 ISSN:2302-8149
e-ISSN:2540-9719

Forensik Jaringan: Analisis Serangan *Client* dan Pengukuran *Quality of Service* oleh ARP *Poisoning* menggunakan *Network Forensic Generic Process* (NFGP) Model

Network Forensic: Analysis of Client Attack and Quality of Service Measurement by Arp Poisoning Using Network Forensic Generic Process (NFGP) Model

[1]Rizdqi Akbar Ramadhan*, [2]Agro Tambas Tira, [3]M. Rizki Fadhilah
[1,2,3]Program Studi Teknik Informatika, Fakultas Teknik, Universitas Islam Riau
Jalan Kaharudin Nasution, Pekanbaru 28284, Riau, Indonesia
*e-mail: ridzqiramadhan@eng.uir.ac.id

(***received*:** 30 December 2024, ***revised*:** 19 February 2024, ***accepted*:** 07 March 2024)

Abstrak

Pada jaringan komputer komunikasi dari satu komputer ke komputer lain sangat bisa dicegat, cara untuk mencegat komunikasi antara perangkat jaringan adalah dengan serangan *address resolution protocol poisoning*. Serangan ini dapat mencuri data seperti username dan password, memodifikasi *traffic*, dan menghentikan *traffic* itu sendiri. Penelitian ini mengimplementasikan model *Network Forensic Generic Proses* sebagai acuan dalam praktek *network forensik*, selain itu penelitian ini juga mengukur *quality of service* untuk membandingkan paramater saat sebelum serangan dan saat serangan terjadi. Tools yang digunakan pada penelitian ini adalah wireshark, XArp, dan snort. Penelitian ini berhasil memperoleh informasi yang autentik dari barang bukti yang diperoleh. Hasil pengukuran *quality of service* didapatkan bahwa parameter *quality of service* mengalami perubahan saat serangan terjadi. Hal ini diharapkan dapat menjadi acuan dalam meningkatkan keamanan jaringan dengan memahami lebih baik tentang ancaman yang mungkin akan dihadapi dan memberikan wawasan yang berharga untuk upaya pencegahan dan tanggapan keamanan di masa depan.

Kata kunci: *Address Resolution Protocol Poisoning, Network Forensic, Network Forensic Generic Process, Quality of Service*

Abstract

In computer network, communication from one computer to another computer can be intercepted, the way to intercept communication between network devices is with Address Resolution Protocol Poisoning attack. This attack can steal data such as usernames and passwords, modify traffic, and stop the traffic itself. This research implements the Network Forensic Generic Process model as a reference in Network Forensics practice. Apart from that, this research also measures quality of service to compare parameters before the attack and when the attack occurred. The tools used in this research are Wireshark, XArp, and Snort. This research succeeded in obtaining authentic information from the evidence obtained. The results of quality of service measurements showed that the quality of service parameters changed when the attack occurred. This research can be a reference in improving network security by better understanding the threats that may be encountered and providing valuable insight for future security prevention and response efforts.

Keywords: *Address Resolution Protocol Poisoning, Network Forensic, Network Forensic Generic Process, Quality of Service*

1 Pendahuluan

Cybercrime merupakan tindakan kejahatan pada dunia maya yang memanfaatkan perangkat komputer yang terhubung dalam suatu jaringan komputer. Beberapa kejahatan pada suatu jaringan

 ISSN:2302-8149
e-ISSN:2540-9719

komputer adalah Distributed *Denial of Service* (DDoS), *Sniffing*, *Spoofing*, dan *Man In The Middle Attack* [1]. Man in the Middle Attack adalah jenis serangan siber di mana pihak ketiga memasuki korespondensi online antara dua pengguna [2]. Dengan cara ini penyerang mampu mengendus data *frame* dan melakukan modifikasi *traffic* (ARP *poisoning*) [3]. Dalam serangan ARP *poisoning*/*spoofing*, paket ARP berbahaya dikirim ke *default gateway* pada jaringan LAN dengan maksud untuk mengubah IP *address* dan MAC *address* pada tabel ARP *cache* [4].

Network forensic didefinisikan sebagai penangkapan, pencatatan, dan analisis peristiwa pada suatu jaringan untuk menemukan sumber serangan atau insiden lainnya [5]. Pengambilan barang bukti digital dapat dilakukan dengan cara *dead forensic* dan *live forensic* [6]. Dead forensic atau static forensics merupakan teknik konvensional untuk melakukan penanganan barang bukti elektronik yang berfokus pada pemeriksaan salinan duplikasi atau image [7]. *Live forensic* adalah teknik analisis yang melibatkan data yang berjalan pada suatu sistem atau data yang rapuh yang umumnya disimpan dalam *Random Access Memory* (RAM) atau dalam transit pada jaringan [8].

Keberadaan ARP *poisoning* menjadi ancaman serius bagi jaringan komputer saat ini [9]. Serangan ini bisa mengganggu lalu lintas jaringan sehingga dapat memutuskan koneksi internet pada perangkat yang terhubung ke jaringan [3].

Oleh Karena itu, penelitian ini bertujuan untuk membahas bagaimana melakukan proses tahapan investigasi *live forensic* terhadap serangan ARP *poisoning* menggunakan metode NFGP (*Network Forensic Generic Process*) untuk menemukan informasi penting yang dapat dijadikan barang bukti seperti MAC *address*, IP *address*, dan waktu terjadinya serangan sekaligus melakukan analisis *Quality of Service* (QoS) saat sebelum serangan dan saat terjadi serangan.

2 Tinjauan Literatur

Beberapa penelitian sebelumnya yang menjadi acuan dalam pembentukan penelitian ini, seperti penelitian yang berjudul Forensik Jaringan Terhadap Serangan ARP *Spoofing* menggunakan Metode *Live Forensic*. Penelitian ini melakukan simulasi serangan menggunakan aplikasi Cain and Abel yang digunakan untuk melakukan serangan ARP *poisoning*. Hasil analisis forensik menggunakan aplikasi XARP yang menampilkan beberapa informasi terkait dengan proses identifikasi penyerang [1].

Selanjutnya adalah penelitian yang berjudul Analisis *Address Resolution Protocol Poisoning Attack* Pada Router WLAN Menggunakan Metode *Live Forensics*. Penelitian ini melakukan simulasi serangan dengan aplikasi Netcut yang digunakan untuk memutus koneksi pada perangkat korban, Hasil analisis live forensics menggunakan aplikasi wireshark didapatkan beberapa informasi mengenai identitas penyerang [3].

Selanjutnya adalah penelitian yang berjudul ARP *Poisoning Detection and Prevention using* Scapy. Penelitian ini melakukan simulasi *virtual machine*. Algoritma pendeteksian serangan ARP *poisoning* diimplementasikan menggunakan python dengan scapy *library* [4].

Selanjutnya adalah penelitian yang berjudul *Network Forensics Analysis of Man in the Middle Attack Using Live Forensics Method*, Penelitian ini melakukan simulasi serangan menggunakan aplikasi Ettercap. Impelementasi IDS (*Intrusion Detection System*) Snort pada *web server* dapat membantu dalam Mendeteksi serangan MITM. *File Log* diambil dan dianalisis untuk menemukan anomali yang terjadi pada *web server*. Hasil analisis didapatkan berupa IP *address* dan *port* yang digunakan penyerang untuk mengakses *web server* [10].

