BAB 2
KAJIAN PUSTAKA & DASAR TEORI

2.1 KAJIAN PUSTAKA

Dalam penyusunan tugas akhir ini, penelitian dilakukan dengan cara membandingan beberapa jurnal terdahulu untukk mengetahui metode apa saja yang digunakan dalam topik yang diambil yaitu *honeypot*. Berikut merupakan jurnal terdahulu yang berkaitan dengan topik tersebut:

Pada penelitian [6] yang dilakukan pada tahun 2020 ini menjelaskan mengenai implementasikan *honeypot Dionaea* di salah satu komputer labor dalam menganalisa serangan atau *malware* yang dilakukan oleh seorang administrator jaringan dengan *honeypot Dionaea* menjadi lebih terorganisir dan tepat sasaran, sehingga informasi data *malware* dapat digunakan oleh *administrator* jaringan untuk mempelajari perilaku *malware* serta pencegahan yang dapat dilakukan.

Penelitian selanjutnya [7] dilakukan pada tahun 2019 membahas tentang Analisis dan implementasi *honeypot* menggunakan *kippo* sebagai penunjang keamanan jaringan. Berdasarkan skenario uji coba sistem, implementasi *kippo* dapat membaca setiap serangan dari *attacker* dan mampu mendeteksi serangan *brute force* sesuai indikator serangan yang dicantumkan pada penelitian ini dan dalam penggunaan *honeypot* ini dapat melindungi *resource* utama dari percobaan serangan yang dilakukan oleh pihak eksternal (*attacker*).

Kemudian untuk penelitian [8] dilakukan pada tahun 2018 dengan judul Honeypot Menggunakan Honeyd Sebagai Solusi Keamanan Jaringan Dari Aktivitas Serangan. Pada penelitian ini peneliti melakukan beberapa jenis serangan yang dilakukan pengujian diantaranya serangan *Denial of service, file transfer protocol dan internet control message protocol*. *Honeyd* berhasil menampilkan notifikasi di *log honeyd* Ketika serangan masuk. Berdasarkan hasil pengamatan diperoleh rata-rata waktu yang diperlukan untuk jenis serangan *DoS* 2,33 detik, serangan *FTP* 2 detik dan serangan *ICMP* 5 detik.

Selanjutnya pada penelitian [9] yang dilakukan pada tahun 2022 dengan judul "Analisis Perbandingan Serangan *Hydra, Medusa,* dan *Ncrack* pada *Password Attack*". Penelitian ini membahas tentang serangan *brute force.* Dalam

penelitian ini penulis menguji serangan dengan menggunakan *tools hydra, medusa* dan *ncrack* kemudian membandingkan ketiga *tools* dari kecepatan dalam menebak dan menyerang kata sandi dan efektivitas waktu yang dibutuhkan dalam proses serangan berhasil. Penelitian ini menunjukan keberhasilan yaitu dengan berhasil menebak kata sandi korban dengan waktu yang sangat singkat, terdapat dua *tools* yang termasuk efektif untuk digunakan dalam melakukan *password attacking. Tools* tersebut merupakan *hydra* dan *medusa*.

Untuk penelitian [10] dilakukan pada tahun 2023 membahas tentang penerapan keamanan jaringan menggunakan *honeypot snare* dan *tunner*. Pada penelitian ini peneliti menggunakan metode *NDLC* dan melakukan pengujian dengan menggunakan tiga jenis serangan yaitu *XSS, Brute Force* dan *SQL Injection.* Hasil dari pengujian tersebut *honeypot* mampu melakukan deteksi penyerangan dalam bentuk catatan atau *log* yang kemudian disimpan dan dapat ditampilkan pada antar muka web.

2.2 DASAR TEORI

2.2.1 *Server*

Server merupakan sebuah sistem komputer yang menyediakan jenis layanan (*sevice)* tertentu dalam sebuah jaringan komputer. *Server* didukung dengan prosesor yang bersifat *scalable*, RAM dan juga dilengkapi dengan sistem operasi khusus atau yang biasanya disebut sebagai sistem operasi jaringan (*network operating system*). Selain itu, *server* dapat menjalankan perangkat lunak administrasi yang mengatur akses ke jaringan dan sumber daya yang terdapat di dalamnya, seperti halnya berkas dan alat pencetak (*printer*) serta memberikan akses kepada klien [11]. Tugas utama *server* adalah melayani komputer klien [12].

