International Journal of Engineering Business and Social Science
Vol. 3 No. 4, April 2025, pages: xxx-xxx
e-ISSN: 2980-4108, p-ISSN: 2980-4272
https://ijebss.ph/index.php/ijebss

Analysis of Address Resolution Protocol Poisoning Attacks on Mikrotik Routers Using Live Forensics Methods

Herman[1*], Rusyadi Umar[2], Agus Prasetyo[3]
Universitas Ahmad Dahlan, Indonesia
Email: hermankaha@mti.uad.ac.id[1], rusydi@mti.uad.ac.id[2], agus2008048039@webmail.uad.ac.id[3]

Corresponding Author: Herman

Keywords
ARP spoofing, ARP poisoning, live forensics, wireshark, mikrotik, network security

ABSTRACT

The rapid development of wireless technology has made network communication more accessible but also increasingly vulnerable to security threats. One of the major threats is the Man-in-the-Middle (MitM) Attack, particularly ARP Spoofing, which manipulates the Address Resolution Protocol (ARP) to intercept or alter network traffic. ARP Spoofing, also known as ARP Poisoning, allows attackers to associate incorrect MAC addresses with IP addresses, enabling unauthorized access and potential data interception. This research focuses on the detection and investigation of ARP Spoofing on MikroTik routers using live forensic methods. The study utilizes Wireshark as a primary tool to monitor ARP-based network activity and identify anomalies indicative of ARP Spoofing attacks. The National Institute of Standards and Technology (NIST) forensic framework, which includes Collection, Examination, Analysis, and Reporting, is employed as a methodology for analyzing forensic evidence. The research also incorporates a virtualized attack simulation environment using VirtualBox, where a PC Client acts as the target, an attacker PC executes an ARP Spoofing attack using Ettercap, and Wireshark captures network traffic for forensic examination. The simulation results reveal that an ARP Spoofing attack can successfully manipulate network traffic by altering ARP table entries. The attacker assumes the identity of IP Address 192.168.0.1 with MAC Address e8-cc-18-41-3f-fb, while the target's identity is duplicated as 192.168.0.19 with MAC Address 08:00:27:15:4c:3c, as confirmed through Wireshark analysis and ARP table inspection using the command prompt. These findings emphasize the importance of implementing proactive security measures, such as Dynamic ARP Inspection (DAI), encryption protocols, and continuous network monitoring, to mitigate the risks associated with ARP Spoofing attacks.

INTRODUCTION

The development of computer networks raises security problems that require a network forensic approach. Network forensics is the process of collecting, analyzing, and interpreting digital evidence (Sunardi, Riadi, & Akbar, 2020) related to activities that occur in computer networks.

Address Resolution Protocol (ARP) This is a protocol used in computer networks to associate physical addresses or Media Access Control (MAC) addresses in network devices with logical addresses (IP addresses) in the network. ARP is useful when data must be transmitted between devices in the network, but the physical address of the destination is unknown to the sending device. When a device wants to send data to a destination IP address, it checks its local ARP table to see if there is already an IP address it wants to go to. If there is none, the device will issue an ARP request sent broadcast to all local networks. PC devices that have a matching IP address will respond by providing their MAC address. Once the sender device responds, it will update the ARP table with this information so that the data can be sent correctly to the destination MAC address. ARP facilitates the process of associating between IP addresses and MAC addresses in a network to transmit data correctly at the data link or layer 2 level in the Open Systems Interconnection (OSI) model.

ARP poisoning (spoofing) is the use of ARP weaknesses to interfere with the assignment of MAC-IP to other devices on the network. In 1982, when the ARP protocol was introduced, security was not a major concern so protocol developers never used authentication mechanisms to validate ARP messages. Any device on the network can respond to an ARP request, regardless of whether it is a receiver or not. For example, if computer A requests the MAC address of computer B, an attacker on computer C can respond and computer A will receive the response as a valid form. Due to this vulnerability, a large number of attacks are carried out using available tools, attackers can poison the ARP cache of the targeted PC Client in the local network by filling it with incorrect data.

Figure 1. Mekanisme ARP Poisoning

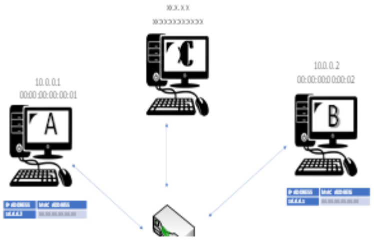


Figure 1. describes the mechanism of ARP Poisoning which illustrates the ARP Poisoning method. Computer A sends an ARP packet destined for computer B. Computer A scans the network for ARP packets on a broadcast basis. The MAC address of computer B can then be changed to the MAC address of the computer to perform computer poisoning. It is evident that Computer X has access to every data packet traffic that occurs on computers A and B, including pin numbers, password information, and more. ARP is matching IP addresses with MAC addresses so that data packets can be sent precisely to the intended device in the computer network. ARP can be used to launch attacks. ARP is also known as spoofing, which is an attack method in which an attacker sends a fictitious ARP packet to a local area network . The association of the attacker's MAC address with the target's IP address (such as the default gateway) is a common goal of these attacks. The attacker manipulates the ARP message to the host on the network, and the data traffic that should be directed to the target's IP address is instead sent to the attacker. Poisoning and spoofing are two related concepts but have important differences in the context of security and technology. Live Forensics can be used to identify digital evidence features to ascertain when an attack is launched against the system. Live Forensics

can be an effective approach to obtain digital evidence from the Router in real-time situations. Using this method, data can be retrieved directly from the device involved in the attack, in a literature review that will analyze on a network

a. Network Protocol Analyzer

Network Protocol Analyzer also known as packet analyzer or packet sniffer is a tool used to analyze and monitor network traffic. This tool works by capturing and examining data packets transmitted over a computer network. A data packet is the smallest unit of information sent over a network. Each data packet consists of a header (controller information) and a payload (actual data). Network Protocol Analyzer (Landri Elusi Hutagalung, Informatika, Teknik, Komputer, & Batam, 2018) can analyze and extract information from packet headers, such as the source and destination of the sender, the protocol used, the port number, and so on.

b. Attack

An attack refers to a set of steps and actions taken by an attacker or attacker to damage or gain unauthorized access to a computer system, network, or other entity. The purpose of the attack is to steal sensitive data, damage infrastructure, disrupt services, or otherwise gain benefits in a way that violates the law or ethics (Mukkamala, 2013).

c. Forensics

A subfield of research known as "network forensics" examines network security and works with digital evidence obtained from crime scenes to investigate network attacks. To confirm the attacker's attack, digital evidence will then be recognized. These attacks include DDoS, Remote to Local, U2R, and probing.(Yahya, Dirman, Buru, & Sugiantoro, 2022) The process of collecting, recording, and examining network activity to find digital traces or evidence of crimes or attacks committed through computer networks and enable perpetrators to face legal consequences, is known as network forensics**.** Finding attack patterns, anomalies in network behavior, or deviations from existing network standards can help uncover digital evidence. There are several tasks and methods of analysis in network forensics. Examples of this type of work include the examination of Network Intrusion Detection System (NIDS) procedures, network traffic analysis, and network device analysis, all of which fall under the category of network forensics (Dedy Hariyadi, 2022)

Live forensics, also known as live forensics, is a forensic method used to analyze and obtain digital evidence of a system that is running or in a living state. It involves the process of collecting and analyzing digital evidence of an active system without disrupting or stopping its operations (Pradhana, 2020)**.** In live forensics, the main objective is to obtain relevant and accurate digital evidence without compromising the integrity or continuity of the ongoing system. This process involves gathering information, analyzing data, identifying suspicious activity, and taking appropriate action.

d. Man-In-The-Middle (MITM)

ARP also known as the Address Resolution mechanism is a mechanism that uses a device's IP address to determine its MAC address on a network. Between the two and three layers of the OSI layer network layer model, ARP works. Changing a 32-bit IPv4 address to a 48-bit MAC address is the main goal of this protocol. To communicate over Ethernet or a local area network, by using ARP. Before data transfer, this network requires the MAC addresses of the sender and receiver. Users cannot share, send, receive, or communicate with other devices over a data plan without a MAC address (Mallik, Ahsan, Shahadat, & Tsou, 2019).

A MitM attack is a type of security attack in which an attacker infiltrates a communication line between two entities that are communicating, such as between two devices or between a user and a website. These attackers can monitor, or even manipulate the data that is being sent between two parties without their knowledge. MitM attacks involve exploiting weaknesses in communication systems or misappropriating certain protocols.

e. Mikrotik Router

MikroTik Router is a series of hardware and software developed by the MikroTik company. These devices are designed for use in computer and telecommunications networks to manage and direct data traffic between local area networks (LANs) and wide networks (WANs).

MikroTik Router OS is the operating system used by MikroTik Router devices. Router OS is a feature-rich Linux-based operating system designed for use on MikroTik network devices. OS routers provide a variety of features and functions that allow users to configure, manage, and monitor the network with ease (Haeruddin, 2021).

MikroTik Router provides a variety of hardware that can be used in a network, including routers, switches, access points, and other devices.

f. Kalilinux

Kali Linux is a Linux distribution devoted to computer security and penetration testing purposes. Developed by Offensive Security, one of the companies known for creating pentesting tools such as Kali Linux, a Linux distribution popular among security and penetration testing professionals.

The company also conducts trainings such as OSCP (Offensive Security Certified Professional) certification, which is highly respected in the information security industry.Kali Linux provides a wide range of tools and utilities used by security professionals, hacker ethics, and security researchers to test system security, perform forensic analysis, and conduct penetration tests on networks and applications (Rasyidah, Setyawan, & Amnur, 2022).

By providing more than 600 pre-installed tools related to information security, Kali Linux allows users to perform a variety of tasks, such as penetration testing, security auditing, data recovery, password cracking, and more. As an operating system designed specifically for security purposes, Kali Linux

g. Wirshark

Wireshark is a network protocol analysis software that allows users to monitor and analyze the data traffic flowing on a computer network (Ardiyasa, 2021). With its ability to record data packets sent and received by devices on a network, Wireshark allows users to inspect, analyze, and troubleshoot network-related, issues by viewing details of the protocols used in those communications. It is a useful tool both for network troubleshooting and for security purposes (Hanipah, 2020).

METHOD

In this research test simulation there are several stages that have the purpose and result for ARP poisoning attacks and see network security is carried out with the parameters of Test Scenario 1. Target Determination, 2. Perform the Arp Poisoning Technique on the target, 3. Checking Target Data Traffic, 4. Target Data Collection, 5. Report

Figure 2. Testing Scenario

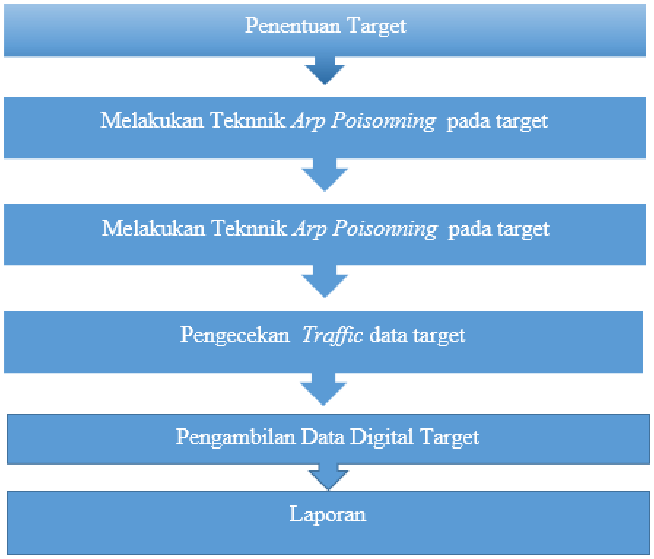
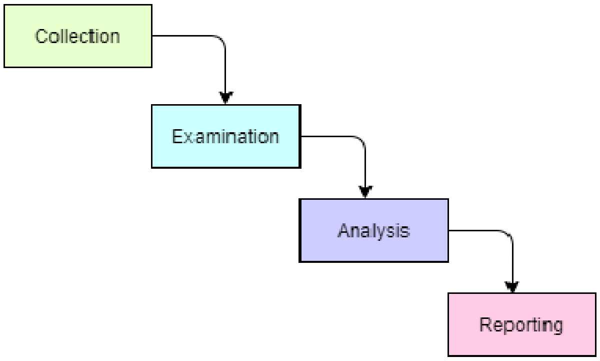

Penentuan Target
Melakukan Teknnik *Arp Poisonning* pada target
Melakukan Teknnik *Arp Poisonning* pada target
Pengecekan *Traffic* data target
Pengambilan Data Digital Target
Laporan

Figure 2. describes the test scenario from the study conducted starting with identifying the system that includes checking the system to be tested such as the vendor, MAC address, and the default IP gateway of the network device. After the identification process is carried out, it is continued with the determination of the test site, including the determination of several test points that will be carried out to obtain several samples of test result data. This test simulation connects from the network to which the target is connected by a WLAN (Wireless Local Area Network) connection. From the testing stage, and the research using the NIST Framework method can be known the framework of the forensic process in a network.