Selanjutnya adalah penelitian yang berjudul Analisis Bukti Serangan *Address Resolution Protocol Spoofing* menggunakan Metode *National Institute of Standard Technology*. Penelitian melakukan simulasi serangan sebanyak dua serangan terhadap satu perangkat laptop dan satu perangkat routerboard, investigasi forensik dilakukan menggunakan kerangka *National Institute of Standards and Technology* (NIST) [11].

Selanjutnya adalah penelitian yang berjudul Analisis Pendeteksian Serangan ARP *Poisoning* Dengan Menggunakan Metode *Live Forensic*. Penelitian melakukan simulasi serangan dilakukan saat *client* mengakses *server* menggunakan protokol SSL dan FTP, penelitian ini juga melakukan investigasi forensik untuk mengindentifikasi serangan ARP *poisoning*. Hasil diketahui berupa identitas penyerang dan *port* yang diserang [12].

Selanjutnya adalah penelitian yang berjudul Network Forensics for Detecting Flooding Attack on Internet of Things (IoT) Device, penelitian ini menyajikan model Network Forensic Generic Proses

 ISSN:2302-8149
e-ISSN:2540-9719

(NFGP) untuk mendeteksi dan mengidentifikasi serangan. Hasil pada penelitian ini ditemukannya infeksi terhadap perangkat Bluetooth Arduino dan ditemukannya beberapa IP address yang melakukan tindakan ilegal [13]

Kemudian penelitian yang berjudul ARP *Cache Poisoning* sebagai Teknik Alternatif untuk Membatasi Penggunaan *Bandwidth* berbasis Waktu. Penelitian ini melakukan serangan ARP *poisoning* untuk membatasi penggunaan *bandwidth* dengan menggunakan aplikasi TuxCut. Hasil dari penelitian ini didapatkan bahwa aplikasi TuxCut dapat digunakan untuk membatasi penggunaan *bandwidth* tanpa harus memutus koneksi secara total [14].

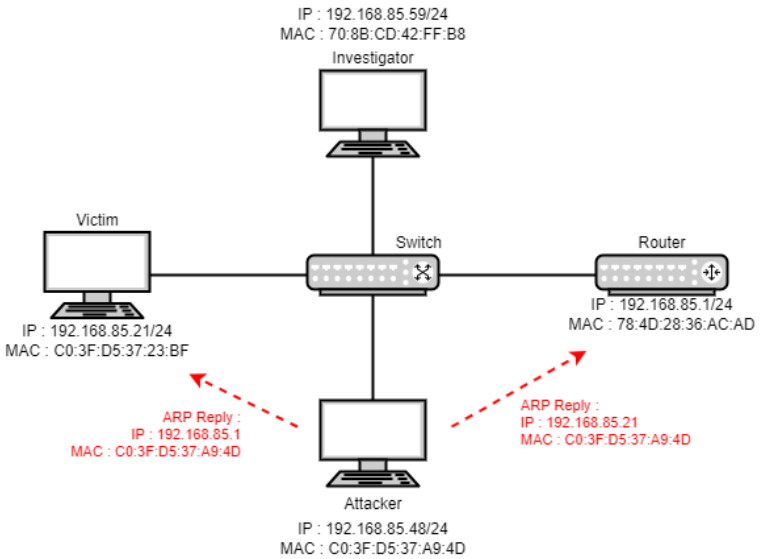
Berdasarkan beberapa penelitian diatas, penelitian ini dilakukan dengan menggunakan kerangka *Network Forensic Generic Proses* (NFGP) model. Selain itu penelitian ini melakukan analisis terhadap parameter *Quality of Service* saat sebelum serangan dan saat serangan terjadi untuk mengetahui apakah terjadi perubahan saat serangan ARP *poisoning* terjadi.

3 Metode Penelitian

Metode Penelitian yang diterapkan pada penelitian ini adalah dengan melakukan impelentasi dan simulasi serangan terhadap sebuah perangkat. Penelitian ini menggunakan aplikasi NetCut untuk memodifikasi trafik sehingga membatasi koneksi sebuah perangkat dan akan disimulasikan.

3.1 Skenario Penelitian

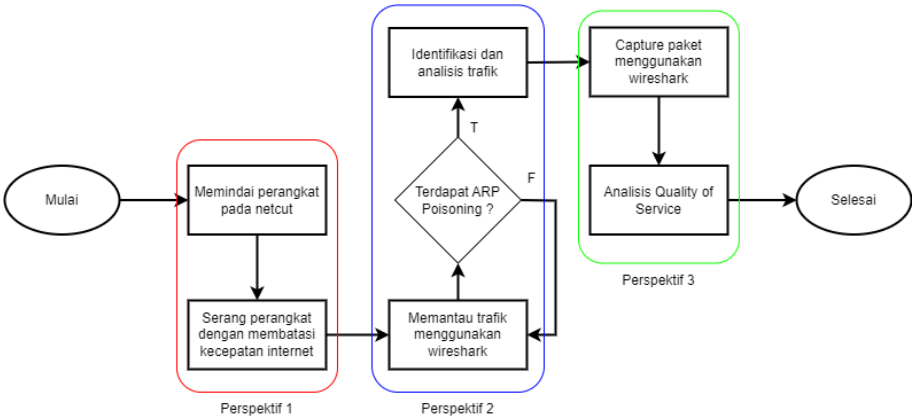
Rancangan serangan yang dilakukan pada penelitian ini adalah dua PC *client* yang terrhubung dalam suatu LAN. Serangan yang dilakukan adalah ARP *Poisoning* yang dapat memodifikasi *traffic* jaringan sehingga dapat membatasi kecepatan internet. Rancangan simulasi serangan dapat dilihat pada Gambar 1.



Gambar 1. Rancangan Simulasi Serangan

Skenario simulasi digambarkan dalam 3 perspektif. Perspektif petama adalah perspektif *attacker*, Perspektif *attacker* melakukan serangan terhadap perangkat *victim*. Perspektif kedua adalah perspektif investigator, perspektif investigator melakukan investigasi untuk mencari dan menganalisis bukti digital, perspektif ketiga melakukan analisis parameter *quality of service* sebelum serangan dan saat terjadi serangan. Skenario simulasi penelitian dapat dilihat pada Gambar 2.

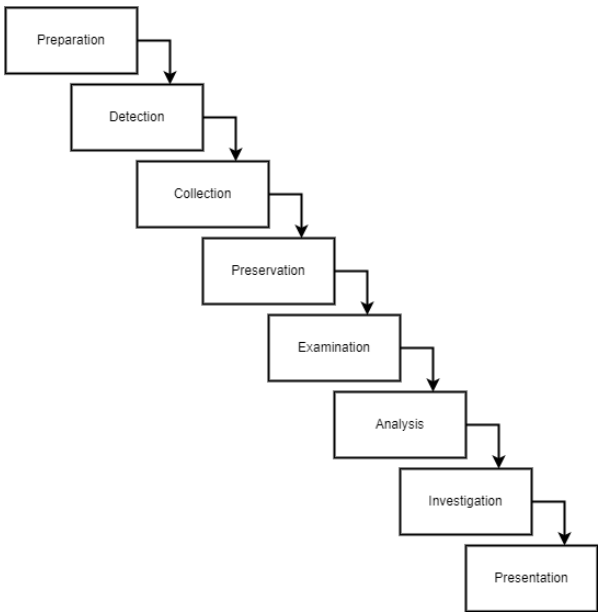
 ISSN:2302-8149
e-ISSN:2540-9719



Gambar 2. Skenario Simulasi Penelitian

3.2 Tahapan Penelitian

Penelitian ini mengadaptasi metode *Network Forensic Generic Process* Model (NFGP). NFGP sendiri merupakan model yang digunakan untuk melakukan tahapan pendeteksian, akuisisi data, dan analisis terhadap apa yang terjadi dalam suatu jaringan computer [15]. Tahapan penelitian dapat dilihat pada Gambar 3.