2.2.2 *Daniel* of *Service* (DoS)

Daniel of Service (DoS) merupakan salah satu serangan yang bertujuan untuk membuat komputer atau *server* yang ditarget serangan menjadi *down* atau mati dikarenakan penyerang membanjiri komputer tersebut dengan pesan-pesan sampah dalam jumlah besar dan dilakukan secara terus menerus [13]. Beberapa jenis serangan DoS sebagai berikut:

1. ICMP *Flooding*

Prinsip ICMP *Flooding* adalah dengan banjir ICMP membanjiri sumber daya target dengan paket *Echo Request* (ping) ICMP yang umumnya mengirim paket secepat mungkin tanpa menunggu balasan. Jenis serangan ini dapat menghabiskan *bandwidth* keluar dan masuk, karena *server* korban akan sering mencoba merespon dengan paket ICMP *Echo Reply*, mengakibatkan perlambatan sistem secara keseluruhan yang signifikan

2. UDP *Flooding*

UDP *Flooding* menurut definisi, adlah serangan DoS yang membanjiri target dengan paket *User Datagram Protokol* (UDP). Tjuan serangan ini adalah untuk membanjiri *port* secara acak pada *host* dengan jarak jauh. Hal ini menyebabkan *host* berulang kali memeriksa paket data pada *port* tersebut dan jika tidak ditentukan membalas dengan paket ICMP '*Destination Unreachable*'. Proses ini menghabiskan sumber daya *host* yang pada akhirnya dapat menyebabkannya tidak dapat diakses.

3. HTTP *Flooding*

Serangan HTTP *flooding* adalah serangan DoS dengan penyerangan yang mengekploitasi permintaan HTTP GET atau POST yang tampaknya sah untuk menyerang *server* web atau aplikasi. HTTP *flooding* tidak membutuhkan lebih sedikit *bandwidth* dari pada serangan lain untuk menjatuhkan situs atau *server* yang ditargetkan. Serangan tersebut paling efektif jika memaksa *server* atau aplikasi untuk mengalokasikan sumber daya semaksimal mungkin sebagai respon untuk setiap permintaan.

4. SYN *Flooding*

Serangan SYN *flooding* adalah serangan DDoS yang mengeksplotasi kelemahan yang diketahui dalam urutan koneksi TCP ("Tri *way handshake*"). Pertama adalah permintaan SYN untuk memulai koneksi TCP, dimana *host* menjawab dengan respon SYN-ACK, dan kemudian dikonfirmasi oleh respon ACK dari penyerang. Dalam skenario banjir SYN, penyerang mengirim beberapa permintaan SYN, tetapi tidak menaggapi respon SYN-ACK *host*, atau mengirim permintaan SYN dari Alamat IP *address* palsu. Bagaimanapun, ssistem *host* terus menunggu

pegakuan untuk setiap permintaan, mengikat sumber daya hingga tidak ada koneksi baru yang dapat dibuat, dan pada akhirnya mengakibatkan penolakan layanan.

2.2.3 *Brute Force*

Brute force adalah serangan yang berusaha untuk membobol *username* dan *password* dengan cara terus menerus mencoba memasukannya sampai berhasil masuk. Dalam hal ini *attacker* menggunakan kamus sandi sebagai perantara untuk mendapatkan *password* yang ingin di carinya. *Attacker* akan mencoba password satu per satu untuk di ontentikasi. Jika sandi tersebut terdapat *password* yang benar maka *attacker* akan berhasil membobol *server* tersebut. Semakin rumit *password* maka akan semakin sulit juga *attacker* membobol *password* tersebut dan semakin banyak juga percobaan yang dilakukan oleh *attaker* [14].

1. SSH *Brute Force*

   SSH merupakan protokol administrasi yang memungkinkan *user* untuk mengakses dan memodifikasi berbagai macam pengaturan *file* yang ada dalam *server*. SSH digunakan untuk memudahkan *user* dalam mengelola data dalam *server* tanpa harus menyentuh langsung di *server* fisiknya. SSH terbagi menjadi dua yaitu SSH *client* dan SSH *server*. SSH *client* adalah komputer yang digunakan untuk berbagi aktivitas seperti mengunduh *file* atau menghapus *file*. SSH *server* adalah komputer yang menyediakan layanan pengelola *file* nya untuk menyambungkan antara *client* dengan *server* diperlukan kunci yang sama untuk proses verivikasi. Ketika kunci yang dipakai SSH *client* sama dengan SSH server maka *client* tidak akan pernah tersambung. SSH *brute force* adalah serangan yang dilakukan terhadap port SSH. Ketika proses verifikasi maka *attacker* akan mencoba memasukkan kunci yang dianggap benar sampai sukses. Jika berhasil maka *attacker* dapat di anggap bahwa *attacker* adalah *client* asli.

