BAB 2
DASAR TEORI

2.1. KAJIAN PUSTAKA

Penelitian dari Syaifuddin, Diah Risqiwati, Eko Ari Irawan pada tahun 2018 yang berjudul "*Realtime* Pencegahan Serangan *Brute force* dan DDoS Pada Ubuntu *Server*" meneliti tentang resiko terhadap keamanan dalam sistem jaringan [13]. Penelitian ini menggunakan *fail2ban* dan *fail2sql*. *Fail2ban* adalah aplikasi untuk mengamankan jaringan dengan cara memblokir alamat IP yang mencoba melanggar keamanan sistem. Sedangkan *fail2sql* berfungsi untuk mendapatkan *log* dari setiap *server* dan mengelola semua hasil *log* dan dikirim ke *database* hasil data. Selanjutnya *database* dalam semua paket *log* serangan akan dibuat dalam bentuk *highcharts* supaya proses analisa akan lebih mudah terbaca. *Server* yang terpasang menggunakan ubuntu *server* versi 14.04 dan beberapa aplikasi seperti *open*SSH dan apache. Sedangkan untuk *host* penyerang terpasang sistem operasi kali linux dan aplikasi *hydra*, *xerves*, *browser* dan *medusa*.

Percobaan pada penelitian ini menggunakan serangan *brute force* dan DDoS. Pada saat melakukan serangan *brute force* terhadap *server*, *fail2ban* yang ada di *server* akan di *enabled*. Serangan *brute force* dan DDoS akan menyerang SSH dan HTTP. Semua serangan yang akan dilakukan berasal dari *host attacker*. Hasil serangan dapat terbaca secara *realtime* dan tersimpan ke dalam *database* melalui program *fail2sql* sehingga *output* mudah di baca maupun di analisa seorang administrator. Pengujian pada penelitian ini ada 3 tahap, seperti pada tahap 1 *fail2ban* dalam keadaan tidak berjalan dan menggunakan *tool medusa* untuk melakukan serangan *brute force* dengan perintah "*Medusa* -h 10.0.2.15 -U /*root*/*Desktop*/*userlist* -p /*root*/*Desktop*/*pass.list* -M ssh". Sedangkan pada tahap 2 menggunakan *tool xerxes* untuk melakukan serangan DDoS dengan perintah "./*xerxes* 192.168.1.9 22". Sedangkan pada tahap 3 akan menyerang apache dengan serangan *brute force*. Setelah melakukan 3 uji coba didapatlah hasil uji coba dari semua serangan *brute force* dan DDoS yaitu ketika *fail2ban disable* maka semua

serangan akan berhasil menemukan *password* yang benar. Selanjutnya, ketika *fail*2*ban enable* maka semua serangan tidak akan berhasil menemukan *password* yang benar [13].

Penelitian dari Seda Yüksel pada tahun 2018 yang berjudul "*Analyzing the Medium-Interaction Honeypot*: *a Case Study*" meneliti tentang *log* data *kippo* yang diambil dalam 6 bulan periode pemantauan dalam infrastruktur jaringan di Turki [14]. *Honeypot* merupakan layanan palsu yang memberikan tanggapan logis untuk membantu mengambil informasi tentang seluruh interaksi *shell* penyerang. *Kippo* yaitu sebuah *honeypot* SSH berinteraksi menengah yang ditulis menggunakan *python* dan juga digunakan untuk mencatat serangan *brute force*. Untuk menggambarkan statik dari *file log* merupakan tugas dari *kippo graph* yang digunakan untuk menganalisis *log* secara *real time*. Hasil uji coba dari penelitian ini didapatkan sebanyak 37982 data yang berisikan *username* dan *password*. Untuk *username* unik yang berhasil tercatat ada 3831 dan 3831 untuk *password* unik. Upaya mencebak dengan kata kunci "*root*" sebagian besar banyak dicoba. Dengan demikian kata kunci "*root*" adalah kata kunci terlemah dari sebagian *username* dan *password* yang tercatat. *Username* dan *password* yang sering digunakan ialah "*root*/*root*" dan "*admin*/*admin*". Persentase kata kunci "*root*" yang digunakan sebesar 10,7 % dari sampel yang diuji, berikutnya kata kunci "*admin*" dan "*ubnt*". *Password* yang banyak digunakan ialah "123456" dengan persentase 2,6 %, selanjutnya diikuti *password* "*admin*" dengan persentase 2,2 %, "*ubnt*" dengan persentase 1,7 % dan "*root*" dengan persentase 1,7 %.