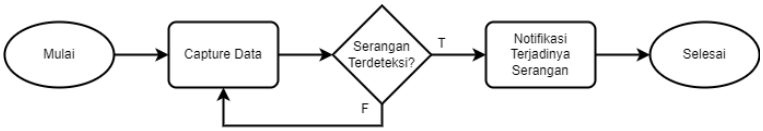
Gambar 3. *Network Forensic Generic Proses (NFGP) Model*

1. *Preparation*

Tahap ini merupakan tahap merancang sebuah sistem monitoring sebuah jaringan untuk mendeteksi serangan ARP *Poisoning*.

2. *Detection*

Tahap ini merupakan tahap mendeteksi serangan ARP *poisoning* menggukanan *tool* monitoring dan deteksi yang sudah dipasang. Tahap *detection* dapat dilihat pada Gambar 4 dibawah.

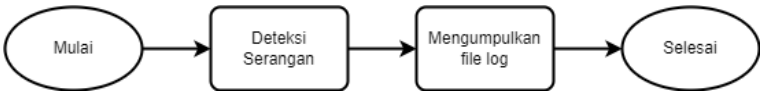


Gambar 4. Tahap Deteksi Serangan

 ISSN:2302-8149
e-ISSN:2540-9719

3. *Collection*

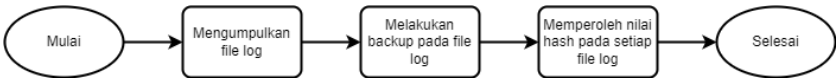
Tahap ini merupakan tahap mengumpulkan informasi berupa *file log* dimana *tool* yang dipasang berhasil mendeteksi aktivitas mencurigakan yang terjadi pada sebuah jaringan. Tahap *collection* dapat dilihat pada Gambar 5 di bawah.



Gambar 5. Tahap Mengumpulkan *File Log*

4. Preservation

Tahap preservasi merupakan tahap yang dilakukan saat data sudah terkumpul, data tersebut dijaga dan dipelihara sekaligus menghitung nilai hash pada setiap *file log* yang dikumpul. Tahap *preservation* dapat dilihat pada Gambar 6 di bawah.



Gambar 6. Tahap Preservasi

5. *Examination*

Tahap ini merupakan tahap pemeriksaan data apakah data yang sudah dikumpulkan merupakan data yang berhubungan dengan kejadian yang terjadi agar bisa dilakukan analisis lebih lanjut.

6. *Analysis*

Tahap ini merupakan tahap dimana data yang sudah diperoleh dan diperiksa akan dianalisis sehingga didapatkan informasi yang berguna untuk investigasi.

7. *Investigation*

Tahap ini merupakan tahap mengidentifikasi hasil dari analisis data seperti kapan serangan terjadi, siapa penyerangnya, bagaimana serangan itu terjadi, dan lain-lain

8. *Presentation*

Tahap presentasi merupakan tahap untuk menampilkan hasil investigasi dengan bahasa yang mudah dipahami.

3.3 Alat dan kebutuhan

Persiapan alat dan kebutuhan sebelum melakukan simulasi serangan dapat dilihat pada Tabel 1.

Tabel 1. Alat dan Bahan

No	Alat dan Bahan	Spesifikasi
1.	Laptop (*Investigator*)	• ASUS X456UR • Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz 2.40 GHz • 8,00 GB (7,87 GB usable) RAM • 1 TB HDD 64-bit operating system, x64-based processor • Serial Number : G8N0CX20F351345 • Hash : 251F800597C6E8801F954256F5102294
2.	PC 1 (*Attacker*)	• Lenovo • Intel(R) Core(TM) i5-3470 CPU @ 3.20GHz 3.20 GHz • 4,00 GB (3,83 GB usable) RAM 64-bit operating system, x64-based processor • Serial Number : PBTH67Z • Hash : 8E785788EA0560645418C73A2F94F47B
3.	PC 2 (*Victim*)	• Lenovo • Intel(R) Core(TM) i5-3470 CPU @ 3.20GHz 3.20 GHz • 4,00 GB (3,83 GB usable) RAM 64-bit operating system, x64-based processor

 ISSN:2302-8149
e-ISSN:2540-9719

		• Serial : PBTH44V • Hash : 32EB71D138CC8EC1D3E8161FE153684A
4.	Routerboard	Mikrotik Routerboard RB3011UiAS-RM
5.	Switch	Switch 3Com 3C16792C Office Connect Fast Ethernet Switch 16 Port
6	Kabel UTP	Straigth Cat 5e
7	Windows (Laptop, PC 1, PC 2)	Windows 10 Home
8	Netcut	Versi 3.0.186
9	Wireshark	Versi 3.6.2
10	XArp	Versi 2.2.2
11	Snort	Versi 2.9.20

4 Hasil dan Pembahasan

Terdapat 2 hal yang dilakukan dalam penelitian ini yaitu adalah melakukan investigasi terhadap perangkat yang diserang dan melakukan perhitungan parameter *quality of service* untuk mengetahui apakah ada perubahan saat serangan terjadi. Penelitian mencakup dua kegiatan penting: pertama, melakukan investigasi terhadap perangkat yang diserang, dan kedua, melakukan perhitungan parameter kualitas layanan (quality of service) untuk menilai apakah terjadi perubahan saat serangan terjadi. Dengan pendekatan ini, penelitian dapat menggali lebih dalam tentang dampak serangan terhadap perangkat dan layanan yang terkait, serta menyediakan pemahaman yang lebih baik tentang kerentanan sistem terhadap serangan.

4.1 Network Forensic

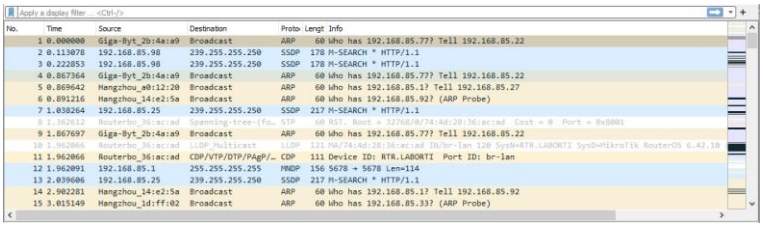
Metode *network forensic* yang diimplementasikan pada penelitian ini adalah *network forensic generic process* model yang terdiri dari 8 tahapan. Model ini dirancang untuk memberikan panduan sistematis dalam melakukan analisis forensik pada jaringan komputer. Dengan mengikuti model ini, penelitian dapat memastikan bahwa proses investigasi dilakukan secara menyeluruh dan terstruktur.