2. TELNET *Brute Force*

   Telnet adalah suatu prokol client-server yang dapat memberikan akses remote (jarak jauh) untuk lIogin ke komputer tertentu dalam sebuah jaringan. Berbeda dengan SSH, Telnet tidak memiliki enkripsi dalam

sistem operasi sehingga secara keamanan. SShH jauh lebih aman di bandingkan dengan Telnet.

3. FTP *Brute Force*

   FTP merupakan kepanjangan dari *File Transfer Protocol*. FTP adalah protol yang berfungsi sebagai jembatan dalam pertukaran informasi antara *client* dengan *server* salam suatu jaringan. FTP berbagi menjadi dua yaitu FTP *client* dan FTP *server*. FTP *server* adalah server yang menjalankan *software* yang bertugas memberikan layanan tukar menukar fIile. Sedangkan FTP *client* digunakan untuk berbagi aktivitas pada saat *client* ingin mengunduh atau mengungah *file* maka *client* harus memasukkan *username* dan *password* untuk bisa mengakses data tersebut. Pada saat itulah *attacker* mencoba *password* dengan bantuan kamus sandi.

2.2.4 *Secure Shell* (SSH)

SSH merupakan protokol untuk *login* dari suatu komputer ke komputer lainnya dengan jarak jauh yang aman. Sehingga memungkinkan pengguna memiliki beberapa pilihan alternatif untuk otentikasi yang kuat serta melindungi komunikasi, keamanan dan integritas dengan menguatkan enkripsi. Dikatakan bahwa SSH dapat melakukan *login* ke komputer lain dalam suatu jaringan dan mengeksekusi perintah pada mesin secara *remote* dan juga memindahkan *file* dari satu mesin ke mesin lainnya. Layanan ini biasanya bekerja pada port 22 [15].

2.2.5 *Honeypot*

Honeypot merupakan sumber daya yang menyediakan layanan *server* untuk diselidiki, diserang atau dikompromikan. Honeypot menciptakan layanan, data, atau situs jaringan yang terlihat seperti bagian dari jaringan asli, tetapi sebenarnya mereka terisolasi dan diawasi. [7]. Terdapat 5 unsur atau elemen pada *honeypot* antara lain, *Monitoring* atau *logging tools* merupakan elemen untuk melakukan *monitoring* yang terjadi di dalam sistem. *Alerting mechanism* yaitu elemen untuk melakukan pemberitahuan Ketika sistem diserang. *Keystroke logger* yaitu elemen untuk melakukan pencatatan jejak aktivitas yang ada di sistem *honeypot*. *Packet analyzer* yaitu elemen untuk memberikan laporan tentang apa saja yang sudah terjadi dalam periode tertentu. *Forestic tool* yaitu elemen

pembantu dalam melakukan *forestic* [26]. *Honeypot* dapat diklasifikasikan berdasarkan tingkat interaksi yang dimiliki yaitu:

1. *Low-interacton Honeypot*

   Honeypot jenis ini memiliki tingkat interaksi *honeypot* rendah dan dirancang untuk mengemulasikan layanan tiruan yang mudah diinstal dan mirip dengan *server* yang asli.

2. *Medium-interraction Honeypot*

   Medium-interaction honeypot mempunyai kelebihan dibandingkan dengan *low-interaction* seperti memiliki kemampuan untuk berinteraksi lebih banyak dengan penyerang, jenis emulasi layanan *honeypot* ini dapat menambahkan berbagai macam fitur tambahan sehingga penyerang merasa seolah-olah mereka sedang berinteraksi dengan layanan yang sebenarnya.

3. *High-interaction Honeypot*

   Salah satu kelebihan dari *honeypot* ini adalah penyerang dapat berinteraksi secara langsung dengan *server*, sistem keamanan yang kompleks, yang siap untuk ditargetkan. Penyerang dapat mengakses sistem operasi secara penuh jika mereka memiliki akses penuh ke root *server* [16].

Adapun jenis *honeypot* yang digunakan dalam penelitian kali ini:

A. Kippo

   Kippo adalah salah satu jenis honeypot dengan tingkat *medium interaction* yang didesain menggunakan bahasa python untuk menyimpan informasi *brute force* dan informasi aktivitas penyusup di dalam *server* [20].

B. Dionaea

   Dionaea adalah salah satu jenis *honeypot* yang tidak banyak berinteraksi yang bertujuan untuk mengumpulkan salinan data dari *malware* dan juga memiliki kemampuan untuk mendeteksi serangan *Denial of Service*. *Dionaea* menggunakan python sebagai bahasa *scripting* dan libemu untuk mendeteksi *shellcodes*. Kemudian, dia akan mencoba mengunduh file dengan protokol tftp dan ftp, yang digunakan dalam python dengan format fttp.py dan ftp.py agar Dionaea dapat membacanya [6].

