BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Pada penelitian yang akan dilakukan oleh peneliti berkaitan dengan pemanfaatan metode dan *tools* yang nantinya akan digunakan untuk menganalisa sebuah keamanan jaringan. Adapun perbedaan dari segi masalah, metode, *tools* serta jenis serangan *cyber* yang digunakan. Berikut beberapa penelitian sebelumnya yang berkaitan dengan penelitian yang akan diangkat oleh peneliti.

Dalam penelitian yang dilakukan Yudhi Arta, dkk., pada tahun 2018 mengenai Simulasi Implementasi *Intrusion Prevention System* (IPS) Pada *Router Mikrotik*. Pada masalah yang diangkat pada penelitian ini adalah melakukan simulasi dan pengujian dengan melakukan serangan dari *attacker* untuk menguji sistem *Intrusion Prevention System* (IPS) pada *Router Mikrotik*. Penelitian ini menggunakan dua *tools* yaitu *Zenmap* dan *Brutus* untuk melakukan serangan. Hasil dari penelitian ini adalah sebuah hasil yang menggunakan dua *tools* dalam pengujian serangan *Port Scanning* dan *Bruteforce* yang dilakukan, serangan *Port Scanning* menggunakan *Zenmap* memperlihatkan bahwa *client* 1 telah melakukan P*ort Scanning* namun tidak mampu mendeteksi *port* 21 karena telah diterapkan IPS pada *server Mikrotik*. Sedangkan serangan *Brute force* menggunakan *Brutus* memperlihatkan proses penyerangan dan pembacaan *username* dan *password*, IPS mampu mencegah serangan yang terjadi dan menampilkan pesan "*Unable to verify* 10.10.10.2, *check connection setting/time out"* yang menyatakan bahwa intruder mengalami *timeout* [9].

Pada penelitian yang dilakukan Amarudin dan Faruk Ulum pada tahun 2018 tentang Desain keamanan jaringan pada *Mikrotik Router* OS menggunakan metode *Port Knocking*. Pengujian penelitian ini dilakukan dengan dua tahap, berdasarkan hasil pengujian pertama yang didapatkan bahwa *admin Router* tidak bisa diakses dari PC1 karena PC1 hanya *PING Request* ke PC2, maka *Admin Router* tetap tidak bisa diakses. Adapun pengujian kedua dilakukan dengan cara mengakses admin *Router* dari PC1 dengan cara *PING Request* ke PC3 terlebih dahulu baru kemudian

bisa *login* ke *admin Router*. Dengan demikian *admin Router* hanya bisa diakses dari PC1 jika PC1 telah melakukan *PING Request* ke PC3 terlebih dahulu [5].

Dalam penelitian yang dilakukan oleh Adesty, dkk., pada tahun 2020 mengenai penerapan *Intrusion Prevention System* (IPS) *Suricata* sebagai pengamanan dari *Distributed Denial of Service Attack* (DDoS). Pada masalah yang diangkat pada penelitian ini adalah melakukan pengujian dengan melakukan serangan dari *attacker* untuk menguji sistem *Intrusion Prevention System* (IPS) *Suricata* dan *firewall IPTables*. Metode yang digunakan adalah melakukan serangan terhadap *website* yang sudah di konfigurasikan. Penelitian ini menggunakan dua *tools* yaitu *HPING3* dan *Loic* untuk melakukan serangan *request flooding*. Hasil dari penelitian ini adalah sebuah hasil yang menggunakan dua *tools* dalam pengujian serangan DDoS yang dilakukan, serangan DDoS berhasil membuat *web* uji coba menjadi tidak dapat diakses, dan *tools* IPS *Suricata* dengan fitur *inline* dan *firewall IPTables* berhasil dan mampu menolak akses serangan DDoS. Sedangkan pada penelitian yang akan dilakukan penulis menggunakan *tools Slowloris* dan *Zenmap* sebagai penyerang DDoS dan *Port Scanning* serta *Snort* sebagai pengaman dari serangan DDoS [3].