4.1.1 Preparation

Tahap *preparation* merupakan tahapan pertama dalam penelitian ini. Pada tahap ini mempersiapkan *tools* yang digunakan untuk mendeteksi serangan. Pada tahap preparation, persiapan alat dan perangkat lunak yang digunakan untuk mendeteksi dan menganalisis serangan menjadi fokus utama. Ini termasuk memastikan bahwa semua alat yang diperlukan telah diunduh, diinstal, dan dikonfigurasi dengan benar. Dengan mempersiapkan alat-alat yang tepat dan menyusun prosedur yang baik, penelitian dapat memastikan bahwa analisis forensik dilakukan dengan efisien dan efektif.

4.1.1.1 Implementasi Wireshark

Tool Wireshark digunakan untuk memonitoring *traffic* jaringan. Wireshark menampilkan beberapa informasi seperti sumber paket dan kemana paket tersebut akan dikirim yang sangat berguna untuk mendeteksi searngan ARP *poisoning*. Gambar 7 menunjukkan bahwa *tool* wireshark sudah dijalankan dan dapat memonitoring *traffic* jaringan

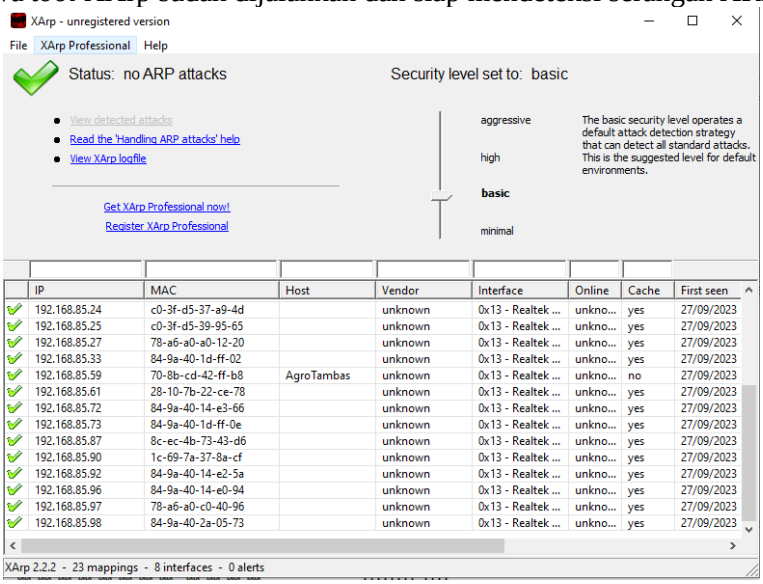


Gambar 7. Tampilan Wireshark

 ISSN:2302-8149
e-ISSN:2540-9719

4.1.1.2 Implementasi XArp

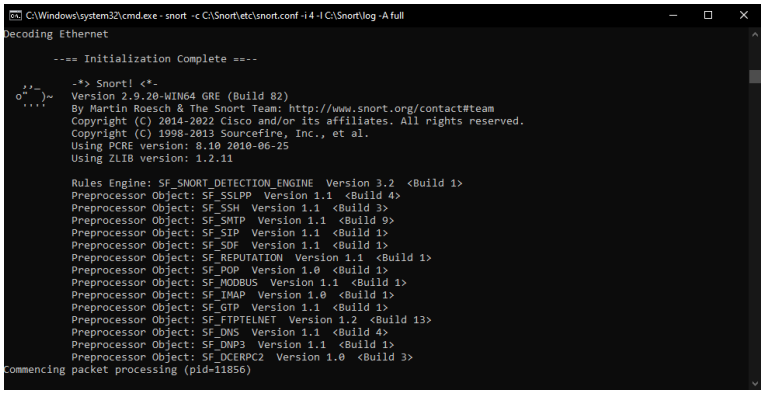
XArp adalah *tool* yang dibuat khusus untuk mendeteksi serangan ARP *poisoning*. XArp juga menampilkan daftar perangkat yang terkoneksi dalam suatu jaringan LAN. Gambar 8 di bawah menunjukkan bahwa *tool* XArp sudah dijalankan dan siap mendeteksi serangan ARP *Poisoning*.



Gambar 8. Tampilan XArp

4.1.1.3 Implementasi Snort

Snort merupakan salah satu sistem yang bisa mendeteksi anomali pada suatu jaringan. Tidak hanya serangan ARP *poisoning* tetapi snort bisa mendeteksi serangan pada jaringan seperti DDoS, SQL *Injection*, *Flooding Attack*, *Port Scanning*, dan lain sebagainya. Gambar 9 di bawah menunjukkan bahwa Snort sudah dijalankan.



Gambar 9. Tampilan Snort

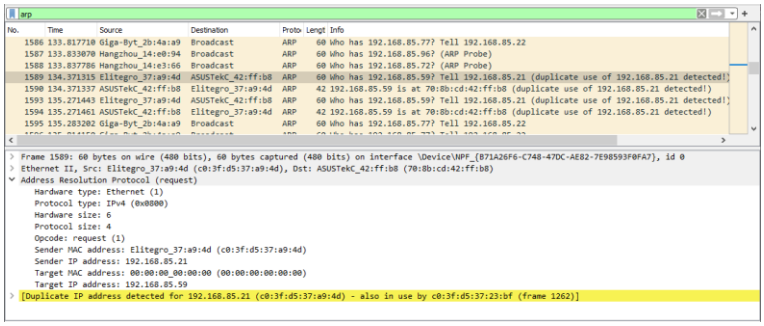
4.1.2 Detection

Tahap detection merupakan tahap yang dilakukan saat *tool* wirshark, XArp dan Snort yang sudah disiapkan mendeteksi serangan. Wireshark digunakan untuk menganalisis lalu lintas jaringan dan menangkap paket data, sementara XArp dapat digunakan untuk mendeteksi serangan berbasis ARP (Address Resolution Protocol) yang mungkin terjadi di jaringan. Selain itu, Snort adalah sebuah sistem deteksi intrusi yang dapat digunakan untuk mendeteksi serangan jaringan yang berbeda, seperti serangan DoS (Denial of Service), serangan buffer overflow, dan lainnya.

4.1.2.1 Deteksi Menggunakan Wireshark

Wireshark mendeteksi adanya duplikasi IP *address client* yang berarti serangan ARP *Poisoning* sedang berlangsung Deteksi pada wireshark dapat dilihat pada gambar 10.

