JOURNAL OF
SOFTWARE ENGINEERING AND
COMPUTATIONAL INTELLIGENCE (JSECI)
INFORMATICS ENGINEERING, FACULTY OF COMPUTER SCIENCE
UNIVERSITAS INDO GLOBAL MANDIRI

Implementasi *Honeypot* Sebagai Pendeteksi Serangan Pada *Virtual Private Server (VPS)*

M. Alamsyah Pratama[*1], Herri Setiawan[1], Zaid Romegar Mair[2]
[1,2,3]Fakultas Ilmu Komputer, Teknik Informatika
[1,2,3]Universitas Indo Global Mandiri
[1,2,3]Kota Palembang, Indonesia
E-mail : [1]alamsyah.map27@gmail.com ,[2] herri@uigm.ac.id ,[3] zaidromegar@uigm.ac.id

Abstrak

VPS (Virtual Private Server) merupakan teknologi virtualisasi server yang saat ini banyak digunakan oleh pengguna untuk berbagai keperluan, antara lain personal, kantor, kantor, dan bisnis. Virtual Private Server (VPS) adalah teknologi yang menggunakan VPS. Karena menyewa server pribadi virtual (VPS) lebih murah daripada membeli atau menyewa server khusus, VPS banyak digunakan. Namun, VPS juga memiliki beberapa kekurangan, salah satunya menyangkut keamanan. Intrusion Detection System (IDS), Virtual Private Network (VPN), dan Honeypot hanyalah beberapa opsi lain yang dapat digunakan untuk server yang aman. Aplikasi Honeypot diperlukan untuk memproteksi Virtual Private Server (VPS) yang membutuhkan aplikasi pendeteksi yang dapat meredam serangan terhadap VPS. Cowrie mampu mendeteksi dan merekam semua serangan yang terjadi pada server berdasarkan hasil pengujian serangan yang dilakukan sebanyak 25 kali untuk setiap pengujian serangan, dimulai dengan pemindaian port dan berlanjut melalui serangan DDoS, serangan brute force, dan individual, double, dan beberapa tes. dianggap sangat efektif, namun ketika beberapa serangan dilakukan secara bersamaan, server mengalami downtime atau lag.

Kata Kunci: *virtual private server(vps), honeypot, cowrie*

Abstract

VPS (Virtual Private Server) is a server virtualization technology that is currently widely used by users for a variety of purposes, including personal, work, office, and business. Virtual Private Server (VPS) is a technology that uses VPS. Since renting a virtual private server (VPS) is less expensive than purchasing or renting a dedicated server, VPS is widely used. However, VPS has some disadvantages as well, one of which concerns security. The Intrusion Detection System (IDS), the Virtual Private Network (VPN), and the Honeypot are just a few of the other options that can be utilized to secure servers. The Honeypot application is needed to protect Virtual Private Server (VPS), which requires a detection application that can reduce attacks on VPS. Cowrie is able to detect and record all attacks that occur on the server based on the results of attack testing that was carried out 25 times for each attack test, beginning with port scanning and progressing through DDoS attacks, brute force attacks, and individual, double, and multiple tests. considered to be very effective, but when multiple attacks were carried out simultaneously, the server experienced a downtime or lag.

***Keyword**: virtual private server(vps), honeypot, cowrie*

1. Pendahuluan

Pesatnya kemajuan teknologi di berbagai ranah kehidupan telah menimbulkan konsekuensi positif dan negatif. Salah satu dampak positifnya adalah kenyamanan yang dibawanya untuk mendukung aktivitas sehari-hari dan pekerjaan kantor. Namun, disamping dampak positifnya, ada juga dampak negatifnya. Salah satu dampak negatifnya adalah perlunya kesadaran dalam menjaga dan melindungi data pribadi kita agar tidak disalahgunakan oleh pihak-pihak yang tidak dapat dipercaya.

Teknologi *virtualisasi server* yang dikenal dengan *VPS (Virtual Private Server)* saat ini banyak digunakan oleh pengguna untuk berbagai keperluan, antara lain penggunaan pribadi, profesional, perkantoran, dan bisnis. Karena menyewa *VPS* lebih murah daripada membeli atau menyewa *server* khusus, banyak orang menggunakannya untuk berbagai keperluan.

