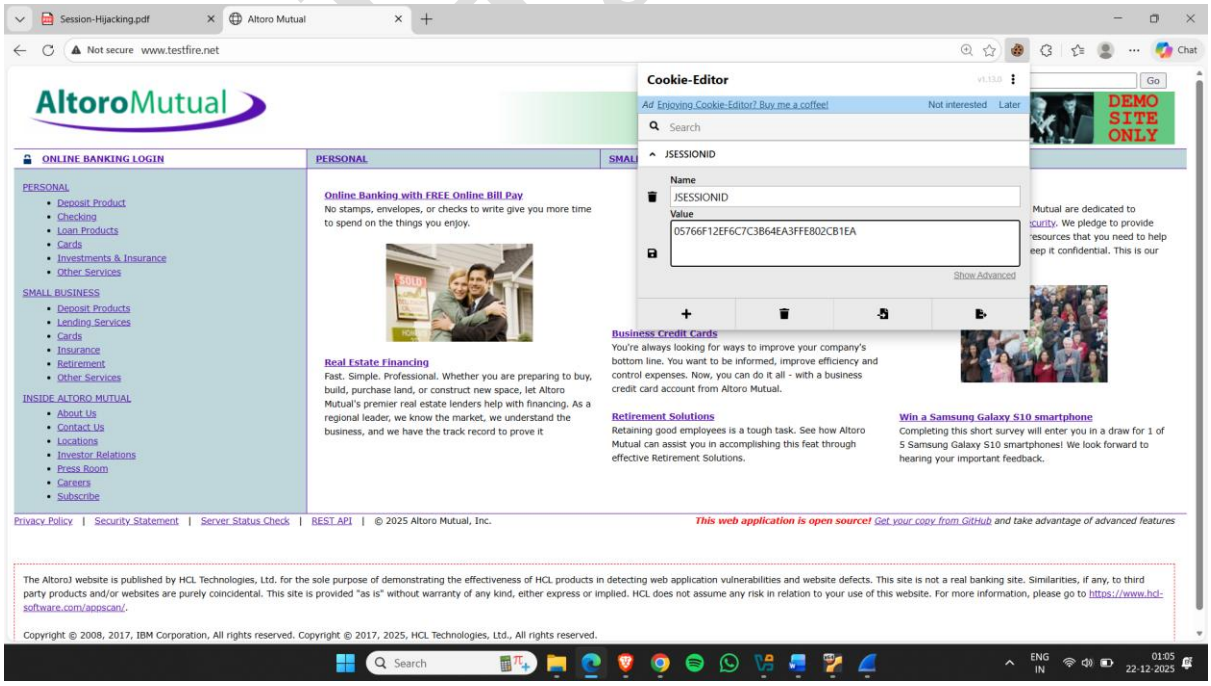


Figure 8

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

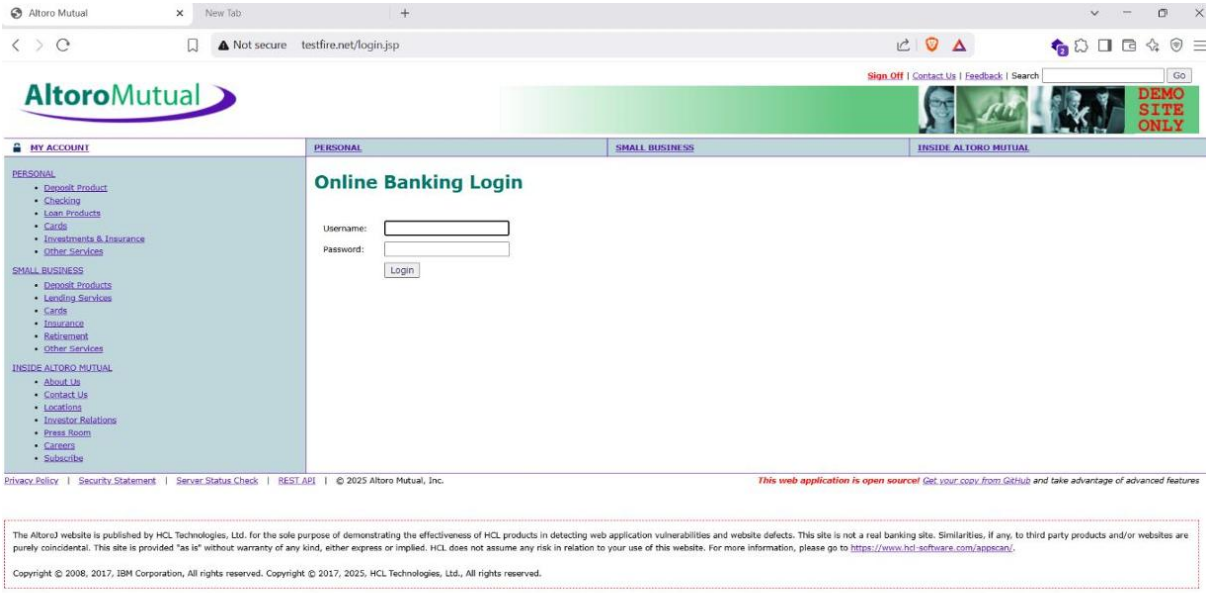


Figure 9

- **Logged in successfully..**

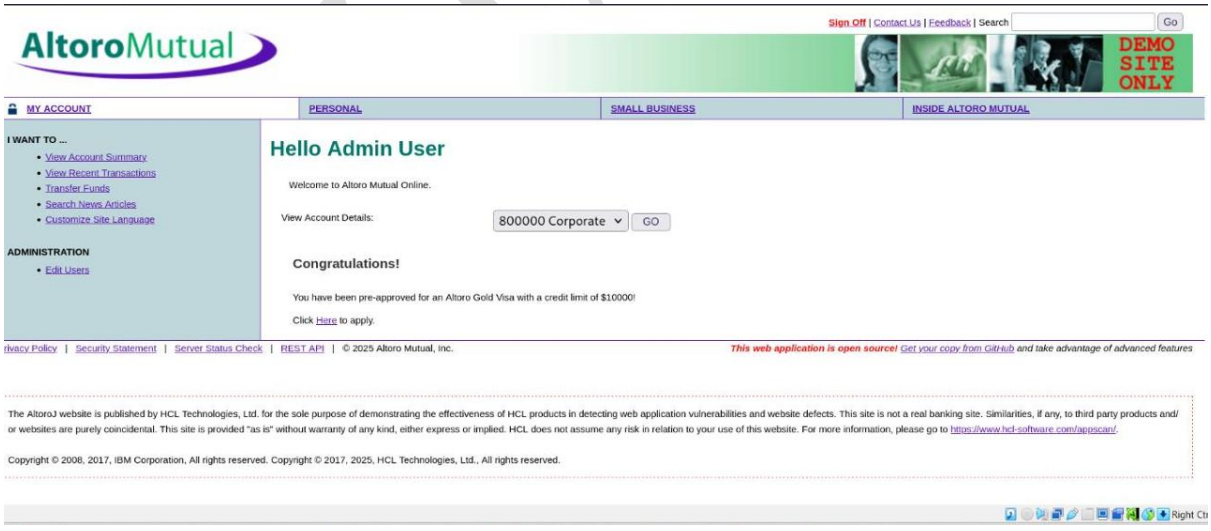


Figure 10

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.

Target Website

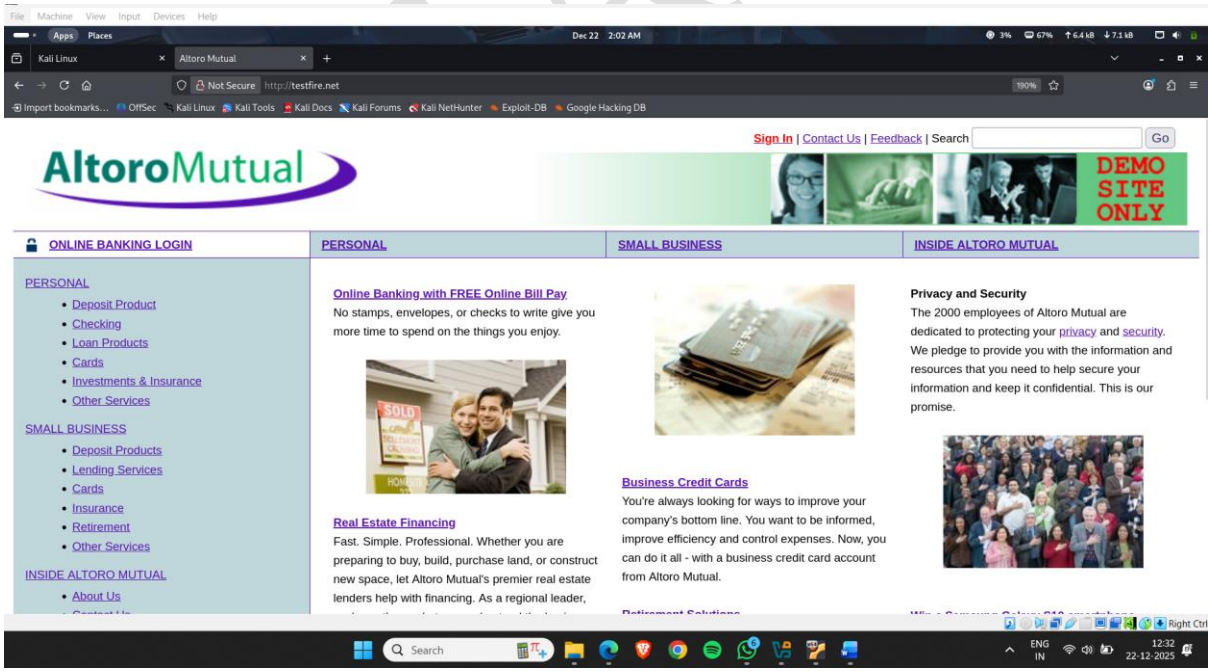


Figure 11

- Enter username and password and click on login
- Account Logged in..