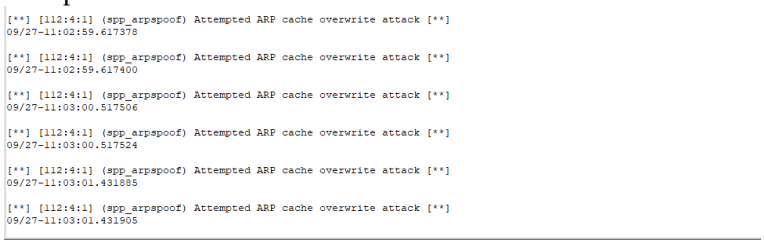
 ISSN:2302-8149
e-ISSN:2540-9719



Gambar 10. Wireshark Mendeteksi Serangan ARP *Poisoning*

4.1.2.2 Deteksi Menggunakan Snort

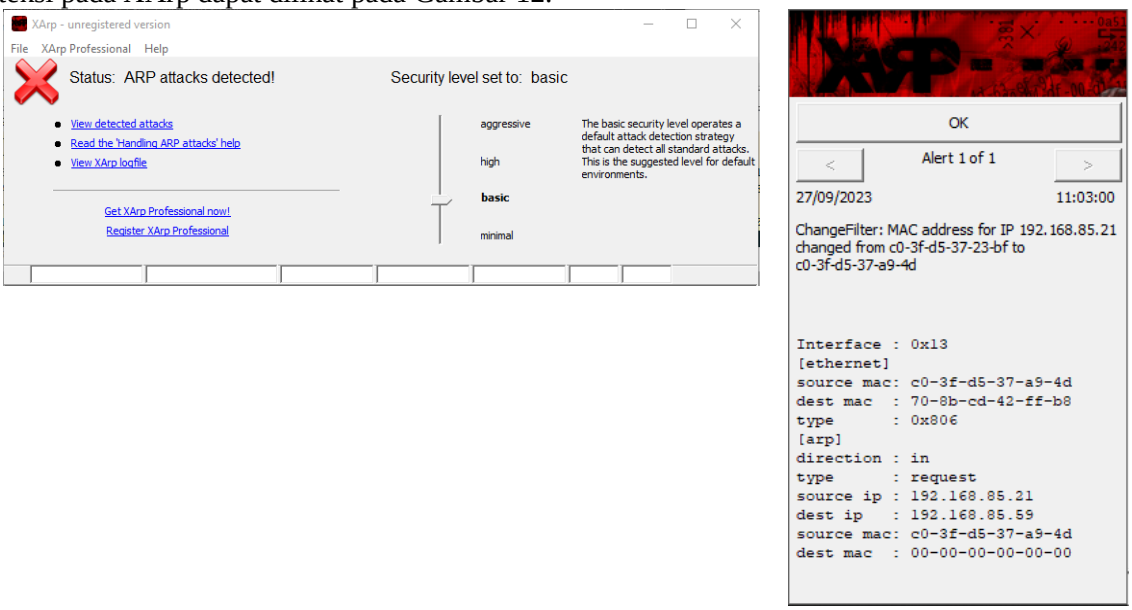
Snort mendeteksi serangan ARP *Poisoning* dengan menampilkan *alert* terus-menerus. Deteksi pada snort dapat dilihat pada Gambar 11.



Gambar 11. Snort Mendeteksi Serangan ARP *Poisoning*

4.1.2.3 Deteksi Menggunakan XArp

XArp mendeteksi serangan ARP *poisoning* dengan memunculkan *alert* yang memberi tahu bawha MAC *address client* telah diganti dan status XARP berubah menjadi ARP *attacks detected*!. Deteksi pada XArp dapat dilihat pada Gambar 12.



Gambar 12. XArp Mendeteksi Serangan ARP *Poisoning*

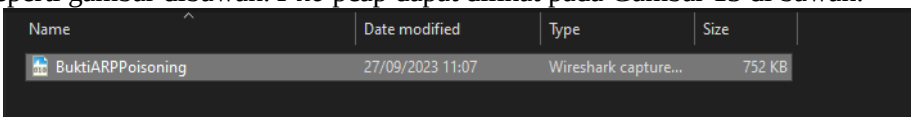
4.1.3 Collection

Tahap *collection* melakukan pengumpulan *file* berupa *log* dan hasil monitoring trafik yang akan dianalisis lebih lanjut. Dengan mengumpulkan data ini secara teliti, tim penelitian akan memiliki dasar yang kuat untuk melakukan analisis mendalam pada tahap selanjutnya, dan memungkinkan mereka untuk memperoleh pemahaman yang lebih baik tentang sifat dan dampak dari serangan atau insiden keamanan yang terjadi.

 ISSN:2302-8149
e-ISSN:2540-9719

4.1.3.1 *File Log* Wireshark

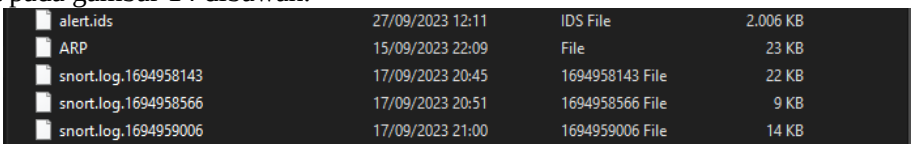
Menyimpan *file* pcap hasil dari monitoring jaringan yang mendeteksi serangan ARP *Poisoning* seperti gambar dibawah. *File* pcap dapat dilihat pada Gambar 13 di bawah.



Gambar 13. *File Log* Wireshark

4.1.3.2 *File Log* Snort

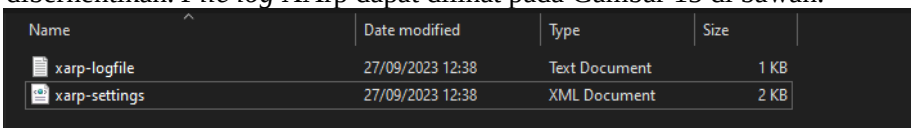
Snort mendeteksi serangan berupa pesan *alert* yang tercatat pada *file* alert.ids. *file* alert.ids dapat dilihat pada gambar 14 dibawah.



Gambar 14. *File Log* Snort

4.1.3.3 *File Log* XArp

File log XArp mencatat perubahan MAC *address* pada sebuah IP *address* walaupun aplikasi XArp sudah diberhentikan. *File log* XArp dapat dilihat pada Gambar 15 di bawah.



Gambar 15. *File Log* Xarp

4.1.4 Preservation

Tahap ini merupakan tahap memelihara *file* yang sudah di dapatkan dengan membuat *back-up*, dan menghitung nilai *hash* pada *file-file* tersebut dengan tipe md5 seperti pada Tabel 1 di bawah.

Tabel 1. Nilai Hash

No	File	Hash
1.	BuktiARPPoisoning.pcapng	95696C31250039113824BE9F3C0B19D1
2.	alert.ids	73FA347C4244CEBF4CA7DD5083D39A6B
3	xarp-logfile	C91DA8AE3FF8C7B198924EF66E6F1201

4.1.5 Examination

Pada tahap ini *file log* yang telah diambil dan akan diperiksa apakah *log* aktivitas yang tercatat pada file tersebut adalah aksi serangan ARP *Poisoning* yang sesuai dengan insiden yang terjadi.

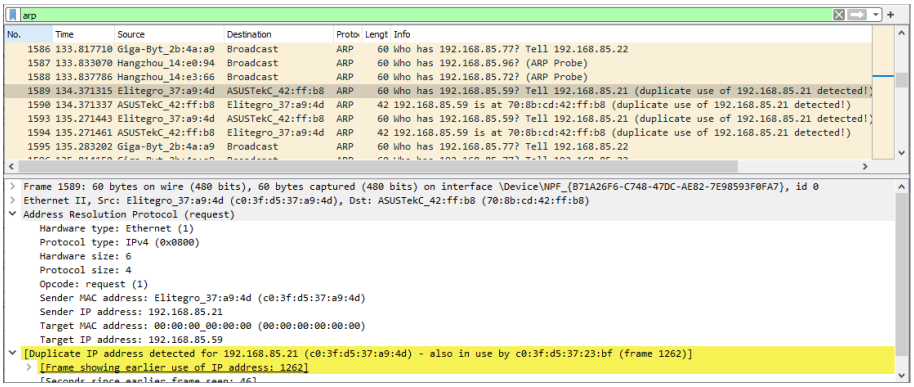
4.1.6 Analysis dan *Investigation*

Analysis dan *investigation* digabung menjadi satu tahap dan diimplementasikan ke dalam 3 perspektif yaitu perspektif menggunakan *tool* wireshark, perspektif menggunakan *tool* XArp, dan perspektif menggunakan *tool* Snort. Dengan mengadopsi tiga perspektif ini, penelitian akan mendapatkan sudut pandang yang beragam dalam menganalisis serangan atau aktivitas mencurigakan dalam jaringan. Ini akan membantu dalam mengidentifikasi, memahami, dan merespons serangan secara lebih efektif.