Where the process of these stages is carried out, namely; 1 Collection, 2 Examination, 3 Analysis, 4 Report, the method can be seen below:

Figure 3. NIS Framework Methods



From figure 3 above, the NIST (National Institute of Standards and Technology) Framework Method for ARP (Address Resolution Protocolv) poisoning attacks is a series of steps or procedures that are designed to assist organizations in identifying, protecting, detecting, responding, and recovering from such attacks. Method is a method used to collect information data and electronic data evidence on a network The computer is turned on, this method aims for faster handling.

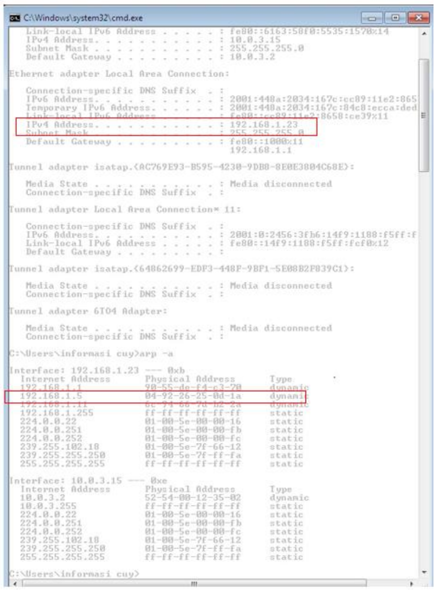
RESULTS AND DISCUSSION

This study uses the NIST forensic framework with the simulation of ARP poisoning attacks using the Virtual Box Application. This forensic research will use a simulation of an Arp attack using the Operating System (Kalilinux) as an Attacker, which is aimed at the target Clinet PC by using a Mikrotik Router configuration on a network that has been forwarded. In this study using the Live Forensic method.

a. Collection (*Pengumpulan Data*)

Collections are the earliest stages in the NIST framework. Things that are done at this stage are collecting, documenting, identifying, recording or retrieving data from relevant data sources according to procedures to maintain data integrity. In this study, the evidence used is Searching for IP Addrass and ARP Tables that will be targeted for attack, which previously used CMD (Command Prompt).

Figure 4. IP Address and MAC Address Client on CMD



```
   Link-local IPv6 Address . . . . . : fe80::6163:58f0:5535:1570%14
   IPv4 Address. . . . . . . . . . . : 10.0.3.15
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.0.3.2

Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . :
   IPv6 Address. . . . . . . . . . . : 2001:448a:2034:167c:cc89:11e2:865
   Temporary IPv6 Address. . . . . . : 2001:448a:2034:167c:84c8:ecca:ded
   Link-local IPv6 Address . . . . . : fe80::cc89:11e2:8658:ce39%11
   IPv4 Address. . . . . . . . . . . : 192.168.1.23
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : fe80::1000%11
                                       192.168.1.1

Tunnel adapter isatap.{AC769E93-B595-4230-9DB8-8E0E3804C68E}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Tunnel adapter Local Area Connection* 11:

   Connection-specific DNS Suffix  . :
   IPv6 Address. . . . . . . . . . . : 2001:0:2456:3fb6:14f9:1188:f5ff:f
   Link-local IPv6 Address . . . . . : fe80::14f9:1188:f5ff:fcf0%12
   Default Gateway . . . . . . . . . :

Tunnel adapter isatap.{64862699-EDF3-448F-9BF1-5E08B2F839C1}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Tunnel adapter 6TO4 Adapter:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

C:\Users\informasi cuy>arp -a

Interface: 192.168.1.23 --- 0xb
  Internet Address      Physical Address      Type
  192.168.1.1           90-55-de-f4-c3-70     dynamic
  192.168.1.5           04-92-26-25-0d-1a     dynamic
  192.168.1.11          6c-94-66-7d-b2-2a     dynamic
  192.168.1.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.102.18        01-00-5e-7f-66-12     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

Interface: 10.0.3.15 --- 0xe
  Internet Address      Physical Address      Type
  10.0.3.2              52-54-00-12-35-02     dynamic
  10.0.3.255            ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.102.18        01-00-5e-7f-66-12     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

C:\Users\informasi cuy>
```

In Figure 4 IP Address and Mac Address Client on CMD (Command Prompt.) In the Specification Analysis from CMD (Command Prompt.) that the client has an IP Address of 192.168.1.23 which has been detected by checking the ARP table 192.168.1.5 and mac Address 04-92-26-25-0d-1a, then recorded for the purpose of IP data evidence and information.

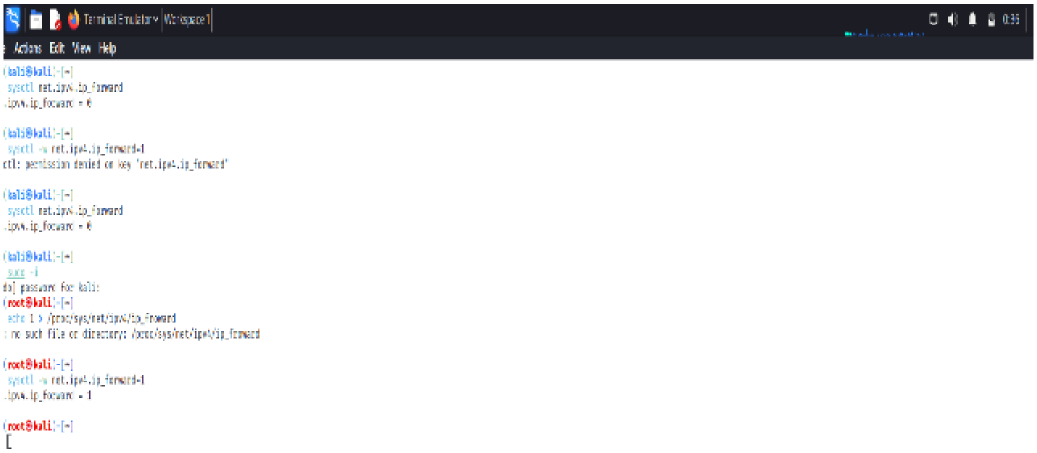
Address, which is contained in the client and ARP, to ensure proof of the information that will be targeted by the Attacker/MiTM, by using Ettercup kalilinux by activating the ARP Poisoning/spoofing Tools so that the data can be duplicated.

Table 1. *Address* PC *Client*

No	IP *Adress*	Mac *Address*
1	192.168.1.23	08-00-27-7B-86-14
2	10-0.3.15	08-00-27-7F-1D-7D
3	192.168.1.5	04-92-26-25-0d-1a
4	10-0.3.3	52-54-00-12-35-04

Analyze the table above that has been detected in CMD which is known target data to be in ARP *Poisoning.* After Enabling Forwarding as one of connecting in a network on a mikrotik router by using the terminal in KaliLinux in the image below:

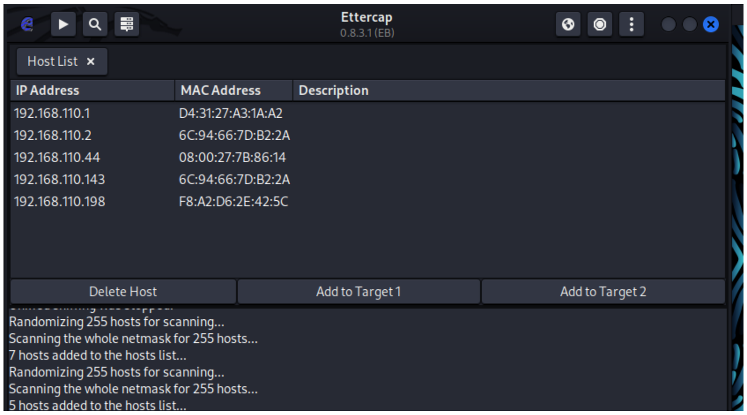
Figure 5. Frowading Activation in kalilinux terminal



```
(kali@kali)-[~]
 sysctl net.ipv4.ip_forward
.ipv4.ip_foward = 0

(kali@kali)-[~]
 sysctl -w net.ipv4.ip_forward=1
ctl: permission denied on key 'net.ipv4.ip_forward'

(kali@kali)-[~]
 sysctl net.ipv4.ip_forward
.ipv4.ip_foward = 0

(kali@kali)-[~]
 sudo -i
do] password for kali:
(root@kali)-[~]
 echo 1 > /proc/sys/net/ipv4/ip_froward
: no such file or directory: /proc/sys/net/ipv4/ip_forward

(root@kali)-[~]
 sysctl -w net.ipv4.ip_forward=1
.ipv4.ip_foward = 1

(root@kali)-[~]
[
```

Is to enable Activate Kalilinux PC as Man in The Middle (MiTM) / Attacer to test the digital proof of Ip Address by means of Current Target Progress where the Ip Address to be detected as a target to be snafed, which has been found whether it can be detected on an Ettercup application tool on an operating system seen in the Figure here

Figure 6. Current target IP Address Progress



IP Address	MAC Address	Description
192.168.110.1	D4:31:27:A3:1A:A2	
192.168.110.2	6C:94:66:7D:B2:2A	
192.168.110.44	08:00:27:7B:86:14	
192.168.110.143	6C:94:66:7D:B2:2A	
192.168.110.198	F8:A2:D6:2E:42:5C	

```
Randomizing 255 hosts for scanning...
Scanning the whole netmask for 255 hosts...
7 hosts added to the hosts list...
Randomizing 255 hosts for scanning...
Scanning the whole netmask for 255 hosts...
5 hosts added to the hosts list...
```

Above is ARP Poisoning by scanning network traffic using Wireshark. ARP manipulates the MAC address of the device connected to the network so that what the host sends will first go to the attacker's MAC address. Figure 4.1 shows the Attacker ARP reply to the user/client with the IP Address 192.168.110.5 and the router to manipulate the MAC address of the recipient so that the packet sent will pass through the attacker.

