BAB 2
DASAR TEORI

2.1 KAJIAN PUSTAKA

Penelitian oleh Firmansyah, Abdul Fadlil, Rusydi Umar pada tahun 2019 yang berjudul "Analisis Forensik *Metarouter* pada Lalu Lintas Jaringan Klien". Pada penelitian ini, menggunakan fitur *metarouter* yang ada pada mikrotik dengan skenario jaringan yaitu membuat dua *unit router* virtual yang akan terkoneksi dengan internet dan *router* asli. *Metarouter* digunakan untuk memecah jaringan *router* menjadi beberapa unit *router* virtual untuk pengujian, dilakukan serangan badai ARP pada *router* virtual dengan cara membanjiri protokol ARP agar permintaan terhadap server tidak dapat terpenuhi. IP *address* penyusup yang melakukan serangan diketahui dengan menggunakan *tool* forensik jaringan yaitu *Wireshark*. Hasil dari penelitian ini adalah *Wireshark* dapat digunakan untuk menganalisis lalu lintas jaringan jika terjadinya penyusupan selain itu serangan badai ARP dapat diketahui dengan melihat tujuan protokol ARP pada *wireshark*, uji coba serangan pada penilitian ini menggunakan serangan badai ARP, jika terdapat *protocol* ARP terdeteksi dengan membanjiri server dalam rentan waktu 10s, maka dapat disimpulkan telah terjadinya serangan badai ARP sehingga menyebabkan terputusnya koneksi antar komputer *client* [4].

Penelitian oleh Gede E A Kamajaya, Imam Riadi, Yudi Prayudi pada tahun 2020 yang berjudul "Analisa Investigasi *Static Forensics* Serangan *Man In The Middle* Berbasis ARP *Poisoning*". Pada penelitian ini, dilakukan pengujian terhadap serangan *Man In The Middle* berbasis ARP *Poisoning* yang akan dideteksi dengan menggunakan metode *static forensics*, yaitu dengan cara menganalisis sistem secara forensik dengan mengambil *dump* memori. Serangan yang dijadikan pengujian dalam penelitian ini adalah *Man In The Middle* yang memanfaatkan *broadcast* ARP untuk melakukan *poisoning*, ARP *Poisoning* bekerja dengan cara mendaftarkan pemetaan alamat palsu pada *cache* ARP *node* satu ke *node* yang lain. Proses identifikasi serangan ARP *Poisoning* yaitu menggunakan *tool wireshark* untuk menganalisis lalu lintas yang terjadi pada jaringan dan digunakan untuk melihat aktivitas-aktivitas yang lewat pada port

ARP. Hasil dari penelitian ini adalah *tool Wireshark* dan metode *static forensic* mampu mengidentifikasi serangan ARP *poisoning* dengan menampilkan aktivitas pada *port* [2].

Penelitian oleh M. Nasir Hafizh, Imam Riadi, Abdul Fadlil pada tahun 2020 yang berjudul "Forensik Jaringan Terhadap Serangan ARP *Spoofing* menggunakan Metode *Live Forensic*". Pada penelitian ini, menggunakan metode *live forensic* untuk mengidentifikasi serangan ARP *spoofing*. Topologi yang digunakan dalam penelitian ini yaitu menggunakan topologi *star* dengan PC sebanyak 4, *router* dan *switch* masing-masing berjumlah 1. Proses pengujian serangan ARP *spoofing* menggunakan *tools Cain and Abel*, sedangkan untuk pengumpulan data dan proses identifikasi penyerang menggunakan *tools Wireshark*. Metode yang digunakan adalah metode *live forensic*, dimana pengujian dan pendeteksian dilakukan ketika sistem dalam kondisi menyala. Hasil dari penelitian ini adalah dengan menggunakan metode *live forensic*, investigator dapat dengan cepat mendeteksi suatu serangan dan mengidentifikasi IP *address* penyerang [5].

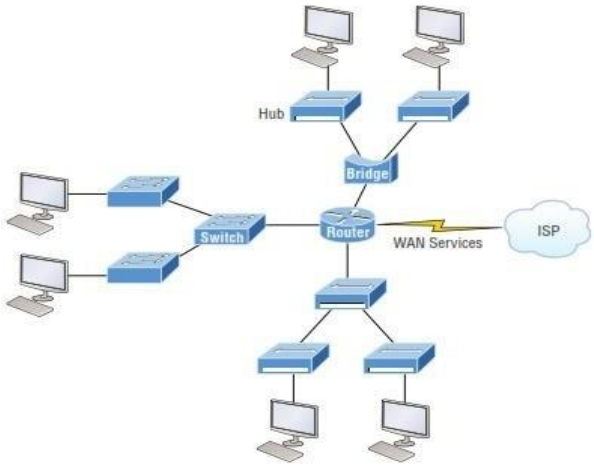
2.2 DASAR TEORI

Pada penelitian ini, penulis membutuhkan teori-teori yang berkaitan dengan topik penelitian, pada sub bab dasar teori, penulis menggali teori dari berbagai referensi, seperti buku, jurnal, skripsi, dan *website*. Dimana pada sub bab ini, penulis akan memaparkan teori tentang jaringan, forensik pada jaringan, serta metode yang berkaitan dengan penelitian ini.