Dalam penelitian yang dilakukan oleh Farid Wahyudi dan Listanto Tri Utomo pada tahun 2021 tentang Perancangan *Security network Intrusion Prevention System* Pada PDTI Universitas Islam Raden Rahmat Malang. Pada masalah yang diangkat pada penelitian ini adalah melakukan pengujian dengan merancang sistem IPS menggunakan *Snort Mode Inline*. Metode yang digunakan untuk penelitian ini adalah dengan melakukan pengujian terhadap serangan DDoS. Hasil dari penelitian ini memperlihatkan serangan DDoS *flooding* menggunakan *LOIC* terhadap *server* IPS dan sistem IPS mampu mendeteksi adanya serangan DDoS [10].

Pada penelitian yang dilakukan Ahmad Zafrullah Mardiansyah, dkk., pada tahun 2021 tentang Optimasi *Port Knocking* dan *Honeypot* menggunakan *IPTables* Sebagai Keamanan Jaringan Pada *Server*. Metode yang digunakan dalam penelitian ini menggunakan *Port Knocking* dan *Honeypot*. Penelitian ini menggunakan *tools LOIC* dan *Hydra* untuk melakukan serangan DoS dan *Bruteforce*. Hasil dari

penelitian ini memperlihatkan serangan dari *bruteforce* dapat diantisipasi oleh *Honeypot* sedangkan penyerangan DoS menggunakan *LOIC* tidak dapat diantisipasi oleh *server* sehingga untuk mengatasi serangan DoS dibuat *rules IPTables* yang berfungsi untuk memblokir dan mengijinkan akses masuk atau keluar *server* [11].

Tabel 2.1 Kajian Pustaka

No.	Peneliti	Judul	Metode	Kelebihan	Kelemahan
1.	Yudi Arta, Abdul Syukur, Roni Kharisma (2018)	Simulasi Implementasi *Intrusion Prevention System* (IPS) Pada *Router Mikrotik*	Pengujian menggunakan *Brutus* dan *Zenmap*	Serangan berhasil dilakukan, dengan dibuktikan pada serangan brute force dalam proses penyerangan dan pembacaan *username* dan *password*, IPS mampu mencegah serangan yang terjadi dan pada *port scan*ning tidak mampu mendeteksi *port* yang telah diterapkan IPS pada *server Mikrotik*.	- Menggunakan *Routerboard* versi lama - Terbatasnya *rules* serangan agar tidak hanya terbatas pada serangan brute force, contohnya serangan Dos
2.	Amarudin, Faruk Ulum (2018)	Desain keamanan jaringan pada *Mikrotik Router* OS menggunakan metode *Port Knocking*	Menganalisa proteksi sebuah *port*	Mampu menjaga *Router* dari akses orang lain yang tidak berhak mengaksesnya	Sistem keamanan jaringan yang sangat simple sehingga lebih mudah untuk diserang
3.	Istiana Adesty, Wahyu Adi Prabowo, Muhammad	Penerapan *Intrusion Prevention System* (IPS) Suricata Sebagai Pengamanan Dari Serangan Distributed	Pengujian menggunakan HPING3 dan Loic.	Serangan berhasil dilakukan, dengan dibuktikan *web* yang di uji tidak dapat diakses	Kurang notifikasi jika serangan terdeteksi

No.	Peneliti	Judul	Metode	Kelebihan	Kelemahan
	Fajar Sidiq (2020)	Denial of *Service* (DDoS)		dan mampu memblokir serangan DDoS	
4.	Farid Wahyudi, Listanto Tri Utama (2021)	Perancangan *Security network Intrusion Prevention System* Pada PDTI Universitas Islam Raden Rahmat Malang	Merancang sebuah sistem keamanan jaringan menggunakan IPS	- Mampu mendeteksi serangan yang terjadi dalam jaringan - Mampu memonitoring jaringan ketika terjadi sebuah *intruder*	Hanya melakukan serangan tanpa memberikan solusi untuk keamanan yang lebih lanjut
5.	Ahmad Zafrullah Mardiansyah, Yayank Muhammad Abdussyakur, Andy Hidayat Jatmika (2021)	Optimasi *Port Knocking* dan *Honeypot* menggunakan IPTables sebagai keamanan jaringan pada *server*	Metode keamanan jaringan yang diterapkan adalah *Port Knocking*, *Honeypot* dan IPTables	Mampu mencegah serangan DoS dan Bruteforce	- Kurang basis data yang dapat memberikan respon terhadap penyerangan pada *Honeypot* - Aplikasi atau *script* tidak dapat berjalan di background *server*

