BAB II
TINJAUAN PUSTAKA

2.1. Profil Perusahaan PT Jabarmaya Kriya Sentosa

2.1.1. Sejarah Singkat Perusahaan PT Jabarmaya Kriya Sentosa

Pendiri JABAR MAYA adalah pelaku *Internet* yang telah berpengalaman sejak Tahun 2000. JABAR MAYA telah berkecimpung di bidang *Internet Service Provider* sejak Tahun 2002. PT. Sarana Insan Muda Selaras (SIMS), mempunyai lisensi nasional dalam bidang jasa layanan multimedia sesuai dengan Izin Prinsip Penyelenggaraan Jasa Multimedia 3015/PT.003/Tel/DJPT-2000 dari Direktur Jenderal Pos dan Telekomunikasi sebagai berikut :

a. *Broadcasting* TV & Radio,
b. *High Speed Internet Access*,
c. VPN (*Virtual Private Network*),
d. *Broadband Multimedia Access*,
e. *Video on Demand*, *Video Conferencing*,
f. *E-Commerce*, *E-Government*,
g. *Telecommunication*, *Telemedicine*, *E-Learning*,
h. *Interactive Game*
i. *Web Design & Development*,
j. *Information System Application Software*,
k. *System Design*.

Sejalan dengan perkembangan pembangunan infrastruktur layanan, maka lisensi diperbarui:

a. 1177/PT.003/Tel/DJPT-2002: Ijin Prinsip penyelenggaraan Jasa Akses Internet
b. 1672/PT.003/Tel/DJPT-2002: Ijin Prinsip penyelenggaraan Jasa Interkoneksi Internet (*NAP*)
c. 141/Dittel/KTS/V/2002: Uji Layak Operasi Jasa Multimedia
d. 1270/PT.003/dittel/srt/2002: Surat Keterangan Laik Operasi Multimedia

e. 1531/PT.003/Dittel/Srt/2003: Surat Keterangan Laik Operasi *ISP*
f. 1532/PT.003/Dittel/srt/2003: Surat Keterangan Laik Operasi *NAP/CISCO*
g. 315/Dirjen/2006 tanggal 11 September 2006 : Izin Penyelenggaraan Jasa Interkoneksi Internet dari Direktur Jendral Pos dan Telekomunikasi
h. 83 tahun 2013 tanggal 19 Februari 2013 : Izin Penyelenggaraan Jasa Interkoneksi Internet dari Dirjen Penyelenggaraan Pos dan Informatika Kementerian Komunikasi dan Informatika
i. 309/Dirjen /2006 tanggal 07-Sep-06 : Ijin Penyelengaraan Jasa Akses Internet dari Direttur Jenderal Pos dan Telekomunikasi
j. 278/KEP/DJPPI/KOMINFO/7/2012 tanggal 30 Juli 2012 : Ijin Penyelenggaraan Jasa Akses Internet Dirjen Penyelenggaraan Pos dan Informatika Kementerian Komunikasi dan Informatika
k. 2014 : Ijin Jartup dan Jartaplok , Kementrian Kominfo

2.1.2. Profil PT Jabarmaya Kriya Sentosa

PT. JABARMAYA KRIYA SENTOSA atau lebih dikenal dengan JABARMAYA adalah perusahaan yang memfokuskan pada Jasa *Internet Service Provider*, *Webhosting* dan *Data Center*. yang mencakup *Dedicated Internet*, *Broadband Internet*, *Shared Hosting*, *Colocation Server*, *Dedicated Server* dan *Managed Server*. Komitmen dari JABAR MAYA yaitu *SLA* 99,5%, *IP Public*, *Network Monitoring*, dan *Support* 24 Jam 7 Hari 365 Hari.

a. Nama Lembaga : PT. JABARMAYA KRIYA SENTOSA
b. Badan Hukum : Perseroan Terbatas
c. Merk Dagang : JABARMAYA
d. Alamat Perusahaan : Jalan Peta No. 168E Kota Bandung 40231
e. Telepon/*Fax* : +62-22-6613539
f. *E-mail* : Sales@jabarmaya.net.id, cs@jabarmaya.net.id, info@jabarmaya.net.id
g. *Website* : www.jabarmaya.net.id

2.1.3. Logo PT Jabarmaya Kriya Sentosa

Logo perusahaan dari PT. Jabarmaya Kriya Sentosa adalah seperti pada gambar 2.1 berikut ini

Gambar 2.1 Logo PT. Jabarmaya Kriya Sentosa

2.2. Landasan Teori

2.2.1. Server

Server secara sederhana dapat berupa satu buah komputer untuk beberapa layanan aplikasi, atau jika jaringannya lebih komplek dan rumit, maka *server* dapat disetting hanya untuk memberikan satu atau beberapa layanan saja [1]. Dengan kata lain *server* adalah sebuah komputer dengan spesifikasi yang tinggi yang digunakan untuk melayani *user* dalam jaringan tertentu.