2.2.1 KONSEP JARINGAN KOMPUTER

Jaringan komputer adalah sekumpulan peralatan komputer yang dihubungkan agar dapat saling berkomunikasi dengan tujuan membagi sumber daya. Dalam sebuah jaringan komputer dibutuhkan aturan-aturan (*protocols*) yang mengatur komunikasi dan layanan-layanan secara umum untuk seluruh sistem jaringan [6]. Jaringan komputer merupakan interkoneksi antara 2 komputer *autonomous* atau lebih, yang terhubung dengan media transmisi kabel atau tanpa kabel (*wireless*). *Autonomous* adalah apabila sebuah komputer tidak melakukan

kontrol terhadap komputer lain dengan akses penuh, sehingga dapat membuat komputer lain, *restart, shutdowns,* kehilangan *file* atau kerusakan sistem [7].



Gambar 2.1 *Internetworking* [6]

Sistem koneksi antar *node* (komputer) ada dua, yakni [7]:

a. *Peer to peer*

   Peer artinya rekan sekerja. *Peer-to-peer network* adalah jaringan komputer yang terdiri dari beberapa komputer (biasanya tidak lebih dari 10 dengan 1-2 *printer*).

b. *Client-Server*

   Sistem ini bisa diterapkan dengan teknologi internet di mana ada suatu unit komputer yang berfungsi sebagai server yang hanya memberikan layanan bagi komputer lain, dan client yang hanya meminta layanan dari server.

2.2.2 PROTOKOL JARINGAN KOMPUTER

Protokol adalah sebuah aturan atau standar yang mengatur atau mengijinkan terjadinya hubungan, komunikasi, dan perpindahan data antara dua atau lebih titik komputer. Fungsi protokol jaringan komputer secara umum adalah untuk menghubungkan pengirim dan penerima dalam berkomunikasi dan bertukar informasi supaya dapat berjalan dengan akurat dan lancar. Sistem protokol ini menentukan aturan-aturan yang perlu ditaati oleh perangkat pengirim dan juga penerima agar hubungan jaringan berlangsung dengan baik. Aturan-aturan

protokol termasuk di dalamnya petunjuk yang berlaku bagi cara-cara atau metode mengakses sebuah jaringan, topologi fisik, tipe-tipe kabel dan kecepatan transfer data. Macam-macam protocol pada jaringan komputer adalah [8]:

1. TCP/IP

   Transmission Control Protocol (TCP) dan *Internet Protocol* (IP) merupakan standar dari komunikasi data yang dipakai oleh komunitas internet. Standar ini mengatur dalam proses tukar-menukar data atau informasi dari satu komputer ke komputer lain di dalam jaringan internet. TCP cocok digunakan untuk koneksi yang membutuhkan keandalan tinggi seperti Telnet, HTTP, FTP, SSH.

2. UDP (*User Datagram Protocol)*

   User Datagram Protocol (UDP) adalah transport TCP/IP yang dapat mendukung komunikasi yang *unreliable*, tanpa adanya koneksi antar *host* di dalam suatu jaringan. UDP untuk koneksi yang tidak terlalu kritis seperti transmisi audio/video (VoIP dan audio/video *streaming*). UDP kurang baik untuk pengiriman packet berukuran besar karena mengakibatkan banyak *packet loss.*

3. DNS (*Domain Name System*)

   Domain Name System (DNS) adalah *distribute database system* yang digunakan untuk pencarian nama komputer (*name resolution*) di jaringan yang mengunakan TCP/IP (*Transmission Control Protocol/Internet Protocol*). DNS biasa digunakan pada aplikasi yang terhubung ke Internet seperti web *browser* atau *e-mail*, dimana DNS membantu memetakan *host name* sebuah komputer ke IP *address.*

4. *Internet Control Message Protocol* (ICMP)

   Internet Control Message Protocol (ICMP)adalah salah satu protokol inti dari keluarga. ICMP berbeda tujuan dengan TCP dan UDP dalam hal ICMP tidak digunakan secara langsung oleh aplikasi jaringan milik pengguna. salah satu pengecualian adalah aplikasi ping yang mengirim pesan ICMP *Echo Request* (dan menerima *Echo Reply*) untuk menentukan apakah komputer tujuan dapat dijangkau dan berapa lama paket yang dikirimkan dibalas oleh komputer tujuan.

5. HTTP (*Hypertext Transfer Protocol*)

   HTTP (*Hypertext Transfer Protocol*) suatu protokol yang digunakan oleh WWW (*World Wide Web*). HTTP mendefinisikan bagaimana suatu pesan bisa diformat dan dikirimkan dari server ke *client*.

6. SSH (*Secure Shell*)

   SSH (*Secure Shell*) adalah protocol jaringan yang memungkinkan pertukaran data secara aman antara dua komputer. SSH dapat digunakan untuk mengendalikan komputer dari jarak jauh mengirim *file*, membuat *tunnel* yang terenkripsi dan lain-lain.

7. *File Transfer Protocol* (FTP)

   FTP (*File Transfer Protocol*) adalah sebuah *protocol* internet yang berjalan di dalam lapisan aplikasi yang merupakan standar untuk pentransferan berkas (*file*) *computer* antar mesin-mesin dalam sebuah *internetworking.*

8. *Secure Socket Layer* (SSL)

   SSL (Secure Socket Layer) adalah cara untuk sebuah website untuk membangun koneksi yang aman (terenkripsi) antara webserver (website) dengan client (Browser) atau antara mail server dengan mail client. Sehingga koneksi antara client dan server dapat berjalan secara aman dari pihak lain yang tidak berkepentingan [8].