2.2 Dasar Teori

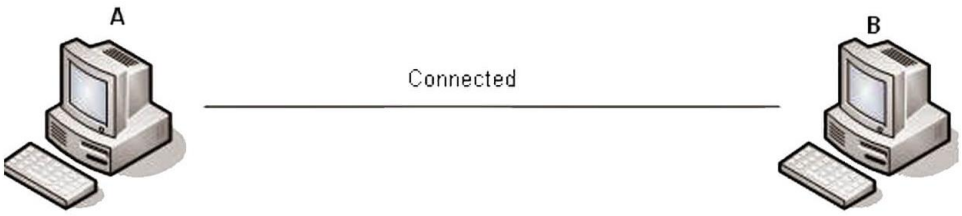
2.2.1 Jaringan Komputer

Jaringan komputer adalah himpunan interkoneksi antara dua komputer autonomous atau lebih yang terhubung dengan media transmisi kabel atau tanpa kabel (*wireless*). Bila sebuah komputer dapat melakukan kontrol terhadap komputer yang lainnya, maka komputer-komputer tersebut bukan autonomous (tidak melakukan kontrol terhadap komputer lain dengan akses penuh). Dua unit komputer dikatakan terkoneksi apabila keduanya bisa saling bertukar data/informasi, berbagi *resource* yang dimiliki, seperti *file*, *printer*, media penyimpanan (*hard disk, floppy disk, cd-rom, flash disk*). Data yang berupa *teks*, *audio*, maupun *video* bergerak melalui media kabel atau tanpa kabel sehingga memungkinkan pengguna komputer dalam jaringan komputer dapat saling bertukar *file*/data dan menggunakan *Hardware*/*Software* yang terhubung dalam jaringan secara bersama-sama.

Tiap komputer, *printer* atau peripheral yang terhubung dalam jaringan disebut dengan *node*. Sebuah jaringan komputer terdiri dari sekurang-kurangnya dua unit komputer atau lebih, dapat berjumlah puluhan komputer, ribuan, bahkan jutaan *node* yang saling terhubung satu sama lain [12]. Di dalam jaringan komputer dikenal sistem koneksi antar *node*, yaitu [12] :

1. *Peer-to-Peer*

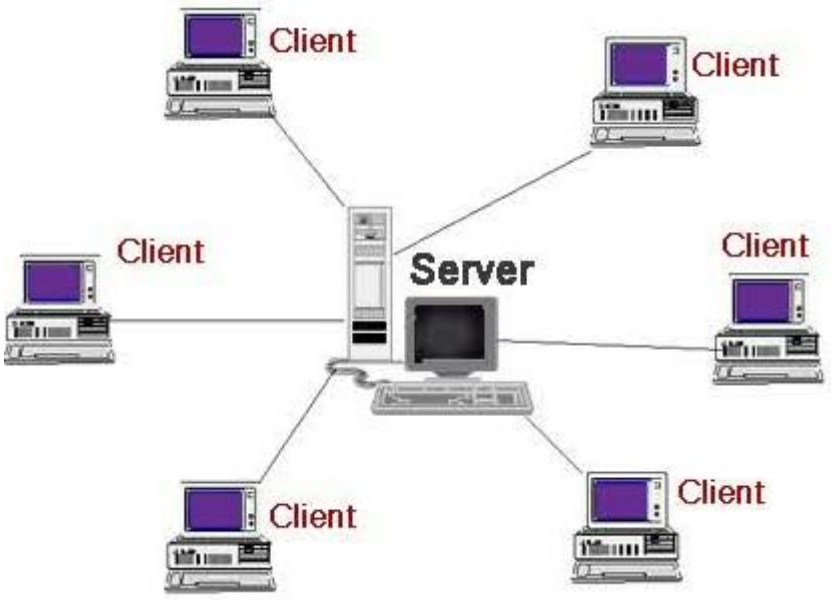
   Peer-to-peer network adalah jaringan komputer yang terdiri dari beberapa komputer (biasanya tidak lebih dari 10 komputer dengan 1-2 *printer*). *Peer-to-peer* adalah suatu model dimana setiap PC dapat memakai *resource* pada PC lain atau memberikan *resource* nya untuk dipakai PC lain. Dengan kata lain dapat berfungsi sebagai *client* maupun *server* pada periode yang sama.