Simulasi Serangan ARP

Figure 7. Illustration/Simulation of ARP Poisoning/Spoofing Attack

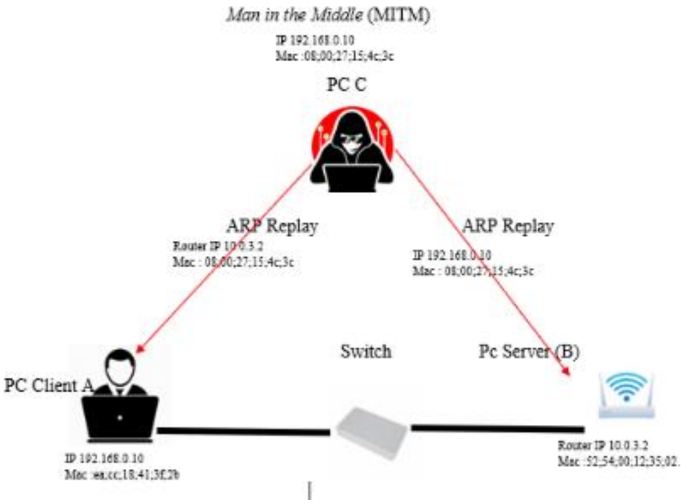
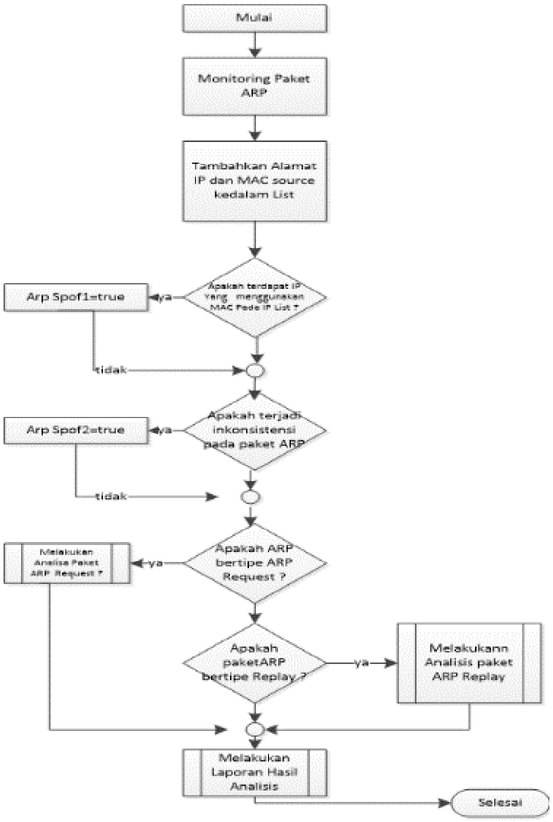


Figure 7 above shows the simulation of the Attacker performing ARP Poisoning on the target PC Client or User.

Figure 8. Flow Chart Simulation of ARP Poisoning/Spoofing Series Approach


Mulai
Monitoring Paket ARP
Tambahkan Alamat IP dan MAC source kedalam List
Apakah terdapat IP Yang menggunakan MAC Pada IP List ?
ya
Arp Spof1=true
tidak
Apakah terjadi inkonsistensi pada paket ARP
ya
Arp Spof2=true
tidak
Apakah ARP bertipe ARP Request ?
ya
Melakukan Analisa Paket ARP Request ?
Apakah paketARP bertipe Replay ?
ya
Melakukann Analisis paket ARP Replay
Melakukan Laporan Hasil Analisis
Selesai

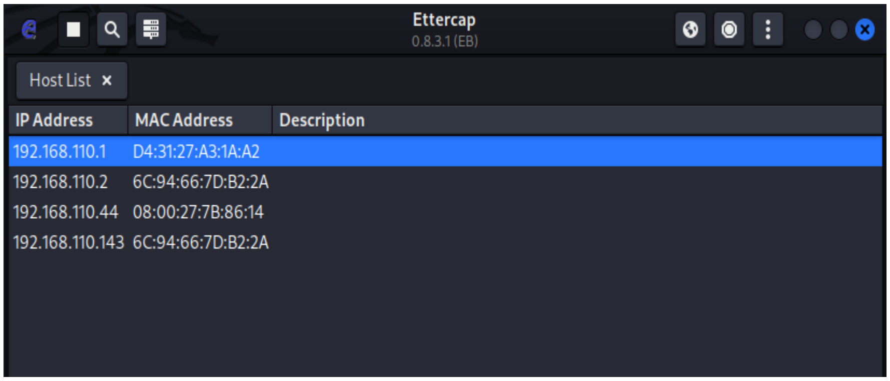
1. Monitor ARP packets using CMD (*Command Prompt*).
2. Forward on the network to connect from the intended target.
3. If there is an ARP packet, the IP address and MAC *source* of the packet are stored in *the* IPMAC dictionary. An inspection is carried out to see if there are IP-MAC pairs stored in the *dictionary*. If so, there is no need to do storage. If not, a search is carried out to see if the MAC address on the IP has been used by another IP address. If yes, there is a duplicate MAC *address*. ARPING is performed to the two IP addresses to ensure the real IP address using the MAC address.
4. ARP packet consistency check where checks are carried out whether the MAC *source address* on the *Ethernet header* and the ARP header are the same. If a difference is found, there is an inconsistency in the ARP package.
5. Analysis based on the type of ARP packet. In the ARP request package*, the* analysis flow diagram on the package that has been attacked and *captured* on *Wiresharak.*
6. If the ARP package meets the conditions as a fake ARP package or a *spoofed* ARP package , the package information is stored in *a log file*.
7. At this stage, data is collected from simulated attacks before the *attacker* is created. Data collection is carried out

b. Examintation

The examination stage is the stage of examining and retrieving Ip Address and Mac Addess data using experiments on the Kalilinux operating system using the predetermined Ettercup Grafhical tools.

Examination Serching Retrieval of IP Address and Mac to Target Snaffing Using Etterstamp Grafhical on the Kalilinux Operation System

Figure 9. IP Address and MAC Address Serching Data

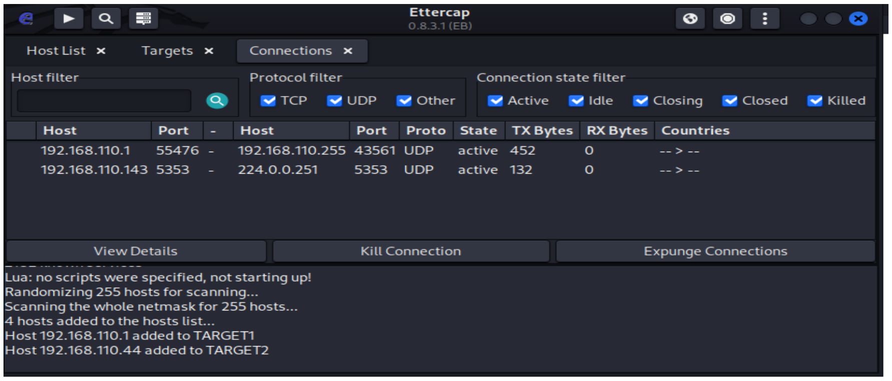


IP Address	MAC Address	Description
192.168.110.1	D4:31:27:A3:1A:A2	
192.168.110.2	6C:94:66:7D:B2:2A	
192.168.110.44	08:00:27:7B:86:14	
192.168.110.143	6C:94:66:7D:B2:2A	

Figure 10. Target 1 and Target 2 Client Connection/Host Activation

The image above shows a list of targets that we found on the network. Want and see that Ip Address and Mac Addres will be able to narrow down the target space? And start to connect those IPs.

Figure 11. Hosts to be Targeted with ARP Poisoning/Spoofing

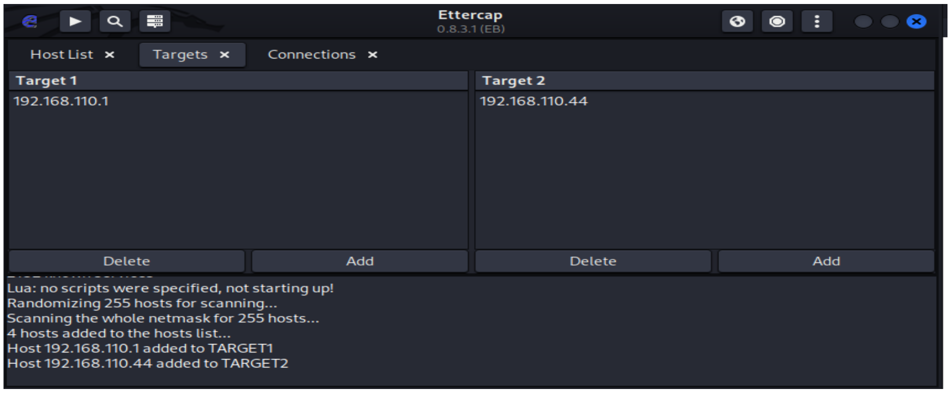


Host	Port	-	Host	Port	Proto	State	TX Bytes	RX Bytes	Countries
192.168.110.1	55476	-	192.168.110.255	43561	UDP	active	452	0	-- > --
192.168.110.143	5353	-	224.0.0.251	5353	UDP	active	132	0	-- > --

```
Lua: no scripts were specified, not starting up!
Randomizing 255 hosts for scanning...
Scanning the whole netmask for 255 hosts...
4 hosts added to the hosts list...
Host 192.168.110.1 added to TARGET1
Host 192.168.110.44 added to TARGET2
```

The image above shows that from targets I and II in the table above, it will be ready to be attacked on the MiTM (*Man-in-The-Middle*) menu, host 192.168.110.1 which is input to target I, and host 192,168.110.44 input to target II.

Figure 12. Attack on Target with ARP Poisoning/Spoofing



Target 1	Target 2
192.168.110.1	192.168.110.44

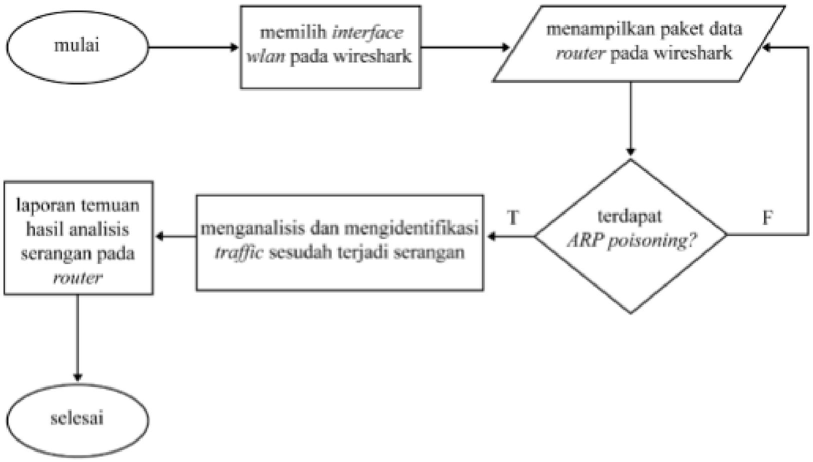
```
Lua: no scripts were specified, not starting up!
Randomizing 255 hosts for scanning...
Scanning the whole netmask for 255 hosts...
4 hosts added to the hosts list...
Host 192.168.110.1 added to TARGET1
Host 192.168.110.44 added to TARGET2
```

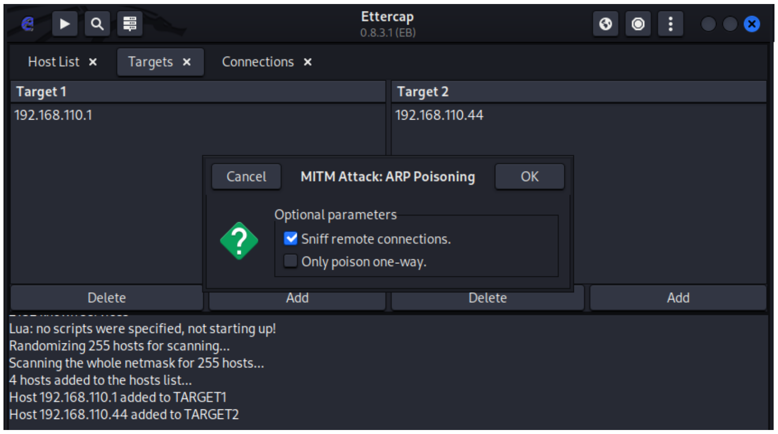
From the image above, the template explains the simulation of the target that has been entered as a Client host, which is run using ARP Poisoning Tools for sniffing. After this occurs, it is possible to capture the login credentials of the targeted user if they are logged into a website that does not use HTTPS.

c. Analysis (Analisis)

Analysis is a stage to see the results of the examination stage in detail to uncover digital evidence. This study limits the search for digital evidence of ARP poisoning attacks using Wireshark and XARP *Flowchart* forensic analysis shown in the Figure below :

Figure 13. Live Forensics Process Flowchart


mulai
memilih *interface wlan* pada wireshark
menampilkan paket data *router* pada wireshark
terdapat *ARP poisoning?*
T
F
menganalisis dan mengidentifikasi *traffic* sesudah terjadi serangan
laporan temuan hasil analisis serangan pada *router*
selesai



Target 1: 192.168.110.1

Target 2: 192.168.110.44

MITM Attack: ARP Poisoning

Optional parameters

☑ Sniff remote connections.