2.2.3 *OPEN SYTEM INTERCONNECTION* (OSI) *LAYER*

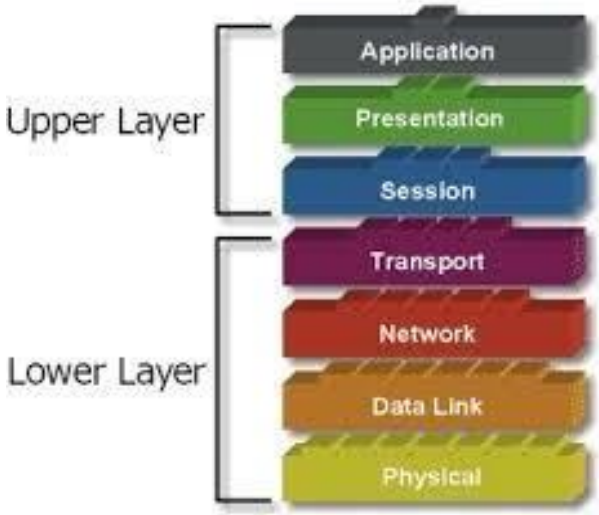
OSI merupakan singkatan dari *Open System Interconnection* adalah *standard* komunikasi yang diterapkan di dalam jaringan komputer. *Standard* itulah yang menyebabkan seluruh alat komunikasi dapat saling berkomunikasi melalui jaringan. Ketika OSI belum digunakan, perangkat komunikasi yang berasal dari vendor berbeda tidak dapat saling berkomunikasi. Alat komunikasi yang diciptakan oleh IBM tidak dapat berkomunikasi dengan vendor lain [9].

Open Systems Interconnection (OSI) model juga merupakan suatu referensi untuk memahami komunikasi data antara dua buah sistem yang saling terhubung. Model *Layer* OSI dibagi dalam dua group yaitu *upper layer* dan *lower layer*. OSI *layer* membagi proses komunikasi menjadi tujuh lapisan. Setiap lapisan berfungsi untuk melakukan fungsi-fungsi spesifik untuk mendukung

lapisan diatasnya dan sekaligus juga menawarkan layanan untuk lapisan yang ada dibawahnya [9].

Gambar 2.2 Lapisan OSI *Layer* [9]

Pada lapisan OSI *layer,* terdapat beberapa lapisan yaitu [9]:



1. *Physical layer*

   Physical layer adalah *layer* yang paling sederhana yang berkaitan dengan *electrical* dan *optical* koneksi antar peralatan. Data *biner* dikodekan dalam bentuk yang dapat ditransmisi melalui media jaringan, sebagai contoh kabel, *transceiver* dan *konektor* yang berkaitan dengan *layer* p*hysical.*

2. *Data-Link Layer*

   Layer ini menyediakan transfer data yang lebih nyata. Sebagai penghubung antara media *network* dan *layer protocol* yang lebih *high-level, layer* data *link* bertanggung jawab pada paket akhir dari data binari yang berasal dari *level* yang lebih tinggi ke paket *diskrit* sebelum ke *layer physical.*

3. *Network Layer*

   Tugas utama dari *network layer* adalah menyediakan fungsi *routing* sehingga paket dapat dikirim keluar dari *segment network lokal* ke suatu tujuan yang berada pada suatu *network* lain. IP, Internet *Protocol,* umumnya digunakan untuk tugas ini.

4. *Transport Layer*

Layer trans*port* data, menggunakan *protocol* seperti UDP, TCP dan/atau SPX (*Sequence Packet eXchange*, yang satu ini. digunakan oleh *NetWare*, tetapi khusus untuk koneksi berorientasi IPX). *Layer transport* adalah pusat dari model OSI.

5. *Session Layer*

   Layer Session menyediakan layanan ke dua *layer* diatasnya, melakukan koordinasi komunikasi antara *entiti layer* yang diwakilinya. Fungsi *session layer* antara lain untuk mendefinisikan bagaimana koneksi dapat dibuat, dipelihara, atau dihancurkan.

6. *Presentation Layer*

   Layer presentation dari model OSI melakukan hanya suatu fungsi tunggal translasi dari berbagai tipe pada *syntax* sistem. Fungsi presentation *layer* antara lain untuk mentranslasikan data yang hendak ditransmisikan oleh aplikasi ke dalam format yang dapat ditransmisikan melalui jaringan.

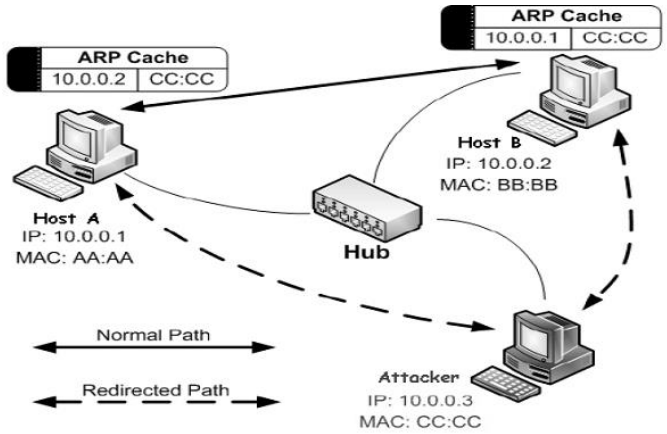
7. *Aplication Layer*

   Layer Application adalah penghubung utama antara aplikasi yang berjalan pada satu komputer dan *resources network* yang membutuhkan akses padanya. *Layer Application* adalah *layer* dimana *user* akan beroperasi padanya, *protocol* seperti FTP, *telnet*, SMTP, *HTTP* , POP3 berada pada *layer Application* [9].