Gambar 2.1 Peer-to-peer

2. *Client-Server*

   Selain pada jaringan lokal, sistem *Client-Server* bisa diterapkan dengan teknologi internet dimana ada suatu unit komputer yang berfungsi sebagai *server* yang hanya memberikan layanan bagi komputer lain, dan *client* yang juga hanya meminta layanan dari *server*. Akses dilakukan secara transparan dari *client* dengan melakukan *login* terlebih dahulu ke *server* yang dituju. *client* hanya bisa menggunakan *resource* yang disediakan *server* sesuai dengan otoritas yang diberikan oleh *administrator*. Aplikasi yang dijalankan pada sisi *client* bisa jadi merupakan *resource* yang tersedia di *server* atau aplikasi yang di-*install* di sisi *client* namun hanya bisa dijalankan setelah terkoneksi ke *server* [12].



Gambar 2.2 *Client Server*

Secara umum, jaringan komputer terbagi menjadi 3 jenis, yaitu [12] :

1. *Local Area Network* (LAN) adalah jaringan komputer yang dibatasi oleh area yang relatif kecil, umumnya dibatasi oleh area lingkungan, seperti sebuah kantor pada sebuah gedung. Jarak antar *node* tidak lebih jauh dari sekitar 200 m.
2. *Metropolitan Area Network* (MAN) meliputi area yang lebih besar LAN, misalnya antar gedung dalam suatu daerah (wilayah provinsi atau negara bagian). Dalam hal ini, jaringan menghubungkan beberapa buah jaringan kecil ke dalam lingkungan area yang lebih besar.
3. *Wide Area Network* (WAN) adalah jaringan yang jangkauannya lebih luas dari jaringan LAN dan WAN. Jaringan WAN biasanya menggunakan media *wireless*, sarana satelit, ataupun kabel serat optik. Bukan hanya meliputi satu kota atau antar kota dalam satu wilayah, tetapi menjangkau area atau wilayah otoritas negara lain.

2.2.2 Keamanan Jaringan

Keamanan jaringan atau *network security* dalam jaringan komputer 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 komputer yang benar-benar aman. Sifat dari jaringan melakukan komunikasi. Setiap komunikasi dapat jatuh ketangan orang lain dan disalahgunakan, sistem keamanan mampu mengamankan jaringan tanpa menghalangi penggunaan dan menempatkan antisipasi ketika jaringan berhasil ditembus. Pastikan bahwa *user* dalam jaringan memiliki pengetahuan yang cukup mengenai keamanan dan pastikan bahwa mereka menerima dan memahami rencana keamanan yang di buat [13]. Ada dua elemen utama pembentuk keamanan jaringan [13]:

1. Tembok pengaman, baik secara fisik maupun maya, yang ditaruh di antara 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 [13]:

1. Keamanan yang bersifat fisik (*physical security*) termasuk 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. *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).
    c. *Syn Flood Attack*, dimana sistem (*host*) yang dituju dibanjiri oleh permintaan sehingga dia menjadi terlalu sibuk dan bahkan dapat berakibat macetnya sistem.
2. Keamanan yang berhubungan dengan orang (personel), contoh:
    a. Identifikasi *user* (*username* dan *password*)
    b. Profil resiko dari orang yang mempunyai akses (pemakai dan pengelola).
3. Keamanan 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 *Availability*. Tiga aspek itulah yang dianggap sebagai tiga komponen *Cyber Security* paling penting di seluruh *platform*, terutama pada *Web App*.

Gambar 2.3 CIA TRIAD

Berikut ini penjelasan lengkap tentang 3 aspek tersebut [14] :

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, konsistensi, akurasi, dan validitas data tersebut masih terjaga. Artinya, *integrity* mencoba memastikan data yang disimpan benar adanya, tidak ada pengguna yang tidak berkepentingan atau *Software* berbahaya yang mengubahnya.