Virtual Private Server (VPS) adalah teknologi virtualisasi yang menyediakan *server virtual* dengan sumber daya *CPU, RAM,* dan penyimpanan yang dialokasikan, menghilangkan kebutuhan akan *server* fisik. Ini memungkinkan pengguna untuk memiliki akses root dan menyesuaikan server mereka sesuai dengan kebutuhan mereka. Dibandingkan menyewa server khusus, *VPS* menawarkan solusi yang lebih hemat biaya. Dalam layanan ini, server utama yang kuat dengan prosesor ganda dan banyak inti dipartisi menjadi beberapa *server virtual (VPS*), masing-masing berjalan secara independen dengan sistem operasi *(OS)* yang sama, memastikan bahwa mereka tidak mengganggu satu sama lain. [1].

Namun, meski memiliki kelebihan dengan biaya yang lebih murah, *virtual private server (VPS*) juga memiliki sejumlah kekurangan, salah satunya adalah keamanan. Firewall adalah ukuran keamanan yang paling banyak digunakan untuk server. *Firewall* adalah sistem keamanan untuk jaringan yang dirancang untuk melindungi komputer dari berbagai ancaman yang diketahui. Mekanisme keamanan *firewall* dapat dibandingkan dengan dinding, yang menghalangi serangan masuk tetapi tidak sepenuhnya mengamankan sistem (*firewal*l hanyalah penghalang). Namun, firewall sering dipilih untuk menghentikan penyerang jaringan dari merusak target. [2].

Ada lebih banyak pengguna *VPS* daripada *server* khusus karena biayanya jauh lebih rendah. Namun, untuk menggunakan firewall sebagai media keamanan, pengguna harus berlangganan lisensi atau membeli perangkat mahal yang harus sering diperbarui. Ini menyulitkan pengguna *VPS* dan membutuhkan opsi lain. dalam keamanan *VPS*. Anda dapat menggunakan *IDS (Intrusion Detection System), VPN (Virtual Private Network),* dan *Honeypot* sebagai metode keamanan *server* tambahan.

Sebuah sistem atau komputer yang dikenal sebagai honeypot adalah salah satu yang sengaja "dikorbankan" untuk menjadi sasaran serangan hacker. Untuk setiap serangan yang diluncurkan peretas terhadap server, sistem dapat menyediakan layanan. Strategi ini dimaksudkan agar pimpinan server dapat mengetahui cara-cara penetrasi yang dilakukan oleh programmer, dan dimaksudkan untuk menjaga server yang sebenarnya. Honeypot adalah aset penanganan aset yang dibuat untuk menyerang, mengambil kendali, mengakses, dan menggunakan dengan berbagai cara yang tidak disetujui. Penyergapan, serangan, atau sumber daya keamanan yang berfokus pada penghancuran dikenal sebagai honeypot. [3].

Honeypot adalah alternatif yang dapat digunakan oleh pengguna server pribadi virtual (VPS) untuk mengamankan server mereka. Karena bersifat open source, pengguna dapat menggunakan layanan ini dengan mudah, gratis, dan dengan cara yang paling sesuai dengan kebutuhan mereka. Sistem pendukung cowrie dan kippo-graph diperlukan untuk implementasi honeypot. Honeypot Cowrie dikenal karena jenis koneksi media, di mana Honeypot Cowrie dapat menganalisis dan merekam aktivitas yang dilakukan oleh penyerang di organisasi yang menggunakan layanan Protected Shell (SSH) [4].

Dalam melakukan implementasi *honeypot* menggunakan aplikasi *honeypot cowrie* karena bersifat *open source* yang membuat penulis dapat menggunakan dengan gratis serta dimodifikasi

sesuai kebutuhan, selain itu *cowrie* hanya terfokus terhadap 2 *port* yaitu *port telnet* dan *SSH*, karena pada dasarnya untuk mengakses *remote VPS* menggunakan *SSH* dan *telnet* hal tersebut menjadi jantung utama karena apabila *SSH* berhasil terjebol maka data yang disimpan pada *VPS* dapat dengan mudah diretas, dengan adanya *honeypot attacker* akan masuk dan terperangkap kedalam *server* jebakan *honeypot*.