2.2.4 *ADDRESS RESOLUTION PROTOCOL* (ARP) *POISONING*

Address Resolution Protocol (ARP) merupakan protokol dalam TCP/IP *Protocol Suite* yang bekerja diantara *network layer* dan *data link layer* dan bertanggung jawab dalam melakukan resolusi pencatatan dan pencocokan alamat IP ke dalam alamat *Media Access Control* (MAC Address) lalu hasilnya letakkan didalam ARP *cache* [10]. *Address Resolution Protocol* (ARP) merupakan protokol yang bekerja dengan menyatukan alamat IP dan alamat MAC sehingga dapat menjalankan komunikasi pada jaringan *Local Area Network* (LAN) dan menghubungkan peralatan yang digunakan sehingga terjadi transaksi komunikasi. ARP adalah protokol yang dapat menangani proses pengalamatan berdasarkan alamat IP ke alamat MAC. ARP didefinisi berdasarkan aturan yang ditetapkan

pada RFC 826 dan kemudian diperbarui pada RFC 5494. ARP lazimnya digunakan oleh peralatan jaringan untuk merawat/menjaga *internal cache* dari alamat MAC untuk dipetakan ke alamat IP. Dalam praktiknya, ARP tidak dapat memberikan jaminan keamanan dan merupakan bagian celah yang dapat digunakan oleh *hacker*. Seorang *hacker* dapat menggunakan protokol ARP untuk menangkap sebuah client server FTP session melalui akses pada jaringan *switch* [2].



Gambar 2.3 Serangan ARP *Poisoning* [11]

Dengan teknik ARP *Poisoning*, seorang penyerang dapat mengirimkan pesan ARP palsu ke dalam jaringan area lokal. Untuk itulah suatu jaringan perlu mendapatkan proteksi dari penyalahgunaan. ARP *poisoning* merupakan salah satu aktifitas yang dapat dilakukan untuk penyerangan terhadap suatu jaringan. ARP *poisoning* merupakan sebuah teknik yang efektif untuk menangkap, mendengarkan, dan membajak koneksi antar komputer dalam jaringan. ARP *poisoning* merupakan cara untuk memanipulasi pemetaan ARP *Cache*. ARP *poisoning* akan membuat paket ARP *Reply* palsu dan dikirimkan secara terus-menerus [10].

2.2.5 KEAMANAN JARINGAN

Keamanan jaringan atau *network security* dalam jaringan computer sangat penting dilakukan untuk *monitoring* akses jaringan dan mencegah penyalahgunaan sumber daya jaringan yang tidak sah. Suatu hal yang perlu

diingat bahwa tidak ada jaringan yang anti sadap atau tidak ada jaringan computer yang benar-benar aman. Sifat dari jaringan melakukan komunikasi. Setiap komunikasi dapat jatuh ketanganorang lain dan disalah gunakan, sistem kemanan mampu mengamankan jaringan tanpa menghalangi penggunaan dan menempatkan antisipasi Ketika jaringan berhasil ditembus. Pastikan bahwa *user* dalam jaringan memiliki pengetahuan yang cukup mengenai keamanan yang dibuat [10]. Ada dua elemen utama pembentuk keamanan jaringan [10]:

1. Tembok pengaman, baik secara fisik maupun maya, yag ditaruh diantara piranti dan layanan jaringan yang digunakan dan orang-orang yang akan berbuat jahat.
2. Rencana pengamanan, yang akan diimplementasikan Bersama dengan *user* lainnya, untuk menjaga agar sistem tidak bisa ditembus dari luar.

Berdasarkan lubang keamanan, keamanan dapat diklasifikasikan menjadi empat yaitu [10]:

1. Kemanan yang bersifat fisik (*physical security*) termasuk akses akses orang ke Gedung, peralatan dan media yang digunakan. Contoh :
   a. *wiretapping* atau hal-hal yang berhubungan dengan akses ke kabel atau komputer yang digunakan juga dapat dimasukkan ke dalam kelas ini.
   b. ARP *Poisoning*, dilakukan dengan memanfaatkan protokol ARP (*Address Resolution Protocol*) untuk melakukan *scanning host* dan memanipulasi MAC *address.*
   c. *Denial of Service*, dilakukan misalnya dengan mematikan peralatan atau membanjiri saluran komunikasi dengan pesan-pesan (yang dapat berisi apa saja karena yang diutamakan adalah banyaknya jumlah pesan).
2. Kamanan yang berhubungan dengan (personal), contoh :
   a. Identifikasi *user* (*username* dan *password*)
   b. Profil resiko dari orang yang mempunyai akses (pemakai dan pengelola)
3. Kemanan dari data dan media serta Teknik komunikasi (*communications*).
4. Keamanan dalam operasi, adanya prosedur yang digunakan untuk mengatur dan mengelola sistem keamanan, dan juga termasuk prosedur setelah serangan (*post attack recovery*)

Tujuan keamanan jaringan terdiri dari 3 aspek yaitu *confidentiality, Integrity* dan *avaibility.* Tiga aspek itulah yang dianggap sebagai tiga komponen *Cyber Security* paling penting diseluruh *platform,* terutama pada *Web App*

Gambar 2.4 CIA TRIAD [11]

Berikut ini penjalasan lengkap tentang 3 aspek tersebut [11] :

1. *Confidentiality*

   Confidentiality adalah kerahasiaan. Kerahasiaan dalam hal ini adalah informasi yang dimiliki pada sistem/*database* adalah hal yang rahasia dan pengguna atau orang yang tidak berkepentingan tidak dapat melihat atau mengaksesnya.