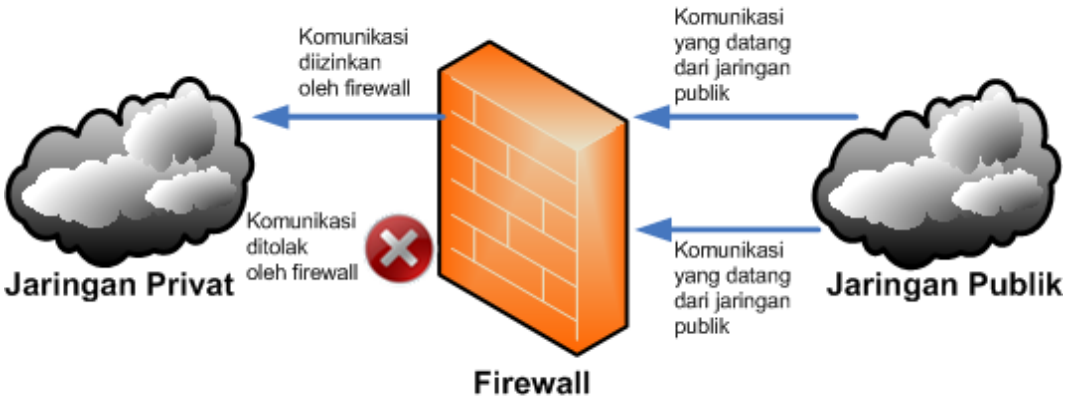
3. *Availability*

   Availability adalah memastikan sumber daya yang ada siap diakses kapan pun oleh *user*/*application*/sistem yang membutuhkannya. Sama seperti

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

2.2.3 *Firewall*

Firewall merupakan suatu piranti atau *Software* yang dirancang spesifik untuk melindungi komputer dan jaringan dari berbagai macam aktivitas jaringan yang berbahaya atau merusak [15]. Agar dapat berfungsi dengan efektif, sebuah *firewall* jaringan harus memenuhi standar-standar tertentu dan mampu menjalankan berbagai skema pencegahan dasar. Gagasan utama penerapan teknologi *firewall* adalah mendirikan suatu bentuk pagar pengaman di sekeliling sebuah jaringan privat, untuk mencegah akses tanpa izin dan berbagai gangguan terhadap aktivitas data sebuah jaringan [16].



Gambar 2.4 Ilustrasi *Firewall*

Beberapa kemampuan utama *firewall* adalah sebagai berikut [15]:

1. Membatasi jenis-jenis trafik tertentu yang keluar masuk dalam jaringan, misalnya memblok usaha pembacaan *file-file* dalam *hard disk* oleh *user* yang tidak sah atau memblok trafik berdasarkan nomor *port* atau jenis protokol koneksinya (*TCP/UDP*).
2. Mencegah komputer-komputer terhubung langsung ke internet atau jaringan publik yang tidak diizinkan.

3. Membantu mencegah penyerang, penyusup, *hacker* atau *Software-Software* perusak seperti *worm* untuk memperoleh akses ke komputer melalui jaringan atau internet.
4. Mencegah komputer mengirim *Software-Software* berbahaya ke komputer lain.
5. Secara aktif mencatat aktivitas jaringan yang berguna untuk penelusuran, misalnya saat terjadi usaha penyusupan atau penyerangan ke komputer.

2.2.4 *Cyber Attack*

Serangan terhadap keamanan sistem informasi (*security attack*) dewasa ini seringkali terjadi. Kejahatan komputer (*cyber crime)* pada dunia maya seringkali dilakukan oleh sekelompok orang yang ingin menembus suatu keamanan sebuah sistem. Aktivitas ini bertujuan untuk mencari, mendapatkan, mengubah, dan bahkan menghapus informasi yang ada pada sistem tersebut jika memang benar-benar dibutuhkan. Ada beberapa kemungkinan tipe dari serangan yang dilakukan oleh penyerang yaitu :

1. *Interception* yaitu pihak yang tidak mempunyai wewenang telah berhasil mendapatkan hak akses informasi
2. *Interruption* yaitu penyerang telah dapat menguasai sistem, tetapi tidak keseluruhan. *Admin* asli masih bisa *login*.
3. *Fabrication* yaitu penyerang telah menyisipkan objek palsu ke dalam sistem target
4. *Modification* yaitu penyerang telah merusak sistem dan telah mengubah secara keseluruhan [15].