Ubuntu *server* adalah sistem operasi berbasis Linux yang digunakan untuk komputer *server* tanpa dukungan *Grafical User Interface (GUI)* secara *default* [2]. Ubuntu *server* hadir dengan dukungan *Command Line Interface (CLI)* secara *default*. Ubuntu Server tersedia dalam dua kategori seperti pada versi Ubuntu Desktop yaitu Ubuntu Server 32-bit dan Ubuntu Server 64-bit.

2.2.2. Keamanan Jaringan Komputer

Keamanan dalam jaringan komputer sangat penting dilakukan untuk memonitor akses jaringan dan mencegah penyalahgunaan sumber daya jaringan yang tidak sah. Tugas keamanan jaringan dikontrol oleh administrator jaringan, segi-segi keamanan didefinisikan lima poin, yaitu:

1. *Confidentiality*, Mensyaratkan bahwa informasi (data) hanya bisa di akses oleh pihak yang memiliki wewenang.
2. *Intergrity*, mensyaratkan bahwa informasi hanya dapat di ubah oleh pihak yang memiliki

3. *Availability*, mensyaratkan bahwa informasi tersedia untuk pihak yang memiliki wewenang ketika di butuhkan.
4. *Authentication*, Mensyaratkan bahwa pengirim suatu informasi dapat diindentifikasi dengan benar dan ada jaminan bahwa identitas yang didapat tidak palsu.
5. *Nonrepudiation*, Mensyaratkan bahwa baik pengirim maupun penerima informasi tidak dapat menyangkal pengiriman dan penerimaan pesan[3].

2.2.2.1 Macam-macam keamanan jaringan

Keamanan jaringan dibagi menjadi 2 yaitu keamanan *physical* dan *logic*, serangan langsung maupun serangan tidak langsung.

1. Keamanan jaringan secara *physical* atau keamanan fisik yang sering mengacu pada tindakan yang diambil untuk melindungi sistem, gedung dan infrastuktur pendukung yang terkait terhadap ancaman yang berhubungan dengan lingkungan fisik, contoh-contoh dalam keamanan fisik yaitu *company surroundings, reception, server, workstation area, wireless access point, access control, computer equipment maintence, wiretapping, remote access.*
2. Kemanan jaringan secara *logic* adalah hal yang paling rawan terjadi sehingga harus lebih memperhatikan lagi keamanan *logic* dalam jaringan komputer kita, dalam hal ini keamanan *logic* ini sering mendapatkan serangan secara tidak langsung seperti yang di alami oleh fisik tetapi serangan secara *logic* menimbulkan dampak yang cukup serius.

2.2.2.2 Ancaman keamanan

Ancaman keamanan yang terjadi bisa di kelompokan menjadi beberapa ancaman yaitu:

a. *Interruption* adalah ancaman terhadap *availability* informasi, data yang ada dalam sistem komputer dirusak atau di hapus sehingga jika data atau informasi tersebut di butuhkan maka pemiliknya akan mengalami kesulitan mengaksesnya, bahkan mungkin informasi itu hilang.

b. *Interception* adalah ancaman terhadap kerahasiaan (*secrecy*). Informasi disadap sehingga orang yang tidak berhak dapat mengakses komputer dimana informasi tersebut disimpan.
c. *Modification* adalah ancaman terhadap integritas. Orang yang tidak berhak berhasil menyadap lalu-lintas informasi yang sedang dikirim dan kemudian mengubahnya sesuai keinginan orang tersebut.
d. *Fabrication* adalah ancaman terhadap integritas. Orang yang tidak berhak berhasil meniru atau memalsukan informasi sehingga orang yang menerima informasi tersebut menyangka bahwa informasi tersebut berasal dari orang yang dikehendaki oleh si penerima informasi[4].

2.2.2.3 Penyebab serangan

Penyerang menggunakan teknik yang telah dipublikasikan, yang biasa di akses dengan mudah ke suatu sistem . jika sytem terhubung dengan jaringan publik tidak mempunyai pertahanan yang baik, Karena tidak dikoreksi dengan baik, walaupun *vendor* memiliki produk yang mempunyai lubang-lubang keamanan, yang apabila tidak di perhatikan dengan baik maka akan banyak situasi yang menyebabkan ada begitu banyak lubang keamanan, seperti