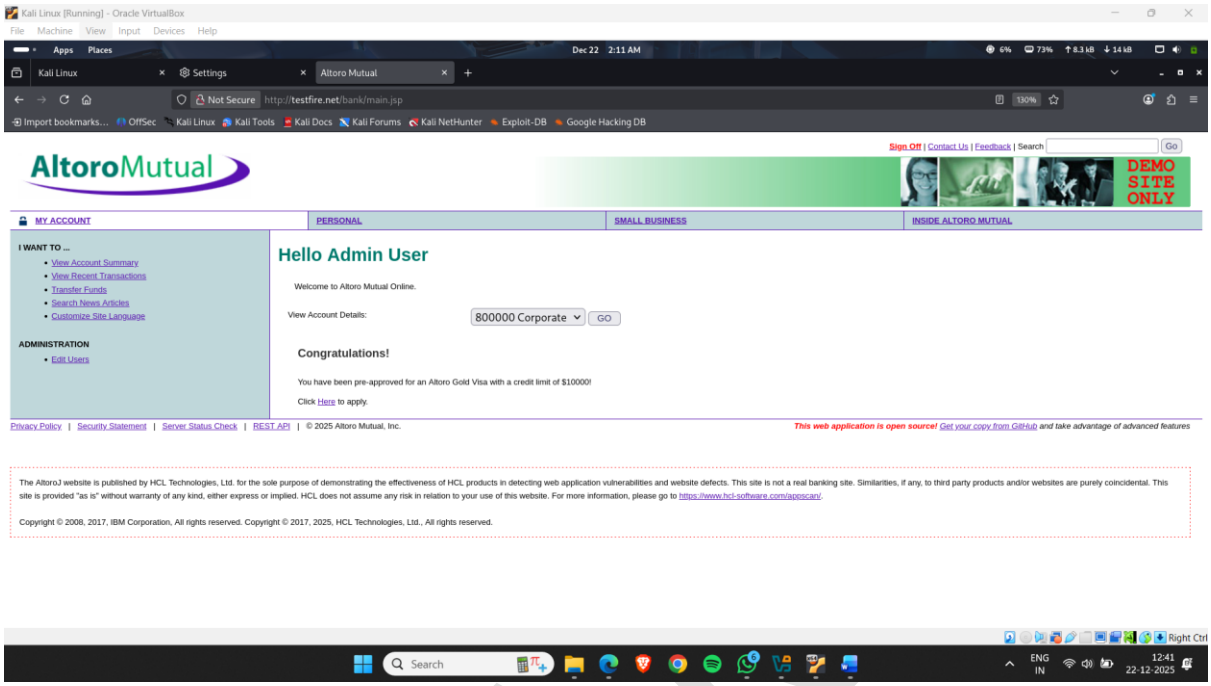


Figure 12

- Open burp suite and click on target option
- Now click on doLogin Option

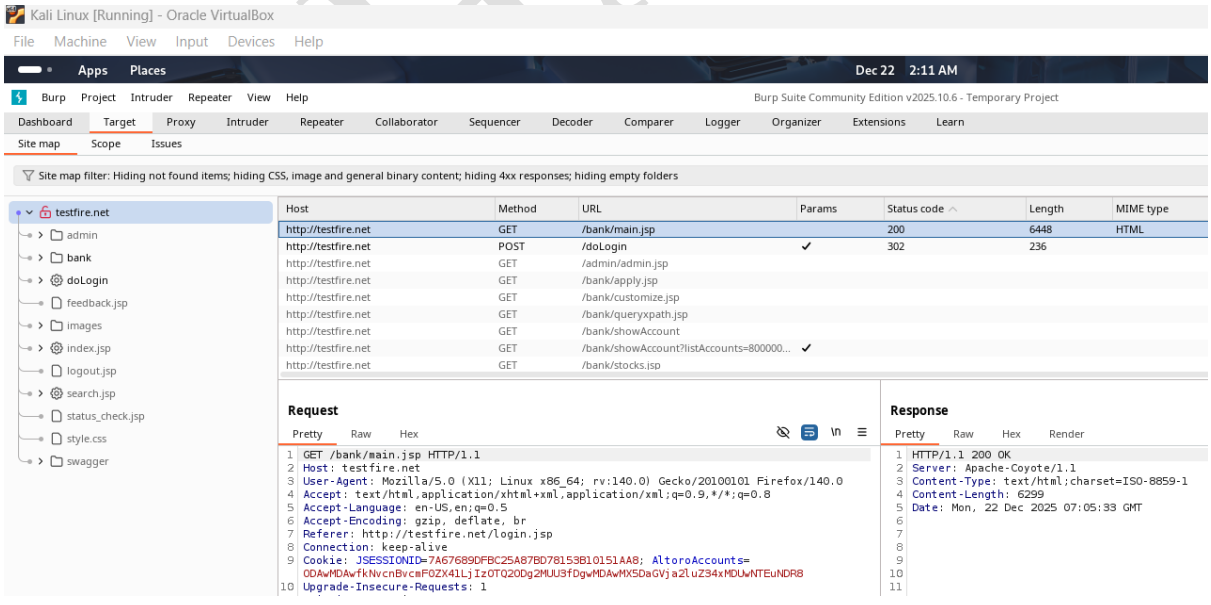


Figure 13

- Now , right click on uid and passw section & Send to Sequencer

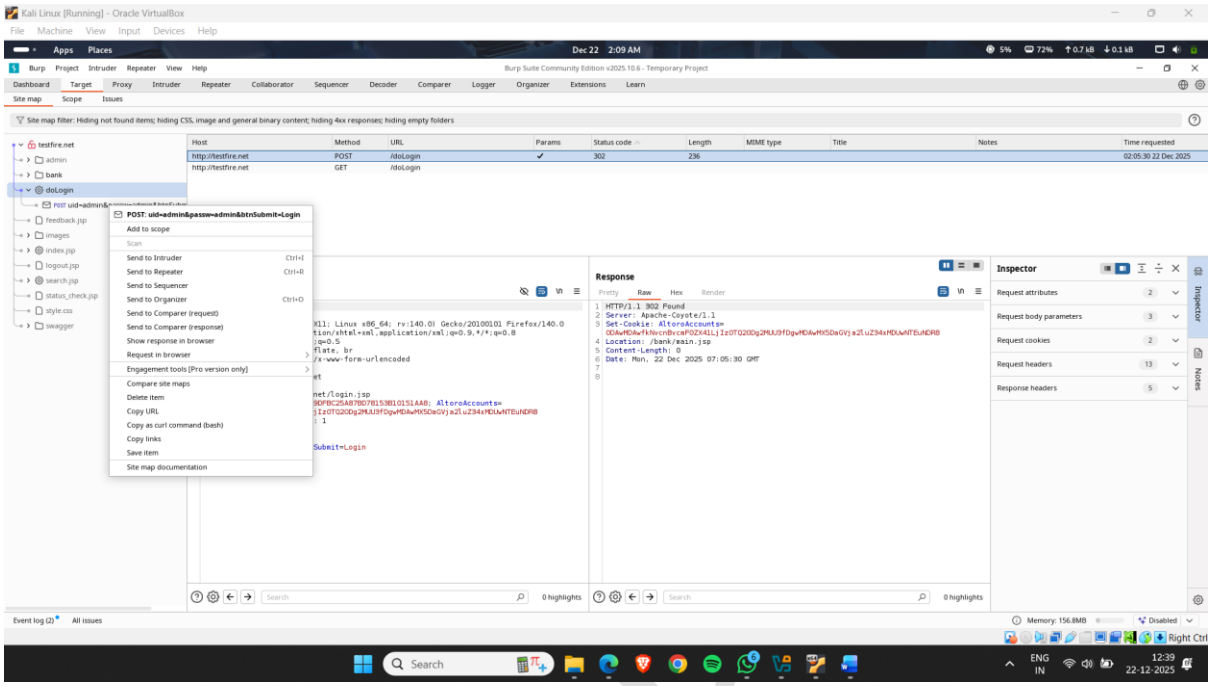


Figure 14

- Go on Sequencer & Click on Start live capture

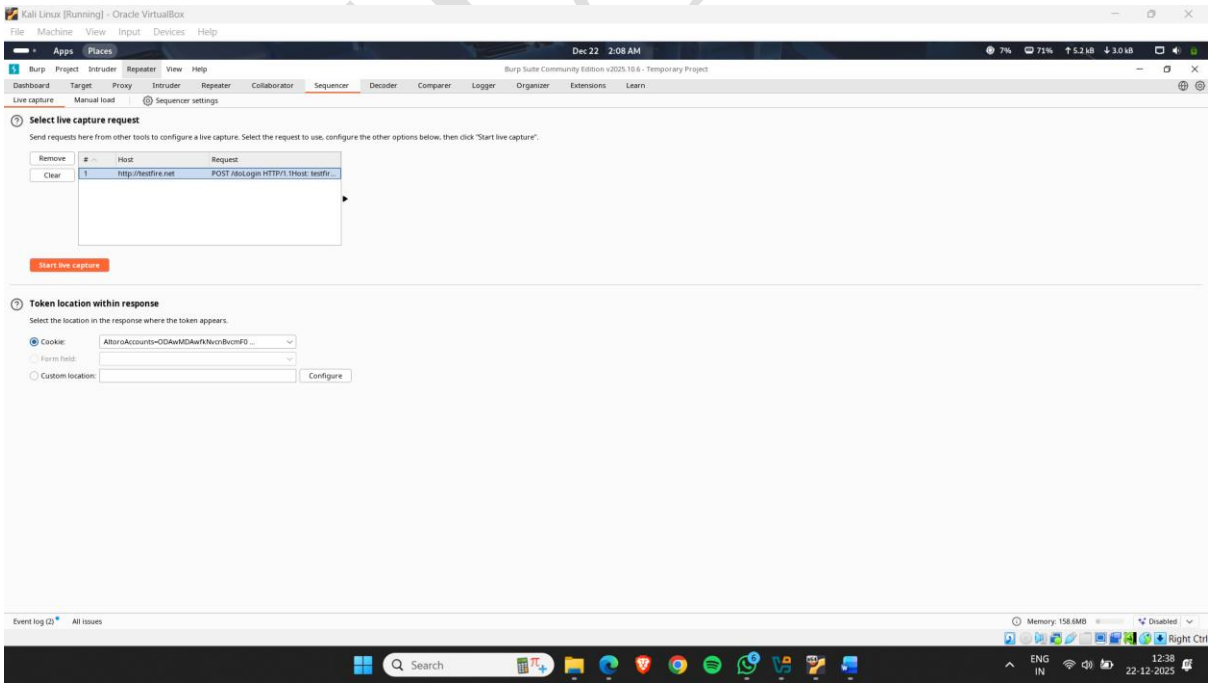


Figure 15

- Now click on Analyze now

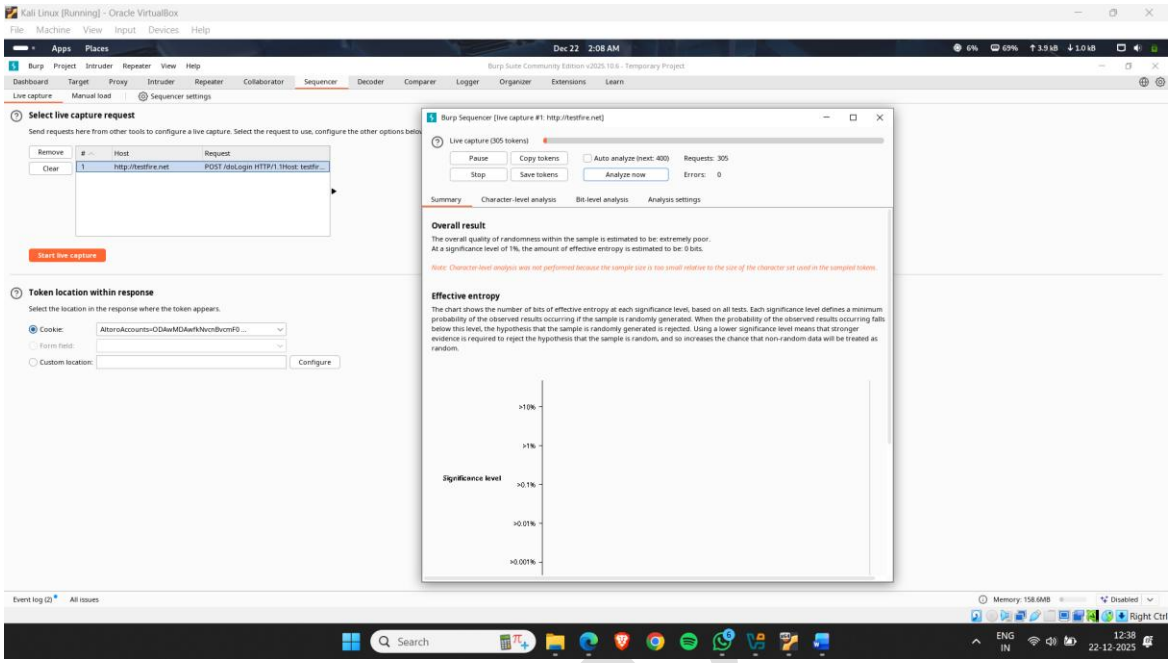


Figure 16

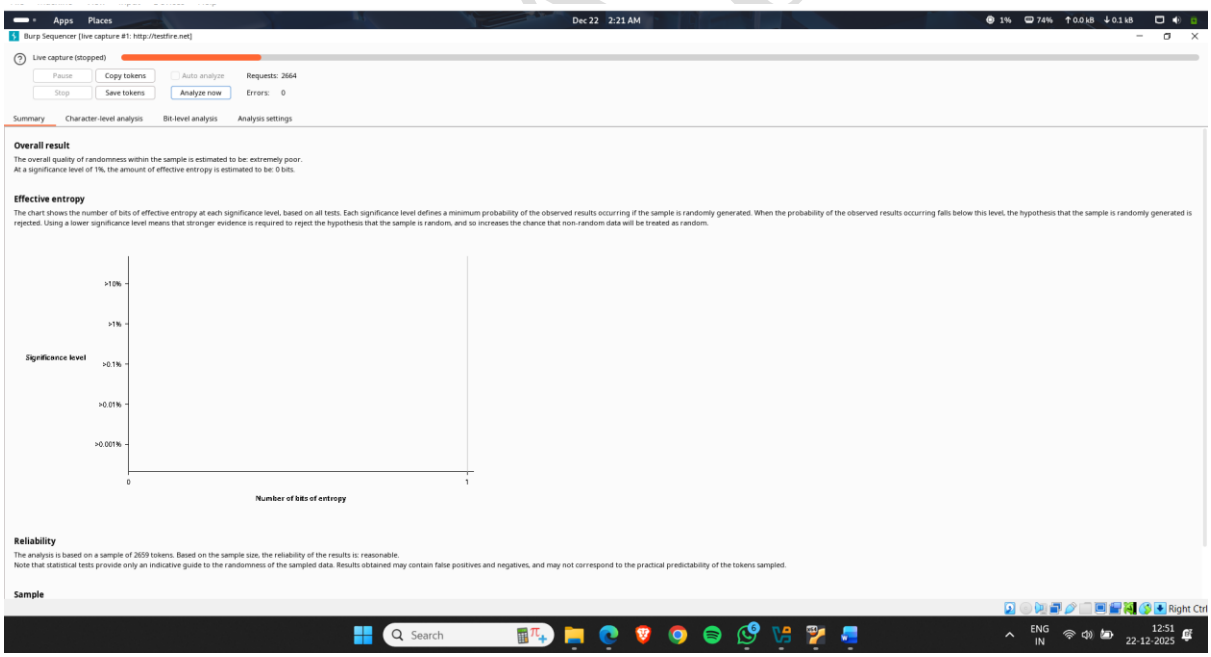


Figure 17

Overall Result

- Randomness quality: Extremely poor

👉 This means the captured tokens are highly predictable and not random enough.

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.

- Open Ettercap tool

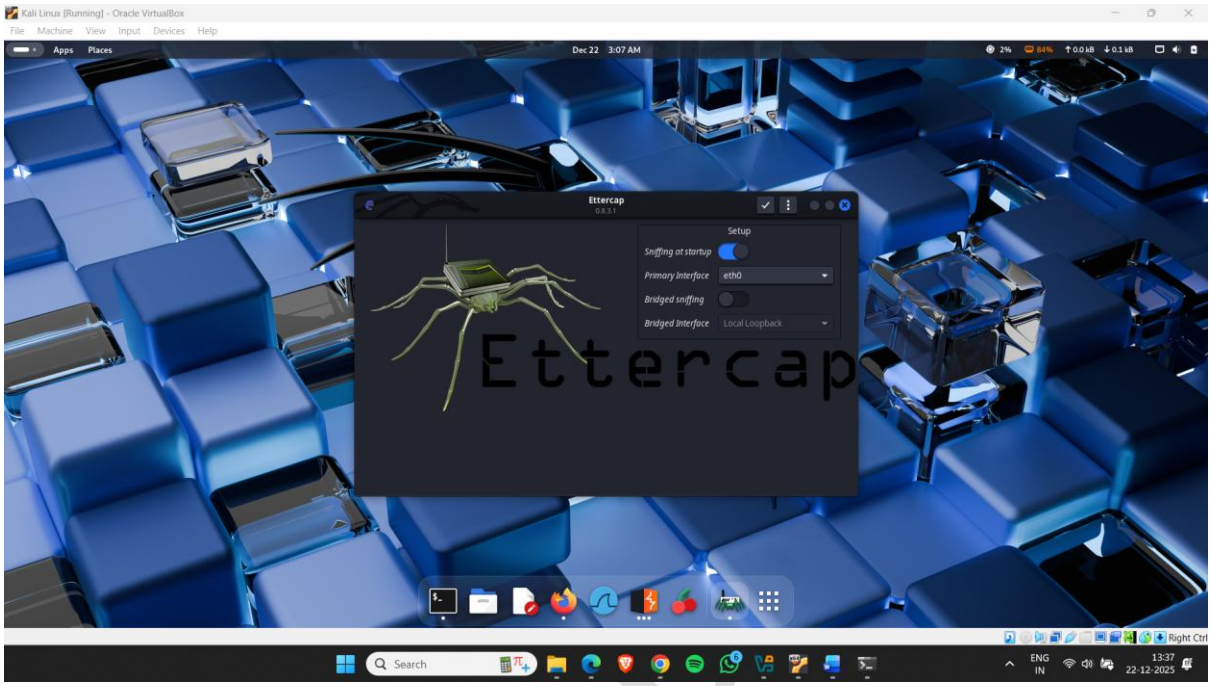


Figure 18

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

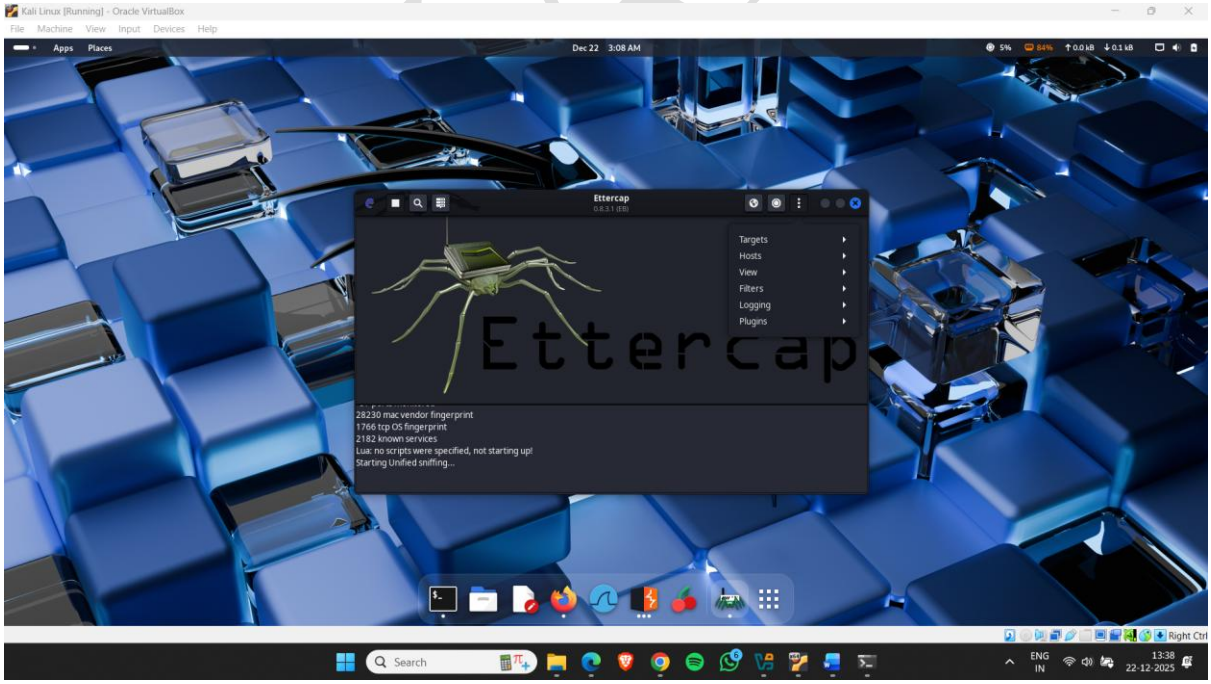


Figure 19

- Click on Scan For Hosts

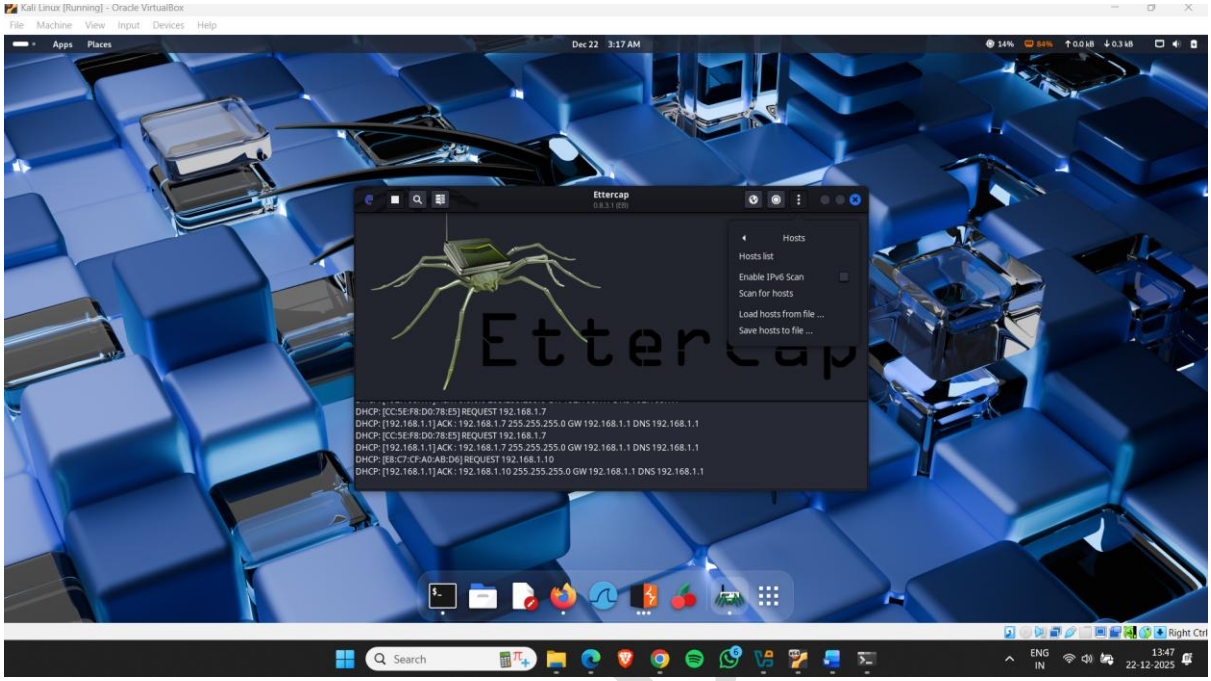


Figure 20

- 32 Hosts Scanned..