2. *Integrity*

   Integrity adalah data tidak diubah dari aslinya oleh pengguna yang tidak berhak. Sehingga, konsistansi, akurasi, dan validitas data tersebut masih terjaga. Atinya, *integrity* mencoba memastikan data yang disimpan benar adanya, tidak ada pengguna yang tidak berkepentingan atau *software* berbahaya yang mengubahnya.

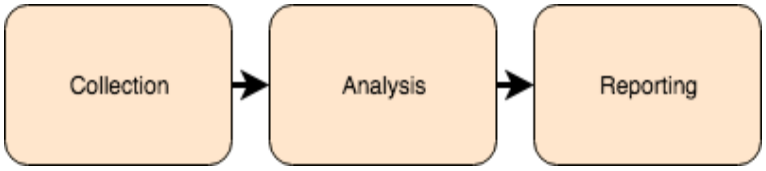
3. *Avaibility*

   Avaibility adalah memastikan sumber daya yang ada siap diakses kapan punoleh *user/application/ system* yang membutuhkannya. Sama seperti aspek

integrity, rusaknya aspek pada *avaibility* dari sistem juga bisa diakibatkan karena faktor kesengajaan dan factor *accidental.*

2.2.6 *NETWORK FORENSIC*

Istilah *forensics* dikenal sebagai proses ilmiah untuk mendapatkan kembali fakta yang tersembunyi dari sebuah lingkungan kejadian dan disajikan ke pengadilan. *Network forensics* adalah teknologi investigasi yang mencakup tugas penangkapan, perekaman, pemantauan, dan analisis paket *traffic* dalam jaringan untuk menentukan ada atau tidaknya lalu lintas yang mengindikasikan serangan. *Network forensics* memiliki dua fungsi yakni keamanan jaringan (mendapatkan barang bukti untuk proses penyidikan) dan aspek hukum untuk menemukan bukti pengiriman *file*, kata kunci, dan rincian komunikasi dalam *email* atau *chatting* [12].



Gambar 2.5 Tahapan *Network Forensic* [13]

Berbeda dari forensik pada umumnya, forensik komputer adalah kegiatan mengumpulkan dan menganalisis data dari berbagai sumber daya komputer. *Log* yang berasal dari komputer (forensik komputer) adalah log antivirus, log database atau log dari aplikasi yang digunakan. Forensik jaringan merupakan bagian dari forensik digital, dimana bukti ditangkap dari jaringan dan di interpretasikan berdasarkan pengetahuan dari serangan jaringan. Hal ini bertujuan untuk menemukan penyerang dan merekonstruksi tindakan serangan penyerang melalui analisis bukti penyusupan [13].

Network forensics juga meng-*capture*, merekam, dan menganalisis kejadian didalam jaringan untuk menemukan sumber serangan keamanan. Menangkap lalu lintas jaringan melalui jaringan itu sederhana secara teori, tetapi dalam praktiknya relatif kompleks. Ini dikarenakan besarnya jumlah data yang mengalir melalui jaringan dan sifat kompleks dari protokol internet atau *Network Forensic Process* merupakan suatu metode yang dapat digunakan untuk kegiatan

investigasi dan analisa aktivitas *cyber crime*. Dimana bukti ditangkap dari jaringan dan diinterpretasikan berdasarkan pengamatan [14].

2.2.7 *LIVE FORENSIC*

Live forensic yaitu suatu teknik analisis dimana menyangkut data yang berjalan pada sistem atau data *volatile* yang umumnya tersimpan pada *Random Access Memory* (RAM) atau transit pada jaringan [15]. Teknik *live forensics* memerlukan kecermatan dan ketelitian, dikarenakan data *volatile* pada RAM dapat hilang jika sistem mati, dan adanya kemungkinan tertimpanya data penting yang ada pada RAM oleh aplikasi yang lainnya. Karena itu diperlukan metode *live forensics* yang dapat menjamin integritas dan keaslian data *volatile* tanpa menghilangkan data yang berpotensial menjadi barang bukti [16].

Live Forensics pada dasarnya memiliki kesamaan pada teknik forensik tradisional dalam hal metode yang dipakai yaitu identifikasi, penyimpanan, analisis, dan presentasi, hanya saja *live forensics* merupakan respon dari kekurangan teknik forensik tradisonal yang tidak bisa mendapatkan informasi dari data dan informasi yang hanya ada ketika sistem sedang berjalan misalnya aktifitas *memory, network proses, swap file, running system proses,* dan informasi dari *file* sistem dan ini menjadi kelebihan dari teknik *live forensics* [17].