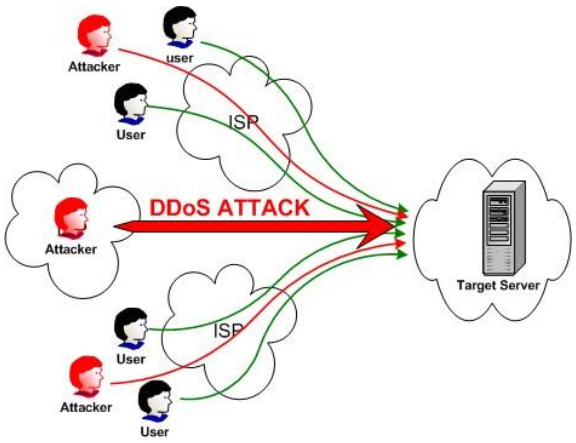
a. Banyak dari *legacy* sistem, sistem oprasi tidak *patch* maupun *update*.
b. *Patch* yang digunakan tidak diperhatikan dengan baik kebanyakan dari *patch* yang ada tidak diperiksa keamanannya sehingga menambahkan masalah baru dalam hal keamanan.
c. *User* yang melakukan kegiatan oprasional terhadap permintaan *service* jaringan dan protokol tidak memahami sistem, langkah-langkah yang harus dilakukan sehingga jaringan dan protokol yang mereka gunakan memiliki lubang keamanan.
d. *User* dan administrator membuat kesalahan dalam konfigurasi dan dalam menggunakan sistem.
e. Mekanisme konfigurasi akses kontrol sistem tidak mempunyai *policy* yang baik.[12]

2.2.3. Jenis Serangan Terhadap Jaringan Komputer

Jenis jenis serangan / tipe tipe serangan adalah macam macam teknik penyerangan terhadap keamanan jaringan yang memiliki tujuan yang berbeda-beda berikut adalah jenis jenis serangan.

2.2.3.1 *Denial of service* (DOS)

Serangan *Denial Of Service* (DOS) merupakan suatu serangan yang dimana bertujuan untuk mendapatkan hak akses penuh pada server misalnya penyerangan ingin mendapatkan suatu data di *database server*, meng-*install* program jahat (*malware*), dan lain-lain yang pada intinya melumpuhkan komputer target. *Denial of service* (DOS) dilakukan dengan cara membanjiri *server* target dengan pesan-pesan ataupun permintaan secara terus-menerus pada satu waktu, sehingga mengakibatkan server menjadi *overload* Karena sumber daya (*memory, CPU useg, bandwith*) yang memiliki *server* tersebut habis terpakai melayani pesan dan permintaan datang[5]. Seperti pada gambar 2.2.



Gambar 2.2 Serangan DOS

Tujuan dari DoS *attack* tidak untuk mendapatkan hak akses yang legal untuk masuk ke dalam sistem, melainkan untuk mencegah *user* untuk mengakses *service*. Agar serangan tersebut dapat berjalan, penyerang akan melakukan berbagai cara antara lain :

1. Mencoba untuk membajiri *traffic* data pada jaringan yang ditargetkan, sehingga membuat jaringan tidak dapat memberi *user* konektivitas.

2. Mencoba untuk mengganggu koneksi antara dua mesin dalam suatu jaringan yang berdampak *service* tidak dapat dipergunakan.
3. Mencoba untuk mencegah beberapa individu dari mengakses *service* atau mengganggu salah satu *service* yang spesifik agar tidak dapat diakses.

2.2.3.2 *Scanning*

Scanning adalah proses dimana penyerang akan menggali informasi tentang alamat IP target, sistem operasi yang digunakan, arsitektur jaringan yang digunakan, dan *service* yang sedang berjalan pada komputer pun dapat di peroleh. Tidak seperti *footprinting* yang hanya mengali informasi secara pasif dari pihak ketiga dalam berbagai sumber, *scanning* secara aktif berhadapan dengan target untuk menperoleh informasi.

Dalam melakukan *scanning*, terdapat beberapa tipe *scanning* diantaranya sebagai berikut :

1. *Port scanning*

   Port scanning meliputi pengiriman beberapa pesan pada komputer target untuk memperoleh tipe *service* jaringan apakah yang sedang berjalan pada jaringan tersebut. Dikarenakan *service* tersebut berkaitan dengan nomor *port*, maka dengan melakukan *port scan* pada target akan menampilkan seluruh *port* yang terbuka.

2. *Network scanning*

   Network scanning adalah suatu prosedur untuk mengidentifikasi *host* yang sedang aktif pada jaringan target yang bertujuan untuk melakukan serangan pada target ataupun untuk menilai keamanan jaringan. Dalam langkah ini akan memungkinkan penyerang untuk membuat daftar *host-host* yang dapat diserang secara langsung atau menggunakan *host* tersebut untuk menyerang *host* lain secara tidak langsung.

3. *Vulnerability scanning*

   Vulnerability scanning berhubungan dengan penggunaan *tools* otomastis yang dikenal dengan *vulnerability scanner* yang secara otomatis mengidentifikasi kelemahan-kelemahan keamanan pada sistem komputer di

jaringan tersebut. *Tools* ini akan melacak target untuk mencari tahu celah keamanan manakah yang dapat di eksploitasi.

2.2.3.3 *Brute Force*

Serangan *brute-force* adalah sebuah teknik serangan terhadap sebuah sistem keamanan komputer yang menggunakan percobaan terhadap semua kunci yang mungkin. Pendekatan ini pada awalnya merujuk pada sebuah program komputer yang mengandalkan kekuatan pemrosesan komputer dibandingkan kecerdasan manusia[6].