Figure 21

- Click on Three Dot and click on Host List

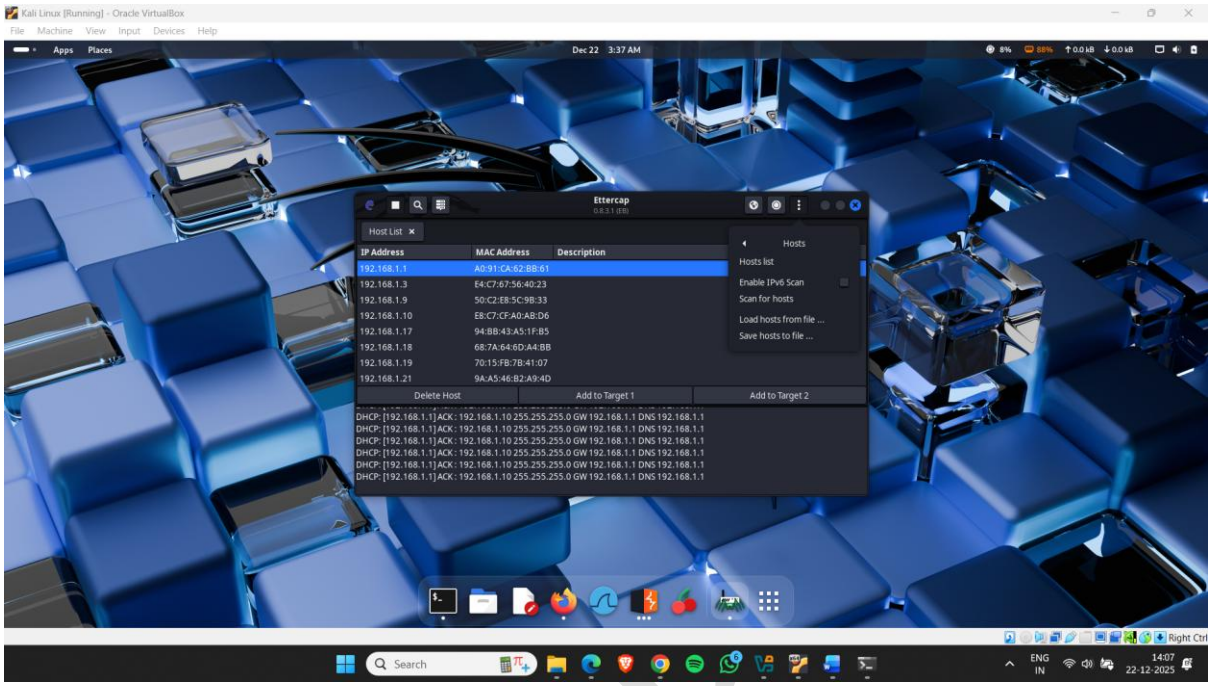


Figure 22

- Host list appears

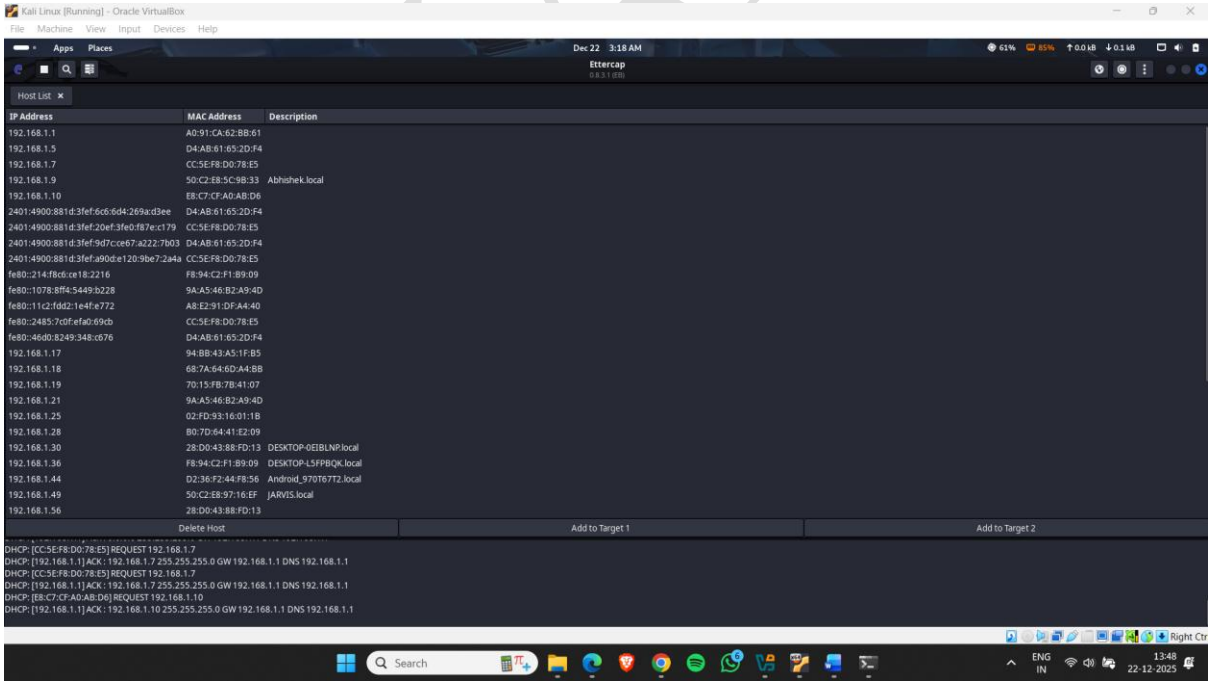


Figure 23

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

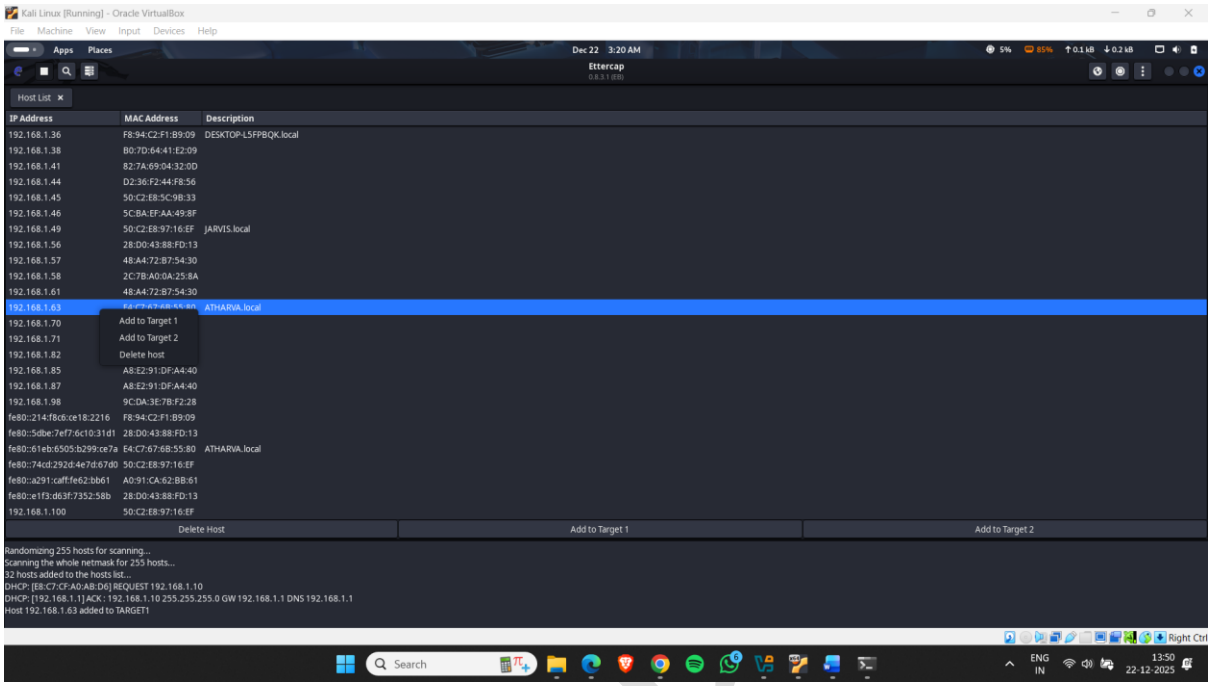


Figure 24

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

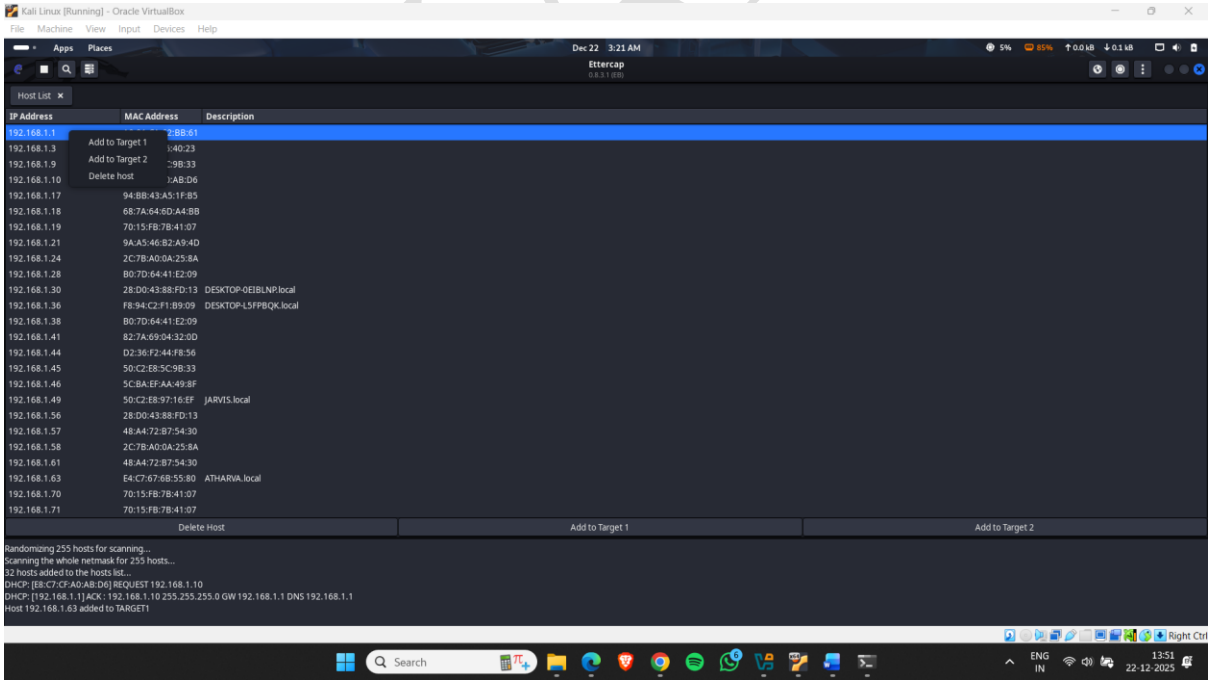


Figure 25

- Then , click on Earth shape icon and then click on ARP Poisoning

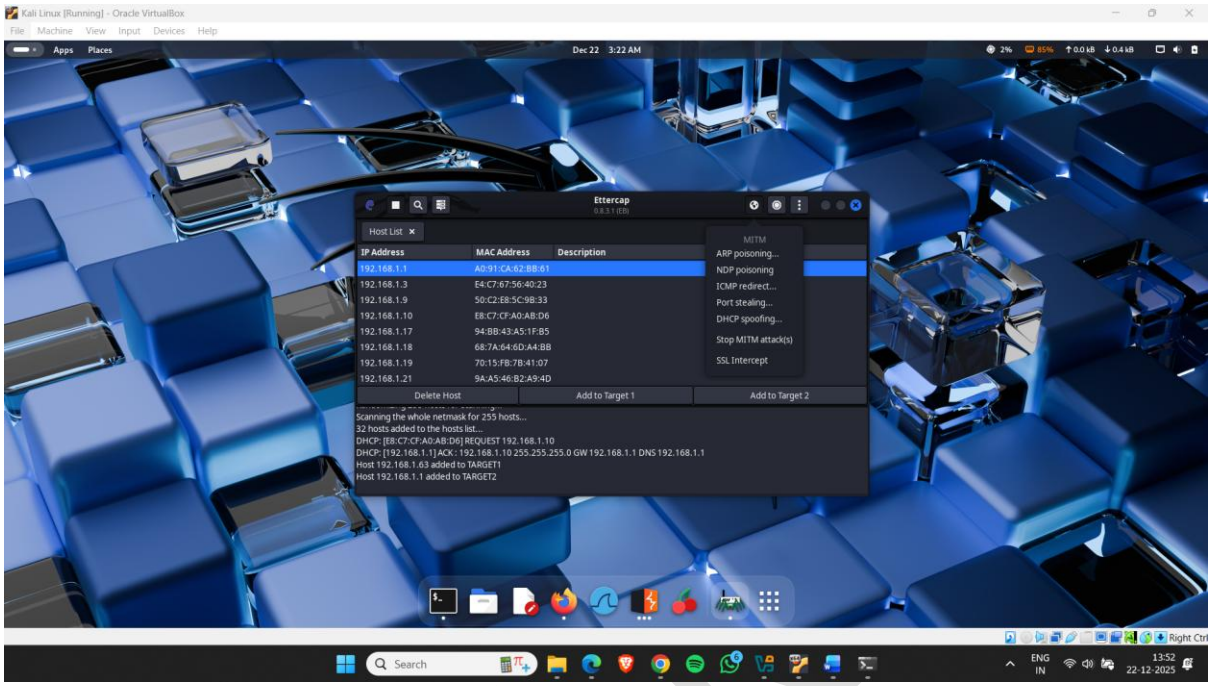


Figure 26

- Check -- Sniff Remote Connections and click on OK

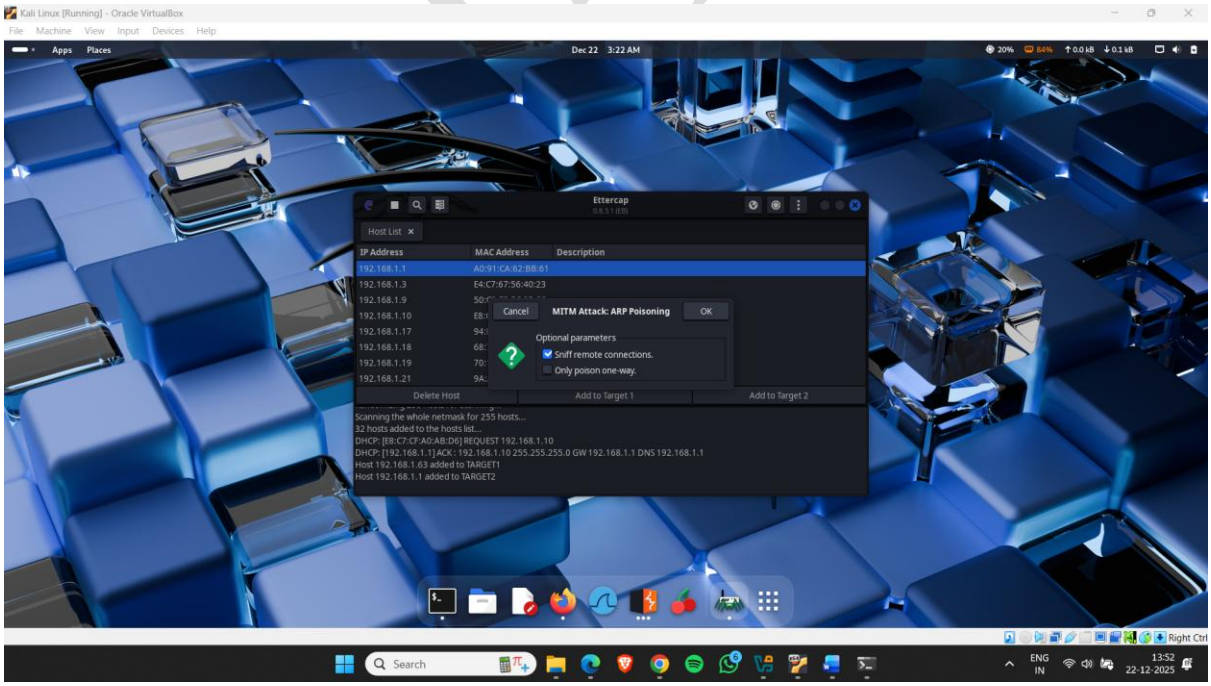


Figure 27

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

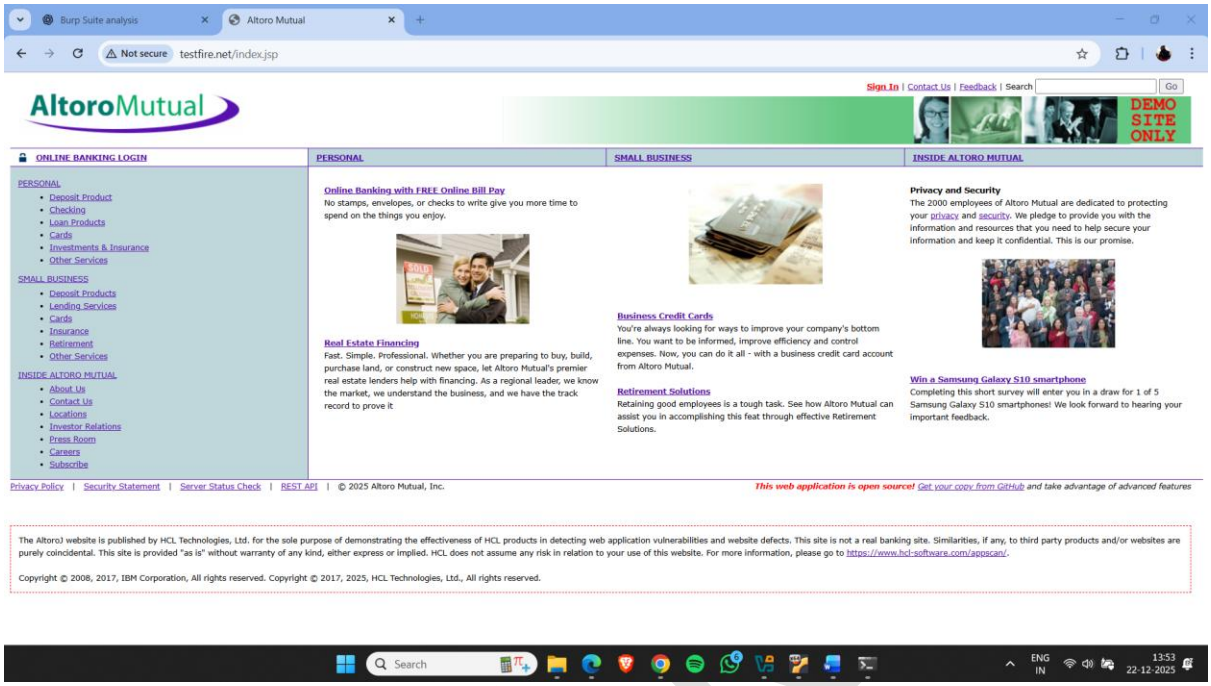


Figure 28

- Sign in with credentials.