Pada metode *live forensics* bertujuan untuk penanganan insiden lebih cepat, integritas data lebih terjamin, teknik enkripsi lebih memungkinkan bisa dibuka dan kapasitas memori yang lebih rendah bila dibandingkan dengan metode forensik tradisional. Banyak *tools* untuk digunakan live forensics untuk analisis data. *Tools* yang dibandingan pada metode live forensics yaitu dari kemampuan penggunaan memory, waktu, jumlah langkah dan akurasi paling baik dalam melakukan *live forensic* [18].

Teknik *live forensics* ini sangat bergantung pada keadaan komputer yang sedang menyala, karena membutuhkan data yang berjalan pada *Random Access Memory* (RAM). Data pada RAM disebut juga data *volatile* atau data sementara yaitu data yang hanya terdapat saat komputer menyala jika komputer mati maka data itu akan hilang. Data *volatile* ini berisi data penting seperti *username, password, file* akses, *file* modifikasi, aplikasi yang digunakan, kata kunci

pencarian. *Username* dan *password* merupakan hal yang penting dalam suatu akun seperti email. Email ini biasanya mengirimkan sesuatu yang penting bahkan data privasi suatu perusahaan atau penggunanya [5].

Live forensics dapat dilakukan ketika sistem belum mati atau *down*, karena hampir kesuluruhan penggunaan sistem tersimpan pada RAM, *page file*, *hibernation file* dan *crash dump file*. Tujuan pentingnya analisis data pada RAM, yaitu dapat mengetahui letak data tersebut dan isi data tersebut. Semua data pada komputer yang berpergian harus melewati RAM, apakah itu membutuhkan jaringan Internet, menyalin atau memindahkan *file*, membuka *file* pada hardisk ataupun menghapusnya semua terekam pada RAM. Perbedaaan RAM dan *Hardisk* yaitu RAM mencatat sesuatu yang terjadi pada waktu dan kondisi tertentu sedangkan hardisk hanya memberikan informasi data yang secara umum. Hal ini sangat penting karena hanya ada data dengan jumlah yang besar dan tidak pernah terdaftar pada *hardisk* yaitu data Internet [5].

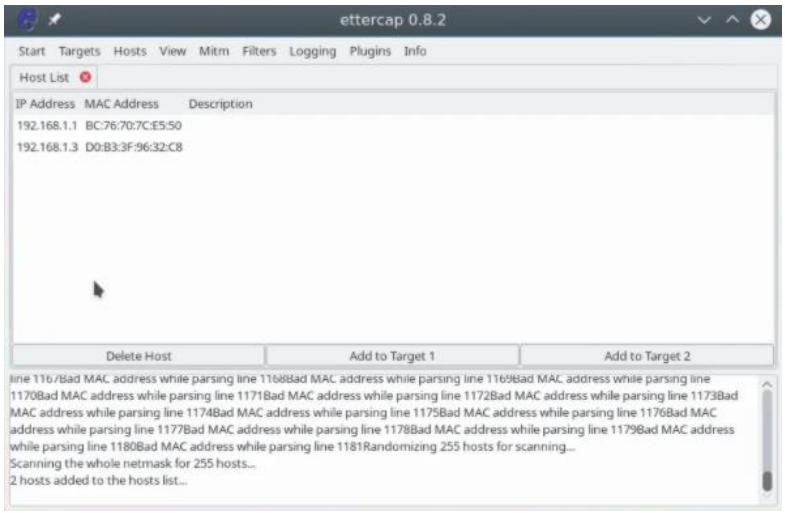
2.2.8 *TOOLS ETTERCAP*

Ettercap adalah sebuah *tools packet sniffer* yang dipergunakan untuk menganalisa protokol jaringan dan mangaudit keamanan jaringan. Dan memiliki kemampuan untuk memblokir lalu lintas pada jaringan LAN, mencuri *password*, dan melakukan penyadapan aktif terhadap protokol-protokol umum, *packet sniffing* juga dapat di salah gunakan oleh pihak yang tidak bertanggung jawab untuk mencuri data penting yang dimiliki oleh user yang sedang terhubung dengan *acces point* [19].

Gambar 2.6 Logo *Ettercap* [19]

Ettercap adalah utilitas untuk menganalisis lalu lintas jaringan yang melewati antar muka komputer, tetapi dengan fungsionalitas tambahan. Program

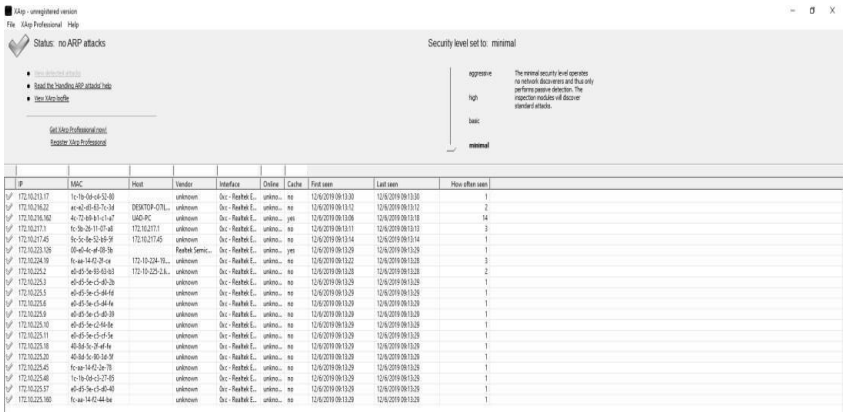
ini memungkinkan untuk melakukan serangan *man-in-the-middle* untuk memaksa komputer lain mengirim paket bukan ke *router*, tetapi kepada penyerang. Dengan *Ettercap*, dapat dilakukan pemeriksaan keamanan jaringan, seberapa rentannya terhadap jenis serangan serta menganalisis lalu lintas dari beberapa komputer, dan bahkan memodifikasinya dengan cepat [19].