2.2.4. Honeypot

Honeypot merupakan *security resource* yang sengaja dibuat untuk diselidiki, diserang, atau dikompromikan. *Honeypot* ini bisa menjadi replika dari *server* yang kita punya, sehingga saat terjadi penyusupan, seorang *hacker* akan mengira dia sedang mengeksploitasi *server* kita padahal dia sedang mengeksploitasi *fake server*. Selain itu *honeypot* akan merekam setiap aktifitas penyusupan, termasuk identifikasi serta menunjukan celah keamanan yang sedang di eksploitasi oleh penyusup[7].

Honeypot dapat diklasifikasikan berdasarkan pada tingkat interaksi yang dimilikinya. Tingkat interaksi dapat didefinisikan sebagai tingkat aktivitas penyerang didalam sistem yang diperbolehkan maka semakin tinggi pula tingkat interaksi honeypot[8].

a) Low Interaction Honeypot

Low-interaction honeypot merupakan *honeypot* dengan tingkat interaksi *honeypot*, yaitu *honeypot* yang didesain untuk mengemulasikan *service* (layanan) seperti pada *server* yang asli. Penyerang hanya mampu memeriksa dan terkoneksi ke satu atau beberapa *port*. Kelebihan dan kekurangan *low interaction honeypot* dapat terlihat pada tabel 2.1.

Tabel 2.1 Kekurangan dan Kelebihan Low Interaction Honeypot

Kelebihan	Kekurangan
Mudah di install, dikonfigurasi, deployed, dan dimaintain	Layanan yang di berikan hanya berupa emulasi, sehingga penyerang tidak

	dapat berinteraksi secara penuh dengan layanan yang diberikan atau sistem operasinya secara langsung
Mampu mengemulasi suatu layanan seperti http, ftp, telnet, dsb.	Informasi yang bisa kita dapatkan dari penyerang sangat minim.
Difungsikan untuk deteksi serangan, khususnya pada proses scanning atau percobaan.	Apabila serangan dilakukan oleh "real person" bukan "automated tools" mungkin akan segera menyadari bahwa yang sedang dihadapi merupakan mesin honeypot, karena keterbatasan layanan yang bisa diakses.

b) High Interaction Honeypot

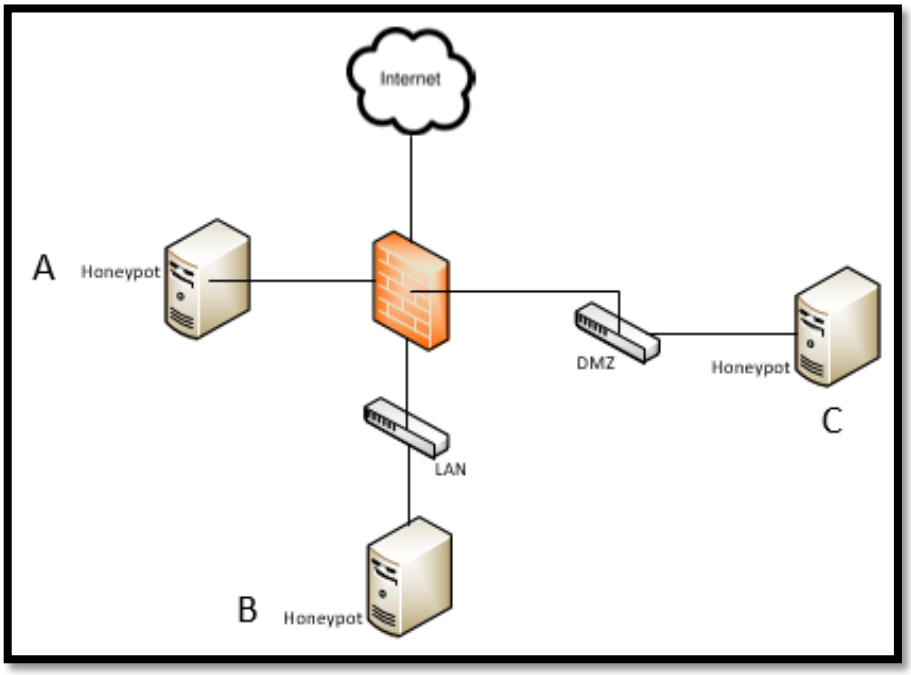
High-interaction honeypot terdapat sistem operasi dimana penyerang dapat berinteraksi langsung dan tidak ada batasan yang membatasi interaksi tersebut. Menghilangkan batasan-batasan tersebut menyebabkan tingkat risiko yang dihadapi semakin tinggi karena penyerang dapat memiliki akses root. Pada saat yang sama, kemungkinan pengumpulan informasi semakin meningkat dikarenakan kemungkinan serangan yang tinggi[8]. Dikarenakan penyerang dapat berinteraksi secara penuh dengan sistem operasi, maka apabila si penyerang telah mendapat akses root. Kelebihan dan kekurangan *high interaction honeypot* dapat terlihat pada tabel 2.2.