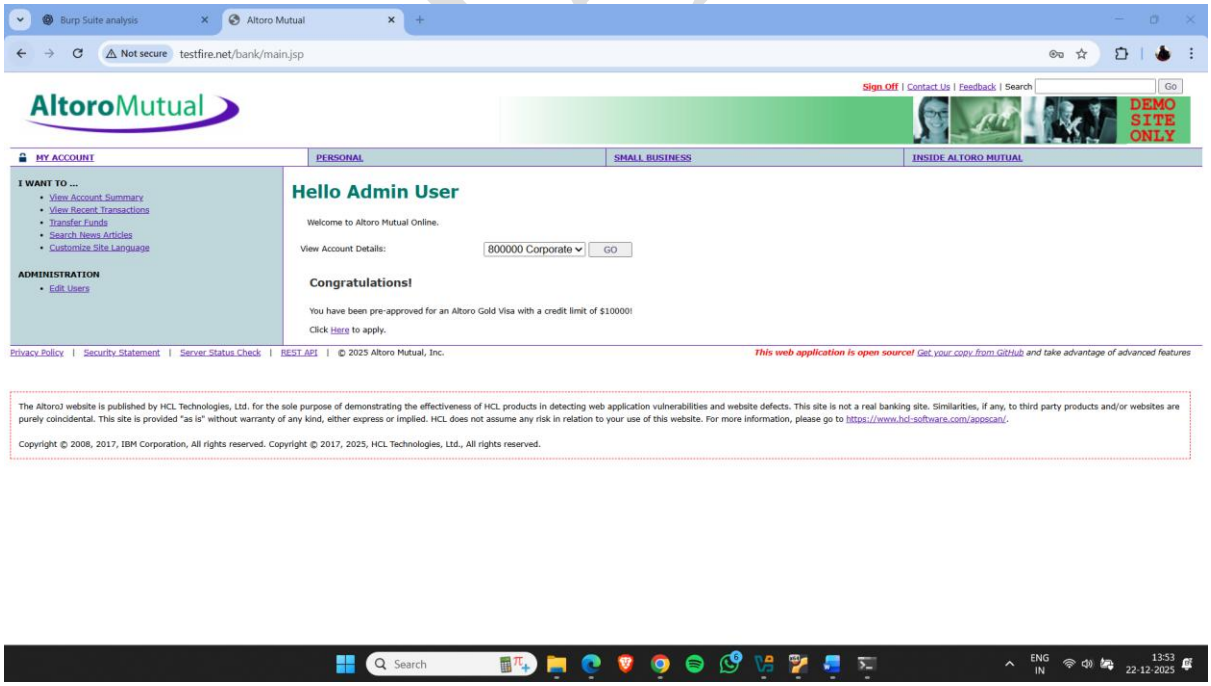


Figure 29

- Now go to Ettercap interface and check that req captured or not.
- Now, Click on Three dot and then click on View

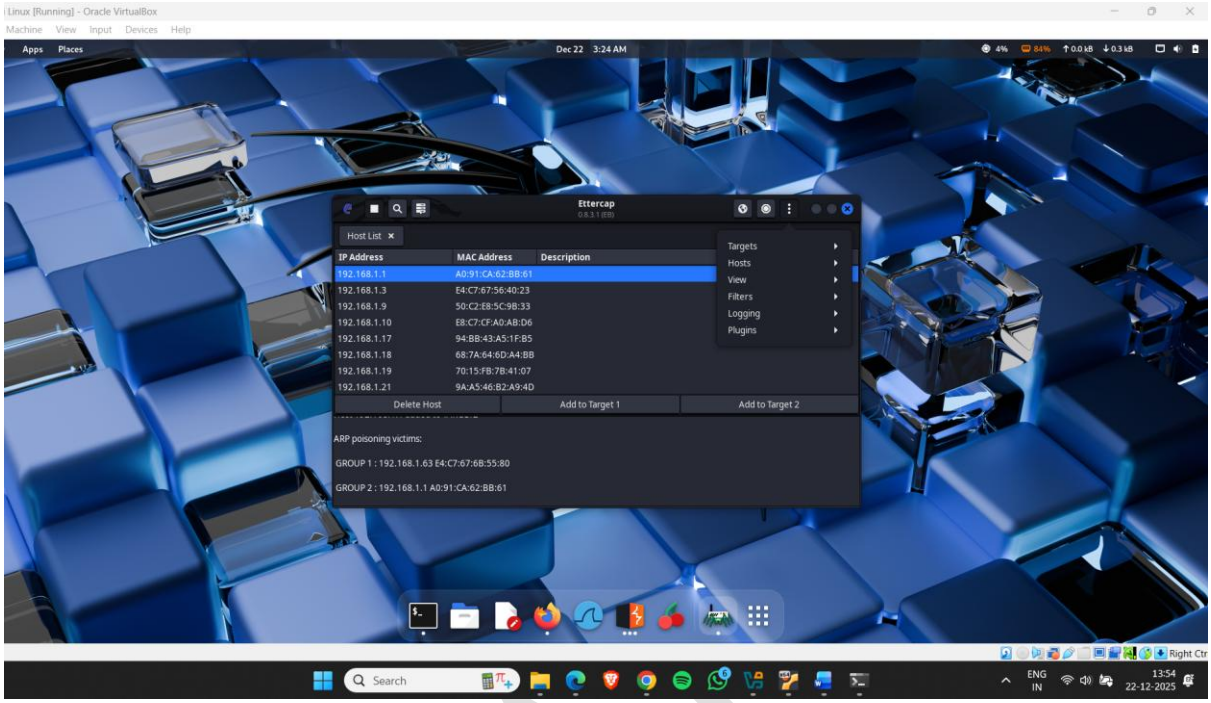


Figure 30

- Click on Connections

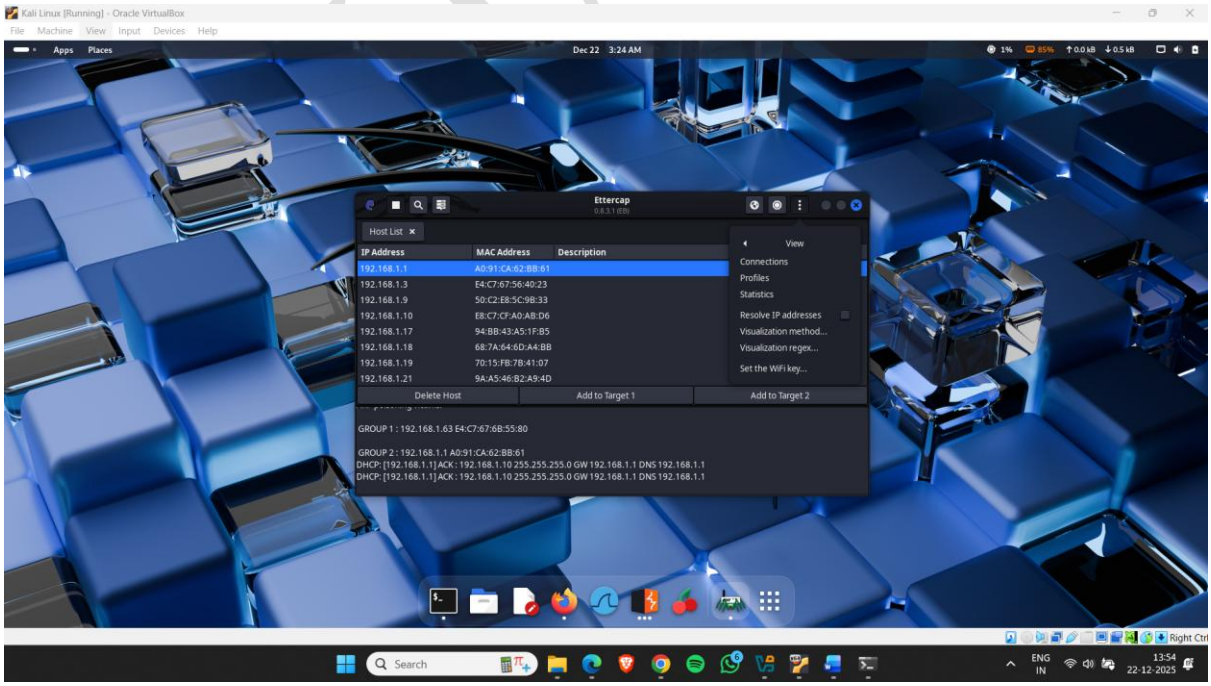


Figure 31

- Here, it captured Many packets
- Search Using Target Ip address & Open Port number 80 packet

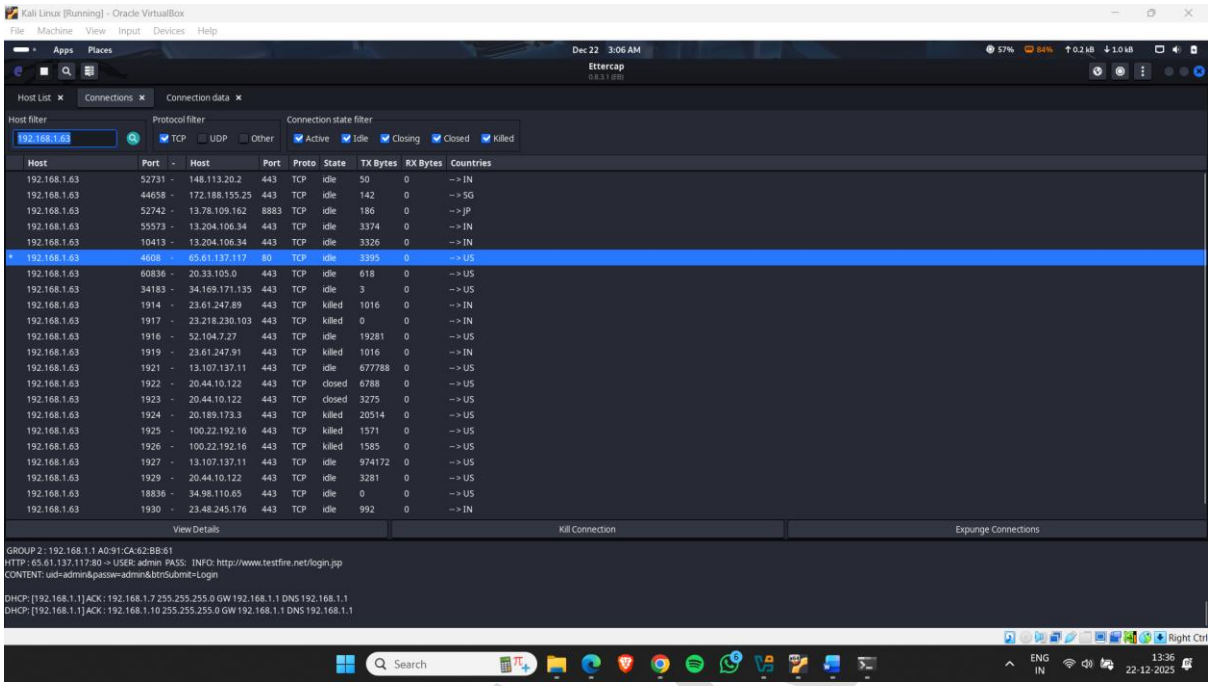


Figure 32

- Here, it captured session id and all like website and other things
- Copy the JSESSION ID.

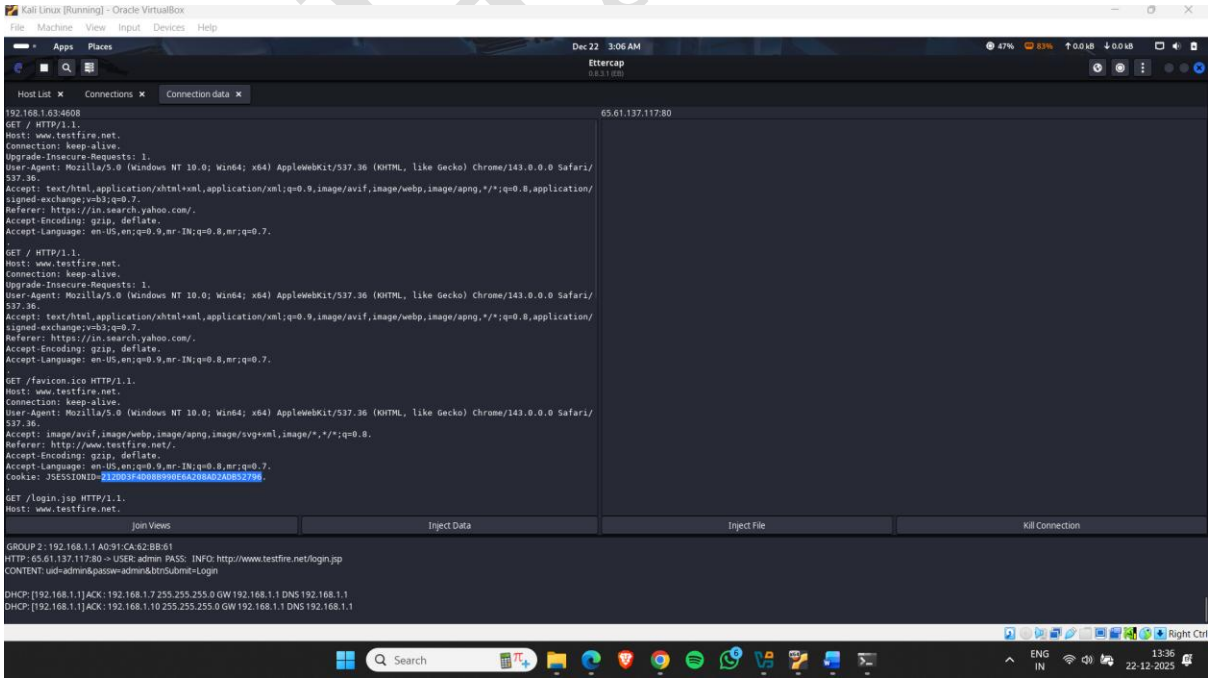


Figure 33

- Open target Website & right click on home page, then click on inspect.

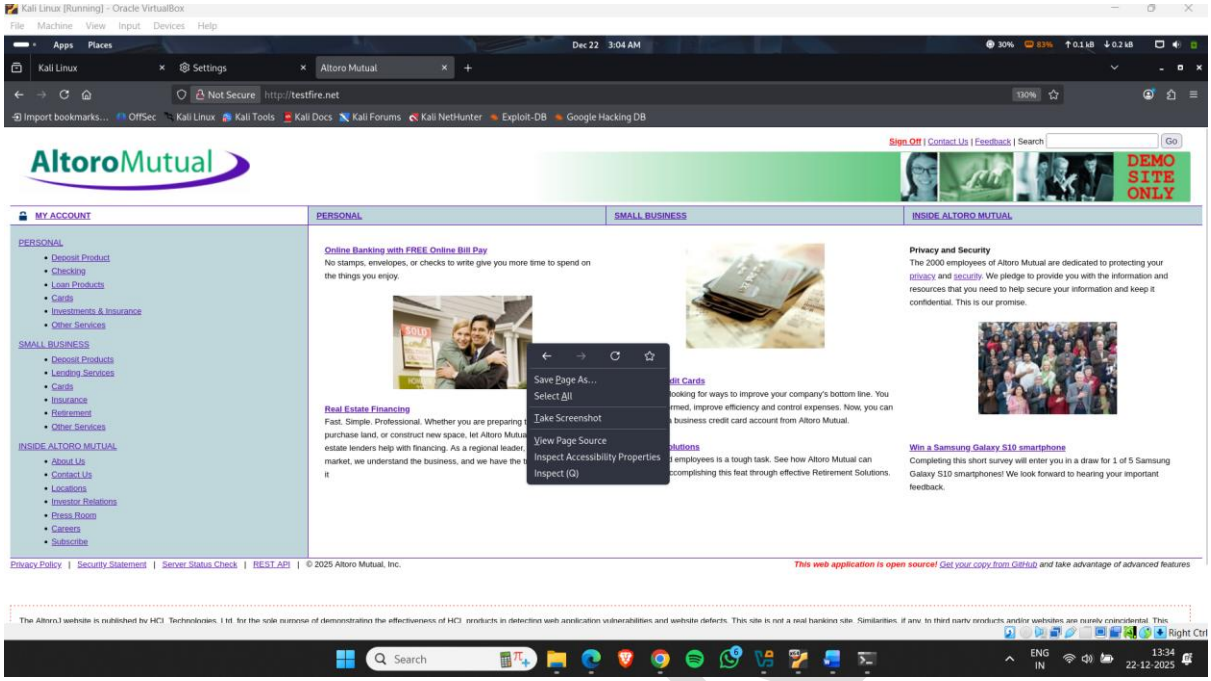


Figure 34

- Paste the JSESSION ID & press enter.