4.1.6.1 Analisis dan Investigasi Menggunakan Wireshark

Duplikasi IP *address client* yang terdeteksi pada wireshark menandakan bahwa serangan ARP *Poisoning* sedang terjadi. Duplikasi IP address dapat dilihat pada Gambar 16 di bawah.

 ISSN:2302-8149
e-ISSN:2540-9719

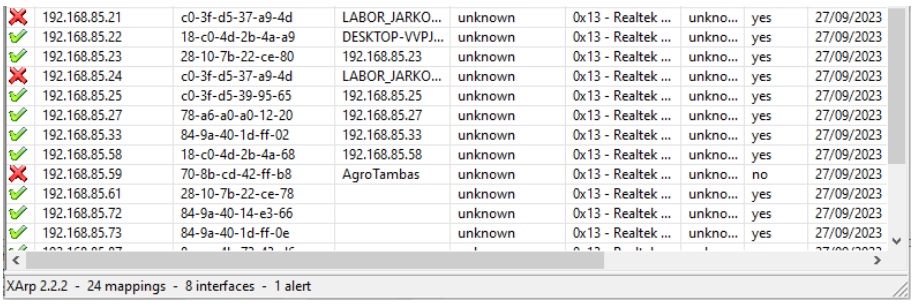


Gambar 16. Analisis dan Investigasi pada Wireshark

Pada Gambar 16 di atas, dapat diketahui bahwa IP *address client* yaitu 192.168.85.21 memiliki MAC *address* C0:3F:D5:37:A9:4D dan C0:3F:D5:37:23:BF, yang berarti salah satu MAC *address* tersebut adalah palsu. Karena duplikasi IP sedang berlangsung dapat diketahui MAC *address* yang digunakan sekarang adalah milik *attacker* yaitu C0:3F:D5:37:A9:4D.

4.1.6.2 Analisis dan Investigasi Menggunakan XARP

Notifikasi XARP menampilkan bahwa MAC *address client* telah diganti dari C0:3F:D5:37:23:BF menjadi C0:3F:D5:37:A9:4D. Diketahui bahwa MAC *address attacker* adalah C0:3F:D5:37:A9:4D, akan tetapi ada dua *host* yang memakai MAC *address* yang sama. Dari kedua MAC *address* yang sama tersebut, salah satu *host* tidak mengalami perubahan MAC *address* yang berarti *host* tersebut adalah *attacker*. Sebelumnya dapat diketahui bahwa IP *client* yang diserang (*victim*) adalah 192.168.85.21 berarti IP yang memiliki MAC *address* yang sama adalah IP *address* milik *attacker* yaitu 192.168.85.24. Perubahan MAC *address* dapat dilihat pada Gambar 17 di bawah.



Gambar 17. Analisis dan Investigasi pada XArp

File log XARP juga mencatat perubahan MAC *address router* yang dapat dilihat pada gambar 18 dibawah.

```
27/09/2023 - 11:01:29 - info - xarp version XArp 2.2.2 -

27/09/2023 - 11:01:29 - info - winpcap version info: Npcap version 1.76, based on libpcap
version 1.10.4 -

27/09/2023 - 11:03:00 - arp - ChangeFilter: MAC address for IP 192.168.85.21 changed from c0-
3f-d5-37-23-bf to c0-3f-d5-37-a9-4d - 0x13 - c0-3f-d5-37-a9-4d - 70-8b-cd-42-ff-b8 - request -
c0-3f-d5-37-a9-4d - 00-00-00-00-00-00 - 192.168.85.21 - 192.168.85.59 - in
```

Gambar 18. Analisis dan Investigasi pada XArp (2)

4.1.6.3 Analisis dan Investigasi Menggunakan Snort

Snort mendeteksi serangan berupa *alert* yang menandakan bahwa serangan ARP *Poisoning* sedang berlangsung, tetapi Snort hanya menampilkan *timestamp* terjadinya serangan tanpa memberikan informasi yang lengkap dari penyerang. Diketahui serangan terjadi pada tanggal 27 September 2023 pada pukul 11:02:59 sampai dengan pukul 11:06:46 yang dapat dilihat pada Gambar 19.

 ISSN:2302-8149
e-ISSN:2540-9719

```
[**] [112:4:1] (spp_arpspoof) Attempted ARP cache overwrite attack [**]
09/27-11:02:59.617378

[**] [112:4:1] (spp_arpspoof) Attempted ARP cache overwrite attack [**]
09/27-11:02:59.617400

[**] [112:4:1] (spp_arpspoof) Attempted ARP cache overwrite attack [**]
09/27-11:03:00.517506

[**] [112:4:1] (spp_arpspoof) Attempted ARP cache overwrite attack [**]
09/27-11:06:45.498962

[**] [112:4:1] (spp_arpspoof) Attempted ARP cache overwrite attack [**]
09/27-11:06:46.402221

[**] [112:4:1] (spp_arpspoof) Attempted ARP cache overwrite attack [**]
09/27-11:06:46.402242
```

Gambar 19. Analisis dan Investigasi pada Snort

4.1.7 *Presentation*

Tahap presentasi merupakan tahap meringkas informasi yang sudah di analisis ke dalam bentuk tabel yang dapat dilihat pada Tabel 2 di bawah.

Tabel 2. *Presentation*

No	Informasi	Keterangan
1	MAC *address victim*	C0:3F:D5:37:23:BF
2	IP *address victim*	192.168.85.21
3	MAC *address attacker*	C0:3F:D5:37:A9:4D
4	IP *address attacker*	192.168.85.24
5	Waktu terjadinya serangan	Tanggal 27 September pukul 11:02:59 – 11:06:46

4.1.8 Hasil

Perbandingan antara *tools* yang digunakan untuk menginvestigasi serangan ARP *Poisoning* dapat dilihat pada Tabel 3 di bawah.

Tabel 3. Perbandingan *Tools*

Tool	IP address attacker	IP address victim	MAC address attacker	MAC address victim	Timestamp
Wireshark		✓	✓	✓	✓
XArp	✓	✓	✓	✓	✓
Snort					✓

4.2 Analisis *Quality of Service*

Pengukuran parameter QoS dilakukan pada PC 2 saat sebelum serangan dan saat serangan terjadi. Pengukuran parameter QoS dilakukan menggunakan *tool* wireshark dengan cara membuka *website* dan men-*capture* paket.