Penelitian dari Naufal Arkaan, Dolly Virgian Shaka Yudha Sakti pada tahun 2019 yang berjudul "Implementasi *Low Interaction Honeypot* Untuk Peningkatan Keamanan *Server* dan Analisa Serangan Pada Protokol SSH" meneliti tentang ancaman serangan terhadap *server* yang sedang berlangsung dan memantau dan menganalisis ancaman tersebut [6]. Untuk mengatasi ancaman serangan tersebut diperlukannya sebuah sistem umpan yang digunakan untuk menyamarkan sebagai sistem yang rentan yaitu sistem dari *honeypot*. *Honeypot* pada penelitian ini digunakan untuk memantau dan manganalisis kegiatan dari penyerang dan juga sebagai *server* tiruan untuk menirukan *server* asli. *Honeypot* yang digunakan pada penelitian ini berjenis *low interaction* dan memanfaatkan dua jenis *honeypot* yaitu

dionaea dan *honeyd*, kedua *honeypot* tersebut berjalan pada protokol SSH. Pada penelitian ini memindahkan protokol SSH asli ke *port* protokol lain yang hanya diketahui oleh administrator *server* saja. Penyerangan menggunakan protokol SSH masuk ke dalam *honeypot* sehingga penyerang mengira masuk ke protokol SSH asli.

Honeypot menghasilkan *log* aktivitas dari penyerang yang ketika berusaha *login* ke dalam SSH yang disediakan oleh *honeypot*. *Log* yang tercatat yaitu IP, *port*, negara asal, kota maupun kode pos milik penyerang. Setelah berhasil menyerang dan berhasil *login* ke *server*, *honeypot* akan mengirimkan email yang berupa pemberitahuan kepada administrator *server*. Dalam pengujian ini diperolehlah *log* dari serangan yang berisikan *username* dan *password* yang telah dicoba *login* ke *server*. Top 5 dari *username* terdiri dari *root*, *admin*, user, 1234, *test* dan *username* lainnya. Sedangkan untuk top 5 dari *password* terdiri dari *password*123, *admin*, 123, *password*, 1234 dan *password* lainnya. Untuk aktivitas yang berhasil terekam pada *log* didapatkan 21 kali aktivitas perintah yang dipakai oleh penyerang [6].

Pada penelitian dari Budi Jaya, Yuhandri Yunus, Sumijan pada tahun 2020 yang berjudul "Peningkatan Keamanan *Router* Mikrotik Terhadap Serangan *Denial of Service* (DoS)" meneliti tentang serangan DoS (*Denial of Service*) pada *router* Mikrotik [7]. Tujuan dari serang DoS yaitu membuat jaringan *router* mengalami *down* sehingga *router* tidak mampu melayani permintaan dari *user* yang memiliki hak akses yang sah. Penelitian ini melakukan simulasi dan analisis serangan DoS menggunakan metode *live forensics* dan meningkatkan keamanan *router* Mikrotik dari segi *hardware* dan *software*. Hasil dari penelitian ini diperoleh bukti digital dari serangan DOS yang berupa alamat IP dan *log* aktivitas dari penyerang. Menggunakan *firewall filter* dan *firewall* raw digunakan untuk meningkatkan keamanan *router* dari segi *software* sedangkan dari segi *hardware* yaitu dengan cara menonaktifkan tombol reset pada *router* dan menambahkan perangkat *hardware firewall* agar *router* bisa terhindar dari serangan fisik oleh orang yang tidak bertanggung jawab. Fungsi dari *router* yaitu dapat menyimpan identitas lalu lintas data berdasarkan tabel-tabel yang tersedia melalui *router*. *Router* sangat dibutuhkan untuk penyedia jasa internet dalam membangun sebuah jaringan maupun

keamanannya. Target dari *attacker* sebelum masuk ke sistem utama adalah mematikan kinerja dari *router*.