2.2.5 *Denial Of Service* (DoS)

Denial of Service (DoS) adalah jenis serangan terhadap sebuah komputer atau *server* di dalam jaringan internet dengan cara menghabiskan sumber (*resource*) yang dimiliki. Bentuk umum dari serangan DoS ini adalah dengan cara mengirim paket data dalam jumlah yang sangat besar terhadap suatu *server* dimana *server* tersebut tidak bisa memproses semuanya. Bentuk lain dari serangan DoS ini adalah

memanfaatkan *port-port* yang rentan dari sistem operasi. Tidak semua DoS merupakan akibat dari serangan keamanan jaringan. Kesalahan dalam *coding* suatu program juga bisa mengakibatkan kondisi seperti serangan DoS. Ada beberapa jenis dari DoS, antara lain :

1. *Distributed Denial* of *Service* (DDoS)

   Terjadi saat penyerang berhasil menggabungkan beberapa layanan *system* dan menggunakannya sebagai pusat untuk menyebarkan serangan terhadap korban.

2. *Distributed Reflective Denial* of *Service* (DRDoS)

   Memanfaatkan operasi normal dari layanan internet seperti *protocol protocol update* DNS dan *Router*. DRDoS ini menyerang fungsi dengan mengirim *update* dalam jumlah yang sangat besar kepada berbagai macam layanan *server* atau *Router* dengan menggunakan *address spoofing* kepada target korban.

3. *SYN flooding*

   Upaya untuk membanjiri sinyal SYN kepada sistem yang menggunakan *protocol TCP/IP* dalam melakukan inisiasi sesi komunikasi.

4. *Smurf Attack*

   Server digunakan untuk membanjiri korban dengan data sampah yang tidak berguna. *Server* atau jaringan yang dipakai menghasilkan respon paket yang banyak seperti *ICMP ECHO* paket atau *UDP* paket dari satu paket yang dikirim.

5. *Ping of Death*

   Dengan menggunakan *tool* khusus, penyerang dapat mengirimkan paket *ping* yang *oversize* yang banyak kepada korban. *Ping of death* tidak lebih dari semacam serangan *buffer overflow*. Serangan ini dapat menyebabkan *crash* sistem, *freeze* atau *reboot.*

6. *Stream Attack*

   Serangan ini terjadi saat banyak jumlah paket yang besar dikirim menuju ke *port* pada sistem korban menggunakan sumber nomor yang random [7].

2.2.6 *Port Knocking*

Port Knocking adalah metode yang dilakukan untuk membuka akses ke *port* tertentu yang telah di*block* oleh *Firewall* pada perangkat jaringan dengan cara mengirimkan paket atau koneksi tertentu. Koneksi bisa berupa *protocol ICMP*, *TCP*, dan *UDP*. Jika koneksi yang dikirimkan oleh *host* tersebut sudah sesuai dengan *rule* autentikasi yang diterapkan, maka secara dinamis *firewall* akan memberikan akses ke *port* yang sudah di*block* [17].

Port Knocking bekerja seperti halnya brankas dengan kunci kombinasi angka putar. Pada brankas tersebut, diharuskan memutar lapisan-lapisan kunci kombinasi beberapa kali hingga tepat seperti yang ditentukan. Sebenarnya memutar lapisan-lapisan di dalam brankas. Dalam lapisan-lapisan kunci tersebut terdapat sebuah lubang kunci. Jika sebuah putaran tepat, maka sebuah lubang terbuka. Jika seluruh putaran dilakukan dengan kombinasi yang benar, maka seluruh lubang terbuka dan menciptakan sebuah jalur khusus yang bebas tidak ada hambatan sama sekali. Jalur lubang kunci tadi tidak lagi menjadi penghalang pintu brankas untuk dibuka, sehingga pintu dapat terbuka dengan mudah [11].