4.2.1 Pengukuran Sebelum Serangan

Pengukuran sebelum serangan dilakukan sebanyak 3 kali yang dijabarkan pada Tabel 4 di bawah.

Tabel 4. Pengukuran QoS Sebelum Serangan

No	Parameter	Pengukuran
1	*Throughput*	*Throughput* = Jumlah *Bytes* / *Timespan* *Throughput* = 1682412 / 19,369 *Throughput* = 86861 x 8 *Throughput* = 694k bits/s
	Packet Loss	*Packet Loss* = [((Paket Dikirim - Paket Diterima) / Paket Dikirim) X 100] *Packet Loss* = ((2562 - 2526) / 2562) x 100 *Packet Loss* = 0,049 x 100 *Packet Loss* = 4,9 %

 ISSN:2302-8149
e-ISSN:2540-9719

	Delay	*Delay* = (*Timespan* / Jumlah Paket) x 1000 *Delay* = (14,628 / 1469) x 1000 *Delay* = 9,9 ms
	Jitter	*Jitter* = [((*Delay 2 – Delay 1*) / Jumlah Paket) x 1000] *Jitter* = (14,638 / 5123) x 1000 *Jitter* = 9,9 ms
2	*Throughput*	*Throughput* = Jumlah *Bytes* / *Timespan* *Throughput* = 2525360 / 11,085 *Throughput* = 227817 x 8 *Throughput* = 1822k bits/s
	Packet Loss	*Packet Loss* = [((Paket Dikirim - Paket Diterima) / Paket Dikirim) X 100] *Packet Loss* = ((3296 - 2829) / 3296) x 100 *Packet Loss* = 0,14 x 100 *Packet Loss* = 14 %
	Delay	*Delay* = (*Timespan* : Jumlah Paket) x 1000 *Delay* = (10,751 : 2731) x 1000 *Delay* = 3,9 ms
	Jitter	*Jitter* = [((*Delay 2 – Delay 1*) / Jumlah Paket) x 1000] *Jitter* = (11,273 / 2731) x 1000 *Jitter* = 4,1 ms
3	*Throughput*	*Throughput* = Jumlah *Bytes* / *Timespan* *Throughput* = 1768681 / 18,534 *Throughput* = 95428 x 8 *Throughput* = 763k bits/s
	Packet Loss	*Packet Loss* = [((Paket Dikirim - Paket Diterima) / Paket Dikirim) X 100] *Packet Loss* = ((2515 - 2356) / 2515) x 100 *Packet Loss* = 0,063 x 100 *Packet Loss* = 6,3 %
	Delay	*Delay* = (*Timespan* : Jumlah Paket) x 1000 *Delay* = (14,696 / 1669) x 1000 *Delay* = 8,8 ms
	Jitter	*Jitter* = [((*Delay 2 – Delay 1*) / Jumlah Paket) x 1000] *Jitter* = (14,682 / 1669) x 1000 *Jitter* = 8,7 ms

4.2.2 Pengukuran Saat Serangan Terjadi

Pengukuran saat serangan terjadi juga dilakukan sebanyak 3 kali seperti yang dijabarkan pada Tabel 5 di bawah.

Tabel 5. Pengukuran QoS Saat Serangan Terjadi

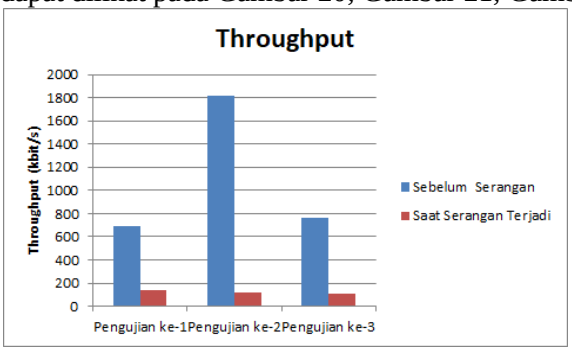
No	Parameter	Pengukuran
1	*Throughput*	*Throughput* = Jumlah *Bytes* / *Timespan* *Throughput* = 1357980 / 76,459 *Throughput* = 17760 x 8 *Throughput* = 142k bits/s
	Packet Loss	*Packet Loss* = [((Paket Dikirim - Paket Diterima) / Paket Dikirim) X 100] *Packet Loss* = ((5041 - 4948) / 5041) x 100 Packet Loss = 0,018 x 100 *Packet Loss* = 1,8 %
	Delay	*Delay* = (*Timespan* : Jumlah Paket) x 1000 *Delay* = (76,458 / 1987) x 1000

ISSN:2302-8149
e-ISSN:2540-9719

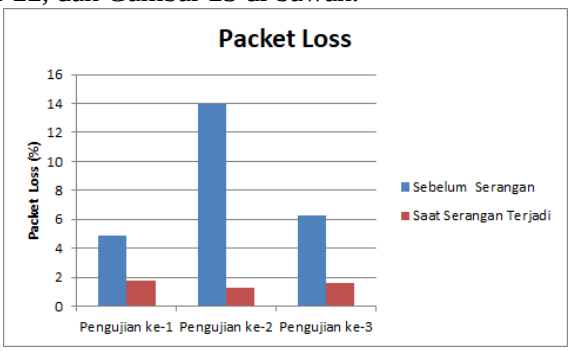
		Delay = 38,4 ms
	Jitter	*Jitter* = [((*Delay 1 – Delay 2*) / Jumlah Paket) x 1000] *Jitter* = [(76,266 / 1987) x 1000] *Jitter* = 38,3 ms
2	*Throughput*	*Throughput* = Jumlah *Bytes* / *Timespan* *Throughput* = 1640849 / 108,800 *Throughput* = 15081 *Throughput* = 120k bits/s
	Packet Loss	*Packet Loss* = [((Paket Dikirim - Paket Diterima) / Paket Dikirim) X 100] *Packet Loss* = ((6079 - 5995) / 6079) x 100 *Packet Loss* = 0,013 x 100 *Packet Loss* = 1,3 %
	Delay	*Delay* = *(Timespan* / Jumlah Paket) x 1000 *Delay* = 106,836 / 1631 *Delay* = 65,5 ms
	Jitter	*Jitter* = [((*Delay 1 – Delay 2*) / Jumlah Paket) x 1000] *Jitter* = (107,412 / 1631) x 1000 *Jitter* = 65,8 ms
3	*Throughput*	*Throughput* = Jumlah *Bytes* / *Timespan* *Throughput* = 1541904 / 110,876 *Throughput* = 13906 *Throughput* = 111k bits/s
	Packet Loss	*Packet Loss* = [((Paket Dikirim - Paket Diterima) / Paket Dikirim) X 100] *Packet Loss* = ((5968 - 5871) / 5968) x 100 *Packet Loss* = 0,016 x 100 *Packet Loss* = 1,6 %
	Delay	*Delay* = *Timespan* / Jumlah Paket *Delay* = 110,600 / 1492 *Delay* = 74,1 ms
	Jitter	*Jitter* = [((*Delay 1 – Delay 2*) / Jumlah Paket) x 1000] *Jitter* = (110,585 / 1492) x 1000 *Jitter* = 74,1 ms

4.2.3 Perbandingan Parameter *Quality of Service*

Grafik Perbandingan parameter *quality of service* dilakukan setelah mengukur parameter dapat dilihat pada Gambar 20, Gambar 21, Gambar 22, dan Gambar 23 di bawah.