Tabel 2.2 Kekurangan dan Kelebihan High Interaction Honeypot

Kelebihan	**Kekurangan**
Penyerang berinteraksi langsung dengan sistem yang nyata termasuk diantaranya sistem operasi, network, hingga layanan yang diberikan (web service, ssh service, mail service, dll)	Perencanaan dan implementasi sistem jauh lebih rumit dan dibutuhkan banyak pertimbangan.

Umumnya dibangun suatu sistem khusus dengan topologi yang telah dipersiapkan.	High-interaction honeypot bersifat tidak efisien karena membutuhkan pengawasan berkala.
Sistem tersebut biasanya terdiri dari berbagai macam implementasi dari teknologi keamanan yang banyak digunakan untuk melindungi suatu sistem, seperti firewall, IDS/IPS, router, dll.	Apabila telah diambil alih oleh penyerang maka honeypot tersebut dapat menjadi ancaman bagi jaringan yang ada.
Target serangan berupa sistem operasi sebenarnya yang siap untuk berinteraksi secara langsung dengan penyerang.	

Penempatan *honeypot* pada suatu jaringan akan memiliki keuntungan dan kerugian tersendiri. Berikut beberapa contoh lokasi penempatan *honeypot*, seperti yang dapat terlihat pada gambar 2.3.



Gambar 2.3 Penempatan Honeypot

a. Penempatan secara langsung dengan menghadapkan *honeypot* dengan internet tanpa adanya *firewall*

   Kelebihan dari penempatan *honeypot* ini adalah *honeypot* akan dianggap *system external* sehingga akan mengurangi resiko jaringan *private* apabila *honeypot* sudah diambil alih oleh penyerang. Kekurangannya tidak bisa menjebak penyerang yang berasal dijaringan *internal*.

b. Penempatan secara tidak langsung, dimana *honeypot* berada diantara *firewall* dan koneksi internet

   Kelebihan dari penempatan *honeypot* ini adalah dapat mendeteksi *firewall* yang tidak terkonfigurasi dengan baik yang menyebabkan adanya trafik tidak sah yang menuju jaringan *private*. Kekurangannya akan menambah resiko pada jaringan *private*, karena pada saat *honeypot* sudah diambil alih oleh penyerang, penyerang akan bisa mengakses jaringan *private.*

c. Penempatan *honeypot* pada DMZ (*Demiliterized Zone)*

Kelebihan dari penempatan *honeypot* ini adalah trafik tidak sah yang menuju *honeypot* akan melewati *firewall* sehingga akan tercatat di *firewall log*. Kekurangannya akan menambah resiko pada jaringan yang ada di DMZ. Karena pada saat *honeypot* sudah diambil alih oleh penyerang. Penyerang akn bisa menggunakan *honeypot* untuk menyerang sistem lain yang berada di DMZ.

2.2.5. HIHAT (*High Interaction Anaysis Tools*)

HIHAT merupakan *software* untuk merubah *script* PHP menjadi *honeypot.* HIHAT melakukan *monitoring* terhadap *honeypot* dan berbasis web. Sebuah *web service* menjadikan *honeypot* berfungsi dengan baik, yang menawarkan keamanan lengkap dari aplikasi untuk pengguna tapi melakukan pencatatan dan pemantauan di balik layar. HIHAT ini dapat mencatat log secara berkala, dapat mencatat IP peretas secara baik, dan dapat memvisualisasikan berapa jumlah *hits* yang dilakukan penyerang dan *file* apa saja yang dapat diakses oleh penyerang. HIHAT memiliki prinsip kerja yaitu menjawab respon yang dilakukan oleh penyerang dengan memberikan respon yang diharapkan oleh penyerang. Penyerang mengirimkan permintaan berbahaya lalu honeypot akan memproses *request* dan menulis ke *database* lalu memberi balasan ke penyerang. Setelah diidentifikasi jenis serangan HIHAT akan menghasilkan respon untuk mensimulasikan hasil dari serangan yang berhasil[8].

2.2.6. Cowrie & Kippo

Cowrie dapat berfungsi sebagai *medium interaction honeypot* ataupun *high interaction honeypot.* Jika Cowrie di *setting* untuk menjadi *high interaction honeypot,* maka Cowrie akan menjadi *proxy,* yang menghubungkan peretas dengan ssh pada *honeypot, username* dan *password* yang disetting pada Cowrie untuk login ke ssh merupakan *username* dan *password* palsu. Cowrie adalah distribusi *honeypot* yang banyak digunakan yang mencatat semua interaksi SSH dalam *database* MySQL. Ini telah dimodifikasi untuk menghasilkan parameter untuk diteruskan ke agen pembelajaran. Bergantung pada tindakan yang dipilih oleh agen pembelajaran, *honeypot* akan memungkinkan, memblokir atau mengganti perintah serangan. Cowrie menyimpan semua interaksi dalam *file log* sebagai standar. Itu juga

menyimpan semua file yang diunduh dalam direktori unduhan. Ini memungkinkan untuk mengakumulasikan jumlah semua perintah yang dicoba pada honeypot[9].