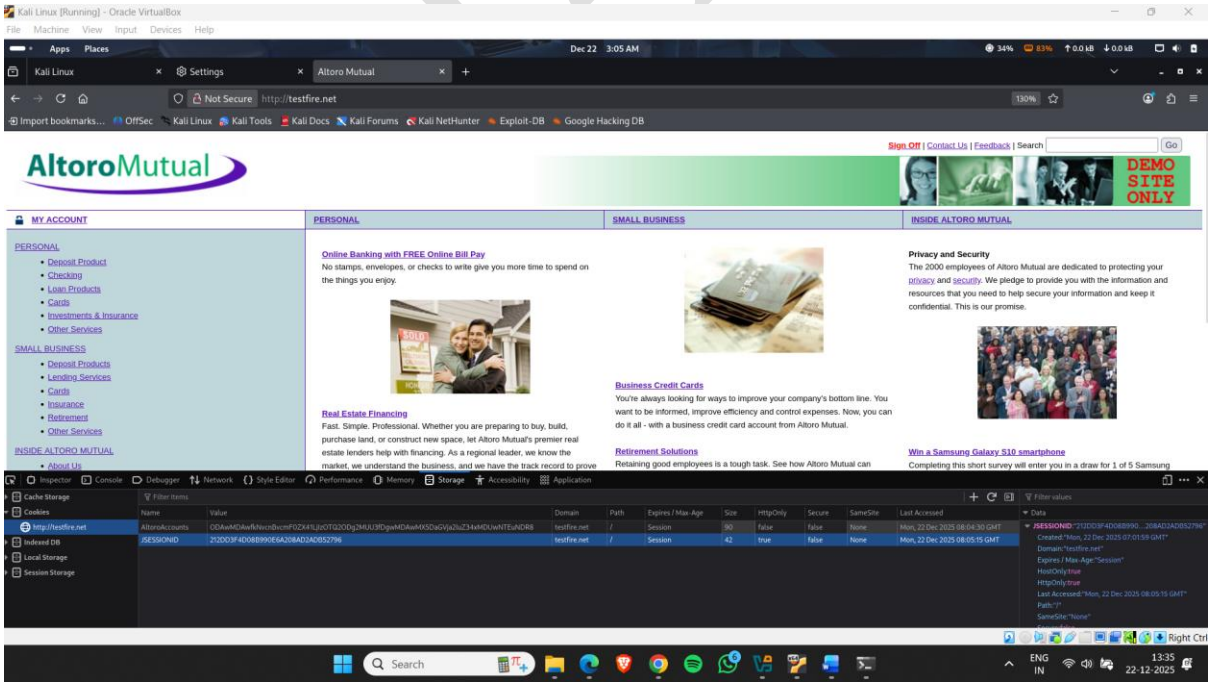
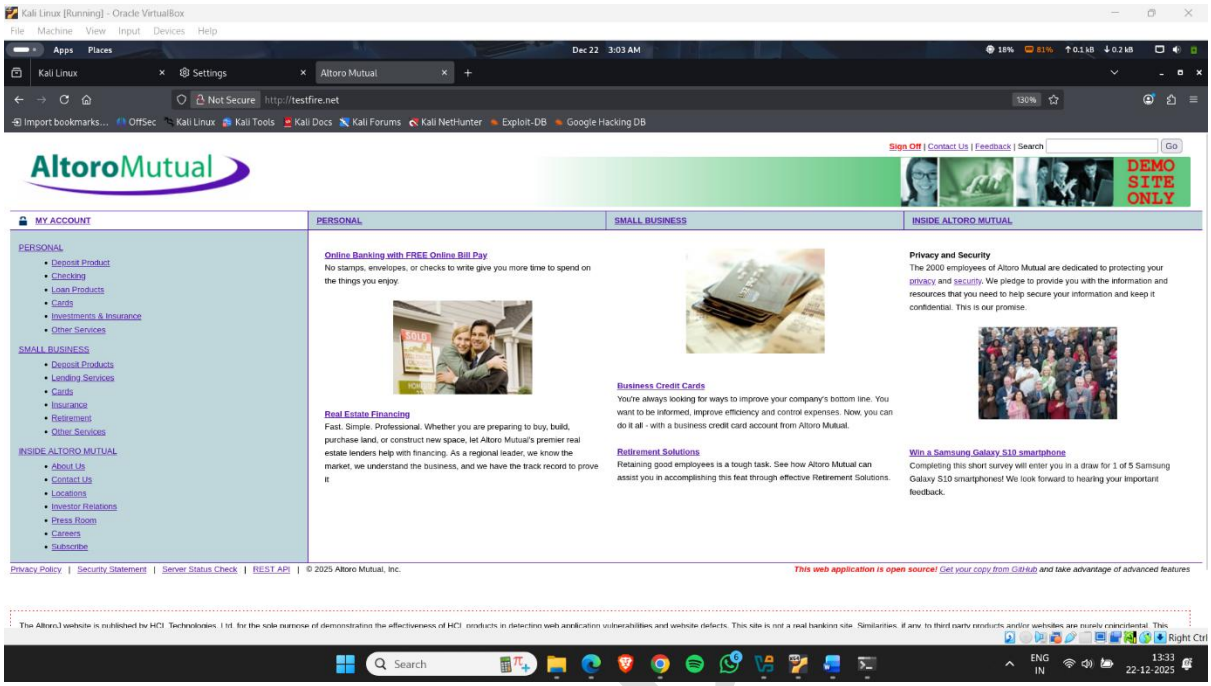


Figure 35

- Logged in successfully



Session Hijacking Using Bettercap Tool

Bettercap is a powerful, modern, and flexible Man-in-the-Middle (MITM) attack framework. It is widely used for performing network-based attacks such as traffic sniffing, session hijacking, and real-time data manipulation. Bettercap is considered the next-generation replacement for Ettercap due to its modular architecture and advanced capabilities.

Features of Bettercap

Bettercap provides several features that make it effective for network attacks:

- **ARP Spoofing:** Redirects network traffic through the attacker's system by poisoning the ARP cache of the victim and gateway.
- **SSL Stripping:** Downgrades HTTPS connections to HTTP, allowing the attacker to sniff unencrypted traffic.
- **Session Hijacking:** Captures session cookies and authentication tokens to take over active user sessions.
- **Live Sniffing:** Enables real-time packet capture and traffic inspection.
- **Modular Architecture:** Supports multiple attack modules and scripting for customized attacks.

Role of Bettercap in Session Hijacking

Bettercap plays a crucial role in session hijacking by positioning the attacker between the victim and the network gateway and capturing sensitive session data.

Steps Involved in Session Hijacking Using Bettercap

Step 1: ARP Spoofing

The attacker performs ARP spoofing to place themselves in a MITM position between the victim and the router.

Step 2: Live Traffic Sniffing

Once MITM is established, Bettercap starts sniffing live network traffic.

Step 3: SSL Stripping (Optional)

SSL stripping is used to force HTTPS traffic to HTTP, making it easier to capture sensitive information.

Step 4: Capturing Session Cookies

Session cookies or authentication tokens are extracted from the intercepted traffic.

Step 5: Session Hijacking

The captured cookies are reused to hijack and access the victim's active session without authentication.

Target Website

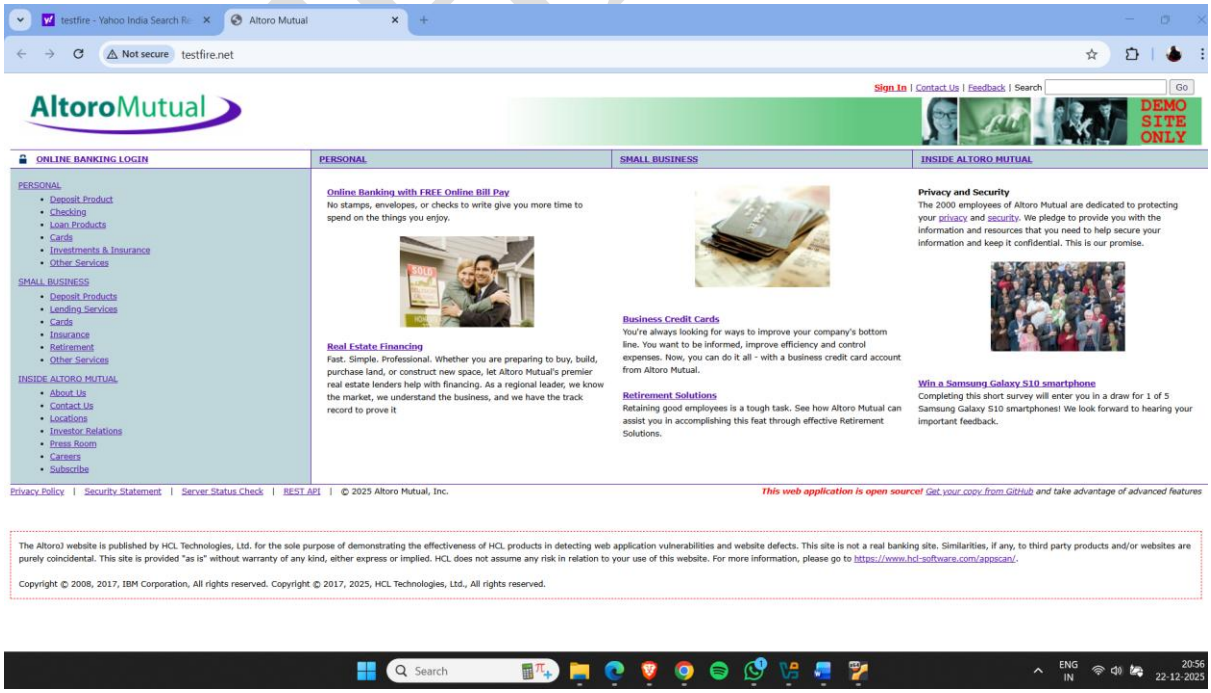


Figure 36

How to use it:

- Cmd- bettercap
- Command -: net.probe on

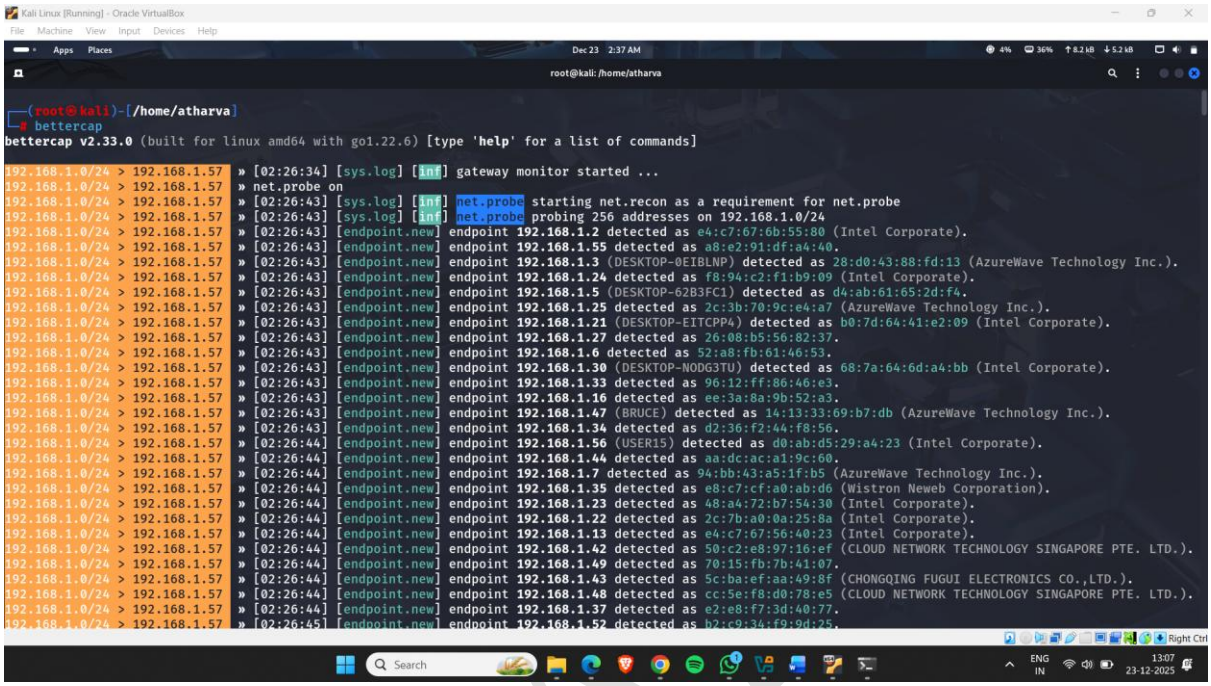


Figure 37

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

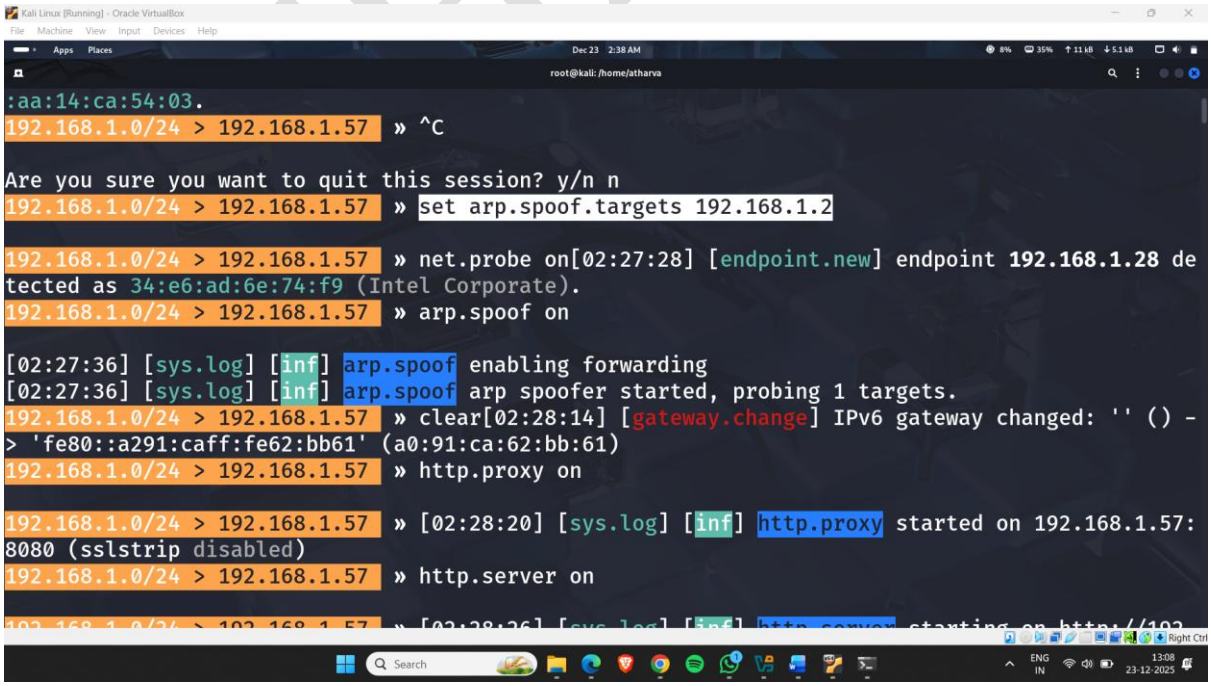


Figure 38

- Command-: arp.spoof on

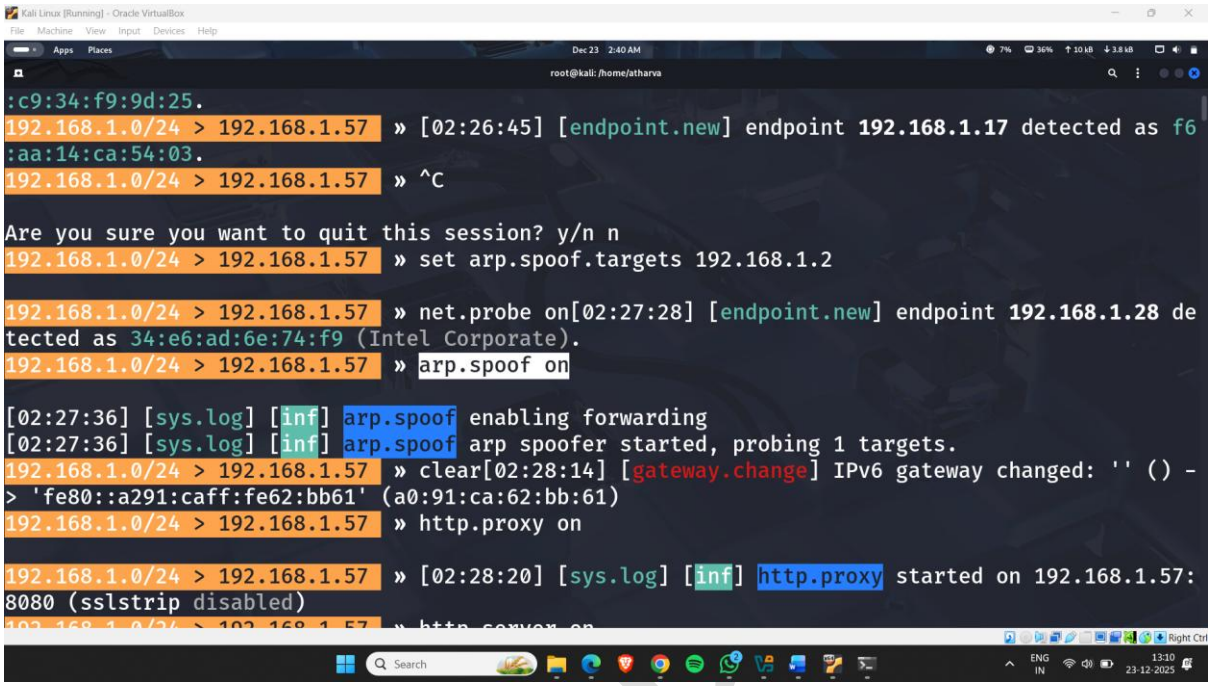


Figure 39

- Cmd-http.proxy on
- Cmd-http.server on
- Cmd-net.sniff on

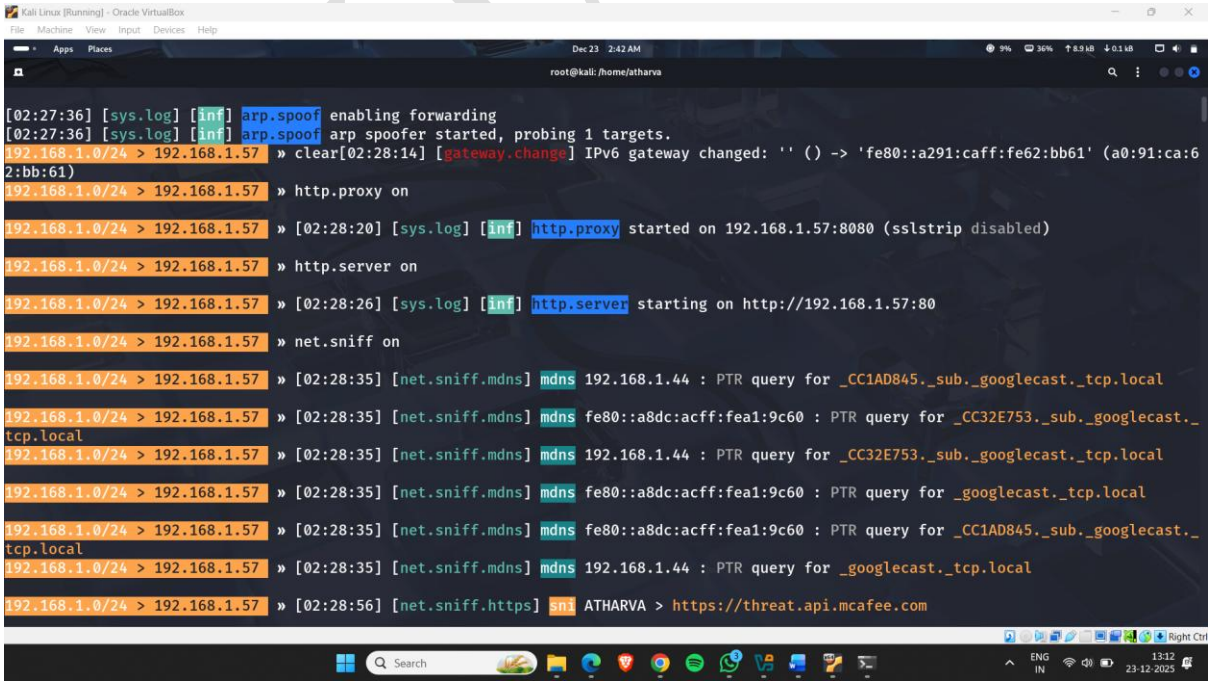


Figure 40

- Now open victim's Browser

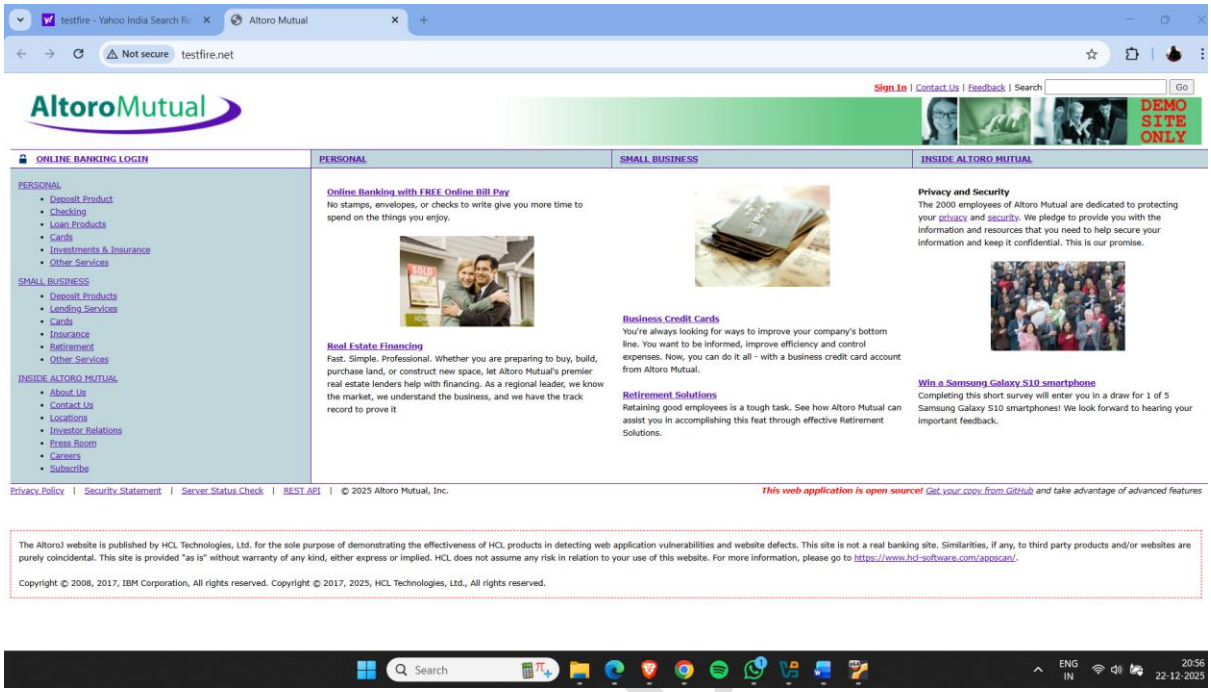


Figure 41

- Enter username and password & login

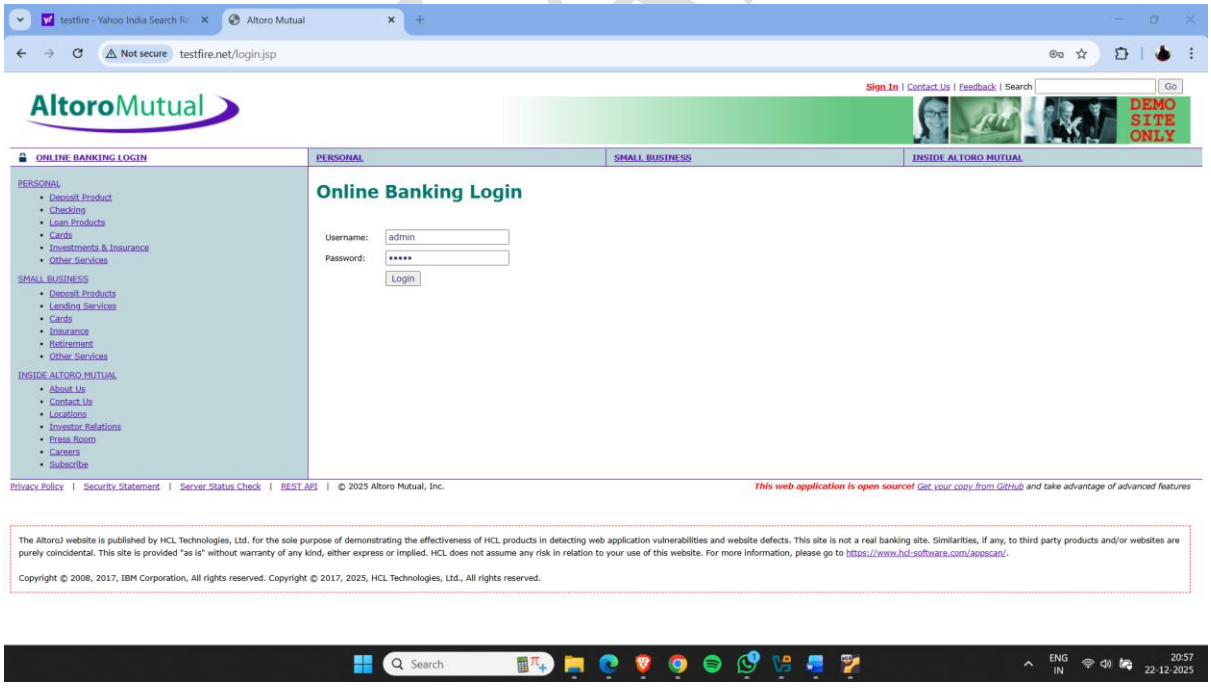


Figure 42

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

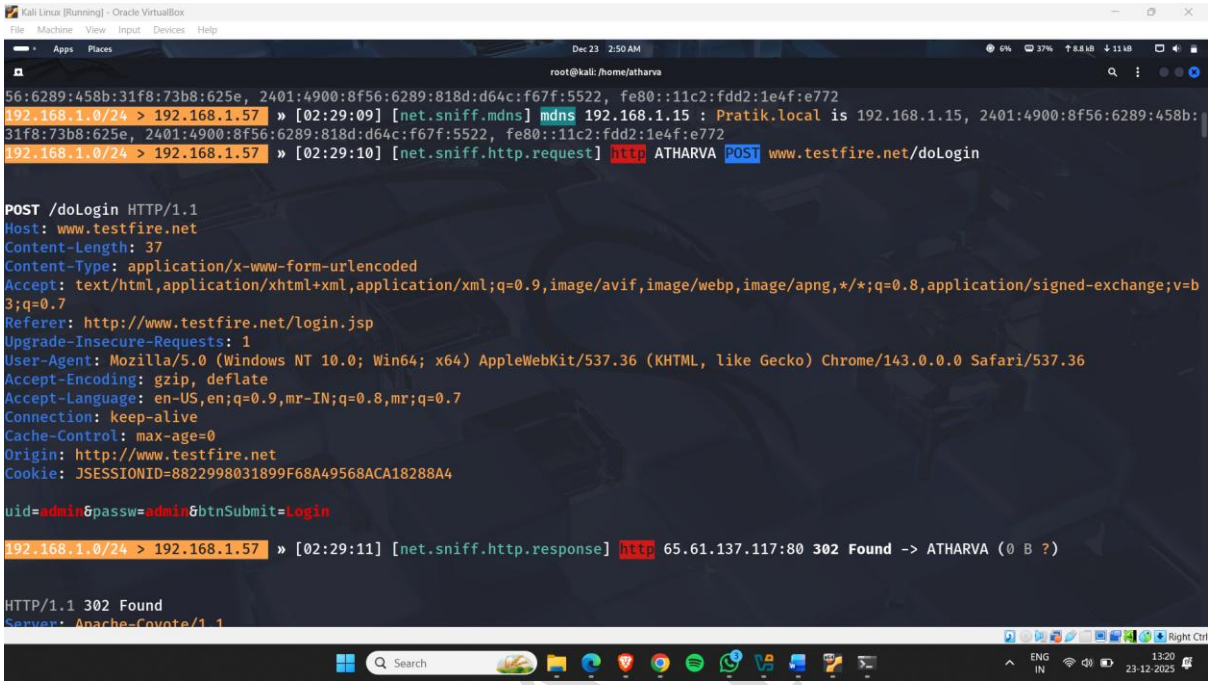


Figure 43

- Now copy this session Id

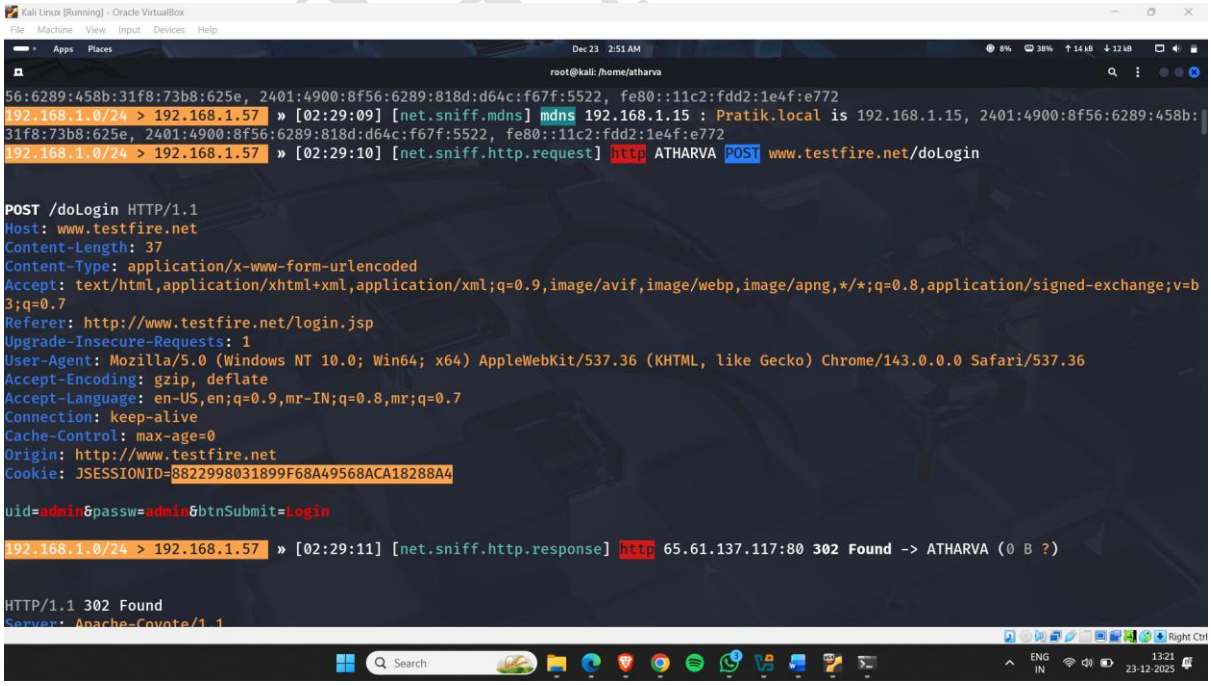


Figure 44

- Open target website and open cookies editor
- Now replace this session id
- Click on save & refresh the browser

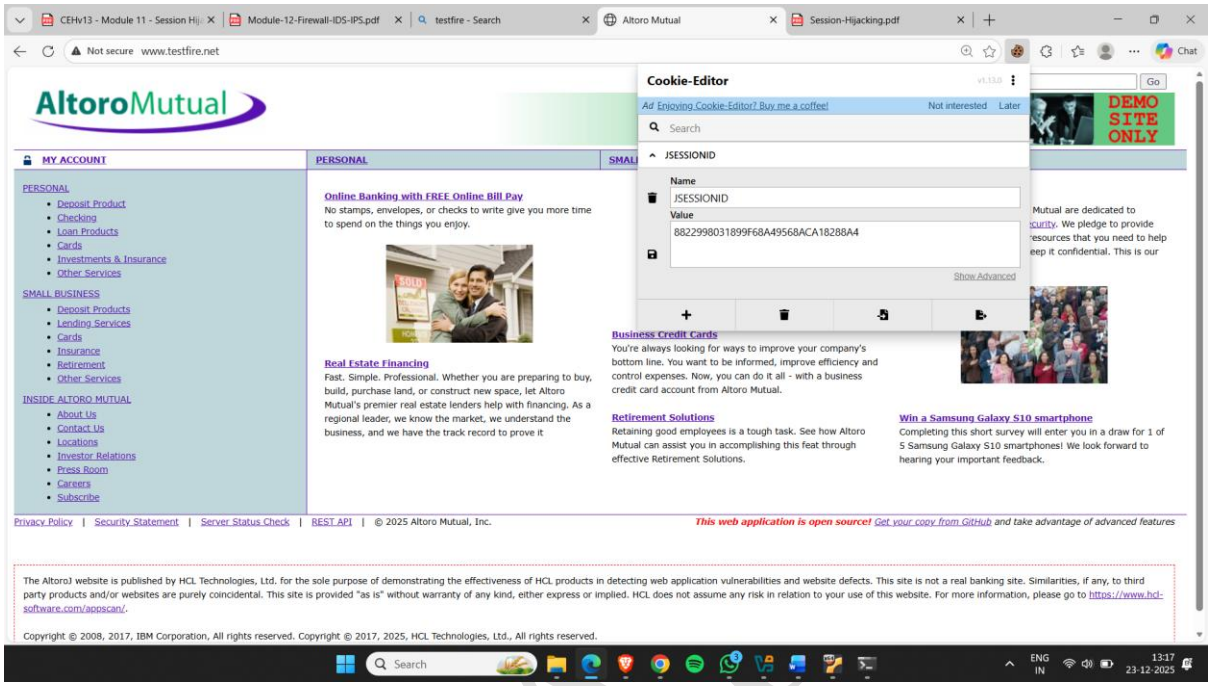


Figure 45

- Logged in successfully.

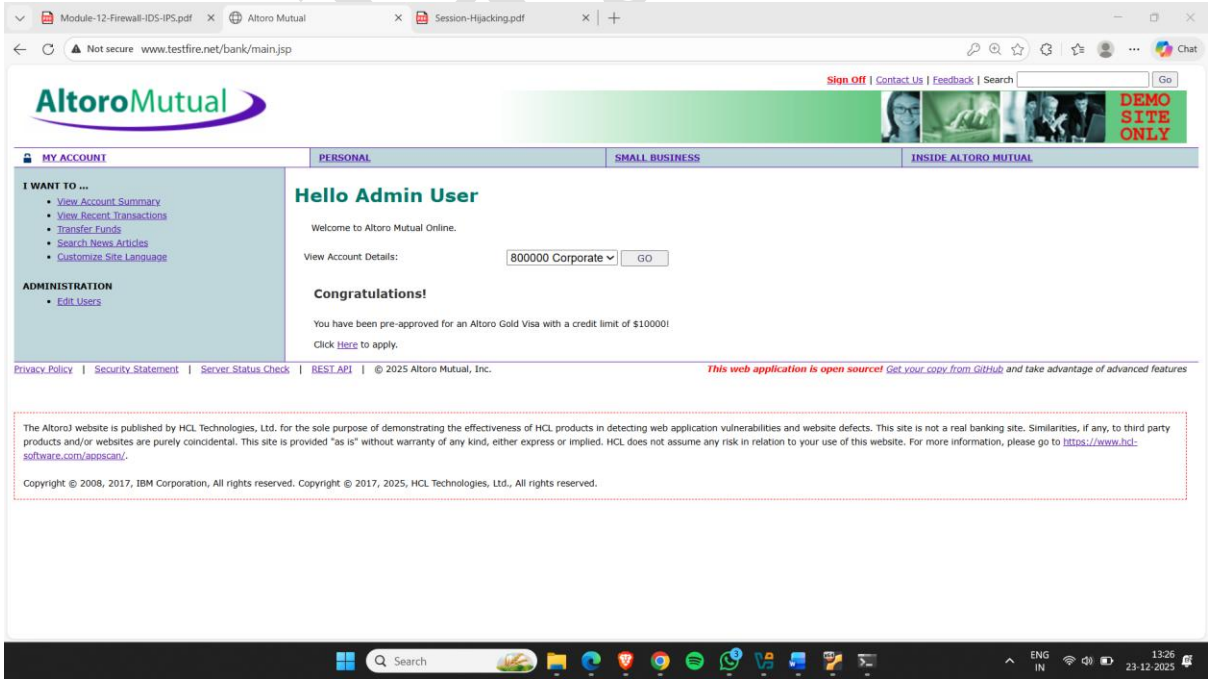


Figure 46

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.