Serangan DoS dilakukan dengan mengubah data *size* yang dikirimkan ke target DoS sehingga *router* yang dilewatinya mengalami peningkatan beban kerja CPU dan peningkatan konsumsi daya listrik. Metode *live forensics* merupakan proses analisis forensik yang dilakukan ketika sistem jaringan komputer beroperasi. Forensik merupakan proses mencatat, mendeteksi, menangkap dan menganalisa aktivitas jaringan untuk menemukan bukti digital dari serangan yang dilakukan melalui jaringan komputer sehingga pelaku kejahatan dapat dituntut sesuai hukum yang berlaku. Hasil percobaan pada penelitian ini mendapatkan kondisi CPU *load* sebesar 1 % dan memori sebesar 7 MB dimana kondisi tersebut merupakan kondisi sebelum terjadinya serangan. Setelah dilakukan serangan kondisi CPU *load* naik menjadi 100 % dan memori naik sebesar 7,6 MB. Protokol yang diserang yaitu protokol TCP dan ICMP. *Port* protokol yang diserang yaitu *port* 443 dan *port* 53. Alamat IP penyerang di 200.200.200.20, untuk IP Mikrotik di 200.200.200.1, untuk IP administrator di 200.200.200.40, untuk IP *forensics* di 192.200.200.20, untuk IP *router* to ISP di 192.168.100.7 dan untuk IP *gateway* internet to ISP di 192.168.100.1. Setelah menggunakan *firewall* filter, CPU dan memori *router* turun menjadi CPU 46 % dan memori 6,5 MB. Setelah menggunakan *firewall* raw, koneksi akan terputus dengan alamat IP penyerang yang diblokir yaitu 200.200.200.20. Kesimpulan dari penelitian ini yaitu *firewall* filter berfungsi untuk menyaring packet data yang masuk pada jaringan *router* dan *firewall* raw berfungsi untuk memblokir alamat IP yang dicurigai mengirim packet data yang tidak wajar pada jaringan *router* [7].

Pada penelitian dari Molavi Arman, Nur Rachmat pada tahun 2020 yang berjudul "Implementasi Sistem Keamanan *Web server* Menggunakan *Pfsense*" meneliti tentang ancaman yang terjadi pada *web server* [15]. Untuk mengatasi ancaman tersebut maka diperlukan suatu sistem dalam menanggulangi dan mengatasi ancaman terhadap *web server*. Sistem yang mendeteksi gangguan tersebut diimplementasikan dengan menggunakan aplikasi *snort* pada *pfsense* dan *pfsense* tersebut bertugas sebagai sistem operasi *router* yang diletakkan berhadapan dengan internet. Metode yang digunakan untuk penelitian ini menggunakan

PPDIOO sebagai metode pengembangan dalam implementasi. *Pfsense* merupakan pendistribusian *firewall network* yang bebas, berdasarkan pada sistem operasi *free*BSD dengan *kernel* khusus dan termasuk paket perangkat lunak bebas dari pihak ketiga untuk fungsionalitas tambahan. Dalam penelitian ini menggunakan *pfsense* sebagai penghubung untuk mengamankan dan melindungi *web server*. PPDIOO (*Prepare Plan Design Implement Operate Optimize*) *network life cycle* merupakan metodologi pengembangan yang dibuat oleh Cisco. Pengujian dalam ini menggunakan aplikasi LOIC (*Low Orbit Ion Cannon*).

Hasil serangan dari komputer penyerang menggunakan aplikasi LOIC didapatkan PC *router pfsense* dan *snort* mendapatkan alamat IP dari komputer penyerang yaitu IP 192.168.0.120. IP tersebut mendeteksi melakukan *ping of death* terhadap alamat IP 192.168.1.1 sebagai komputer yang terpasang *snort psense* dan juga mendeteksi serangan TCP *flooding attack*. Selanjutnya PC *router pfsense* dan *snort* dicoba dengan penyerangan *slowloris attack* dan terdeteksi oleh *snort* sehingga mendapatkan *log* dari penyerang yang terdeteksi dari protokol HTTP dan juga alamat IP dari penyerang. *Snort* dapat mengenali jenis serangan pada *port* TCP dan serangan DDoS dengan jenis serangan *slowloris*. Setiap serangan akan disimpan di *log snort* dan *log* tersebut, *pfsense* dapat mengambil tindakan untuk melakukan pemblokiran dalam durasi yang telah ditetapkan [15].