Kippo adalah *honeypot SSH* interaksi menengah yang ditulis dalam *Python*. Kippo digunakan untuk mencatat serangan *brute force* dan seluruh interaksi shell yang dilakukan oleh penyerang. Kippo terinspirasi oleh honeypot Kojoney. Kippo berisi sistem *file* palsu yang mampu meniru menambahkan dan menghapus *file*. Itu terinspirasi oleh debian Linux OS. Pengguna dapat mengunduh *file* baru (misalkan menggunakan perintah wget) dan menggunakan perintah cat juga. *File* yang diunduh direkam. Namun, sebagian besar perintah dan alat Linux tidak diimplementasikan (ditiru) dan jika penyerang mencoba menggunakannya, pesan kesalahan dihasilkan. Semua serangan dicatat dengan stempel waktu yang tepat. Inilah yang ditiru Kippo honeypot yaitu sistem yang tampak seperti aplikasi mandiri. Di Kippo hanya layanan *SSH* (port 22) yang ditiru dan setelah penetrasi sistem hanya aktivitas dalam sistem yang dianalisis. Tetapi berkat kenyataan bahwa Kippo adalah honeypot interaksi rendah, penyerang diizinkan untuk melakukan apa saja. Tidak ada perintah yang dieksekusi itu sistem nyata, beberapa dari mereka tersedia sebagai perintah *dummy* sementara banyak lainnya menghasilkan pesan kesalahan. Pengukuran memungkinkan untuk membuat tinjauan umum dari penetrasi yang paling sering dicoba melalui protokol SSH. Berkat fakta bahwa tingkat interaksi dalam honeypot tinggi biasanya hanya serangan yang lebih sederhana (misalkan *Bot* dan *skrip-kiddies*) tertangkap[10].

2.2.7. IPS (*Intrusion Prevention System*)

Intrusion Prevention System (IPS) merupakan solusi keamanan yang lebih advance dari IDS, karena IPS dapat melakukan lebih dari ‘sekedar’ menganalisis *traffic/log* dan menghasilkan *alert*. IPS dapat secara proaktif melakukan ‘reaksi’ terhadap intrusi yang terdeteksi. Oleh karena itu IPS secara umum diletakkan secara in-line dengan firewall, agar IPS dapat menganalisis secara *real-time* semua *traffic* yang masuk dan keluar pada *network* untuk mendeteksi *suspicious* atau *malicious activity* dan kemudian secara instan melakukan aksi yang diperlukan untuk mencegah aktifitas (yang merupakan serangan) tersebut berhasil masuk kedalam jaringan atau sistem[11].

IPS juga bukanlah pengganti untuk *firewall*, *strong security policy*, *patching/hardening management* dan teknik *defense-in-depth* lainnya, melainkan digunakan sebagai ekstra *layer security*. Sebagai contoh:

Firewall digunakan untuk menghentikan *service* dengan menutup *port* atau layanan tertentu tetapi tidak dapat menganalisis *traffic* yang berasal dari *port* yang dibuka. IDS dapat mengevaluasi *traffic* yang berasal dari *port* yang dibuka tersebut, namun tidak dapat menghentikannya (block). IPS dapat secara proaktif melakukan *blocking* (terhadap *malicious traffic*) tersebut. Namun begitu IPS tidak bisa menganalisis *traffic* secara mendalam pada layer aplikasi, contohnya saja pada web. Untuk itulah kita menambahkan *Web Application Firewall (WAF)* sebagai tambahan perangkat *defense-in-depth* lainnya.

IPS akan sangat membantu dalam urusan waktu. Contohnya terdapat *vulnerability* pada *server* kita dan kita belum sempat untuk melakukan *hardening*. Karena kita telah mengimplementasikan IPS, IPS akan mendeteksi *vulnerability* tersebut. Lalu apabila ada *attacker* yang mencoba melakukan eksploitasi terhadap *vulnerability* tersebut, IPS akan menghentikannya sebelum eksploitasi tersebut berhasil dilakukan. Dengan begitu, kita memiliki cukup waktu untuk melakukan *patching/hardening* pada *server* kita.

Ada dua jenis IPS yaitu :

a ***Host-based Intrusion Prevention System (HIPS)***

Tidak jauh beda pengertiannya dengan HIDS, dimana HIPS juga ‘diletakkan’ pada spesifik *host*, HIPS akan memproteksi *operating system* dan aplikasi yang berjalan pada *host* tersebut dengan mengidentifikasi dan menghentikan *known attack* dan *unknown attack.*

HIPS *software* menggunakan teknik yang disebut *system call interception*, dimana intinya HIPS *software* memiliki kemampuan dapat melakukan *deny* atau *permit request* ketika request tersebut diidentifikasi sebagai *request* yang berbahaya atau tidak. Pada HIPS ini akan memproteksi *traffic* pada *interface network*, memproteksi *integrity file*, dan memproteksi *behavior* aplikasi.

b ***Network-based Intrusion Prevention System (NIPS)***

Network-based pada IPS hampir sama konsepnya dengan yang ada pada IDS, dimana NIPS ini memproteksi keseluruhan network. Seperti dijelaskan sebelumnya, IPS umunya diletakkan in-line dengan *firewall* dan *network* kita. Sederhananya, semua *traffic* yang melalui NIPS dari internet yang mengarah ke *firewall* dan keluar *firewall*, jika menghasilkan *alert* akan didrop oleh NIPS dan tidak akan diteruskan ke *network* kita. NIPS harus memiliki *ability* yang sama dengan NIDS, dimana secara berkelanjutan melakukan monitoring terhadap abnormal *traffic pattern*, men-*generate event log*, menghasilkan *alert* dan yang terpenting menghentikan intruksi yang masuk ke *network* kita.