Target Website

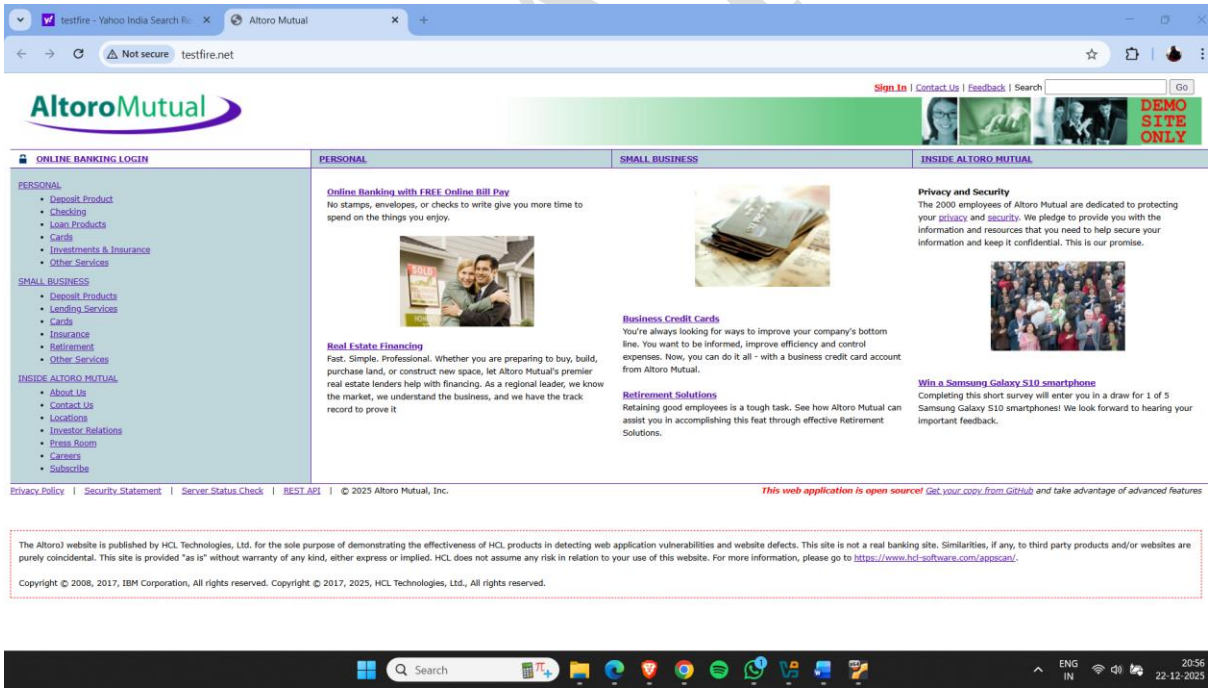


Figure 47

- Enter username and password & Click on Login.

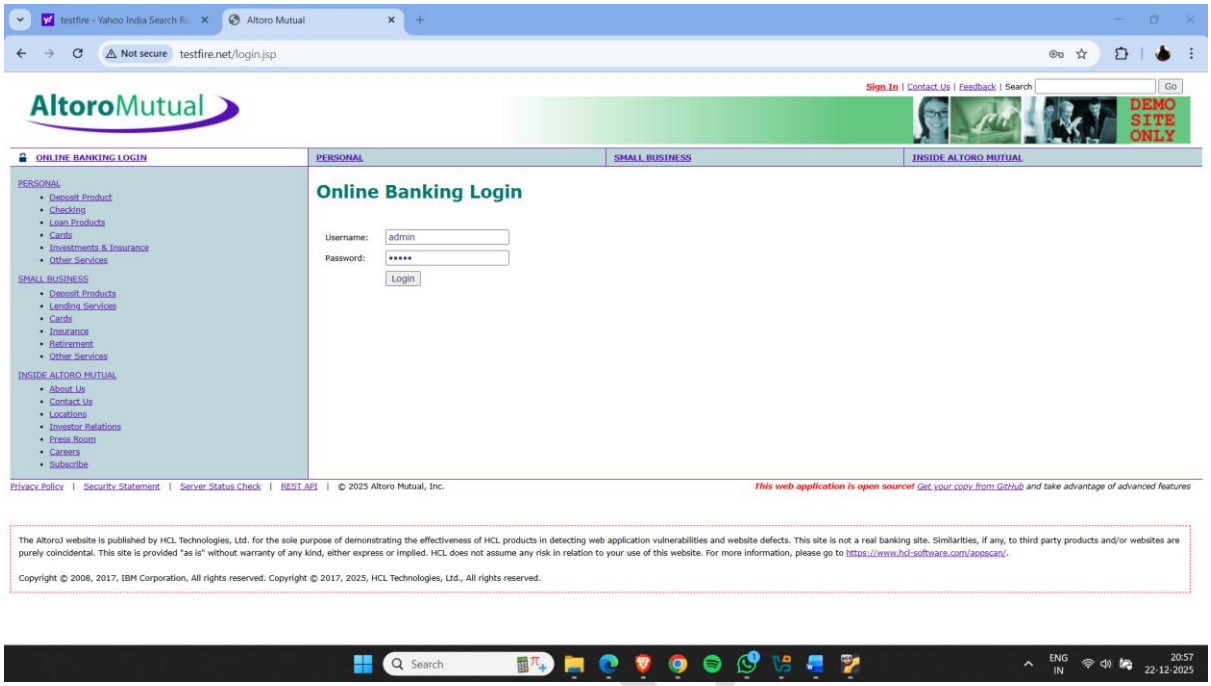


Figure 48

- Logged in Successfully.