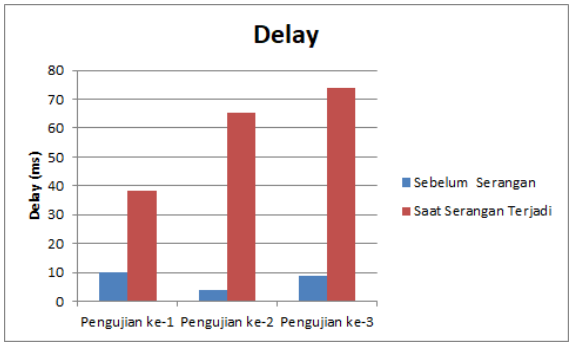


Gambar 20. Perbandingan *Throughput*

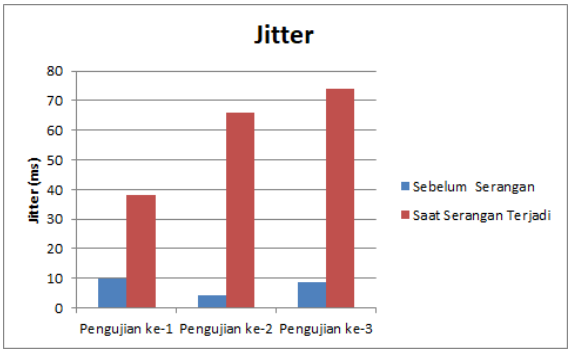


Gambar 21. Perbandingan *Packet Loss*

 ISSN:2302-8149
e-ISSN:2540-9719



Gambar 22. Perbandingan *Delay*



Gambar 23. Perbandingan *Jitter*

5 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan bahwa Wireshark, Snort dan XARP dapat dengan efektif mendeteksi serangan dan dapat menyimpan *log* yang bisa di investigasi lebih lanjut. Analisis *Network forensic* menggunakan metode *Network Forensics Generic Process* menghasilkan beberapa informasi antara lain seperti IP *address* penyerang, MAC *address attacker*, dan kapan waktu terjadinya serangan. Parameter *quality* of *service* mengalami perubahan saat terjadi serangan. Pada pengujian pertama nilai *throughput* turun 73% saat serangan terjadi, nilai *packet loss* turun 63,2%, nilai *delay* naik 74,2%, dan nilai *jitter* juga naik 74,1%. Pada pengujian kedua nilai *throughput* turun 93,4%, nilai *packet loss* turun 90,7%, nilai *delay* naik 94%, dan nilai *jitter* juga naik 93,7%. Pada pengujian ketiga nilai *throughput* turun 85,4%, nilai *packet loss* turun 74,6%, nilai *delay* naik 88,1% dan nilai *jitter* juga naik 88,2%.

Referensi

[1] M. N. Hafizh, I. Riadi, and A. Fadlil, "Forensik Jaringan Terhadap Serangan ARP Spoofing menggunakan Metode Live Forensic," *J. Telekomun. dan Komput.*, vol. 10, no. 2, p. 111, 2020, doi: 10.22441/incomtech.v10i2.8757.

[2] A. Mallik, A. Ahsan, M. Shahadat, and J.-C. Tsou, "Man-in-the-middle-attack: Understanding in simple words," *Int. J. Data Netw. Sci.*, vol. 3, pp. 77–92, Jan. 2019, doi: 10.5267/j.ijdns.2019.1.001.

[3] Syaifuddin, A. D. Regata, and T. A. Gholib, "Analisis Address Resolution Protocol Poisoning Attack pada Router WLAN menggunakan Metode Live Forensics," *J. Komput. Terap.*, vol. 7, no. 1, pp. 62–73, 2021, [Online]. Available: https://jurnal.pcr.ac.id/index.php/jkt/.

[4] A. Majumdar, S. Raj, and T. Subbulakshmi, "ARP Poisoning Detection and Prevention using Scapy," *J. Phys. Conf. Ser.*, vol. 1911, no. 1, 2021, doi: 10.1088/1742-6596/1911/1/012022.

[5] D. Mualfah and I. Riadi, "Network Forensics for Detecting Flooding Attack on Web Server," *Int. J. Comput. Sci. Inf. Secur.*, vol. 15, pp. 326–331, Mar. 2017.

[6] D. T. Yuwono, A. Fadlil, and S. Sunardi, "Perbandingan Kinerja Perangkat Lunak Forensik untuk File Carving dengan Metode NIST," *J. Teknol. dan Sist. Komput.*, vol. 7, no. 3, pp. 89–92, 2019, doi: 10.14710/jtsiskom.7.3.2019.89-92.

[7] S. Sunardi, I. Riadi, and M. Akbar, "Steganalisis Bukti Digital pada Media Penyimpanan menggunakan Metode Static Forensics," *J. Nas. Teknol. dan Sist. Inf.*, vol. 6, pp. 1–8, Jun. 2020, doi: 10.25077/TEKNOSI.v6i1.2020.1-8.

[8] N. Hildayanti and I. Riadi, "Forensics Analysis of Router on Computer Networks using Live Forensics Method," *Int. J. Cyber-Security Digit. Forensics*, vol. 8, pp. 74–81, May 2019, doi: 10.17781/P002559.

[9] G. Kamajaya, I. Riadi, and Y. Prayudi, "Analisa Investigasi Static Forensics Serangan Man in the Middle Berbasis Arp Poisoning," *JIKO (Jurnal Inform. dan Komputer)*, vol. 3, no. 1, pp. 6–12, 2020, doi: 10.33387/jiko.v3i1.1692.

[10] D. Saputra and I. Riadi, "Network Forensics Analysis of Man in the Middle Attack using Live Forensics Method," *Int. J. Cyber-Security Digit. Forensics*, vol. 8, no. 1, pp. 66–73, 2019, doi: 10.17781/p002558.

[11] I. Riadi, A. Fadlil, and M. N. Hafizh, "Analisis Bukti Serangan Address Resolution Protocol

 ISSN:2302-8149
e-ISSN:2540-9719

Spoofing menggunakan Metode National Institute of Standard Technology," *Edumatic J. Pendidik. Inform.*, vol. 4, no. 1, pp. 21–29, 2020, doi: 10.29408/edumatic.v4i1.2046.

[12] M. R. Choiruman, J. G. A. Ginting, and N. Iryani, "Analisis Pendeteksian Serangan ARP Poisoning dengan menggunakan Metode Live Forensic," *InfoTekJar J. Nas. Inform.*, vol. 2, pp. 0–4, 2022.

[13] R. Rizal, I. Riadi, and Y. Prayudi, "Network Forensics for Detecting Flooding Attack on Internet of Things (IoT) Device," vol. 7, pp. 382–390, Sep. 2018.

[14] A. Almaarif and S. Yazid, "ARP Cache Poisoning sebagai Teknik Alternatif untuk Membatasi Penggunaan Bandwidth berbasis Waktu," *J. Rekayasa Sist. Dan Ind.*, vol. 05, pp. 2–7, 2018.

[15] O. Prayogo and I. Riadi, "Router Forensic Analysis against Distributed Denial of Service (DDoS) Attacks," *Int. J. Comput. Appl.*, vol. 175, pp. 19–25, Dec. 2020, doi: 10.5120/ijca2020920944.