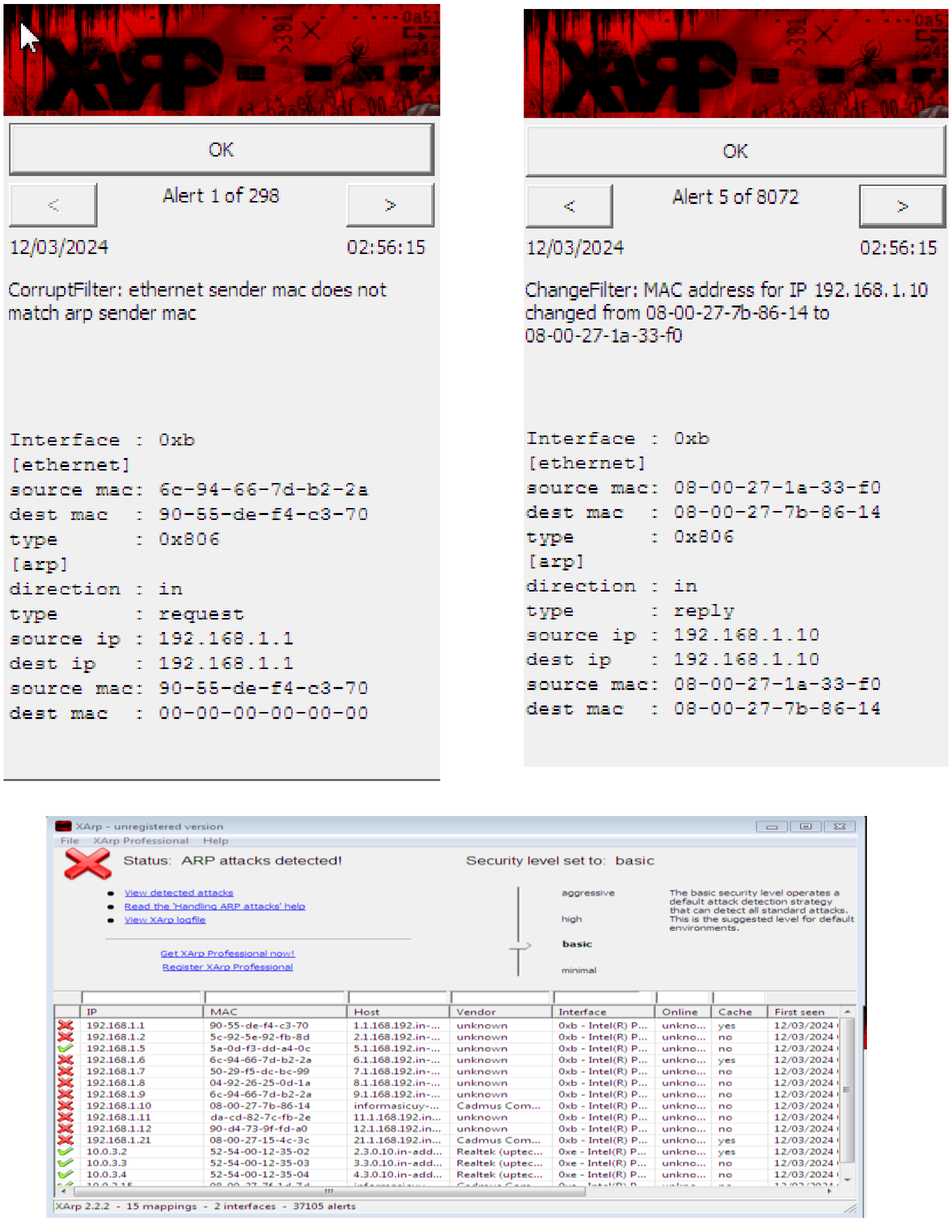
☐ Only poison one-way.

```
Lua: no scripts were specified, not starting up!
Randomizing 255 hosts for scanning...
Scanning the whole netmask for 255 hosts...
4 hosts added to the hosts list...
Host 192.168.110.1 added to TARGET1
Host 192.168.110.44 added to TARGET2
```

1. Analysis Using XARP

Analyze and This can be seen in the XARP tool which constantly provides early warnings (alerts) in real time on the investigator's PC screen display, Capture obtained by the investigator (investigator) through the XARP tool it can be seen that there is an ARP-based attack in which the attacker manipulates the MAC address of the server, this is intended so that the attacker can disguise as if it were a server so that in this case the client sends data packets continuously which are redirected to traffic through the attacker's PC then the data is forwarded back to the server which makes these client data intercepted by the attacker. Based on the digital evidence image on the XARP application display, it can be seen that IP 192.168.1.1 has the same MAC address as the server's IP address, namely 192.168.1.10, both IP addresses have MAC address b4-2e-99-90- 3f-70 as seen below.

Figure 15. XARP Detecting ARP Poisoning Attacks



```
OK
<        Alert 1 of 298        >
12/03/2024                02:56:15
CorruptFilter: ethernet sender mac does not
match arp sender mac

Interface : 0xb
[ethernet]
source mac: 6c-94-66-7d-b2-2a
dest mac  : 90-55-de-f4-c3-70
type      : 0x806
[arp]
direction : in
type      : request
source ip : 192.168.1.1
dest ip   : 192.168.1.1
source mac: 90-55-de-f4-c3-70
dest mac  : 00-00-00-00-00-00
```

```
OK
<        Alert 5 of 8072        >
12/03/2024                02:56:15
ChangeFilter: MAC address for IP 192.168.1.10
changed from 08-00-27-7b-86-14 to
08-00-27-1a-33-f0

Interface : 0xb
[ethernet]
source mac: 08-00-27-1a-33-f0
dest mac  : 08-00-27-7b-86-14
type      : 0x806
[arp]
direction : in
type      : reply
source ip : 192.168.1.10
dest ip   : 192.168.1.10
source mac: 08-00-27-1a-33-f0
dest mac  : 08-00-27-7b-86-14
```

Status: ARP attacks detected!

Security level set to: basic

- View detected attacks
- Read the 'Handling ARP attacks' help
- View XArp logfile

Get XArp Professional now!
Register XArp Professional

The basic security level operates a default attack detection strategy that can detect all standard attacks. This is the suggested level for default environments.

	IP	MAC	Host	Vendor	Interface	Online	Cache	First seen
✗	192.168.1.1	90-55-de-f4-c3-70	1.1.168.192.in-...	unknown	0xb - Intel(R) P...	unkno...	yes	12/03/2024
✗	192.168.1.2	5c-92-5e-92-fb-8d	2.1.168.192.in-...	unknown	0xb - Intel(R) P...	unkno...	no	12/03/2024
✓	192.168.1.5	5a-0d-f3-dd-a4-0c	5.1.168.192.in-...	unknown	0xb - Intel(R) P...	unkno...	no	12/03/2024
✗	192.168.1.6	6c-94-66-7d-b2-2a	6.1.168.192.in-...	unknown	0xb - Intel(R) P...	unkno...	yes	12/03/2024
✗	192.168.1.7	50-29-f5-dc-bc-99	7.1.168.192.in-...	unknown	0xb - Intel(R) P...	unkno...	no	12/03/2024
✗	192.168.1.8	04-92-26-25-0d-1a	8.1.168.192.in-...	unknown	0xb - Intel(R) P...	unkno...	no	12/03/2024
✗	192.168.1.9	6c-94-66-7d-b2-2a	9.1.168.192.in-...	unknown	0xb - Intel(R) P...	unkno...	no	12/03/2024
✗	192.168.1.10	08-00-27-7b-86-14	informasicuy-...	Cadmus Com...	0xb - Intel(R) P...	unkno...	no	12/03/2024
✗	192.168.1.11	da-cd-82-7c-fb-2e	11.1.168.192.in...	unknown	0xb - Intel(R) P...	unkno...	no	12/03/2024
✗	192.168.1.12	90-d4-73-9f-fd-a0	12.1.168.192.in...	unknown	0xb - Intel(R) P...	unkno...	no	12/03/2024
✗	192.168.1.21	08-00-27-15-4c-3c	21.1.168.192.in...	Cadmus Com...	0xb - Intel(R) P...	unkno...	yes	12/03/2024
✓	10.0.3.2	52-54-00-12-35-02	2.3.0.10.in-add...	Realtek (uptec...	0xe - Intel(R) P...	unkno...	yes	12/03/2024
✓	10.0.3.3	52-54-00-12-35-03	3.3.0.10.in-add...	Realtek (uptec...	0xe - Intel(R) P...	unkno...	no	12/03/2024
✓	10.0.3.4	52-54-00-12-35-04	4.3.0.10.in-add...	Realtek (uptec...	0xe - Intel(R) P...	unkno...	no	12/03/2024

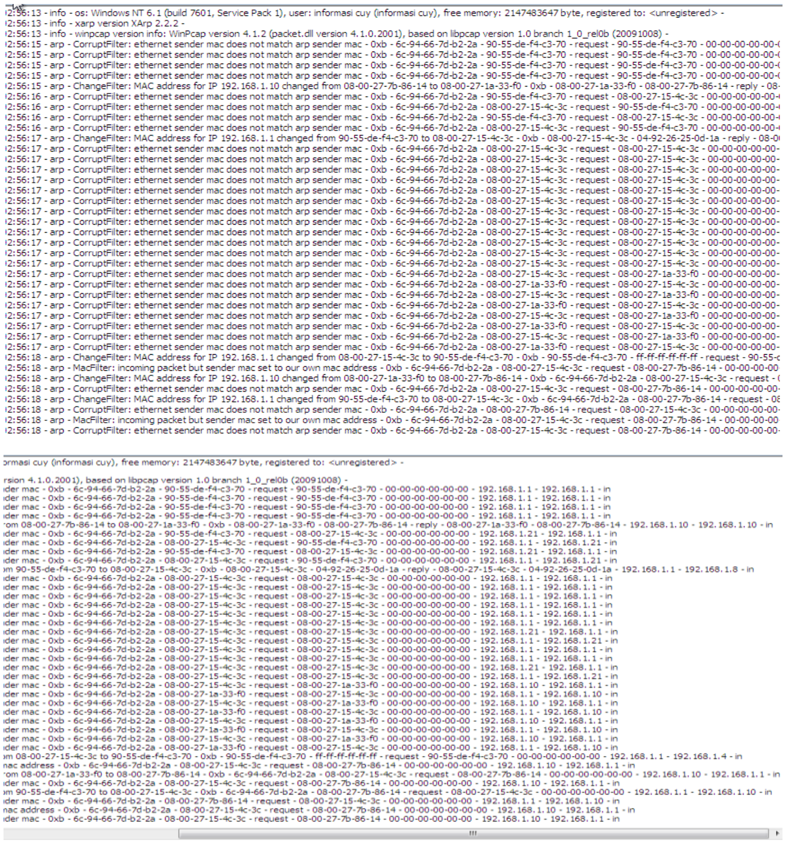
XArp 2.2.2 - 15 mappings - 2 interfaces - 37105 alerts

Once the ARP poisoning attack is detected in the XARP application, it then checks the incoming attack on network traffic using wireshark. Based on the image above, it can be seen that there is suspicious activity that has been caught by the wireshark application, it can be seen in the protocol column that there is one host broadcasting using the ARP protocol type at 32.760586 seconds to 32.944116 seconds, this can be assumed that there is an attack that utilizes the ARP protocol in its attack activities because the only protocols available on the server are the FTP protocol and the SSL protocol.