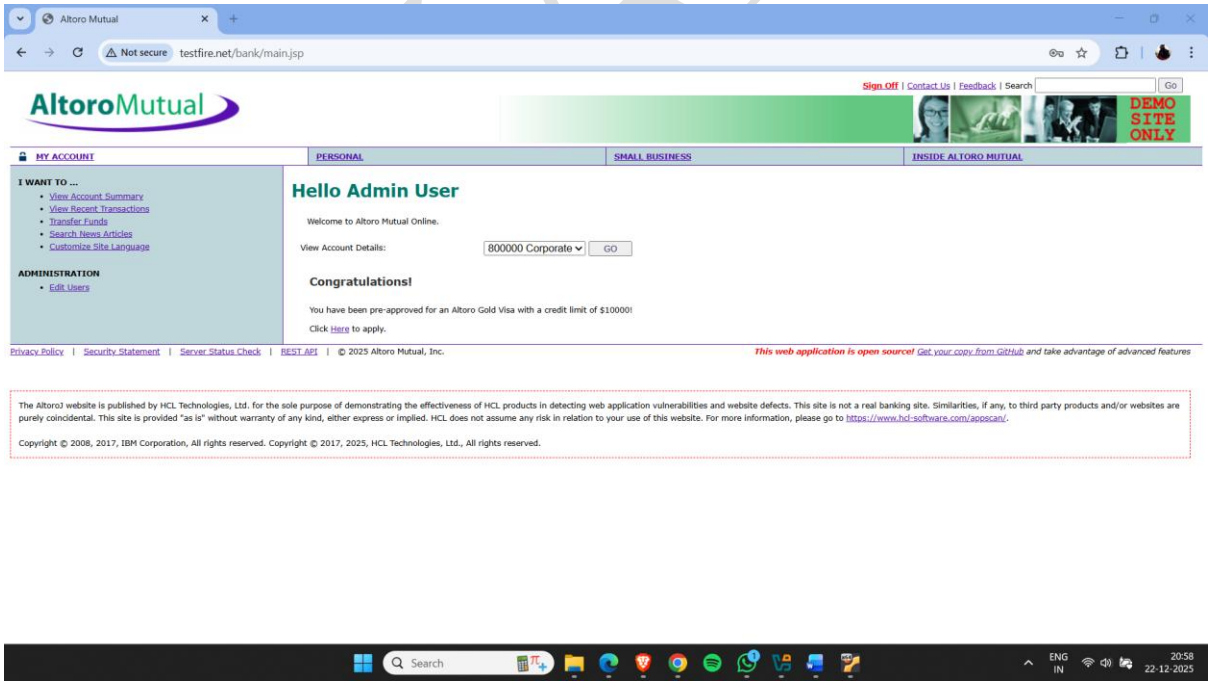


Figure 49

- Right click on page and then click on inspect option

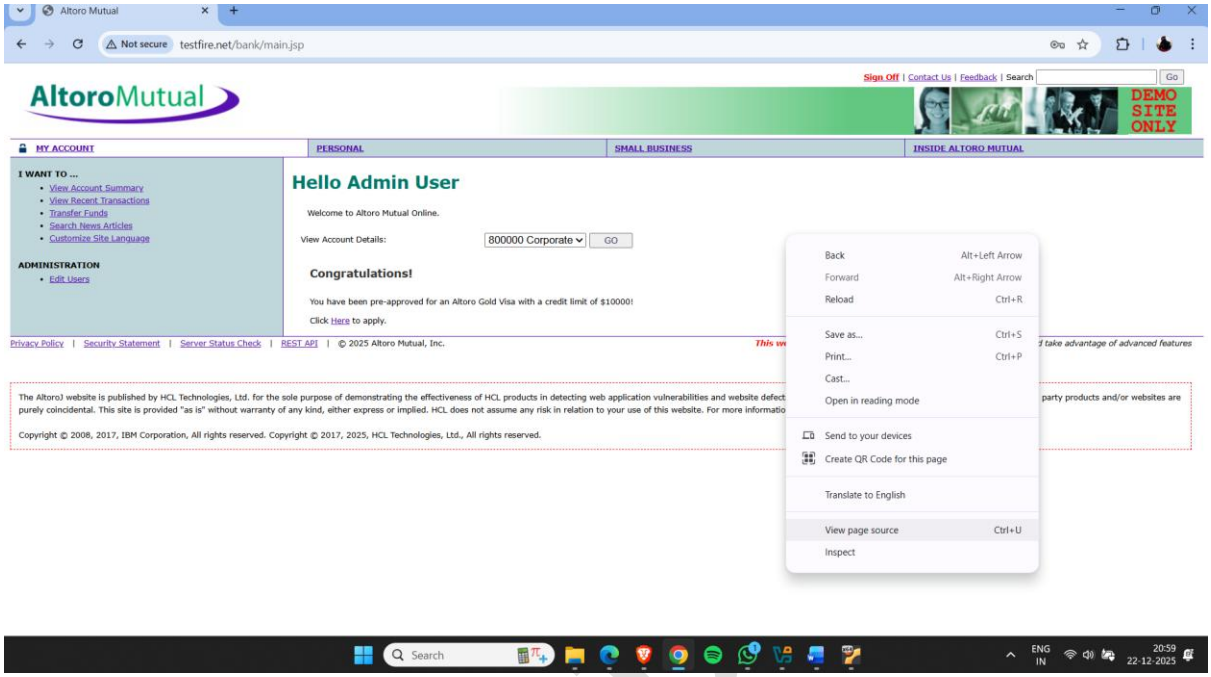


Figure 50

- Now click on Application option

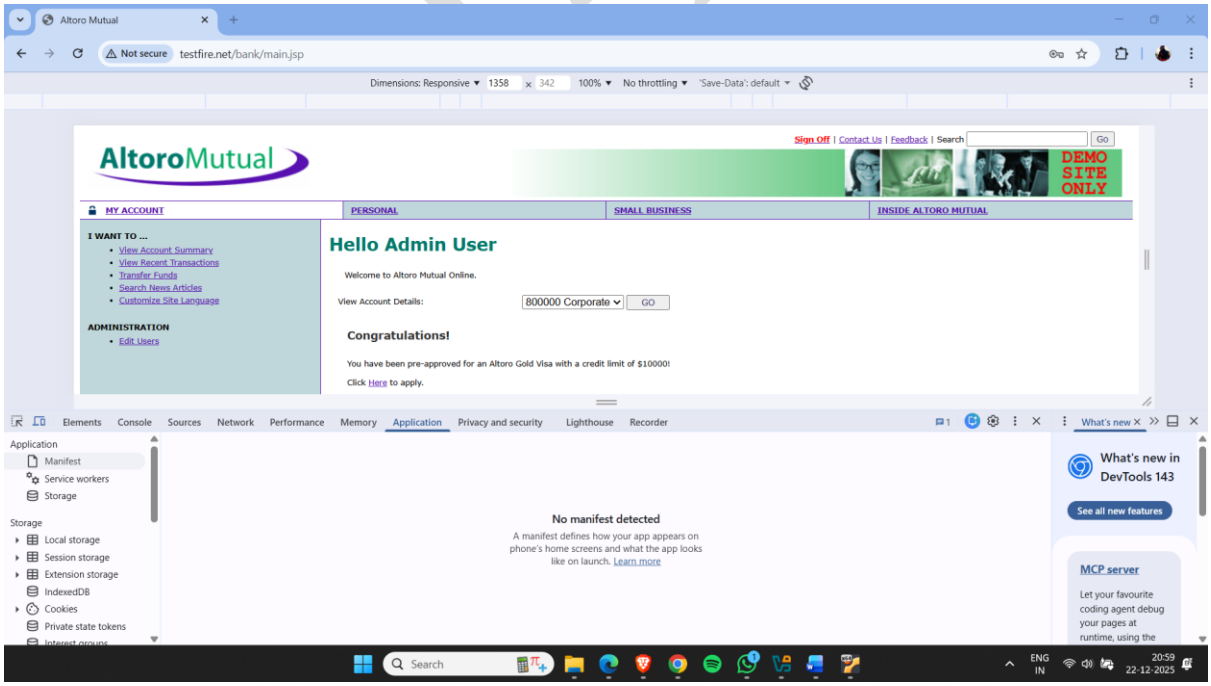


Figure 51

- Now , Click on Cookies Section & Click on Website

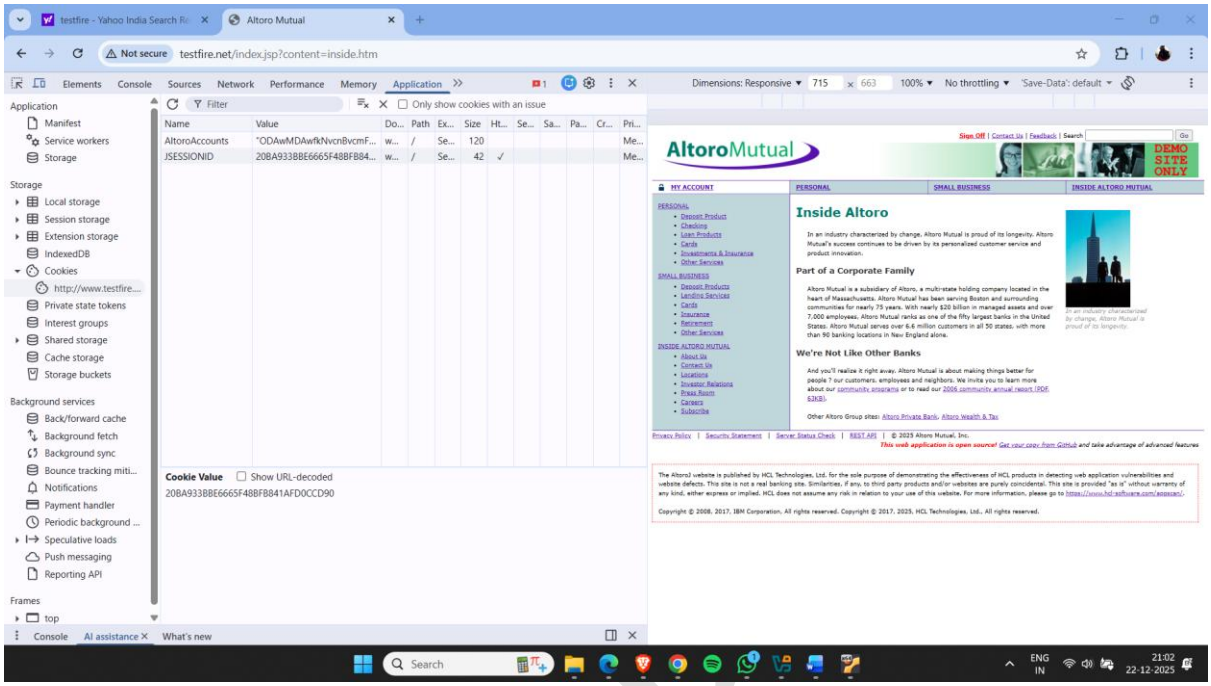


Figure 52

- Click on JSESSIONID & copy this session Id

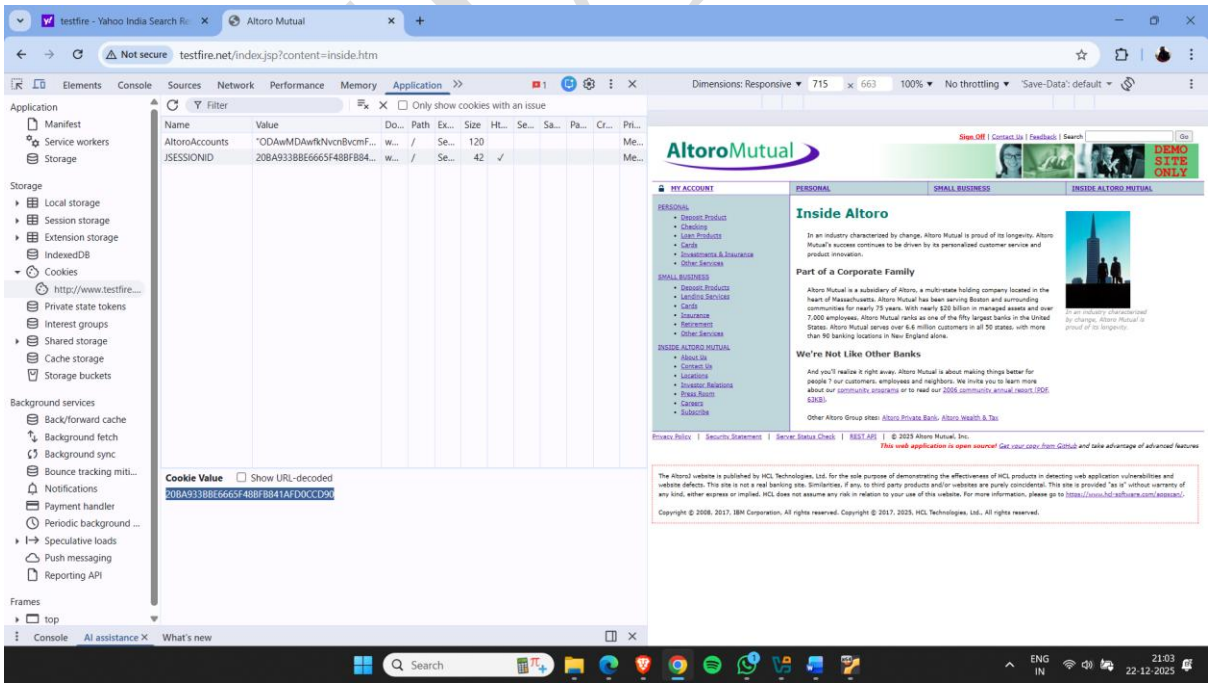


Figure 53

- Now Switch to another browser and open same website

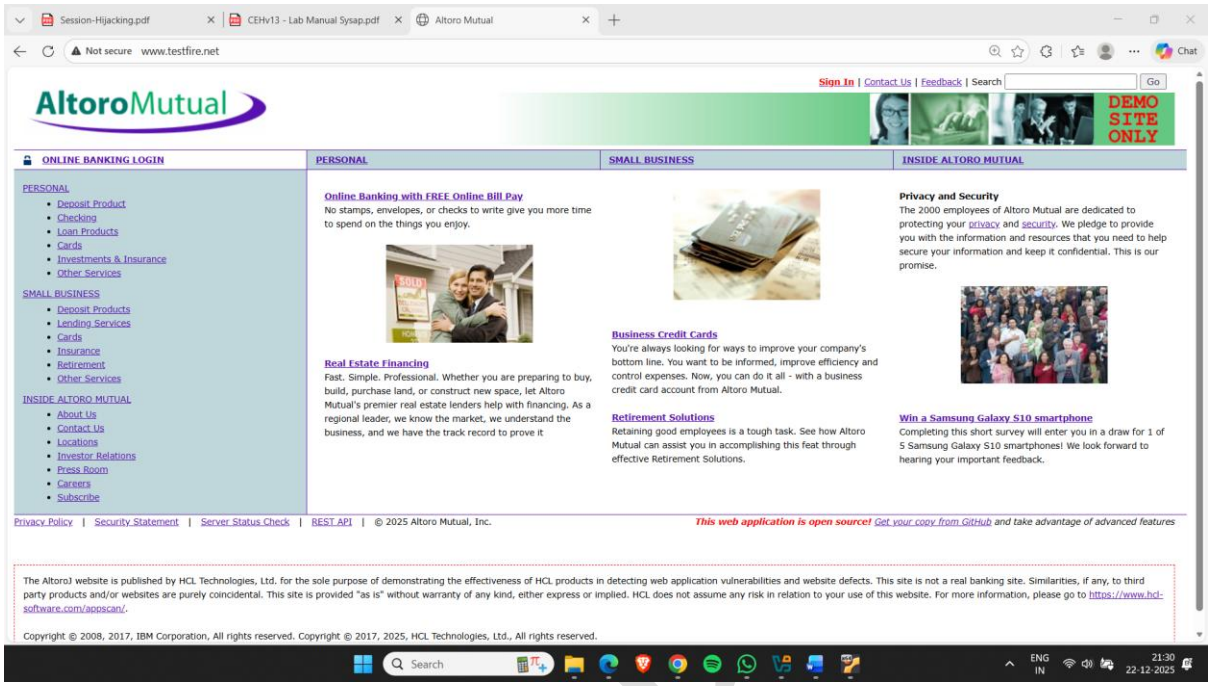


Figure 54

- Assume that we don't have any username and password but we have a session Id

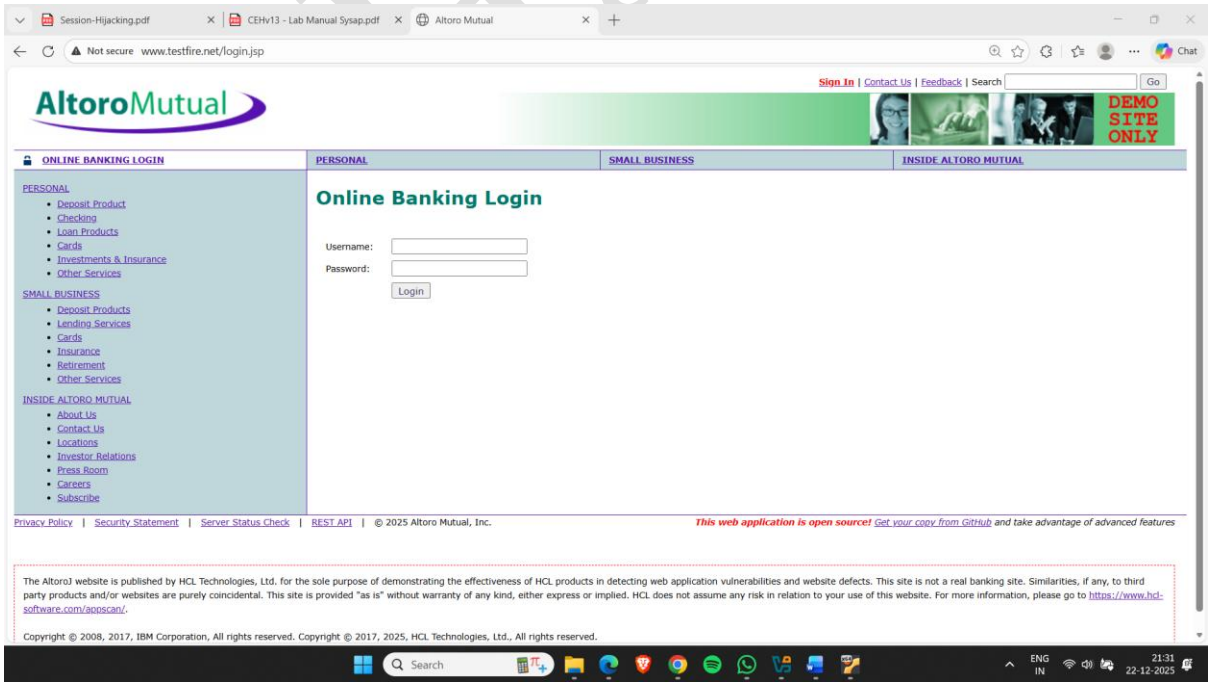


Figure 55

- Open cookies Editor Extension

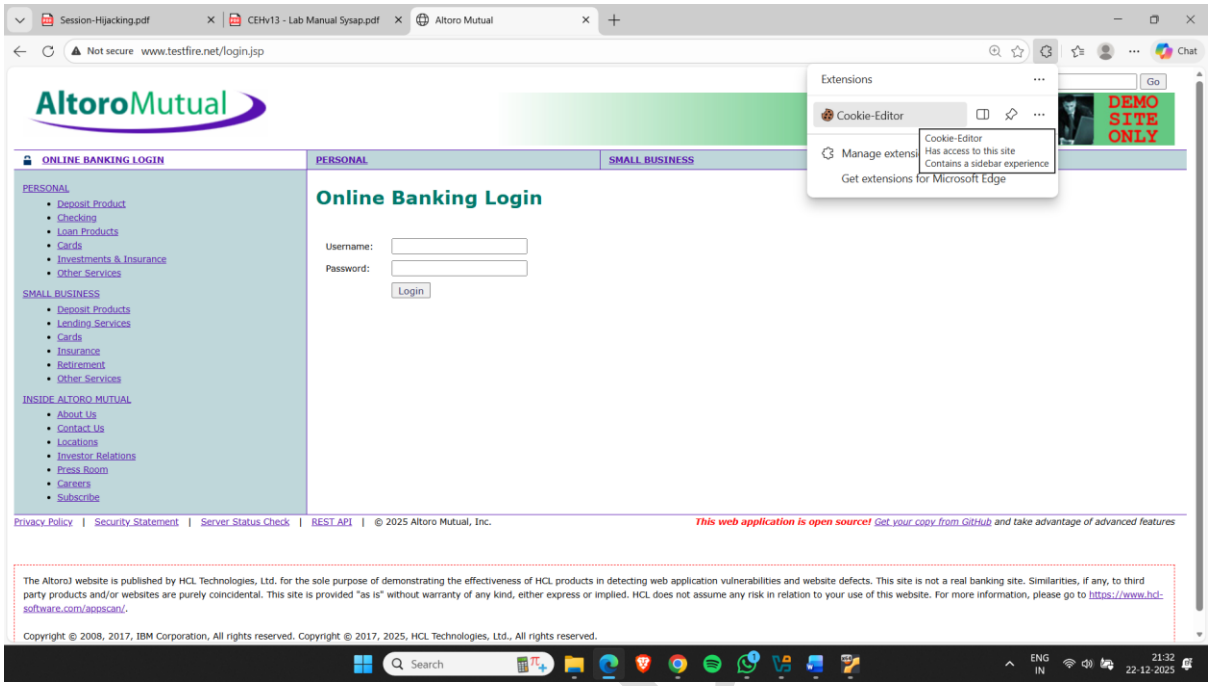


Figure 56

- Replace the Session id & click on save

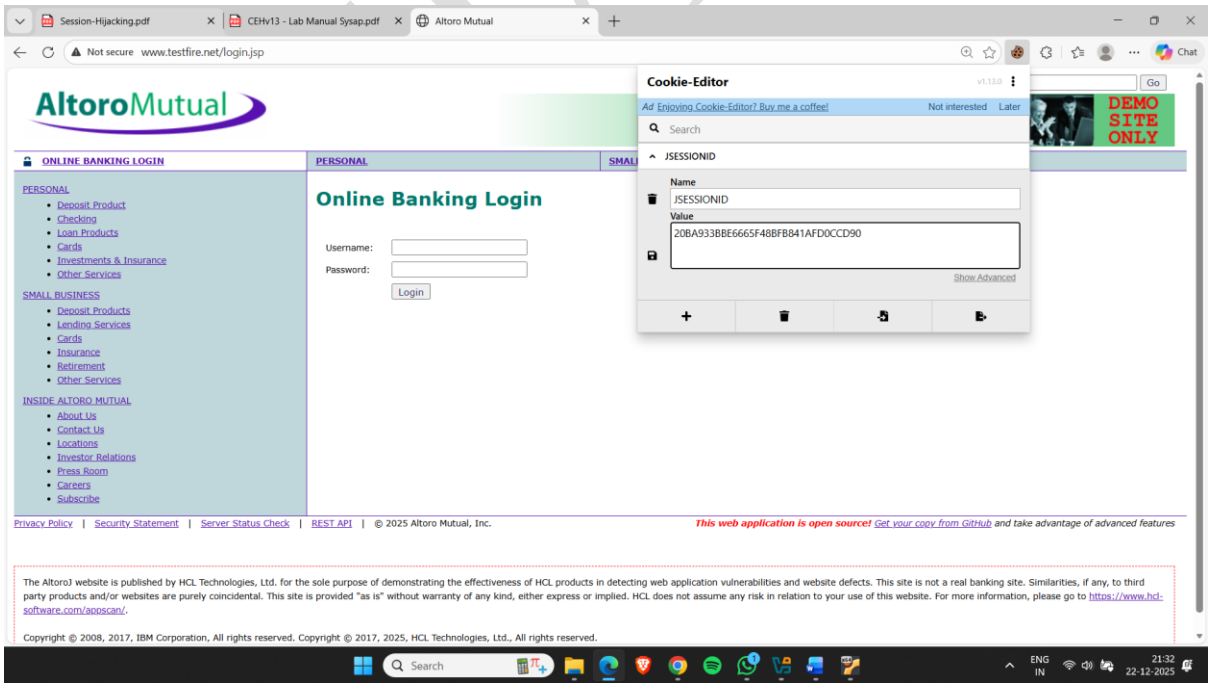


Figure 57

- Logged in Successfully

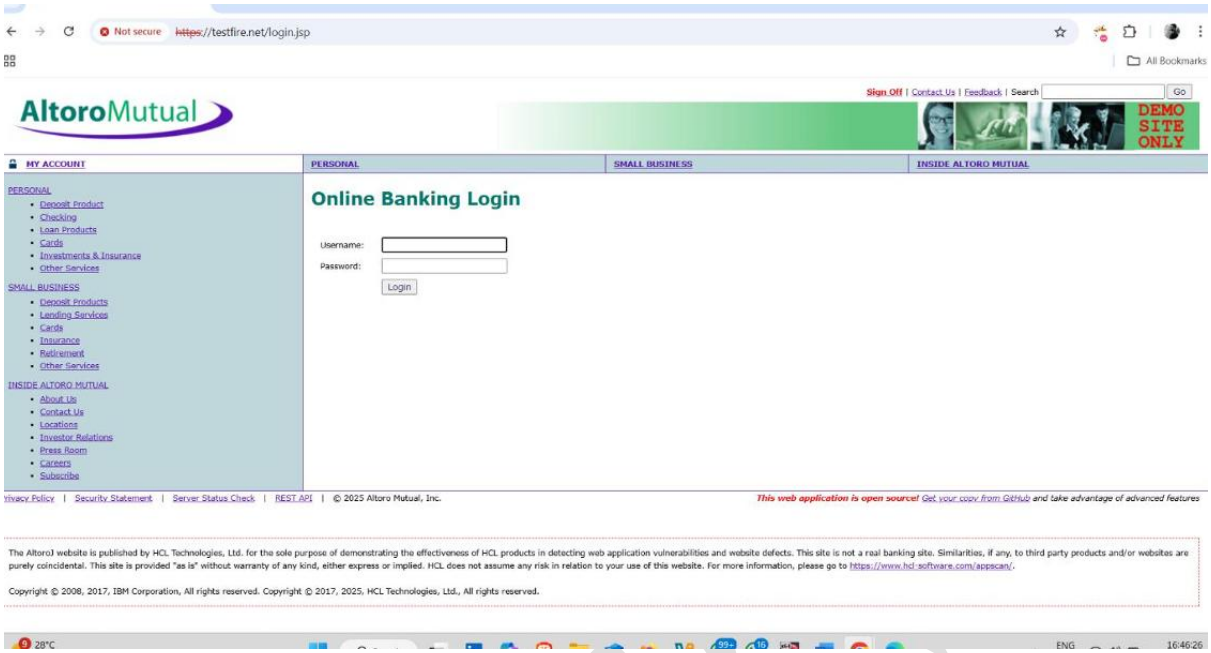


Figure 58

- Click on my account
- Admin User account page will open.

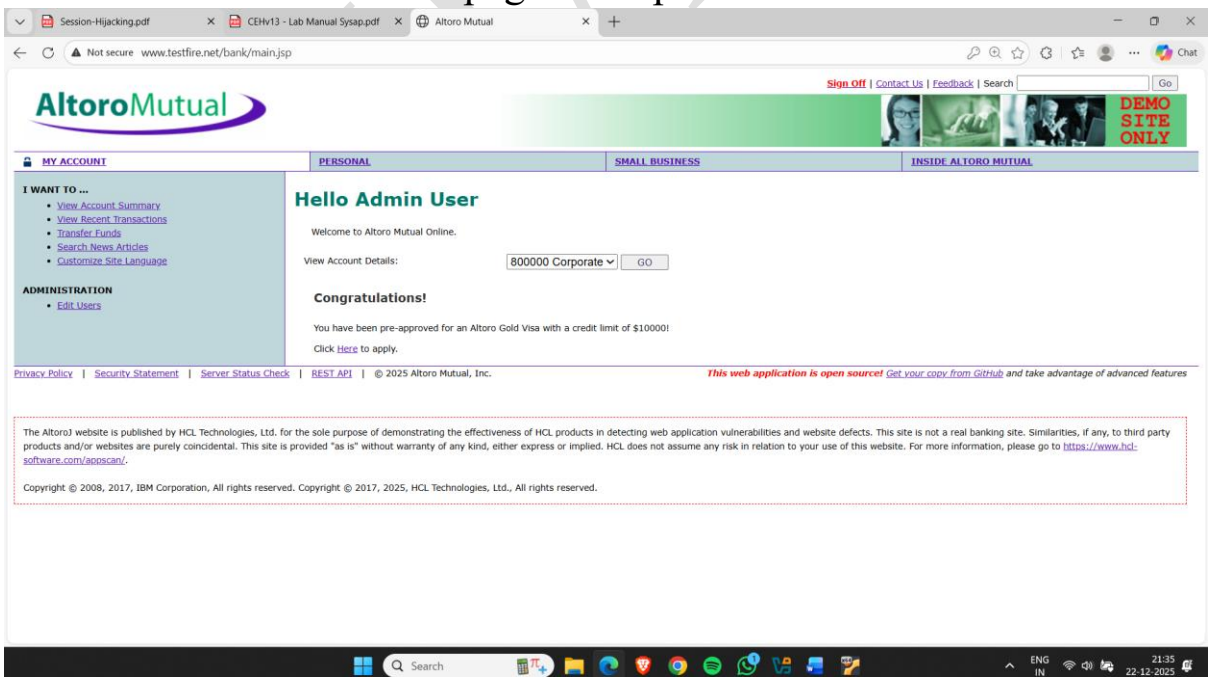


Figure 59

How to Prevent Session Hijacking

What is Session Hijacking?

Session hijacking is a web-based attack in which an attacker takes control of a valid user session by stealing the session ID. The session ID is commonly stored in cookies and, when compromised, allows unauthorized access to a user's account without requiring login credentials.

Session Hijacking Prevention Techniques

1. Use Secure Communication (HTTPS)

Secure communication ensures that session data is encrypted during transmission.

- 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 communicate only over HTTPS.

---

2. Set Secure Cookie Attributes

Proper cookie configuration helps protect session IDs from theft.

- **Secure Flag:** Ensures cookies are sent only over HTTPS connections.
- **HttpOnly Flag:** Prevents JavaScript access to cookies, reducing XSS risks.
- **SameSite Flag:** Restricts cookies from being sent in cross-site requests.

Example:

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

---

3. Implement Session Timeouts

Limiting session duration reduces the risk of hijacked sessions being misused.

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

---

4. Regenerate Session ID

Regenerating session IDs improves security after authentication events.

- Generate a new session ID after login or privilege escalation.
- This helps prevent session fixation attacks.

---

5. Destroy Session on Logout

Proper session termination prevents reuse of old sessions.

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

---

6. Implement IP and User-Agent Validation

Binding sessions to user characteristics reduces unauthorized access.

- Associate sessions with the user's IP address and browser User-Agent.

---

7. User Awareness and Best Practices

User behavior plays a crucial role in session security.

- Avoid leaving logged-in sessions unattended.
- Do not use public Wi-Fi without a VPN.
- Always log out properly after use.
- Regularly clear browser cookies and cache.

---

8. Advanced Security Techniques

Additional security measures provide stronger protection against session hijacking.

- Use Multi-Factor Authentication (MFA).
- Implement Content Security Policy (CSP) to reduce XSS risks.
- Apply subdomain separation for sensitive cookies.
- Monitor session anomalies such as multiple locations or frequent IP changes.