2.2. DASAR TEORI

2.2.1. *Server*

Server adalah sebuah sistem komputer yang menyediakan jenis layanan (*service*) tertentu dalam sebuah jaringan komputer [8]. *Server* didukung dengan prosesor yang bersifat *scalable* dan RAM yang besar, juga dilengkapi dengan sistem operasi khusus, yang disebut sebagai sistem operasi jaringan (*network operating system*). *Server* juga menjalankan perangkat lunak administratif yang mengontrol akses terhadap jaringan dan sumber daya yang terdapat di dalamnya, seperti halnya berkas atau alat pencetak (*printer*) dan memberikan akses kepada *client* anggota jaringan [8]. Tugas utama dari *server* yaitu melayani komputer *client* [16].

2.2.1. DDoS (*Distributed Denial of Service*)

DDoS merupakan serangan DoS yang dilakukan secara serempak dengan jumlah komputer yang lebih banyak dan mempunyai target yang sama [11]. Sedangkan DoS adalah serangan yang bekerja dengan cara mengirimkan *request* ke *server* berulang kali untuk bertujuan membuat *server* menjadi sibuk menanggapi *request* dan *server* akan mengalami kerusakan atau *hang* [17]. Dampak serangan DDoS akan menyebabkan *bandwidth* yang digunakan akan habis yang mengakibatkan terputusnya koneksi antar *server*, bila serangan DDoS tidak segera ditanggulangi dapat menyebabkan kerusakan secara permanen terhadap *hardware* dan *software* korban. Contohnya adalah memori yang lebih besar dan *processor* yang lebih cepat. Sedangkan Peningkatan level serangan DoS dengan melakukan pengubahan pada data *size* yang dikirimkan ke target DoS menyebabkan *router* yang dilewatinya mengalami peningkatan konsumsi daya listrik dan beban kerja CPU [7]. Aplikasi yang digunakan untuk penelitian ini menggunakan LOIC (*Low Orbit Ion Cannon*).

2.2.2. *Brute Force*

Brute force merupakan teknik intensif komputasi yang menghasilkan serangkaian kata sandi dengan menggunakan kombinasi karakter dan kemudian digunakan untuk mencoba memecahkan kata sandi dari *server* [18]. *Brute force* merupakan serangan yang menggunakan algoritma untuk memecahkan masalah secara langsung, sederhana dan dengan cara yang jelas [13]. Pada serangan *brute force*, penyerang melakukan *login* dengan mengungkapkan *password login* dengan menggunakan protokol SSH dan Telnet. Protokol tersebut dapat memungkinkan pertukaran data antara dua perangkat jaringan seperti yang digunakan pada sistem berbasis Linux dan Unix [12]. Aplikasi yang digunakan untuk penelitian ini menggunakan Nmap (*Network mapper*).

2.2.3. SSH (*Secure Shell*)

SSH pertama kali dikembangkan oleh Tatu Yl nen di Helsinki University of Technology [19]. SSH (*Secure Shell*) adalah protokol untuk *login* dari satu komputer ke komputer lain dengan jarak jauh yang aman. Hal ini memungkinkan penggunaan memiliki beberapa pilihan alternatif untuk otentikasi yang kuat dan

melindungi komunikasi, keamanan dan integritas dengan cara menguatkan enkripsi [20]. Layanan SSH bekerja pada *port* 22. Dikatakan SSH adalah sebuah program untuk melakukan *login* terhadap komputer lain dalam suatu jaringan, serta mengeksekusi perintah-perintah lewat mesin secara *remote* dan juga memindahkan *file* dari satu mesin ke mesin lain [19].