1. Forensic investigation on XARP

The investigation stage is carried out to obtain stronger evidence related to the detection of ARP poisoning attacks on a tissue. In this study, the investigator has used wireshark tools and then the investigator identifies through notifications and captures obtained from the XARP tools.

Figure 16. XARP Displaying Scanning Data Victim



```
2:56:13 - info - os: Windows NT 6.1 (build 7601, Service Pack 1), user: informasi cuy (informasi cuy), free memory: 2147483647 byte, registered to: <unregistered> -
2:56:13 - info - xarp version XArp 2.2.2 -
2:56:13 - info - winpcap version info: WinPcap version 4.1.2 (packet.dll version 4.1.0.2001), based on libpcap version 1.0 branch 1_0_rel0b (20091008) -
2:56:15 - arp - CorruptFilter: ethernet sender mac does not match arp sender mac - 0xb - 6c-94-66-7d-b2-2a - 90-55-de-f4-c3-70 - request - 90-55-de-f4-c3-70 - 00-00-00-00-00-
2:56:15 - arp - CorruptFilter: ethernet sender mac does not match arp sender mac - 0xb - 6c-94-66-7d-b2-2a - 90-55-de-f4-c3-70 - request - 90-55-de-f4-c3-70 - 00-00-00-00-00-
2:56:15 - arp - CorruptFilter: ethernet sender mac does not match arp sender mac - 0xb - 6c-94-66-7d-b2-2a - 90-55-de-f4-c3-70 - request - 90-55-de-f4-c3-70 - 00-00-00-00-00-
2:56:15 - arp - CorruptFilter: ethernet sender mac does not match arp sender mac - 0xb - 6c-94-66-7d-b2-2a - 90-55-de-f4-c3-70 - request - 90-55-de-f4-c3-70 - 00-00-00-00-00-
2:56:15 - arp - ChangeFilter: MAC address for IP 192.168.1.10 changed from 08-00-27-7b-86-14 to 08-00-27-1a-33-f0 - 0xb - 08-00-27-1a-33-f0 - 08-00-27-7b-86-14 - reply - 08-
2:56:16 - arp - CorruptFilter: ethernet sender mac does not match arp sender mac - 0xb - 6c-94-66-7d-b2-2a - 90-55-de-f4-c3-70 - request - 08-00-27-15-4c-3c - 00-00-00-00-00-
2:56:16 - arp - CorruptFilter: ethernet sender mac does not match arp sender mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-15-4c-3c - request - 90-55-de-f4-c3-70 - 00-00-00-00-00-
2:56:16 - arp - CorruptFilter: ethernet sender mac does not match arp sender mac - 0xb - 6c-94-66-7d-b2-2a - 90-55-de-f4-c3-70 - request - 08-00-27-15-4c-3c - 00-00-00-00-00-
2:56:16 - arp - CorruptFilter: ethernet sender mac does not match arp sender mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-15-4c-3c - request - 90-55-de-f4-c3-70 - 00-00-00-00-00-
2:56:17 - arp - ChangeFilter: MAC address for IP 192.168.1.1 changed from 90-55-de-f4-c3-70 to 08-00-27-15-4c-3c - 0xb - 08-00-27-15-4c-3c - 04-92-26-25-0d-1a - reply - 08-0
2:56:17 - arp - CorruptFilter: ethernet sender mac does not match arp sender mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-15-4c-3c - request - 08-00-27-15-4c-3c - 00-00-00-00-00-
2:56:17 - arp - CorruptFilter: ethernet sender mac does not match arp sender mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-15-4c-3c - request - 08-00-27-15-4c-3c - 00-00-00-00-00-
2:56:17 - arp - CorruptFilter: ethernet sender mac does not match arp sender mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-15-4c-3c - request - 08-00-27-15-4c-3c - 00-00-00-00-00-
2:56:17 - arp - CorruptFilter: ethernet sender mac does not match arp sender mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-15-4c-3c - request - 08-00-27-15-4c-3c - 00-00-00-00-00-
2:56:17 - arp - CorruptFilter: ethernet sender mac does not match arp sender mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-15-4c-3c - request - 08-00-27-15-4c-3c - 00-00-00-00-00-
2:56:17 - arp - CorruptFilter: ethernet sender mac does not match arp sender mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-15-4c-3c - request - 08-00-27-15-4c-3c - 00-00-00-00-00-
2:56:17 - arp - CorruptFilter: ethernet sender mac does not match arp sender mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-15-4c-3c - request - 08-00-27-15-4c-3c - 00-00-00-00-00-
2:56:17 - arp - CorruptFilter: ethernet sender mac does not match arp sender mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-15-4c-3c - request - 08-00-27-15-4c-3c - 00-00-00-00-00-
2:56:17 - arp - CorruptFilter: ethernet sender mac does not match arp sender mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-15-4c-3c - request - 08-00-27-15-4c-3c - 00-00-00-00-00-
2:56:17 - arp - CorruptFilter: ethernet sender mac does not match arp sender mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-15-4c-3c - request - 08-00-27-15-4c-3c - 00-00-00-00-00-
2:56:17 - arp - CorruptFilter: ethernet sender mac does not match arp sender mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-15-4c-3c - request - 08-00-27-15-4c-3c - 00-00-00-00-00-
2:56:17 - arp - CorruptFilter: ethernet sender mac does not match arp sender mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-15-4c-3c - request - 08-00-27-15-4c-3c - 00-00-00-00-00-
2:56:17 - arp - CorruptFilter: ethernet sender mac does not match arp sender mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-15-4c-3c - request - 08-00-27-1a-33-f0 - 00-00-00-00-00-
2:56:17 - arp - CorruptFilter: ethernet sender mac does not match arp sender mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-1a-33-f0 - request - 08-00-27-15-4c-3c - 00-00-00-00-00-
2:56:17 - arp - CorruptFilter: ethernet sender mac does not match arp sender mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-15-4c-3c - request - 08-00-27-1a-33-f0 - 00-00-00-00-00-
2:56:17 - arp - CorruptFilter: ethernet sender mac does not match arp sender mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-1a-33-f0 - request - 08-00-27-15-4c-3c - 00-00-00-00-00-
2:56:17 - arp - CorruptFilter: ethernet sender mac does not match arp sender mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-15-4c-3c - request - 08-00-27-1a-33-f0 - 00-00-00-00-00-
2:56:17 - arp - CorruptFilter: ethernet sender mac does not match arp sender mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-1a-33-f0 - request - 08-00-27-15-4c-3c - 00-00-00-00-00-
2:56:17 - arp - CorruptFilter: ethernet sender mac does not match arp sender mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-15-4c-3c - request - 08-00-27-1a-33-f0 - 00-00-00-00-00-
2:56:17 - arp - CorruptFilter: ethernet sender mac does not match arp sender mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-1a-33-f0 - request - 08-00-27-15-4c-3c - 00-00-00-00-00-
2:56:18 - arp - ChangeFilter: MAC address for IP 192.168.1.1 changed from 08-00-27-15-4c-3c to 90-55-de-f4-c3-70 - 0xb - 90-55-de-f4-c3-70 - ff-ff-ff-ff-ff-ff - request - 90-55-c
2:56:18 - arp - MacFilter: incoming packet but sender mac set to our own mac address - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-15-4c-3c - request - 08-00-27-7b-86-14 - 00-00-00-00-
2:56:18 - arp - ChangeFilter: MAC address for IP 192.168.1.10 changed from 08-00-27-1a-33-f0 to 08-00-27-7b-86-14 - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-15-4c-3c - request - (
2:56:18 - arp - CorruptFilter: ethernet sender mac does not match arp sender mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-15-4c-3c - request - 08-00-27-7b-86-14 - 00-00-00-00-00-
2:56:18 - arp - ChangeFilter: MAC address for IP 192.168.1.1 changed from 90-55-de-f4-c3-70 to 08-00-27-15-4c-3c - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-7b-86-14 - request - 08
2:56:18 - arp - CorruptFilter: ethernet sender mac does not match arp sender mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-7b-86-14 - request - 08-00-27-15-4c-3c - 00-00-00-00-00-
2:56:18 - arp - MacFilter: incoming packet but sender mac set to our own mac address - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-15-4c-3c - request - 08-00-27-7b-86-14 - 00-00-00-00-
2:56:18 - arp - CorruptFilter: ethernet sender mac does not match arp sender mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-15-4c-3c - request - 08-00-27-7b-86-14 - 00-00-00-00-00-
```