2.2.8. Snort

Snort adalah sebuah *software* ringkas yang sangat berguna untuk mengamati aktivitas dalam suatu jaringan komputer. Snort dapat digunakan sebagai suatu *Network Intrusion Detection System* (NIDS) yang berskala ringan (*lightweight*), dan *software* ini menggunakan sistem peraturan-peraturan (*rules system*) yang relatif mudah dipelajari untuk melakukan deteksi dan pencatatan (*logging*) terhadap berbagai macam serangan terhadap jaringan komputer. Dengan membuat berbagai *rules* untuk mendeteksi ciri-ciri khas (*signature*) dari berbagai macam serangan, maka Snort dapat mendeteksi dan melakukan logging terhadap serangan-serangan tersebut. *Software* ini bersifat *opensource* berdasarkan GNU *General Public License* [GNU89], sehingga boleh digunakan dengan bebas secara gratis, dan kode sumber (*source code*) untuk Snort juga bisa didapatkan dan dimodifikasi sendiri bila perlu. Snort pada awalnya dibuat untuk sistem operasi (*operating system*) berdasarkan Unix, tetapi versi Windows juga sudah dibuat sehingga sekarang ini Snort bersifat *cross-platform*[12]. Snort sendiri merupakan *software* yang masih berbasis *command-line*, sehingga cukup merepotkan bagi pengguna yang sudah terbiasa dalam lingkungan *Graphical User Interface* (GUI). Oleh karena itu, ada beberapa *software* pihak ketiga yang memberikan GUI untuk Snort, misalnya IDScenter untuk Microsoft Windows, dan Acid yang berbasis PHP sehingga bisa diakses melalui *web browser*.

2.2.9. Barnyard2

Barnyard2 adalah tool *open source* sebagai penerjemah *alert unified*. Barnyard2 bekerja dengan membaca file log *unified2* dan memasukannya kedalam database[13]. Jika database tidak terkoneksi maka Barnyard2 akan memasukan semua data ketika database tersedia kembali sehingga tidak ada *alert* atau log yang hilang.

2.2.10. IPTables

Iptables adalah suatu *tools* dalam sistem operasi linux yang berfungsi sebagai alat untuk melakukan *filter* (penyaringan) terhadap (*trafic*) lalu lintas data. Secara sederhana digambarkan sebagai pengatur lalulintas data. Dengan iptables inilah kita akan mengatur semua lalu lintas dalam komputer kita, baik yang masuk ke komputer, keluar dari komputer, ataupun *traffic* yang sekedar melewati komputer kita[11]. Dengan kemampuan *tools* iptables ini, kita bisa melakukan banyak hal dengan iptables. Yang paling penting adalah bahwa dengan iptables ini kita bisa membuat aturan, untuk arus lalu lintas data. Aturan aturan itu dapat mencakup banyak hal, seperti besar data yang boleh lewat, jenis paket/datagram yang dapat diterima, mengatur *trafic* berdasar asal dan tujuan data, *forwarding, nat, redirecting*, pengelolaan *port*, dan *firewall*. Perintah-perintah pada iptables dapat terlihat pada tabel 2.3.

Tabel 2.3 Bentuk Umum Perintah Iptables

Perintah	Keterangan
-A –append	Perintah ini menambahkan aturan di akhir chain. Aturan ditambahkan di akhir baris chain yang bersangkutan, sehingga dieksekusi terakhir kali
-D –delete	Menghapus satu aturan chain. Caranya dengan menyebutkan perintah mana yang ingin dihapus secara lengkap atau menyebutkan nomor baris perintah yang akan dihapus.
-R –replace	Menggantikan aturan chain dengan aturan (entry) baru.
-I –insert	Memasukkan aturan pada suatu baris di chain. Aturan akan dimasukkan ke baris yang ditulis, dan aturan yang tadinya ada di baris tersebut akan bergeser ke bawah bersama dengan baris-baris selanjutnya
-L –list	Menampilkan semua aturan pada tabel. Jika tabel tidak disebutkan, seluruh aturan di semua tabel akan ditampilkan, walaupun tidak ada aturan sama sekali pada tabel. Command ini bisa dikombinasikan dengan option –v (verbose), -n (numeric) dan –x (exact).
-F –flush	Perintah ini mengosongkan aturan pada sebuah chain. Apabila chain tidak disebutkan, maka semua chain akan hilang.
-N –new-chain	Membuat chain baru.