2. Metode

A. Debian

Debian adalah sistem operasi komputer yang bebas (dari kata freedom yang artinya kebebasan). Kerangka kerja adalah kumpulan proyek dan utilitas mendasar yang dibutuhkan PC Anda untuk bekerja. Debian lebih dari sekedar sistem operasi: Debian juga menawarkan lebih dari 59.000 paket tambahan, yang merupakan perangkat lunak terkompilasi yang tersedia dalam berbagai paket yang dikemas dengan baik dan mudah diinstal. [1].

B. Honeypot

Honeypot adalah sistem yang mencoba meniru atau mereplikasi sistem nyata untuk mengalihkan perhatian penyerang dari sistem target sebenarnya dan menuju sistem umpan. Selain menjebak penyerang dan mencegah mereka menyerang server yang sebenarnya, ini juga memberi administrator sistem dan analis keamanan wawasan yang berharga. Dengan mengirimkan honeypot, mereka dapat mempelajari latihan dan strategi yang digunakan oleh penyerang dan malware di dalam kerangka honeypot. [5].

Honeypot menawarkan 3 jenis layanan berbeda yang dapat diubah agar sesuai dengan tingkat interaksi. Ketiga jenis administrasi tersebut adalah sebagai berikut

1. Honeypot Interaksi Rendah

Low Interaction Honeypot merupakan bantuan utama dalam sebuah honeypot dimana Honeypot akan membuat tiruan dari server dan direktur organisasi sebagai pemilik server sebenarnya memiliki komando penuh atas latihan interupsi yang terjadi.

2. Layanan kedua di honeypot, Honeypot Interaksi Sedang membuat sistem operasi fiktif untuk memikat penyerang. Sistem akan melewati beberapa perintah honeypot melalui layanan ini, tetapi informasi penyerang akan dicatat dan dievaluasi oleh manajer jaringan.

3. Layanan ketiga dalam sebuah honeypot adalah High Interaction Honeypot. Dengan High Interaction Honeypot, administrator jaringan tidak perlu lagi mengawasi intrusi karena server asli telah direplikasi secara keseluruhan. Akibatnya, penyerang bebas menyerang server replika yang diisi dengan informasi palsu. Hal ini memungkinkan penyerang merasa puas dengan informasi yang mereka peroleh secara ilegal, meskipun server masih sepenuhnya aman.

C. *Cowrie*

Cowrie adalah perangkat lunak pendukung yang digunakan untuk menyamarkan layanan di server openssh dan mempermudah inisialisasi honeypot. Cowrie dikenang karena klasifikasi tipe asosiasi honeypot menengah, yang digunakan untuk membedakan dan mencatat kekuatan monster setelah itu menyerang server ssh, telnet dan openssh. Ide yang digunakan oleh Cowrie adalah menjual, yaitu setelah dibuka dengan baik, Cowrie akan memandu penyerang untuk masuk ke administrasi honeypot palsu. sehingga meskipun penyerang hanya masuk ke perangkap honeypot, mereka akan percaya bahwa serangan mereka berhasil. Ada baik log atau masuk fitur cowrie. Logging adalah sebuah siklus untuk merekam semua jenis pergerakan yang dilakukan oleh agresor yang terjadi pada kerangka palsu (honeypot). Dengan tujuan agar pengurus organisasi dapat mengetahui kegiatan apa saja yang dilakukan oleh agresor pada kerangka palsu tersebut [2].

D. *Virtual Private Server (VPS)*

Mesin virtual yang ditawarkan sebagai layanan oleh penyedia hosting dikenal sebagai server pribadi virtual (VPS). Dalam VPS, pengguna memiliki akses dan mengelola semua aspek

perangkat lunak server, termasuk akses administrator ke sistem operasi dan aplikasi yang akan diimplementasikan di server. Setiap Mesin Virtual (VM) adalah "server Virtual" yang dapat dikonfigurasi dengan sistem operasinya sendiri. Server pribadi virtual (VPS) dapat dipecah menjadi beberapa VM.

VPS memiliki kesan dedicated server. Menyewa virtual private server (VPS) memiliki resource yang lebih baik dibandingkan shared hosting, sehingga Anda tidak akan terganggu jika website yang dikelola bermasalah. Selain itu, VPS diberikan akses root, memungkinkannya menyesuaikan server dengan lebih mudah. [6].