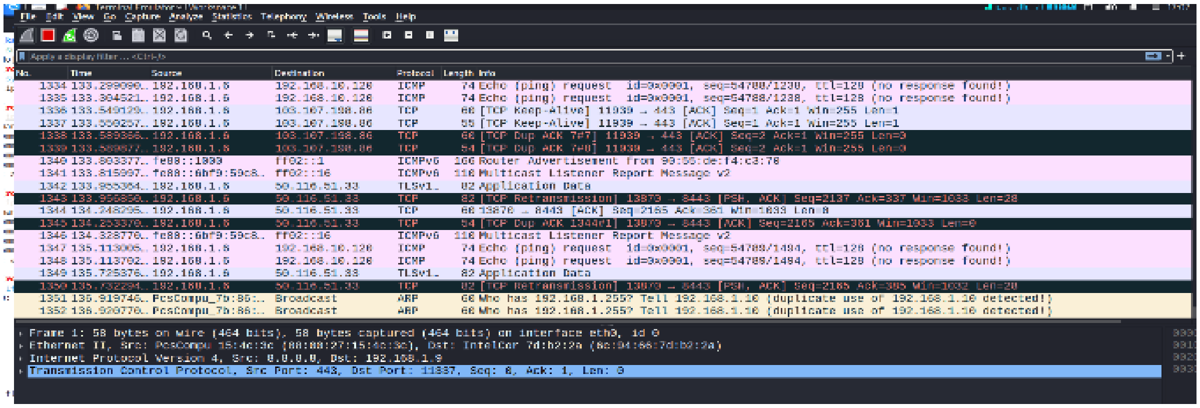
```
ormasi cuy (informasi cuy), free memory: 2147483647 byte, registered to: <unregistered> -

rsion 4.1.0.2001), based on libpcap version 1.0 branch 1_0_rel0b (20091008) -
ider mac - 0xb - 6c-94-66-7d-b2-2a - 90-55-de-f4-c3-70 - request - 90-55-de-f4-c3-70 - 00-00-00-00-00-00 - 192.168.1.1 - 192.168.1.1 - in
ider mac - 0xb - 6c-94-66-7d-b2-2a - 90-55-de-f4-c3-70 - request - 90-55-de-f4-c3-70 - 00-00-00-00-00-00 - 192.168.1.1 - 192.168.1.1 - in
ider mac - 0xb - 6c-94-66-7d-b2-2a - 90-55-de-f4-c3-70 - request - 90-55-de-f4-c3-70 - 00-00-00-00-00-00 - 192.168.1.1 - 192.168.1.1 - in
ider mac - 0xb - 6c-94-66-7d-b2-2a - 90-55-de-f4-c3-70 - request - 90-55-de-f4-c3-70 - 00-00-00-00-00-00 - 192.168.1.1 - 192.168.1.1 - in
rom 08-00-27-7b-86-14 to 08-00-27-1a-33-f0 - 0xb - 08-00-27-1a-33-f0 - 08-00-27-7b-86-14 - reply - 08-00-27-1a-33-f0 - 08-00-27-7b-86-14 - 192.168.1.10 - 192.168.1.10 - in
ider mac - 0xb - 6c-94-66-7d-b2-2a - 90-55-de-f4-c3-70 - request - 08-00-27-15-4c-3c - 00-00-00-00-00-00 - 192.168.1.21 - 192.168.1.1 - in
ider mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-15-4c-3c - request - 90-55-de-f4-c3-70 - 00-00-00-00-00-00 - 192.168.1.1 - 192.168.1.21 - in
ider mac - 0xb - 6c-94-66-7d-b2-2a - 90-55-de-f4-c3-70 - request - 08-00-27-15-4c-3c - 00-00-00-00-00-00 - 192.168.1.21 - 192.168.1.1 - in
ider mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-15-4c-3c - request - 90-55-de-f4-c3-70 - 00-00-00-00-00-00 - 192.168.1.1 - 192.168.1.21 - in
om 90-55-de-f4-c3-70 to 08-00-27-15-4c-3c - 0xb - 08-00-27-15-4c-3c - 04-92-26-25-0d-1a - reply - 08-00-27-15-4c-3c - 04-92-26-25-0d-1a - 192.168.1.1 - 192.168.1.8 - in
ider mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-15-4c-3c - request - 08-00-27-15-4c-3c - 00-00-00-00-00-00 - 192.168.1.1 - 192.168.1.1 - in
ider mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-15-4c-3c - request - 08-00-27-15-4c-3c - 00-00-00-00-00-00 - 192.168.1.1 - 192.168.1.1 - in
ider mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-15-4c-3c - request - 08-00-27-15-4c-3c - 00-00-00-00-00-00 - 192.168.1.1 - 192.168.1.1 - in
ider mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-15-4c-3c - request - 08-00-27-15-4c-3c - 00-00-00-00-00-00 - 192.168.1.1 - 192.168.1.1 - in
ider mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-15-4c-3c - request - 08-00-27-15-4c-3c - 00-00-00-00-00-00 - 192.168.1.1 - 192.168.1.1 - in
ider mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-15-4c-3c - request - 08-00-27-15-4c-3c - 00-00-00-00-00-00 - 192.168.1.1 - 192.168.1.1 - in
ider mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-15-4c-3c - request - 08-00-27-15-4c-3c - 00-00-00-00-00-00 - 192.168.1.21 - 192.168.1.1 - in
ider mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-15-4c-3c - request - 08-00-27-15-4c-3c - 00-00-00-00-00-00 - 192.168.1.1 - 192.168.1.21 - in
ider mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-15-4c-3c - request - 08-00-27-15-4c-3c - 00-00-00-00-00-00 - 192.168.1.1 - 192.168.1.1 - in
ider mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-15-4c-3c - request - 08-00-27-15-4c-3c - 00-00-00-00-00-00 - 192.168.1.1 - 192.168.1.1 - in
ider mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-15-4c-3c - request - 08-00-27-15-4c-3c - 00-00-00-00-00-00 - 192.168.1.21 - 192.168.1.1 - in
ider mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-15-4c-3c - request - 08-00-27-15-4c-3c - 00-00-00-00-00-00 - 192.168.1.1 - 192.168.1.21 - in
ider mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-15-4c-3c - request - 08-00-27-1a-33-f0 - 00-00-00-00-00-00 - 192.168.1.10 - 192.168.1.1 - in
ider mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-1a-33-f0 - request - 08-00-27-15-4c-3c - 00-00-00-00-00-00 - 192.168.1.1 - 192.168.1.10 - in
ider mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-15-4c-3c - request - 08-00-27-1a-33-f0 - 00-00-00-00-00-00 - 192.168.1.10 - 192.168.1.1 - in
ider mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-1a-33-f0 - request - 08-00-27-15-4c-3c - 00-00-00-00-00-00 - 192.168.1.1 - 192.168.1.10 - in
ider mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-15-4c-3c - request - 08-00-27-1a-33-f0 - 00-00-00-00-00-00 - 192.168.1.10 - 192.168.1.1 - in
ider mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-1a-33-f0 - request - 08-00-27-15-4c-3c - 00-00-00-00-00-00 - 192.168.1.1 - 192.168.1.10 - in
ider mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-15-4c-3c - request - 08-00-27-1a-33-f0 - 00-00-00-00-00-00 - 192.168.1.10 - 192.168.1.1 - in
ider mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-1a-33-f0 - request - 08-00-27-15-4c-3c - 00-00-00-00-00-00 - 192.168.1.1 - 192.168.1.10 - in
om 08-00-27-15-4c-3c to 90-55-de-f4-c3-70 - 0xb - 90-55-de-f4-c3-70 - ff-ff-ff-ff-ff-ff - request - 90-55-de-f4-c3-70 - 00-00-00-00-00-00 - 192.168.1.1 - 192.168.1.4 - in
nac address - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-15-4c-3c - request - 08-00-27-7b-86-14 - 00-00-00-00-00-00 - 192.168.1.10 - 192.168.1.1 - in
rom 08-00-27-1a-33-f0 to 08-00-27-7b-86-14 - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-15-4c-3c - request - 08-00-27-7b-86-14 - 00-00-00-00-00-00 - 192.168.1.10 - 192.168.1.1 - in
ider mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-15-4c-3c - request - 08-00-27-7b-86-14 - 00-00-00-00-00-00 - 192.168.1.10 - 192.168.1.1 - in
om 90-55-de-f4-c3-70 to 08-00-27-15-4c-3c - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-7b-86-14 - request - 08-00-27-15-4c-3c - 00-00-00-00-00-00 - 192.168.1.1 - 192.168.1.10 - in
ider mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-7b-86-14 - request - 08-00-27-15-4c-3c - 00-00-00-00-00-00 - 192.168.1.1 - 192.168.1.10 - in
nac address - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-15-4c-3c - request - 08-00-27-7b-86-14 - 00-00-00-00-00-00 - 192.168.1.10 - 192.168.1.1 - in
ider mac - 0xb - 6c-94-66-7d-b2-2a - 08-00-27-15-4c-3c - request - 08-00-27-7b-86-14 - 00-00-00-00-00-00 - 192.168.1.10 - 192.168.1.1 - in
```

The scanning data displayed only displays the IP address and MAC address of the victim of the ARP Poisoning attack but does not display the IP address of the attacker.

1. Analysis Using Wirsharak

The analysis stage on Wireshark conducts digital evidence testing by identifying and displaying the captured data from the device in the report document. Based on the forensic process that has been carried out on the PC Client Network in the condition that it has been duplicated, Starting Recording (Capturing) and also observation in the course of network traversal so that it can filter on packets based on IP Address and Mac address so that Protocol data can be analyzed from the findings of analyzing the recorded data seen in the figure below

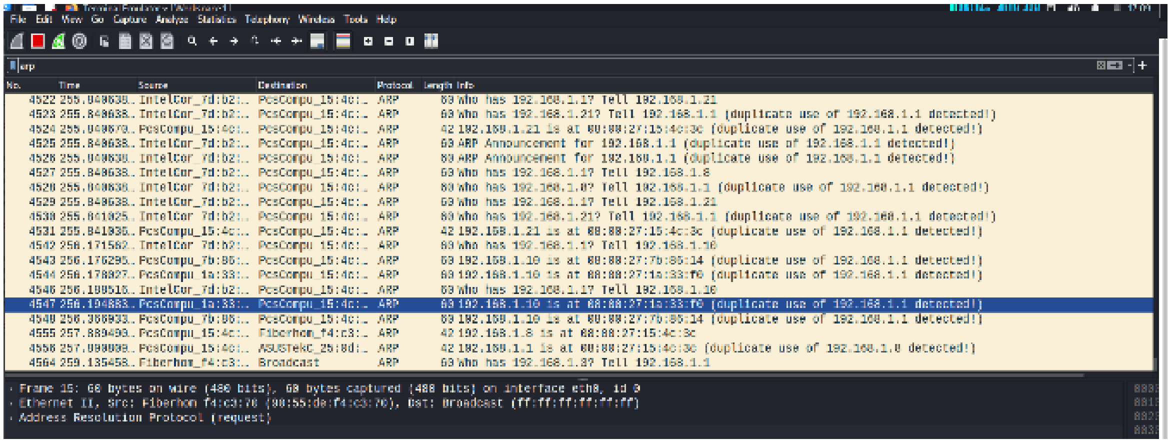
Figure 17. Wireshark Displaying Victim Data Scanning


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
No.   Time         Source              Destination      Protocol Length Info
 1334 133.299090.. 192.168.1.8         192.168.10.128   ICMP      74 Echo (ping) request  id=0x0001, seq=54788/1238, ttl=128 (no response found!)
 1335 133.304521.. 192.168.1.8         192.168.10.128   ICMP      74 Echo (ping) request  id=0x0001, seq=54788/1238, ttl=128 (no response found!)
 1336 133.549129.. 192.168.1.8         103.107.198.86   TCP       66 [TCP Keep-Alive] 11939 → 443 [ACK] Seq=1 Ack=1 Win=255 Len=1
 1337 133.550257.. 192.168.1.8         103.107.198.86   TCP       55 [TCP Keep-Alive] 11939 → 443 [ACK] Seq=1 Ack=1 Win=255 Len=1
 1338 133.589366.. 192.168.1.8         103.107.198.86   TCP       66 [TCP Dup ACK 7#7] 11939 → 443 [ACK] Seq=2 Ack=1 Win=255 Len=0
 1339 133.589677.. 192.168.1.8         103.107.198.86   TCP       54 [TCP Dup ACK 7#8] 11939 → 443 [ACK] Seq=2 Ack=1 Win=255 Len=0
 1340 133.803377.. fe80::1893          ff02::1          ICMPv6   166 Router Advertisement from 90:55:de:f4:c3:70
 1341 133.815997.. fe80::6bf9:59c8..   ff02::16         ICMPv6   116 Multicast Listener Report Message v2
 1342 133.955364.. 192.168.1.8         50.116.51.33     TLSv1..   82 Application Data
 1343 133.955850.. 192.168.1.8         50.116.51.33     TCP       82 [TCP Retransmission] 13879 → 8443 [PSH, ACK] Seq=2137 Ack=337 Win=1033 Len=28
 1344 134.248295.. 192.168.1.8         50.116.51.33     TCP       66 13878 → 8443 [ACK] Seq=2165 Ack=361 Win=1033 Len=0
 1345 134.253379.. 192.168.1.8         50.116.51.33     TCP       54 [TCP Dup ACK 1344#1] 13879 → 8443 [ACK] Seq=2165 Ack=361 Win=1033 Len=0
 1346 134.328770.. fe80::6bf9:59c8..   ff02::16         ICMPv6   116 Multicast Listener Report Message v2
 1347 135.113006.. 192.168.1.8         192.168.10.128   ICMP      74 Echo (ping) request  id=0x0001, seq=54789/1494, ttl=128 (no response found!)
 1348 135.113702.. 192.168.1.8         192.168.10.128   ICMP      74 Echo (ping) request  id=0x0001, seq=54789/1494, ttl=128 (no response found!)
 1349 135.725376.. 192.168.1.8         50.116.51.33     TLSv1..   82 Application Data
 1350 135.732294.. 192.168.1.8         50.116.51.33     TCP       82 [TCP Retransmission] 13879 → 8443 [PSH, ACK] Seq=2165 Ack=385 Win=1032 Len=28
 1351 135.919746.. PcsCompu_7b:86:..   Broadcast        ARP       68 Who has 192.168.1.255? Tell 192.168.1.10 (duplicate use of 192.168.1.10 detected!)
 1352 135.920770.. PcsCompu_7b:86:..   Broadcast        ARP       68 Who has 192.168.1.255? Tell 192.168.1.10 (duplicate use of 192.168.1.10 detected!)
```