2.2.7 *Port Forwarding*

Penggunaan Protokol L2TP (*Layer 2 Tunneling Protocol*)/IPSec (IP *Security*) memberikan perlindungan ganda melalui otentikasi L2TP dan IPSec serta mendapat virtual IP *address* yang satu subnet dengan jaringan *internal*, sehingga *device* lain seperti berada pada *internal network*. Fungsi *port forwarding* adalah membuka akses terhadap perangkat pada jaringan lokal untuk dapat diakses melalui jaringan publik. *Port forwarding* akan mentranslasikan *Remote address* L2TP/IPsec yang didapatkan *Mikrotik client* ke Static IP *Public* VPS dengan penambahan *port* yang akan diakses. Dengan metode tersebut, *Administrator* hanya mengkonfigurasi L2TP/IPSec *client* di *Mikrotik* yang akan di*remote*. Dengan sistem ini diharapkan dapat membantu dan mempermudah *administrator* dalam melakukan *remote* perangkat jaringan melalui laptop maupun *smartphone* pada jaringan publik [18].

2.2.8 *Intrusion Prevention System* (IPS)

Intrusion Prevention System (IPS) adalah sebuah perangkat lunak atau perangkat keras yang bekerja untuk monitoring trafik jaringan, mendeteksi aktivitas yang mencurigakan dan melakukan pencegahan dini terhadap penyusupan atau kejadian yang dapat membuat jaringan menjadi berjalan tidak seperti sebagaimana mestinya. IPS merupakan pendekatan yang sering digunakan untuk membangun sistem keamanan komputer, IPS mengombinasikan teknik *firewall* dan metode *Intrusion Detection System* (IDS) dengan sangat baik. Teknologi ini dapat digunakan untuk mencegah serangan yang akan masuk ke jaringan lokal dengan memeriksa dan mencatat semua paket data serta mengenali paket dengan sensor saat serangan teridentifikasi. Jadi IPS bertindak seperti layaknya *firewall* yang akan mengizinkan atau menghalang paket data.

Secara khusus, IPS memiliki empat komponen utama, yaitu:

1. Normalisasi *Traffic*: menginterpretasikan *traffic* jaringan dan melakukan analisa terhadap paket yang disusun kembali, seperti halnya fungsi *block* sederhana.
2. *Detection Engine*: mendeteksi *traffic* jaringan dan melakukan *pattern matching* terhadap tabel acuan dan respon yang sesuai.
3. *Service Scanner*: membangun suatu tabel acuan untuk mengelompokkan informasi.
4. *Traffic Shaper*: membentuk dan mengatur *traffic* jaringan.

Ada 2 jenis IPS, yaitu *Host Based Intrusion Prevention System* (HIPS) dan *Network Based Intrusion Prevention System* (NIPS).

1. *Host Intrusion Prevention System* (HIPS)

   Host-based Intrusion Prevention System (HIPS) sama seperti halnya *Host Based Intrusion Detection System* (HIDS). Program agent HIPS di*install* secara langsung di sistem yang diproteksi untuk memonitor aktivitas sistem internalnya. HIPS di binding dengan kernel sistem operasi dan *service*s sistem operasi sehingga HIPS bisa memantau dan menghadang *system call* yang dicurigai dalam rangka mencegah

terjadinya intrusi terhadap *host*. HIPS juga bisa memantau aliran data dan aktivitas pada aplikasi tertentu. Sebagai contoh HIPS untuk mencegah intrusi pada *web server* misalnya. Dari sisi *security* mungkin solusi HIPS bisa mencegah datangnya ancaman terhadap *host*. Tetapi dari sisi *performance*, harus diperhatikan apakah HIPS memberikan dampak *negative* terhadap *performance host*. Karena meng*install* dan *binding* HIPS pada *system* operasi mengakibatkan penggunaan *resource* komputer *host* menjadi semakin besar.