-X –delete-chain	Menghapus chain yang dituliskan. Notabene, tidak boleh ada aturan lain yang bersangkutan dengan chain tersebut.
-P –policy	Membuat kebijakan default pada chain. Jika ada paket yang tidak memenuhi aturan pada baris yang diinginkan, paket akan diperlakukan sesuai dengan kebijakan default ini.
-E –rename-chain	Mengubah nama chain.

2.2.11. *Bot*

Bot merupakan kependekan dari Robot yaitu berupa program yang beroperasi sebagai agen untuk pengguna atau program lain atau mensimulasikan aktivitas manusia. Dengan kata lain *bot* menjalankan tugas secara automatis jauh lebih cepat dibanding yang dilakukan manusia normal.

Menurut sebuah jurnal yang diterbikan oleh IJARCCE yang berjudul "*Artificial Intelegent Chatbot in Android System using Open Source Program-O*", *Chatbot* adalah agen percakapan dimana program komputer dirancang untuk mensimulasikan percakapan yang cerdas dengan manusia [14]. *Chatbot* mengotomatiskan proses menangkap detail masalah dari pengguna dan memberikan solusi yang layak kepada pengguna [15].

2.2.12. Telegram Messanger

Telegram Messanger adalah aplikasi *messaging* yang berfokus pada kecepatan, keamanan, sederhana, dan dapat diunduh secara gratis. Telegram dapat digunakan di semua perangkat dalam satu akun dalam waktu yang sama. Telegram juga dapat mengirim pesan, foto, video, dan beberapa jenis *file* (doc, zip, mp3, dll), serta dapat membuat group hingga 5000 orang atau *channel* untuk *broadcasting* untuk khalayak terbatas. Beberapa fitur lain yang terdapat dalam *Telegram Messanger* :

a. Enkripsi pesan, personal dan *bussiness secret*.

b. Menghapus pesan secara otomatis dengan *timer*.

c. Menyimpan media di dalam *cloud*.

d. Membangun sistem/*tools* sendiri dengan menggunakan API telegram.

Gambar 2.4 Logo Telegram Apps

Salah satu fitur telegram yang berbeda dengan aplikasi *messenger* lainnya adalah Telegram menyediakan API (*Application Programming Interface*) yang terbuka 100% untuk publik yang ingin mengembangkan aplikasi menggunakan API Telegram.

2.2.13. Telegram API

Telegram mempunyai dua jenis APIs untuk *developer*, yaitu:

a. Bot API

Bot API memungkinkan *developer* untuk menghubungkan Bot dengan sistem Telegram. Telegram Bots adalah sebuah akun khusus yang tidak memerlukan nomor telepon tambahan dalam pengaturannya. Akun ini berfungsi sebagai antarmuka untuk tempat berjalannya kode di pada suatu *server*.

Gambar 2.5 The Botfather Telegram

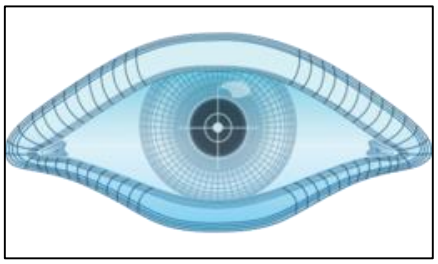
Botfather Telegram adalah salah satu *tools* resmi dari telegram yang digunakan untuk membuat atau menciptakan *bot* telegram dimana nantinya Botfather tersebut akan mengeluarkan token API yang digunakan sebagai kunci mengelola *bot* yang akan dipakai pada kode program.

b. Telegram API

Telegram API memungkinkan *developer* untuk membangun sendiri Telegram *clients* yang diinginkan. Telegram API terbuka 100% untuk semua *developer* yang ingin membuat aplikasi dengan *platform* Telegram.

2.2.14. Zenmap

Zenmap adalah aplikasi multi platform sebagai *interface* sederhana untuk aplikasi nmap. Fungsi zenmap itu sendiri adalah untuk eksplorasi dan audit keamanan jaringan dan memeriksa jaringan besar secara cepat, meskipun ia zenmap dapat pula bekerja terhadap *host* tunggal[16]. *Nmap (Network Mapper)* sendiri adalah sebuah aplikasi *open source* untuk explorasi *network* dan audit keamanannya. Nmap bekerja dengan melakukan scan terhadap komputer (*host*) *stand alone* ataupun *host* yang terhubung dalam sebuah jaringan, *menentukan host-host* yang aktif dalam suatu jaringan, menentukan informasi sistem operasi, *port-port* yang terbuka dan jenis *firewall* yang digunakan.



Gambar 2.6 Logo Zenmap

Zenmap bersifat multi platform, artinya bisa berjalan pada berbagai sistem operasi seperti Linux, Windows, Mac, FreeBSD, openBSD dan Sun OS. Nmap adalah aplikasi berbasis *command line* tetapi untuk kemudahan penggunaan dan analisis hasilnya disertakan aplikasi GUI yang dinamakan zenmap.