```
› Frame 1: 58 bytes on wire (464 bits), 58 bytes captured (464 bits) on interface eth0, id 0
› Ethernet II, Src: PcsCompu 15:4c:3c (08:00:27:15:4c:3c), Dst: IntelCor 7d:b2:2a (8c:04:66:7d:b2:2a)
› Internet Protocol Version 4, Src: 0.0.0.0, Dst: 192.168.1.9
› Transmission Control Protocol, Src Port: 443, Dst Port: 11837, Seq: 0, Ack: 1, Len: 0
```

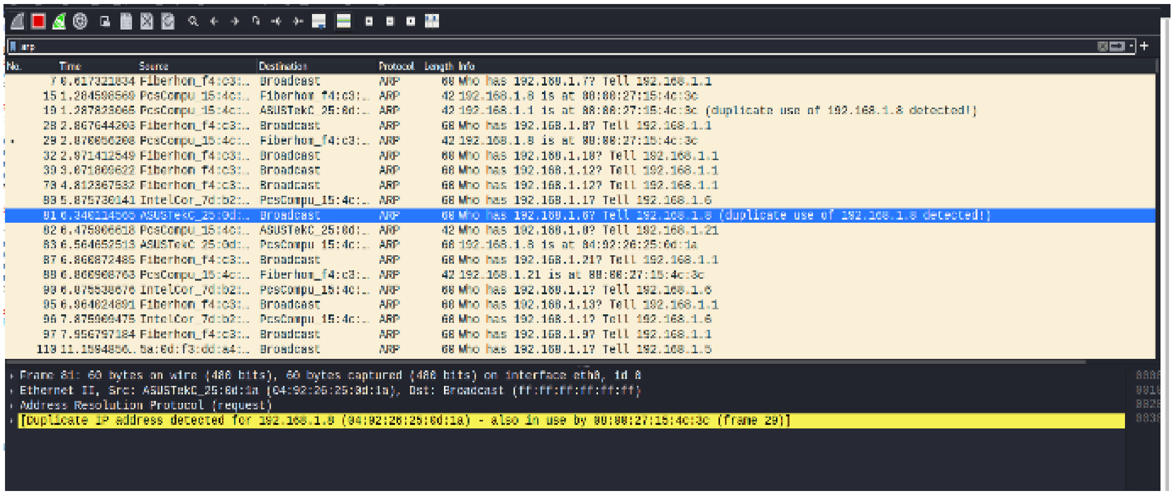
Figure 18. Wireshark Detects ARP Poisoning Attack


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
arp

```
No.   Time         Source              Destination         Protocol Length Info
 4522 255.840638.. IntelCor_7d:b2:..   PcsCompu_15:4c:..   ARP       60 Who has 192.168.1.1? Tell 192.168.1.21
 4523 255.840638.. IntelCor_7d:b2:..   PcsCompu_15:4c:..   ARP       60 Who has 192.168.1.21? Tell 192.168.1.1 (duplicate use of 192.168.1.1 detected!)
 4524 255.840670.. PcsCompu_15:4c:..   PcsCompu_15:4c:..   ARP       42 192.168.1.21 is at 08:00:27:15:4c:3c (duplicate use of 192.168.1.1 detected!)
 4525 255.840638.. IntelCor_7d:b2:..   PcsCompu_15:4c:..   ARP       60 ARP Announcement for 192.168.1.1 (duplicate use of 192.168.1.1 detected!)
 4526 255.840638.. IntelCor 7d:b2:..   PcsCompu 15:4c:..   ARP       60 ARP Announcement for 192.168.1.1 (duplicate use of 192.168.1.1 detected!)
 4527 255.840638.. IntelCor_7d:b2:..   PcsCompu_15:4c:..   ARP       60 Who has 192.168.1.1? Tell 192.168.1.8
 4528 255.840638.. IntelCor 7d:b2:..   PcsCompu 15:4c:..   ARP       60 Who has 192.168.1.8? Tell 192.168.1.1 (duplicate use of 192.168.1.1 detected!)
 4529 255.840638.. IntelCor_7d:b2:..   PcsCompu_15:4c:..   ARP       60 Who has 192.168.1.1? Tell 192.168.1.21
 4530 255.841025.. IntelCor 7d:b2:..   PcsCompu 15:4c:..   ARP       60 Who has 192.168.1.21? Tell 192.168.1.1 (duplicate use of 192.168.1.1 detected!)
 4531 255.841036.. PcsCompu_15:4c:..   PcsCompu_15:4c:..   ARP       42 192.168.1.21 is at 08:00:27:15:4c:3c (duplicate use of 192.168.1.1 detected!)
 4542 256.171582.. IntelCor 7d:b2:..   PcsCompu 15:4c:..   ARP       60 Who has 192.168.1.1? Tell 192.168.1.10
 4543 256.176295.. PcsCompu_7b:86:..   PcsCompu_15:4c:..   ARP       60 192.168.1.10 is at 08:00:27:7b:86:14 (duplicate use of 192.168.1.1 detected!)
 4544 256.178827.. PcsCompu 1a:33:..   PcsCompu 15:4c:..   ARP       60 192.168.1.10 is at 08:00:27:1a:33:f0 (duplicate use of 192.168.1.1 detected!)
 4546 256.188516.. IntelCor_7d:b2:..   PcsCompu_15:4c:..   ARP       60 Who has 192.168.1.1? Tell 192.168.1.10
 4547 256.194883.. PcsCompu_1a:33:..   PcsCompu_15:4c:..   ARP       60 192.168.1.10 is at 08:00:27:1a:33:f0 (duplicate use of 192.168.1.1 detected!)
 4548 256.366933.. PcsCompu_7b:86:..   PcsCompu_15:4c:..   ARP       60 192.168.1.10 is at 08:00:27:7b:86:14 (duplicate use of 192.168.1.1 detected!)
 4555 257.889493.. PcsCompu_15:4c:..   Fiberhom_f4:c3:..   ARP       42 192.168.1.8 is at 08:00:27:15:4c:3c
 4558 257.090009.. PcsCompu_15:4c:..   ASUSTekC_25:8d:..   ARP       42 192.168.1.1 is at 08:00:27:15:4c:3c (duplicate use of 192.168.1.8 detected!)
 4564 259.135458.. Fiberhom_f4:c3:..   Broadcast           ARP       60 Who has 192.168.1.3? Tell 192.168.1.1
```

```
› Frame 15: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface eth0, id 0
› Ethernet II, Src: Fiberhom f4:c3:70 (90:55:de:f4:c3:70), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
› Address Resolution Protocol (request)
```

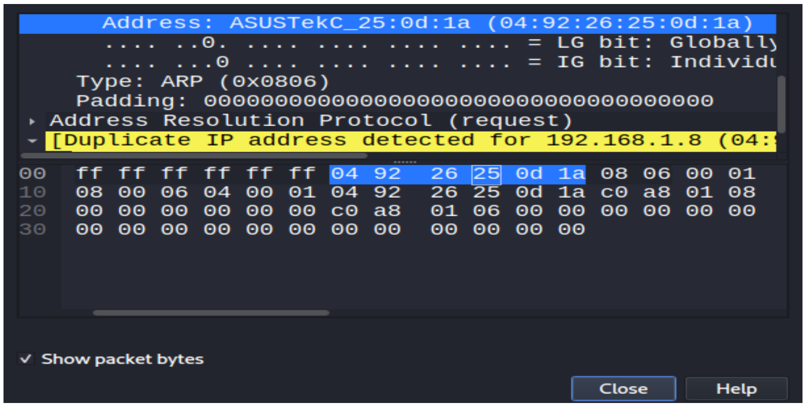
Figure 19. Wireshark Detects ARP Poisoning Attack



```
No. Time        Source              Destination         Protocol Length Info
  7 0.617321034 Fiberhom_f4:c3:..   Broadcast           ARP       60 Who has 192.168.1.7? Tell 192.168.1.1
 15 1.284598560 PcsCompu 15:4c:..   Fiberhom f4:c3:..   ARP       42 192.168.1.8 is at 08:00:27:15:4c:3c
 19 1.287823065 PcsCompu 15:4c:..   ASUSTekC 25:8d:..   ARP       42 192.168.1.1 is at 08:00:27:15:4c:3c (duplicate use of 192.168.1.8 detected!)
 28 2.867644203 Fiberhom_f4:c3:..   Broadcast           ARP       60 Who has 192.168.1.8? Tell 192.168.1.1
 29 2.870056208 PcsCompu_15:4c:..   Fiberhom_f4:c3:..   ARP       42 192.168.1.8 is at 08:00:27:15:4c:3c
 32 2.871412549 Fiberhom_f4:c3:..   Broadcast           ARP       60 Who has 192.168.1.18? Tell 192.168.1.1
 39 3.871809622 Fiberhom f4:c3:..   Broadcast           ARP       60 Who has 192.168.1.12? Tell 192.168.1.1
 73 4.812367532 Fiberhom_f4:c3:..   Broadcast           ARP       60 Who has 192.168.1.12? Tell 192.168.1.1
 83 5.875736141 IntelCor_7d:b2:..   PcsCompu_15:4c:..   ARP       60 Who has 192.168.1.1? Tell 192.168.1.6
 81 6.340114565 ASUSTekC_25:8d:..   Broadcast           ARP       60 Who has 192.168.1.6? Tell 192.168.1.8 (duplicate use of 192.168.1.8 detected!)
 82 6.475906618 PcsCompu_15:4c:..   ASUSTekC_25:8d:..   ARP       42 Who has 192.168.1.8? Tell 192.168.1.21
 83 6.564652513 ASUSTekC 25:8d:..   PcsCompu 15:4c:..   ARP       60 192.168.1.8 is at 04:92:26:25:8d:1a
 87 6.860872485 Fiberhom_f4:c3:..   Broadcast           ARP       60 Who has 192.168.1.21? Tell 192.168.1.1
 88 6.860968763 PcsCompu_15:4c:..   Fiberhom_f4:c3:..   ARP       42 192.168.1.21 is at 08:00:27:15:4c:3c
 90 6.875538676 IntelCor_7d:b2:..   PcsCompu_15:4c:..   ARP       60 Who has 192.168.1.1? Tell 192.168.1.6
 95 6.964024891 Fiberhom f4:c3:..   Broadcast           ARP       60 Who has 192.168.1.13? Tell 192.168.1.1
 96 7.875969475 IntelCor 7d:b2:..   PcsCompu 15:4c:..   ARP       60 Who has 192.168.1.1? Tell 192.168.1.6
 97 7.956797184 Fiberhom_f4:c3:..   Broadcast           ARP       60 Who has 192.168.1.9? Tell 192.168.1.1
113 11.1594856.. 5a:6d:f3:dd:a4:..  Broadcast           ARP       60 Who has 192.168.1.1? Tell 192.168.1.5
```

```
› Frame 81: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface eth0, id 0
› Ethernet II, Src: ASUSTekC_25:8d:1a (04:92:26:25:8d:1a), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
› Address Resolution Protocol (request)
› [Duplicate IP address detected for 192.168.1.8 (04:92:26:25:8d:1a) - also in use by 08:00:27:15:4c:3c (frame 29)]
```

Figure 20. Interface Detecting Duplicate ARP and MAC Address in Frame 29