2. *Network Intrusion Prevention System* (NIPS)

   Network-based Intrusion Prevention System (NIPS) tidak melakukan pemantauan secara khusus di satu *host* saja. Tetapi melakukan pemantauan dan proteksi dalam satu jaringan secara *global*. NIPS menggabungkan fitur IPS dengan *firewall* dan kadang disebut sebagai *In-Line* IDS atau *Gateway Intrusion Detection System* (GIDS). Sistem kerja IPS yang populer yaitu pendeteksian berbasis *signature*, pendeteksian berbasis anomali, dan monitoring *file* pada sistem operasi *host*.

   a. Sistematika IPS yang berbasis *signature* adalah dengan cara mencocokkan lalu lintas jaringan dengan *signature database* milik IPS yang berisi *attacking rule* atau cara-cara serangan dan penyusupan yang sering dilakukan oleh penyerang. Sama halnya dengan antivirus, IPS berbasis *signature* membutuhkan *update* terhadap *signature database* untuk metode penyerangan terbaru. IPS berbasis *signature* juga melakukan pencegahan terhadap ancaman intrusi sesuai dengan *signature database* yang bersangkutan.

   b. Sistematika IPS yang berbasis anomali adalah dengan cara melibatkan pola pola lalu lintas jaringan yang pernah terjadi. Umumnya, dilakukan dengan menggunakan teknik statistik. Statistik tersebut mencakup perbandingan antara lalu lintas jaringan yang sedang di monitor dengan lalu lintas jaringan yang biasa terjadi (*normal state*). Metode ini dapat dikatakan lebih kaya dibandingkan *signature-based* IPS. Karena *anomaly based* IPS dapat mendeteksi

gangguan terhadap jaringan yang terbaru yang belum terdapat di *database* IPS. Tetapi kelemahannya adalah potensi timbulnya *false positive*, yaitu pesan/*log* yang belum semestinya dilaporkan. Sehingga tugas *Network Administrator* menjadi lebih rumit, dengan harus memilah-milah mana yang merupakan serangan yang sebenarnya dari banyaknya laporan *false positive* yang muncul.

c. Teknik lain yang digunakan adalah dengan cara melakukan monitoring berkas-berkas sistem operasi pada *host*. IPS akan melihat apakah ada percobaan untuk mengubah beberapa berkas sistem operasi, utamanya berkas *log*.

Teknik ini diimplementasikan dalam IPS jenis *Host Based Intrusion Prevention System* (HIPS). Teknik yang digunakan IPS untuk mencegah serangan ada dua, yaitu *sniping* dan *shunning*.

a. *Sniping*: memungkinkan IPS untuk menterminasi serangan yang dicurigai melalui penggunaan paket *TCP RST* atau pesan *ICMP Unreachable*.
b. *Shunning*: memungkinkan IPS mengkonfigurasi secara otomatis *firewall* untuk melakukan *drop traffic* berdasarkan apa yang dideteksi oleh IPS. Untuk kemudian melakukan *prevention* atau pencegahan terhadap koneksi tertentu [6].

2.2.9 *Snort*

Snort merupakan salah satu contoh program *Network-based Intrusion Detection System*, yaitu sebuah program yang dapat mendeteksi suatu usaha penyusupan pada suatu sistem jaringan komputer. *Snort* bersifat *open Source* dengan lisensi GNU *General Purpose License* sehingga *Software* ini dapat dipergunakan untuk mengamankan sistem *server* tanpa harus membayar biaya lisensi [9].

Gambar 2.5 Logo *Snort*

Suatu sistem IDS harus bersifat lintas *platform*, mempunyai sistem *footprinting* yang ringan, dan mudah dikonfigurasi oleh *administrator* sebuah sistem yang membutuhkan implementasi dari solusi keamanan dalam waktu yang singkat. Implementasi tersebut dapat berupa seperangkat *Software* yang dapat diasosiasikan dalam melakukan aksi untuk merespon situasi keamanan tertentu. Selain itu. sebuah sistem IDS juga harus *powerfull* dan cukup fleksibel untuk digunakan sebagai bagian permanen dari suatu sistem jaringan. *Snort* memenuhi kriteria tersebut, yaitu dapat dikonfigurasi dan dibiarkan berjalan untuk periode yang lama tanpa meminta pengawasan atau perawatan bersifat administratif sebagai bagian dari sistem keamanan terpadu sebuah infrastruktur jaringan [9].

Snort iuga dapat berjalan pada semua platform sistem operasi di mana *libpcap* dapat berjalan. Sampai saat ini, *Snort* telah teruji dapat berjalan pada sistem

operasi *RedHat Linux, Debian Linux, MkLinux, HP-UX, Solaris (x86 dan Sparc), x86 Free/Net/OpenBSD, Windows* dan *Mac OS X* [9].