E. Secure Socket Shell (SSH)

SSH (Secure Shell atau Secure Attachment Shell) adalah konvensi organisasi yang memberi klien, terutama eksekutif kerangka kerja, metode yang aman untuk mengakses PC dalam organisasi yang goyah. SSH juga merupakan nama untuk bermacam-macam utilitas yang memanfaatkan konvensi SSH. Secure Shell memastikan komunikasi data yang andal antara dua komputer yang terhubung melalui jaringan terbuka seperti internet dengan menyediakan enkripsi yang kuat, kata sandi yang aman, dan autentikasi kunci publik.

Administrator jaringan sering menggunakan SSH untuk menyaring aplikasi dan sistem dari jarak jauh, memberdayakan mereka untuk mengeksekusi perintah, memindahkan dokumen, dan mengakses PC lain di dalam asosiasi. SSH adalah nama untuk konvensi organisasi kriptografi dan utilitas yang menggunakannya.

Model klien-server SSH digunakan untuk menautkan aplikasi klien Secure Shell, yang merupakan titik akhir tempat sesi ditampilkan, dan server SSH, yang merupakan titik akhir tempat sesi berjalan. Tampilan aplikasi yang digunakan untuk transfer file atau emulasi terminal seringkali didukung oleh implementasi SSH.[7].

F. Web Server

Server web adalah perangkat lunak yang berjalan di server dan bertanggung jawab untuk mengirimkan hasil dalam bentuk halaman web — biasanya dokumen HTML — sebagai tanggapan atas permintaan halaman web yang dibuat oleh klien yang dikenal sebagai browser web melalui HTTP atau HTTPS. Dengan cara ini, dapat diasumsikan bahwa kemampuan server web sebagai server luar biasa yang melayani klien web seperti Mozilla, Chrome, Web Explorer, Show, Safari, dan lainnya, sehingga proyek ini dapat menampilkan halaman atau data yang disebutkan oleh klien.[8].

G. Port Scanning

Pemindaian port adalah metode yang digunakan untuk mengidentifikasi port terbuka di jaringan komputer, yang dapat dimanfaatkan oleh penyerang. Dengan memeriksa hasil pemindaian port, kerentanan dalam sistem jaringan dapat ditemukan. Hasil pemindaian port memberikan informasi seperti daftar port terbuka dan tertutup, daftar layanan yang berjalan di port tersebut, bahkan jenis dan versi sistem operasi yang digunakan.[9].

H. Distributed Denial of Service (DDoS)

Serangan "Denial of Service" (DOS) bertujuan untuk merusak atau mengganggu layanan. Serangan DDoS menggunakan sumber daya server sedemikian rupa sehingga tidak dapat diakses atau digunakan untuk tujuan yang dimaksudkan. Serangan DDoS biasanya dilakukan oleh agresor dengan membanjiri lalu lintas dengan banyak bundel yang dikirim dari server. Dalam nada yang sama, ada berbagai macam cara serangan DoS dilakukan, mulai dari transfer yang lamban hingga serangan paket SYN dan serangan paket ICMP hingga keamanan aplikasi, mengirimkan banyak permintaan layanan, dan serangan DDoS gabungan atau super tahan lama. [10].

I. Bruteforce Attack

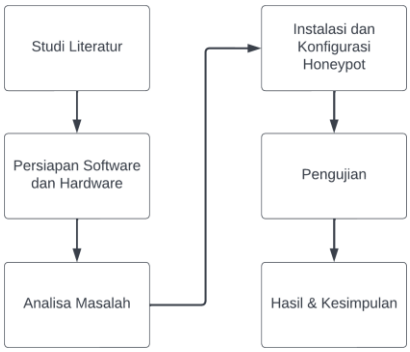
Brute force adalah jenis serangan yang mencoba setiap kemungkinan kombinasi karakter untuk mendapatkan skor besar. Ini adalah salah satu teknik paling umum untuk memecahkan kata sandi. [1].

Serangan kekuatan biadab adalah metodologi yang jelas dan langsung yang digunakan oleh pemrogram untuk memperoleh izin masuk yang tidak disetujui ke suatu kerangka kerja. Untuk menyelesaikan atau menemukan peretasan kata sandi yang valid, serangan tersebut melibatkan percobaan sistematis semua kemungkinan kombinasi kata sandi dengan memasukkan karakter dan panjang kata sandi tertentu. Itu berusaha untuk sepenuhnya menggabungkan kata sandi ini.