2.2.6 VirtualBox

Salah satu aplikasi yang berfokus pada virtualisasi adalah VirtualBox, yang tersedia secara *open source*. Teknologi yang dikenal sebagai virtualisasi memungkinkan Anda membuat komputer pribadi virtual yang dapat beroperasi secara mandiri tanpa sistem operasi. Semua perangkat keras yang terkait dengan mesin virtual disimulasikan oleh komputer *host*. Kemampuan ini sangat penting jika seseorang ingin menguji dan meniru instalasi sistem tanpa kehilangan sistem yang ada [17].

2.2.7 Linux

Linux adalah sistem operasi berbasis Unix yang tersedia secara bebas untuk umum dan diatur oleh GNU *General Public License* (GPL). Banyak distribusi Linux tersedia, masing-masing mengandung paket perangkat lunak yang dapat diinstal. Untuk memanfaatkan fitur-fitur baru, perbaikan *bug*, dan *patch* keamanan, sangat penting agar paket-paket ini tetap *up-to-date*. Linux adalah sistem operasi *server* dan melindungi akses data sehingga dapat menembus dan melindungi jaringan [18]. Berikut ini merupakan sistem operasi yang digunakan pada penelitian ini yaitu:

1. Kali linux

   Offensive Security membuat sistem operasi berbasis Debian untuk menggantikan distribusi linux perusahaan induknya, *BackTrack*. Pengujian penetrasi, penelitian keamanan, forensic komputer, dan rekayasa balik adalah contoh perangkat lunak pengujian penetrasi untuk komputer dan sistem operasi *opensource* yang dirancang untuk berbagi aktifitas keamanan informasi. Selain itu, tersedia berbagai alat kali linux yang dapat digunakan untuk uji coba penelitian keamanan [19].

2.2.8 *Network Mapper* (NMAP)

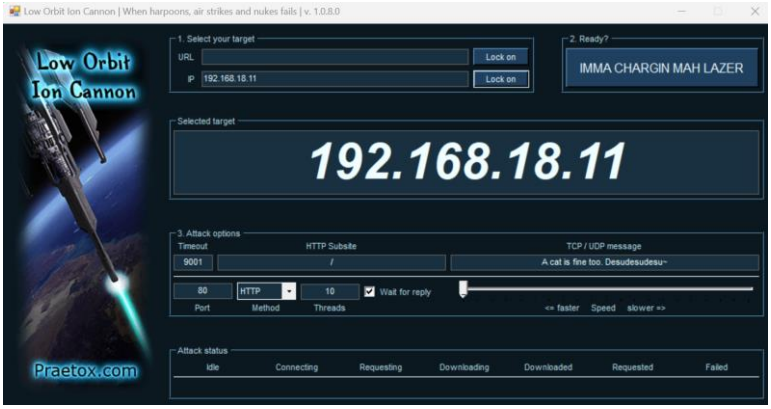
Nmap adalah aplikasi gratis dan *open-source* yang digunakan untuk menemukan dan mengaudit keamanan jaringan. Selain itu, nmap digunakan untuk melakukan tugas lainnya, seperti mengelola inventaris jaringan dan memantau host atau layanan yang aktif. Nmap menggunakan paket IP asli dengan cara baru untuk mengidentifikasi host yang tersedia di jaringan, layanan yang tersedia, sistem operasi yang dijalankan, jenis filter paket atau firewall yang digunakan,

dan fitur lainnya. Dirancang untuk memindai jaringan yang besar dengan cepat, Nmap juga berfungsi dengan baik terhadap host [21].

2.2.9 *Low Orbit Ion Cannon* (LOIC)

LOIC merupakan sebuah *tools* yang digunakan untuk melakukan uji coba pada jaringan dan sering disalahgunakan oleh *attacker* atau seorang yang tidak bertanggung jawab untuk melakukan serangan DoS maupun DDoS [10]. Ada 3 jenis serangan di dalam aplikasi LOIC yaitu:

1. *Serangan User Data Protocol (UDP)* adalah serangan DoS yang menyerang target melalui port secara acak menggunakan datagram UDP karena UDP tidak memiliki hubungan dan dapat dibuan dengan beragan jenis bahasa pemrograman [12].
2. *Transmission Control Protocol Synchronization* (TCP SYN) merupakan jenis serangan DDoS yang bertujuan untuk mencegah server melayani lalu lintas jaringan dan menghentikan akses ke sumber daya. Penyerang dapar membanjiri semua port yang tersedia pada server yang ditargetkan dengan mengirimkan paket permintaan koneksi SYN berulang kali, yang menyebabkan server target mengalami penundaan lalu lintas jaringan [22].
3. *hypertext Transfer Protocol* (HTTP) merupakan jenis serangan DDoS yang bertujuan untuk membanjiri target server dengan permintaan HTTP. Setelah target jenuh dengan menerima banyak permintaan, target tidak dapat menanggapi, menyebabkan lalu lintas jaringan yang tidak teratur atau bahkan hang [23]. Dari 3 serangan tersebut peneliti akan menggunakan serangan HTTP seperti pada gambar 2.1



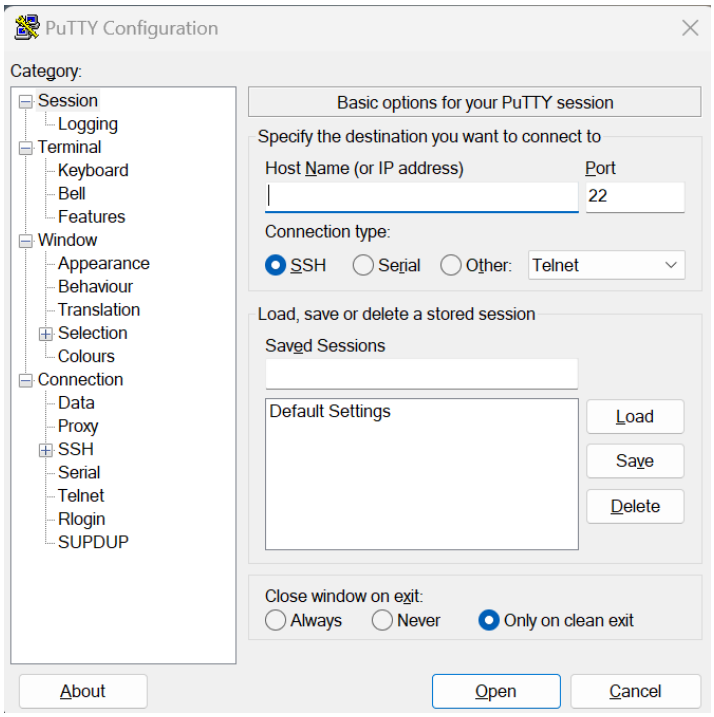
Gambar 2. 1 LOIC

2.2.10 Structured Query Language (SQL)

SQL Injection adalah kerentanan kemanan terjadi ketika penyerang memiliki kemampuan untuk mengubah *query Structured Query Language* (SQL) melalui *back-end database* aplikasi. Dengan menginfeksi melalui *database*, penyerang dapat mempengaruhi sytax dan kapabilitas SQL itu sendiri, termasuk kemampuan untuk menguasai dan mengoperasikan fleksibilitas *database* serta kemampuan sistem operasi yang menyimpan database. Aplikasi web tidak langsung terpengaruh oleh kerentanan keamanan SQL Injection setiap kode yang diterima sebagai masukkan dari sumber yang tidak dapat dipercaya dan digunakan sebagai masukkan pada form dinamis statement SQL dapat berpotensi berbahaya [24].

2.2.11 PuTTy

PuTTy adalah aplikasi yang memungkinkan koneksi antara *server* dan klien dengan SSH dan Telnet. Simon Tatham membuat PuTTY untuk platform Windows, yang merupakan perangkat lunak *opensource* yang dapat dikembangkan dan didukung oleh sekelompok sukarelawan. Berikut merupakan tampilan dari aplikasi PuTTY. Berikut merupakan tampilan PuTTY seperti gambar 2.2



Gambar 2. 2 PuTTY

2.2.12 Port (Jaringan Komputer)

Port adalah mekanisme yang mengizinkan sebuah komputer untuk mendukung beberapa sesi koneksi dengan komputer lainnya dan program di dalam jaringan. *Port* dapat mengidentifikasikan aplikasi dan layanan yang menggunakan koneksi di dalam jaringan TCP/IP. Sehingga, *port* juga mengidentifikasikan sebuah proses tertentu di mana sebuah *server* dapat memberikan sebuah layanan kepada klien atau bagaimana sebuah klien dapat mengakses sebuah layanan yang ada dalam *server*. *Port* dapat dikenali dengan angka 16-bit (dua *byte*) yang disebut dengan *Port Number* dan diklasifikasikan dengan jenis protokol *transport* apa yang digunakan, ke dalam *Port* TCP dan *Port* UDP. Karena memiliki angka 16-bit, maka total maksimum jumlah *port* untuk setiap protokol transport yang digunakan adalah 65536 buah.