2.2.4. *Firewall* NAT (*Network Address Translation*)

Firewall merupakan suatu sistem yang dibuat untuk mencegah akses yang tidak diinginkan dari atau ke dalam suatu jaringan internal [21]. Cara kerja dari *firewall* yaitu dengan melacak dan mengendalikan lalu lintas data serta memutuskan aksi untuk melewatkan (*pass*), menjatuhkan (*drop*), menolak (*reject*), mengenkripsi atau melakukan pencatatan aktivitas (*log*) data. *Firewall* menjamin agar data sesuai dengan aturan (*rule*) yang terdapat di dalam kebijakan keamanannya (*security policy*) yaitu seperangkat aturan yang telah didefinisikan di dalam keamanan jaringan internal [21]. NAT (*Network Address Translation*) merupakan sebuah *firewall* yang berfungsi untuk mengubah alamat IP pengirim seakan-akan menjadi alamat IP dari *router*. NAT biasanya digunakan untuk *router* yang berada diantara 2 jaringan yang berbeda. *Firewall* NAT dibagi menjadi SRCNAT (*source* NAT) dan DSTNAT (*destination* NAT) [22].

2.2.5. *Honeypot Cowrie*

Honeypot Cowrie adalah jenis *honeypot* dengan interaksi medium yang berjalan pada layanan SSH dan Telnet [24]. Pengertian dari *honeypot* merupakan sistem layanan palsu yang berfungsi untuk menjebak penyerang [6]. *Honeypot* dapat mendeteksi serangan dan mencatat serangan berupa alamat IP, *port* dan aksi apa yang dilakukan oleh penyerang [6].

Honeypot cowrie dapat mencatat serangan *brute force* dan interaksi *shell* penyerang. *Honeypot cowrie* berbasiskan *open source* yang dikembangkan oleh Michael Oosterhof [23]. *Cowrie* merupakan pengembangan dari *kippo* yang dikembangkan oleh Michael Oosterhof, *security researcher* berbasis di Dubai [24]. Apabila ada serangan masuk, penyerang akan memiliki akses ke *shell* Linux palsu dimana penyerang dapat menjalankan perintah dan menerima tanggapan seperti masuk ke *server* sesungguhnya. Penyerang tidak akan bisa menjalankan perintah

tersebut di luar ruang lingkup *honeypot* dikarenakan kerangka (*shell*) *cowrie* sebenarnya bukan kerangka (*shell*) Linux sama sekali. Ruang lingkup perintah pada *cowrie* diterapkan sepenuhnya dengan *phython* [25]. Untuk memasang *cowrie* dibutuhkan *software* tambahan seperti *python* 3.6+ dan *python-virtualenv* [26].

Seperti *honeypot* lainnya, *cowrie* melabuhi penyerang agar mengira mereka berada di *server*. *Cowrie* dapat mencatat dan menganalisis serangan yang terjadi. Hal ini memungkinkan administrator mendapatkan gambaran tentang jenis serangan yang terjadi, tingkat keberhasilan atau kegagalan, serta lokasi geografis alamat IP dari serangan yang berasal. *Cowrie* mampu menangkap informasi tentang penyerang seperti SSH *fingerprint* yang tidak sengaja terekspos [25]. Dalam *honeypot* memiliki beberapa tingkatan yang berdasarkan interaksinya terhadap penyusup, seperti *low interaction honeypot*, medium *interaction honeypot* dan *high interaction honeypot*.

2.2.5.1. *Low Interaction Honeypot*

Honeypot jenis ini adalah *honeypot* yang dibuat untuk menirukan layanan (*service*) seperti pada *server* asli. Penyerang dapat menyerang *server* dengan mengakses pada beberapa *port* yang aktif. Mudah dalam pemasangan dan konfigurasi merupakan kelebihan dari *honeypot* jenis ini namun informasi terjadinya serangan sangatlah sedikit [27].

2.2.5.2. *Medium Interaction Honeypot*

Honeypot jenis ini mempunyai kelebihan dibandingkan dengan *low interaction* seperti memiliki kemampunan lebih banyak berinteraksi dengan penyerang. Emulasi layanan *honeypot* jenis ini mampu menambahkan berbagai macam fitur tambahan sehingga penyerang seolah-olah sedang berinteraksi dengan layanan yang sebenarnya [27].