Dengan mencari kelemahan keamanan dan memanfaatkan kelemahan keamanan kata sandi, peretas menggunakan kekerasan untuk mendapatkan akses. Dengan kesembronoan serangan ini, pada akhirnya akan cocok dengan kombinasi dan panjang karakter yang digunakan dalam nama pengguna. Sampai menemukan ekspresi misteri tertentu. Korban disarankan untuk membuat kata sandi yang rumit dengan menambah panjang dan kerumitan kata sandi mereka untuk menghindari serangan brute force, yang sangat berbahaya. [11].

Metode Penelitan

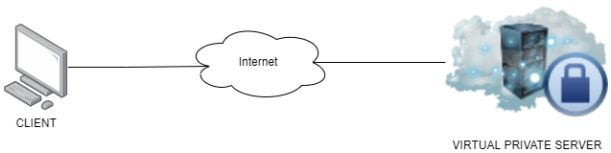
Metode penelitian yang dimulai dari studi literatur hingga hasil & kesimpulan atau lebih lengkapnya bisa dilihat pada Gambar 1, dengan hal tersebut penelitian dapat dilakukan lebih terstruktur dari tahap awal hingga selesai serta tercapai target yang telah ditargetkan diawal.



Gambar 1. Diagram Alir Penelitian

Analisa Masalah

Pada penelitian ini menggunakan *Virtual Private Server (VPS)* untuk menganalisa keamanan yang ada, karena *Virtual Private Server (VPS)* memiliki kekurangan salah satunya dari segi sisi keamanan *server.* Adapun solusi untuk mengamankan *Virtual Private Server (VPS)* diperlukan sebuah Aplikasi untuk meminimalisir serangan pada *Virtual* Private Server *(VPS).* Aplikasi pendeteksi yang diperlukan adalah aplikasi yang dapat mendeteksi serangan yang terjadi pada *Virtual Private Server (VPS)* yaitu aplikasi *Honeypot.* Adapun bentuk topologi dari *Virtual Private Server (VPS)* dapat dilihat pada Gambar 2.



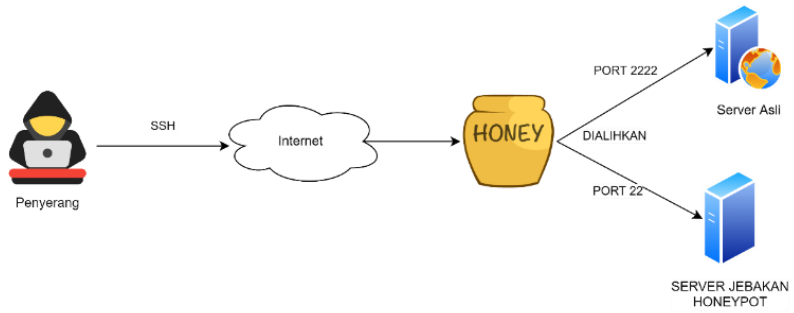
Gambar 2. Topologi *VPS*

Instalasi dan Konfigurasi *Honeypot*

Setelah pada tahapan analisa masalah pada penelitian ini, maka selanjutnya merupakan tahapan instalasi dan konfigurasi *honeypot* yang di implementasikan pada *Virtual Private Server*

(VPS), sebelum melakukan instalasi *honeypot cowrie* kita perlu untuk menginstal beberapa paket pendukung seperti *python virtual environments.*

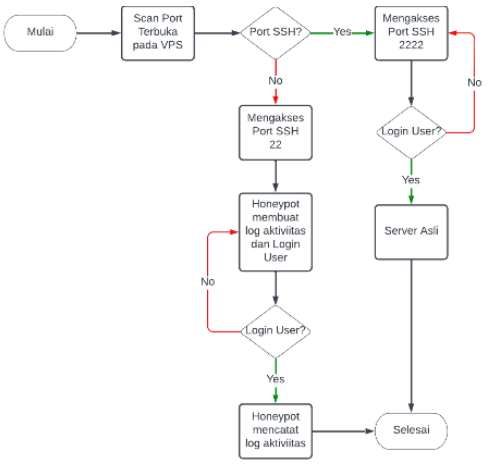
Setelah melakukan instalasi, selanjutnya melakukan konfigurasi pada *port ssh*, *default port ssh* adalah 22 maka akan kita ganti ke port 2222, karena *port ssh 22* akan digunakan untuk *server* jebakan *honeypot*. Adapun bentuk topologi dari *honeypot* pada Gambar 3.