Gambar 2.7 Tampilan Awal *Ettercap* [19]

2.2.9 *TOOLS* XARP

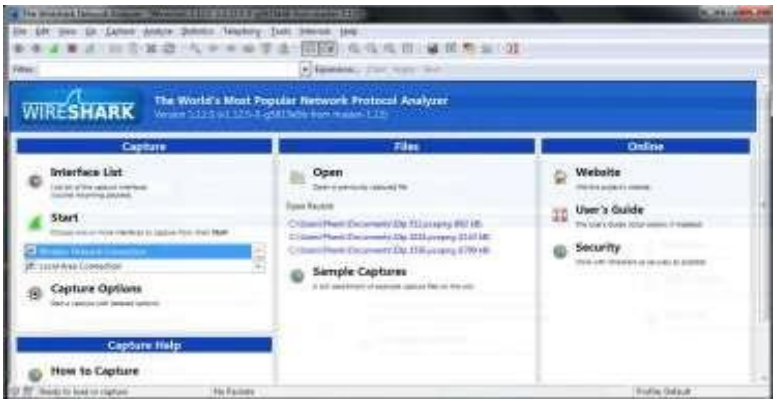
XArp adalah aplikasi keamanan yang menggunakan teknik canggih untuk mendeteksi serangan berbasis ARP. Menggunakan modul aktif dan pasif, *XArp* mendeteksi peretas di dalam jaringan. Serangan *spoofing* ARP tidak terdeteksi oleh *firewall* dan keamanan sistem operasi [20]. Aplikasi *XArp* digunakan untuk mendeteksi dan memberikan peringatan apabila terjadi serangan ARP *Spoofing*. XArp memberikan peringatan saat serangan ARP *Spoofing* terdeteksi dan memberikan IP *Address Victim*, IP *Address Attacker* dan waktu terjadinya serangan. *XArp* memberikan peringatan dini apabila terjadi serangan ARP *Spoofing* dengan memunculkan notifikasi [5].



Gambar 2.8 Tampilan Awal XARP [5]

2.2.10 *TOOLS* WIRESHARK

Wireshark merupakan salah satu dari *tools network analyzer* yang banyak digunakan oleh *network administrator* untuk menganalisa kinerja jaringannya termasuk protokol didalamnya. Interface pada *wireshark* menggunakan *Graphical User Interface* (GUI) atau tampilan grafis. *Wireshark* mampu menangkap paket-paket data atau informasi yang melewati jaringan.semua jenis paket informasi dalam berbagai format *protocol* pun akan dengan mudah ditangkap dan dianalisa [21].



Gambar 2.9 Tampilan Awal *Wireshark* [22]

Wireshark sangat berguna dalam menyediakan jaringan dan protokol serta memberikan informasi tentang data yang tertangkap pada jaringan. *Software wireshark* dapat menganalisa transmisi paket data dalam jaringan, proses koneksi dan transmisi data antar komputer. Dapat mengumpamakan sebuah *Network Packet Analyzer* sebagai alat untuk memeriksa apa yang sebenarnya sedang terjadi

di dalam kabel jaringan [22]. Beberapa fitur kelebihan *Wireshark*, diantaranya [21]:

1. Berjalan pada sistem operasi Linux dan Windows.
2. Menangkap paket (*Capturing Packet*) langsung dari *network interface.*
3. Mampu menampilkan hasil tangkapan dengan *detail.*
4. Dapat melakukan pemfilteran paket
5. Hasil tangkapan dapat di *save*, di *import* dan di export.

Wireshark berguna untuk pekerjaan analisis jaringan. Cara kerjanya yaitu dengan menangkap paket-paket data dari protokol-protokol yang berbeda dari berbagai tipe jaringan yang umum ditemukan di dalam trafik jaringan internet. Paket-paket data tersebut ditangkap lalu ditampilkan di jendela hasil *capture* secara *real-time*. Pada awal proses analisis jaringan menggunakan Wireshark, semua paket data yang berhasil ditangkap tadi ditampilkan semua tanpa pilih-pilih (*promiscuous mode*). Semua paket data tersebut bisa diolah lagi menggunakan perintah *sorting* dan *filter*. Terminologi *sniffing* sebenarnya tidak jauh berbeda dengan capture paket data, namun dalam konotasi yang negatif, karena bisa jadi menimbulkan dampak yang merugikan untuk orang lain terutama dari sisi privasi.

Agar dapat bekerja dengan baik, *Wireshark* membutuhkan aplikasi bernama *WinPcap* atau *Npcap* sebagai pondasinya. WinPcap masih dapat digunakan sampai versi Windows 7, sedang untuk Windows 10 sudah tidak didukung lagi, seterusnya sudah dikembangkan Npcap. Berbeda dengan *pcap* sebagai libcap *library* pada sistem Linux, Windows hanya menggunakan sebuah *port* saja dari *library* libcap tersebut yaitu *Npcap* [22].