2.2.5.3. *High Interaction Honeypot*

Kelebihan dari *honeypot* jenis ini yaitu penyerang mampu berinteraksi secara langsung tanpa ada batasan dalam interaksi tersebut. Apabila penyerang sudah mengetahui akses penuh pada *root server* maka penyerang bisa berinteraksi secara penuh dengan sistem operasi [27].

2.2.6. Mikrotik *Router*OS

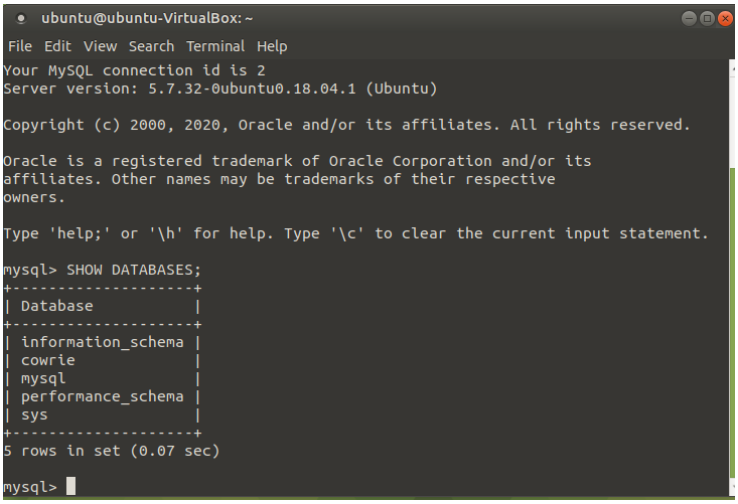
Mikrotik *Router*OS adalah sistem operasi dan perangkat lunak yang dapat digunakan untuk menjadikan komputer biasa menjadi *router network* yang handal dan mencakup berbagai fitur yang dibuat untuk IP *network* dan *wireless network* [28]. Dalam Mikrotik *Router*OS terdapat 2 tipe, yaitu dalam bentuk perangkat keras dan perangkat lunak. Dalam bentuk perangkat keras, Mikrotik *Router*OS biasanya sudah diinstalasi pada suatu *board* tertentu, sedangkan dalam bentuk perangkat lunak, Mikrotik *Router*OS merupakan satu *distro* Linux yang dikhususkan untuk fungsi *router* [28]. Untuk mengakses Mikrotik *Router*OS bisa dilakukan melalui aplikasi *winbox*. Mikrotik *Router*OS memiliki fitur seperti *firewall* dan NAT, *routing*, *hotspot*, *point to point tunneling protocol*, DNS *server*, DHCP *server*, manajemen *bandwidth* dan konfigurasi keamanan [29]. Penelitian ini menggunakan Mikrotik *Router*OS seri RB941. Gambar dari Mikrotik *Router*OS seri RB941 secara fisik ditunjukkan pada gambar 2.1.

Gambar 2.1 Mikrotik *Router*OS [30]

2.2.7. MySQL

MySQL adalah *database server* yang dikembangkan pada lingkungan *open source* dan didistribusikan secara gratis dibawah lisensi GPL. MySQL merupakan aplikasi sistem yang bersifat RDBMS (*Relational Database Management System*) [31]. RDBMS adalah sebuah program untuk membuat dan mengatur *database* dengan model yang saling berhubungan. MySQL tergolong aplikasi sistem *database* yang gratis dan dapat digunakan pada sistem operasi apapun [31].

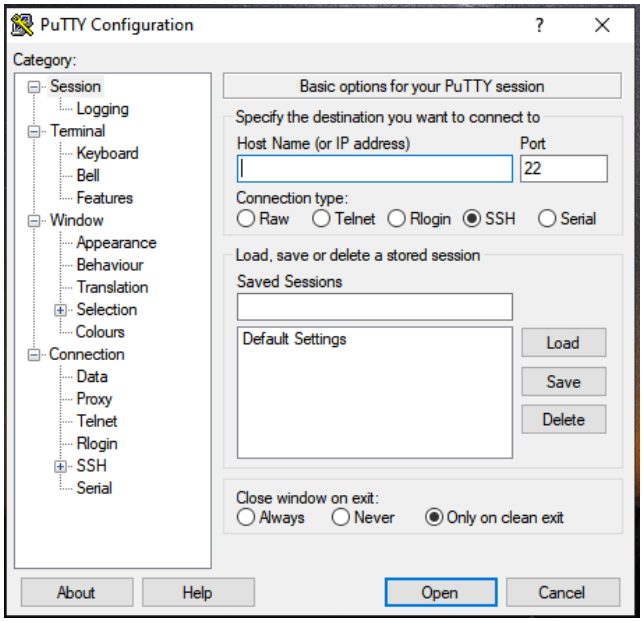
MySQL dalam penelitian ini digunakan untuk menghubungkan *log* serangan dari *honeypot cowrie* ke *kippo graph*. Tampilan dari MySQL ditunjukkan pada gambar 2.2.