```
      Address: ASUSTekC_25:0d:1a (04:92:26:25:0d:1a)
      .... ..0. .... .... .... .... = LG bit: Globally
      .... ...0 .... .... .... .... = IG bit: Individu
    Type: ARP (0x0806)
    Padding: 000000000000000000000000000000000000
 ▸ Address Resolution Protocol (request)
 ▾ [Duplicate IP address detected for 192.168.1.8 (04:
00   ff ff ff ff ff ff 04 92  26 25 0d 1a 08 06 00 01
10   08 00 06 04 00 01 04 92  26 25 0d 1a c0 a8 01 08
20   00 00 00 00 00 00 c0 a8  01 06 00 00 00 00 00 00
30   00 00 00 00 00 00 00 00  00 00 00 00
```

c. Report

Based on the results of the analysis of ARP Poisoning attacks using the Live Forensic method, this study summarizes some information in the form of a table about the investigation report of ARP Poisoning attacks. The data of this report was obtained based on the scenarios carried out during this study, which can be seen in the following table

Table 2. of The Report

NO	INFORMATION ANALYSIS	DESCRIPTION
1	*Wireshark* tool successfully captures suspicious activities and identifies the protocol used by the attacker, which is the ARP protocol (*Address Resolution Protocol*).	Obtaining information on ARP *Poisoning / Spoofing* attacks to be used as network forensic evidence.
2	Capturing ARP *Poisoning* attack information by providing an alert and identifying the attacker's identity.	Obtaining information on ARP *Poisoning* attacks to be used as network forensic evidence.
3	Attacker's IP Address	192.168.1.21
4	Attacker's MAC Address	08:00:27:15:4c:3c
5	Successfully Attacked Protocol	UDP protocol (*User Data Protocol*)
6	Unsuccessful Attacked Protocol	SSL protocol (*Secure Socket Layer*)
7	Time of Attack	The attack occurred on January 9, 2024, starting from 02:14 AM to 06:08 AM WIB.

This report will then be submitted to the network admin for prevention and blocking so that ARP poisoning attacks do not occur on the LAN (Local Area Network) network that has been built.

Discussion

The results of network forensics analysis using Wireshark successfully identified an ARP Poisoning attack. This attack works by manipulating the Address Resolution Protocol (ARP) table to redirect network traffic through the attacker’s device, allowing potential data interception or unauthorized access.

Based on the forensic findings:

1. The attack was successfully captured and identified, proving that Wireshark is an effective tool for detecting suspicious network activities.
2. The attacker's IP address was recorded as **192.168.1.21**, with a MAC address of **08:00:27:15:4c:3c**, allowing for precise identification of the perpetrator.
3. The UDP protocol (User Data Protocol) was successfully attacked, indicating that unencrypted or weakly protected protocols are vulnerable to ARP Poisoning attacks.
4. The SSL (Secure Socket Layer) protocol remained secure, reinforcing the importance of encrypted communications in preventing unauthorized interception.
5. The attack lasted approximately **four hours**, from **02:14 AM to 06:08 AM WIB** on **January 9, 2024**, emphasizing the need for real-time monitoring to mitigate prolonged attacks.

These findings highlight the necessity of implementing security measures such as dynamic ARP inspection (DAI), encrypted communications, and real-time network monitoring to prevent similar attacks. Regular network audits and security awareness are also crucial in mitigating potential threats.

CONCLUSION

Based on the network forensic analysis using Wireshark, it was concluded that an **ARP Poisoning attack** had occurred, successfully compromising the UDP protocol while failing to breach the SSL protocol. The attacker's IP address (**192.168.1.21**) and MAC address (**08:00:27:15:4c:3c**) were identified, demonstrating the effectiveness of network monitoring tools in detecting malicious activities.

This study highlights the **importance of network security measures** to mitigate ARP Poisoning attacks, including:

a. Implementing **Dynamic ARP Inspection (DAI)** to prevent ARP spoofing.
b. Enforcing **strong encryption protocols** like SSL/TLS to secure network communication.
c. Conducting **continuous network monitoring** to detect and respond to suspicious activities in real time.

By applying these security measures, organizations can **reduce vulnerabilities** to ARP-based attacks and enhance overall network protection.

REFERENCES

Akbar, Faisal. (2017). *Arp Poisoning Faisal Akbar Nim : 23216099 Program Studi Magister Teknik Elektro Sekolah Teknik Elektro Dan Informatika Pendahuluan*.

Ardiyasa, I. Wayan. (2021). Analisa Serangan Remote Exploit Pada Jaringan Komputer Dengan Menggunakan Metode Network Forensic. *Explore*, *11*(2), 46. Https://Doi.Org/10.35200/Explore.V11i2.451

Dedy Hariyadi. (2022). *Buku Panduan Dasar Forensik Digital*. Retrieved From Https://Www.Researchgate.Net/Publication/365993681

Dhinda_Maydhita_I.Pdf. (N.D.). 2014.

Haeruddin, Haeruddin. (2021). Analisa Dan Implementasi Sistem Keamanan Router Mikrotik Dari Serangan Winbox Exploitation, Brute-Force, Dos. *Jurnal Media Informatika Budidarma*, *5*(3), 848. Https://Doi.Org/10.30865/Mib.V5i3.2979

Hanipah, Rahma. (2020). *Wireshark*. *4*(1), 11–23.

Landri Elusi Hutagalung, Program, Informatika, Teknik, Teknik, Fakultas, Komputer, D. A. N., & Batam, Universitas Putera. (2018). *Dengan Menggunakan Wireshark Network Protocol Analyzer Dengan Menggunakan Wireshark*.

Mallik, Avijit, Ahsan, Abid, Shahadat, Mhia Md Zaglul, & Tsou, Jia Chi. (2019). Man-In-The-Middle-Attack: Understanding In Simple Words. *International Journal Of Data And Network Science*, *3*(2), 77–92. Https://Doi.Org/10.5267/J.Ijdns.2019.1.001

Mukkamala, Srinivas. (2013). Network Attack. *Nber Working Papers*, 89. Retrieved From Http://Www.Nber.Org/Papers/W16019

Mukkamala, Srinivas, & Sung, A. H. (2003). Identifying Significant Features For Network Forensic Analysis Using Artificial Intelligent Techniques. *International Journal Of Digital Evidence*, *1*(4), 1–17.

Pradhana, I. (2020). *Bab Ii Tinjauan Pustaka Live Analysis Merupakan Cara Terbaik Untuk Menyelidiki Sistem Target*.

Rasyidah, Setyawan, Fajar, & Amnur, Hidra. (2022). Keamanan Jaringan Wireless Dengan Kali Linux. *Jitsi : Jurnal Ilmiah Teknologi Sistem Informasi*, *3*(1), 16–22. Https://Doi.Org/10.30630/Jitsi.3.1.57

Riadi, Imam, Fadlil, Abdul, & Hafizh, Muhammad Nasir. (2020). Analisis Bukti Serangan Address Resolution Protocol Spoofing Menggunakan Metode National Institute Of Standard Technology. *Edumatic : Jurnal Pendidikan Informatika*, *4*(1), 21–29. Https://Doi.Org/10.29408/Edumatic.V4i1.2046

Sunardi, Sunardi, Riadi, Imam, & Akbar, Muh Hajar. (2020). Steganalisis Bukti Digital Pada Media Penyimpanan Menggunakan Metode Static Forensics. *Jurnal Nasional Teknologi Dan Sistem Informasi*, *6*(1), 1–8. Https://Doi.Org/10.25077/Teknosi.V6i1.2020.1-8

Yahya, Arjun Zakari, Dirman, Buru, Dadang Juwoto, & Sugiantoro, Bambang. (2022). Analisis Bukti Digital Pada Random Access Memory Android Menggunakan Metode Live Forensic Kasus Penjualan Senjata Illegal. *Cyber Security Dan Forensik Digital*, *5*(1), 6–11. Https://Doi.Org/10.14421/Csecurity.2022.5.1.1724

Kaya, A., Ozturk, R., & Gumussoy, C. A. (2019). Usability Measurement Of Mobile Applications With System Usability Scale (Sus). Springer Nature Switzerland, 389–400. Https://Doi.Org/10.1007/978-3-030-03317-0_32

Lewis, J. R. (2018). The System Usability Scale: Past, Present, And Future. International Journal Of Human-Computer Interaction, 34(7), 577–590. Https://Doi.Org/10.1080/10447318.2018.1455307

Nafiah, Z., & Hartarini, Y. M. (2022). Efektivitas Aplikasi Tumbasin . Id Pada Masa Pandemi Covid-19 Di Kota Semarang. 2(1), 32–45.

Ningsih, S. R., Suryani, A. I., & Aulia, P. (2019). Aplikasi E-Task Berbasis Student Center Learning Pada Matakuliah Manajemen Proyek Sistem Informasi. Techno.Com, 18(1), 37–49. Https://Doi.Org/10.33633/Tc.V18i1.2064

Prabowo, M., & Suprapto, A. (2021). Usability Testing Pada Sistem Informasi Akademik Iain Salatiga Mengunakan Metode System Usability Scale. Jiska (Jurnal Informatika Sunan Kalijaga), 6(1), 38–49. Https://Doi.Org/10.14421/Jiska.2021.61-05

Pragantha, J., Setyaningsih, E., Orlando, S., & Liman, H. L. (2021). Pembuatan Website Sebagai Sarana Untuk Mempromosikan Organisasi Profesi. Seri Seminar Nasional Ke-Iii Universitas Tarumanagara Tahun 2021, 1819–1828.

Puspitaningtias, R., Widodo, J., & Zulianto, M. (2020). Analisis Loyalitas Merek Laptop Asus Dengan Net Promoters Score (Studi Kasus Pada Mahasiswa Fakultas Ilmu Komputer Program Studi Sistem Informasi Angkatan 2016 Universitas Jember). Jurnal Pendidikan Ekonomi: Jurnal Ilmiah Ilmu Pendidikan, Ilmu Ekonomi Dan Ilmu Sosial, 14(1), 208–212. Https://Doi.Org/10.19184/Jpe.V14i1.12039

Rahmansyah, R., Carudin, & Ridha, A. A. (2021). Perbandingan Hasil Investigasi Barang Bukti Digital Pada Aplikasi Facebook Dan Instagram Dengan Metode Nist. Cyber Security Dan Forensik Digital, 4(1), 49–57. Https://Doi.Org/10.14421/Csecurity.2021.4.1.2421

Setiawan, D., Wicaksono, S. L., & Rafianto, N. (2020). Evaluasi Usability E-Learning Moodle Dan Google Classroom Menggunakan Sus Quisionnare. Jami: Jurnal Ahli Muda Indonesia, 1(1), 55–64. Https://Doi.Org/10.46510/Jami.V1i1.13

Simarmata, B. T. (2019). Mengukur Tingkat Kepuasan Nasabah Dengan Net Promoter Score Pada Pt. Bpd Jawa Barat Dan Banten, Tbk. Cabang Medan. Jurnal Ilmiah Skylandsea, 3(2), 257– 264.

Suharsih, R., Febriani, R., & Triputra, S. (2021). Usability Of Jawara Sains Mobile Learning Application Using System Usability Scale (Sus). Jurnal Online Informatika, 6(1), 41–52. Https://Doi.Org/10.15575/Join.V6i1.700

Syamsuar, & Reflianto. (2019). Pendidikan Dan Tantangan Pembelajaran Berbasis Teknologi Informasi Di Era Revolusi Industri 4.0. E-Tech: Jurnal Ilmiah Teknologi Pendidikan, 6(2), 1–
13. Https://Doi.Org/10.24036/Et.V2i2.101343

Wardana, M. R. W., Angriani, H., & Muawwal, A. (2023). Analisis Pelanggan Smart Catering Menggunakan Metode Net Promoter Score. Kharisma Tech, 18(1), 151–164. Https://Doi.Org/10.55645/Kharismatech.V18i1.329

Widayanti, R., & Maknunah, J. (2021). Analisis Website Stimata Menggunakan System Usability Scale (Sus). Jurnal Ilmiah Komputasi, 20(3), 331–338.

Xiong, J. (2020). Susapp : A Free Mobile Application That Makes The System Usability Scale (Sus) Easier To Administer. Journal Of Usability Studies, 15(3), 135–144.