Pcap adalah sebuah API (*application programming interface*) untuk melakukan *capture* terhadap trafik jaringan internet. *Pcap* bukan sesuatu yang baru, *Pcap* adalah bagian *core*/inti dari program *capture paket* data pendahulunya, *TCPdump*. *Wireshark* menggunakan *pcap* untuk menangkap paket data, sehingga seorang analis jaringan yang menggunakan *Wireshark* hanya dapat melakukan penangkapan tipe-tipe paket data yang hanya didukung oleh *pcap* saja. Pada hasil tangkapan *wireshark*, terdapat tiga bagian jendela, yaitu [23] :

1. Jendela *packet list*

Pada jendela ini, hasil tangkapan paket data disusun di dalam format tabel. Setiap paket yang diterima ditampilkan dalam baris/*row* sesuai nomor korespondennya secara urut. Semakin lama proses *capture*, maka akan semakin banyak baris/*row* data paket yang 'tertangkap'. Setiap baris akan memuat unit-unit informasi paket di antaranya sumber paket (*source*), destinasi (*destination*), protokol (*protocol*), *length* (panjang paket data dalam satuan *bytes*), dan info.

2. Jendela *packet details*

   Jendela yang ini terletak di bagian tengah, fungsinya untuk menyajikan substansi informasi protokol-protokol dari baris paket data yang dipilih pada jendela *packet list*, data tersebut disajikan secara horizontal dan berhirarki.

3. Jendela *packet bytes*

   Pada jendela yang paling bawah ini ditampilkan data *raw* dari paket data yang diseleksi pada jendela paling atas (*packet list*). Data *raw* tersebut tampil dalam *format hexadesimal* (hex). Data hex tersebut memuat 16 *hexadecimal bytes* dan 16 ASCII *bytes* [23].

2.2.11 WEB SERVER

Web Server adalah sebuah perangkat lunak server yang berfungsi menerima permintaan HTTP atau HTTPS dari client yang dikenal dengan web browser dan mengirimkan kembali hasilnya dalam bentuk halaman-halaman web yang umumnya berbentuk dokumen HTML. Salah satu server web yang terkenal di linux adalah Apache. Apache merupakan server web antar *platform* yang dapat berjalan di beberapa platform seperti linux dan windows. Web Server juga merupakan sebuah komputer yang menyediakan layanan untuk internet [24].

2.2.12 PHP *MY ADMIN*

Pada penelitian ini dibangun *webserver* yang digunakan untuk akses komputer *client* menuju komputer server untuk membangun layanan web server tersebut, penelitian ini memerlukan PHP *MyAdmin*. Aplikasi ini merupakan sebuah *website open source* yang ditulis dengan bahasa pemrograman PHP, XHTML, CSS, *JavaScript* dan berfungsi untuk pengolahan database

MySQL dalam bentuk tampilan *website*. Aplikasi ini dikembangkan oleh The Php *MyAdmin Project* dan di rilis pada tanggal 09 September 1998. Pada dasarnya untuk mengelola MySQL untuk membuat sebuah *web server* diperlukan pengetikan secara manual pada *command line* sehingga mengharuskan untuk menghafal *script* untuk membuat, menghapus dan mengedit pada *data base* MySQL. Akan tetapi dengan menggunakan *software* berbasis web ini *(PhpMyAdmin*) akan dapat dengan mudah untuk melakukan manipulasi *Database* MySQL yang dibuat. Fungsi PHP My Admin diantara lain adalah membuat *database*, mengedit *data base*, menghapus *database*, membuat tabel, mengedit tabel, menghapus tabel, membuat relasi antar tabel, menghapus relasi antar tabel dan mensortir data [24].

Gambar 2.10 Logo PHP My Admin [24].

2.2.13 FILEZILLA

FileZilla atau dikenal dengan sebutan FileZilla *Client*, adalah salah satu *software* FTP gratis, *open source*, *cross-platform*. Binari tersedia untuk Windows, Linux, dan Mac OS X. *Software* ini mendukung FTP, SFTP, dan FTPS. FileZilla *Server* adalah produk lain dari FileZilla *Client*. Ini adalah server FTP yang didukung oleh proyek yang sama dan fitur-fitur dukungan untuk FTP dan FTP melalui SSL/TLS. Kode sumber FileZilla ditaruh pada *SourceForge.net*. Fitur utama dari Filezilla adalah [24] :

1. *Site manager*

   Mengizinkan pengguna untuk membuat daftar situs FTP beserta data koneksinya, seperti nomor *port* yang akan digunakan, protokol yang digunakan, dan apakah akan menggunakan log anonim atau normal. Untuk log normal, nama pengguna dan kata sandinya akan disimpan. Penyimpanan kata sandi adalah opsional.

2. *Message log*

   Ditampilkan di bagian atas jendela. Fitur ini menampilkan output berjenis *console type* yang menunjukkan perintah yang dikirim oleh FileZilla dan respon yang diterima dari server.

3. *File and folder view*

   Ditampilkan di bawah *message log*, menyediakan sebuah tampilan grafis antarmuka untuk FTP. Pengguna dapat menavigasi folder serta melihat dan mengubah isinya pada komputer lokal dan server dengan menggunakan tampilan antarmuka gaya *explorer*. Pengguna dapat men-*drag* dan *drop file* antara komputer lokal dan server.

4. *Transfer queue*

   Ditampilkan di sepanjang bagian bawah jendela, menunjukkan status *real time* setiap antrian atau *transfer file* yang aktif [25].

Gambar 2.11 Logo Filezilla [25]