Gambar 2.2 MySQL

2.2.8. PuTTY

PuTTY adalah sebuah aplikasi koneksi antara *server* dan *client* menggunakan SSH dan Telnet. PuTTY dikembangkan oleh Simon Tatham untuk *platform* Windows [32]. PuTTY merupakan perangkat lunak *open source* yang disediakan dengan kode sumber dan dapat dikembangkan serta didukung oleh sekelompok sukarelawan [32]. Tampilan dari aplikasi PuTTY ditunjukkan pada gambar 2.3.



Gambar 2.3 Aplikasi PuTTY

2.2.9. *VirtualBox*

Virtualbox merupakan sebuah aplikasi visualisasi berbasiskan *open source* yang dimana penggunanya dapat menjalankan lebih dari satu sistem operasi dalam satu mesin *virtual* saat menjalankan Windows ataupun Linux [33]. Aplikasi ini dirancang dengan mudah menggunakan antarmuka dan langkah demi langkah yang sangat memukau yang memungkinkan pengguna untuk membuat VM (*Virtual Mechine*) pertama dalam hitungan menit [33].

2.2.10. Ubuntu

Ubuntu merupakan sistem operasi turunan dari distro Linux Debian *unstable* [33]. Ubuntu memiliki prinsip selamanya bersifat gratis dan *open source*, prinsip tersebut merupakan prinsip dari suatu *project* untuk komunitas yang bertujuan untuk menciptakan sebuah sistem operasi dan paket aplikasi yang bersifat gratis dan *open source* [33]. Ubuntu memiliki berbagai kelebihan diantaranya adalah :

1) Pemaketan (*Packaging*)
2) Pemilihan aplikasi yang luas (*Application Choice*)
3) Siklus pembaharuan dilakukan secara rutin (*Updates*)
4) Dikenal stabilitas dan kualitasnya terutama di sisi *server* (*Stability and Quality*) [33].

2.2.11. *Kippo Graph*

Kippo Graph merupakan fitur dari *honeypot kippo* yang menampilkan statistik serangan dalam sebuah *web server* [34]. *Kippo graph* dapat menampilkan 24 jenis grafik termasuk 10 kata sandi penyerang, 10 kombinasi nama pengguna, perbandingan keberhasilan masuk *honeypot*, koneksi per IP (*Internet Protocol*), koneksi per negara, *probe* per hari, *probe* per minggu, SSH *client* dan masih banyak lagi [34]. *Kippo graph* adalah layanan yang menghasilkan grafik dan tabel dari *database* kippo yang ditampilkan pada halaman *web* [35]. Untuk menampilkan hasil *log* serangan dari *honeypot cowrie* yang berupa grafik diperlukan sebuah instalasi *cowrie* dan MySQL [36].

2.2.12. Nmap (*Network Mapper*)

Nmap (*Nerwork Mapper*) adalah aplikasi dengan lisensi gratis dan *open source* untuk *network discovery* dan *security auditing* [37]. Nmap digunakan untuk melakukan tugas-tugas seperti *network inventory*, mengelola jadwal peningkatan layanan dan memantau *host* atau layanan yang aktif. Nmap menggunakan *raw packet* IP dengan cara baru untuk menentukan *host* yang tersedia di jaringan, layanan yang tersedia, sistem operasi yang dijalankan, jenis filter paket atau *firewall* yang digunakan dan karakteristik lainnya. Nmap dirancang untuk memindai jaringan besar secara cepat tetapi memiliki fungsi yang baik terhadap *host*. Nmap berjalan di semua sistem operasi seperti Linux, Windows dan Mac OS X [37].

2.2.13. LOIC (*Low Orbit Ion Cannon*)

LOIC merupakan sebuah *tool* yang digunakan untuk melakukan uji coba pada suatu jaringan dan sering disalahgunakan oleh seseorang yang tidak bertanggung jawab untuk melakukan serangan DoS maupun DDoS [11]. Dalam aplikasi ini terdapat 3 jenis serangan seperti UDP *flood*, TCP *flood* dan HTTP *flood*.

2.2.13.1. UDP (*User Data Protocol*) *Flood*

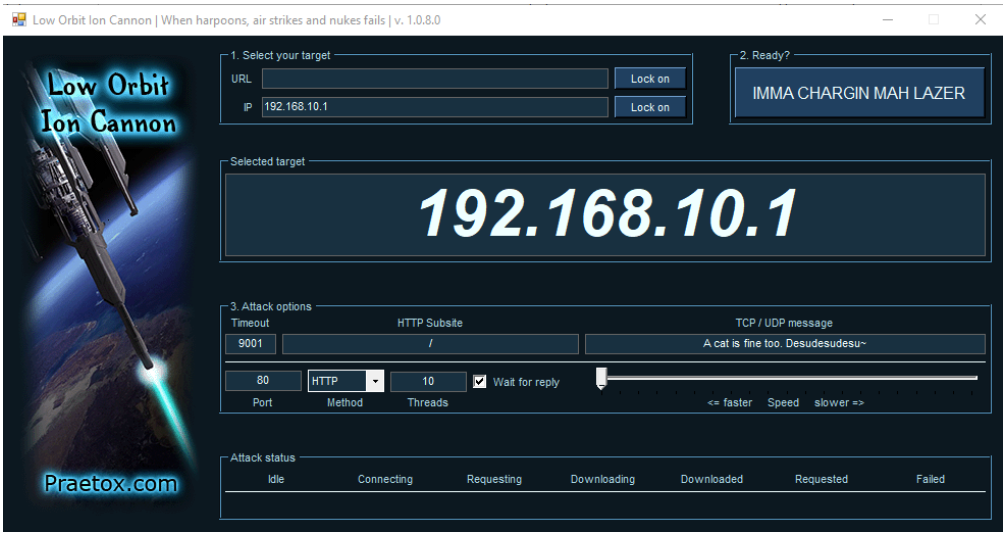
UDP *flood* merupakan serangan DoS dengan menyerang target melalui *Port* secara acak menggunakan UDP datagram karena UDP memiliki sifat *connectionless* dan mudah dibuat menggunakan beragam jenis bahasa pemrograman [11].

2.2.13.2. TCP SYN (*Transmission Control Protocol Synchronization*) *Flood*

SYN *flood* merupakan jenis serangan DDoS (*Distributed Denial of Service*) yang bertujuan untuk membuat *server* tidak bisa melayani lalu lintas jaringan dengan cara membuat pemakaian *resources* menjadi habis atau menjadi tidak tersedia. Dengan berulang kali mengirimkan paket permintaan koneksi (SYN), penyerang dapat membanjiri semua *port* yang tersedia pada *server* yang ditargetkan dan menyebabkan *server* target akan mengalami lambat pada lalu lintas jaringan. SYN *flood* bekerja dengan mengekspoloitasi proses *handshake* dari koneksi TCP [38].

2.2.13.3. HTTP (*Hypertext Transfer Protocol*) *Flood*

HTTP *flood* merupakan jenis serangan dari DDoS (*Distributed Denial of Service*) yang dirancang untuk membanjiri target yaitu *server* dengan permintaan HTTP. Setelah target jenuh dengan menerima banyak permintaan dan target tidak dapat menanggapi permintaan tersebut sehingga lalu lintas jaringan berjalan tidak normal [39]. Dari ketiga pilihan serangan diatas, untuk penelitian ini menggunakan serangan HTTP dengan memilih jenis serangan HTTP *flood* seperti pada gambar 2.4.



Gambar 2.4 Aplikasi LOIC