Gambar 3. Topologi Honeypot

Skenario Pengujian Sistem

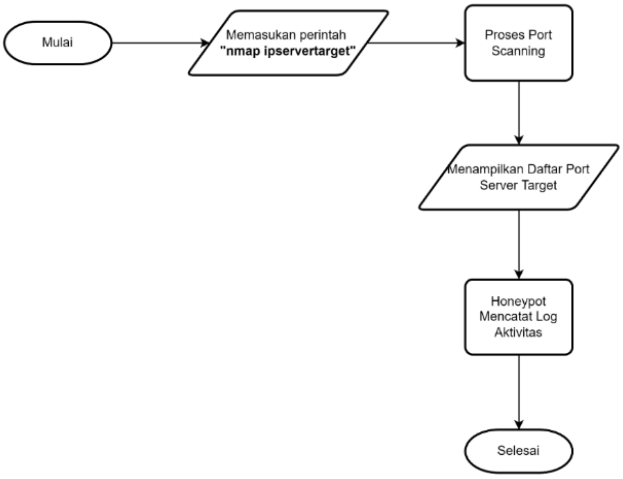
Honeypot merupakan *server* tiruan atau jebakan yang berfungsi mengalihkan dan menipu penyerang, yang mana ketika penyerang telah berhasil menyerang *server* yang asli namun penyerang dialihkan ke *server* tiruan lalu aktifitas penyerang tersebut akan tercatat pada log aktifitas. Log aktifitas tersebut dapat dilihat oleh administrator *server* pada *server* sebenarnya, dan dapat menjadi rujukan untuk mempelajari serangan serta memperbaiki celah yang ada pada *server* yang dimiliki. Adapun diagram alir dari pengujian sistem dapat dilihat pada Gambar 4.



Gambar 4. Diagram Alir Pengujian Sistem

Skenario Pengujian *Port Scanning*

Prosedur pengujian port yang terbuka pada VPS menggunakan alat bernama NMAP. NMAP memanfaatkan IP mentah yang canggih, yang memiliki metode operasi yang canggih, untuk mengidentifikasi host aktif. NMAP juga dapat digunakan untuk melihat port host dinamis, baik dengan menggeser, menutup atau membuka. Adapun diagram alir dari *port scanning* dapat dilihat pada Gambar 5.



Gambar 5. Diagram Alir Pengujian Port Scanning

Gambar 5 merupakan alur dari pengujian *port scanning*, adapun alurnya sebagai berikut :

1. Memasukan perintah *nmap ipservertarget* pada aplikasi *Nmap* ataupun *command line terminal linux.*
2. Tunggu beberapa saat untuk proses dari *scanning port server* target.
3. Setelah itu maka akan tampil daftar *list port server* target.
4. *Cowrie* mendeteksi aktivitas serangan yang terjadi, lalu mencatat ke dalam *log file.*

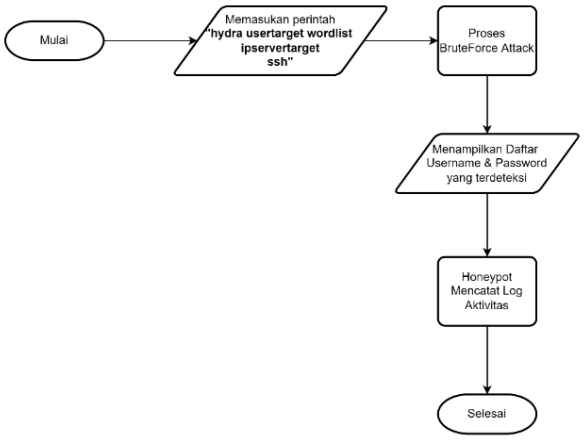
Skenario Pengujian *Brute Force Attack*

Setiap kata sandi yang mungkin dicoba dalam metode serangan Brute Force. Hydra adalah nama alat yang digunakan dalam serangan brute force. Alat ini bekerja dengan mengirimkan banyak nama pengguna dan kata sandi untuk mencoba menebak dengan benar nama pengguna dan kata sandi pengguna ssh. Jumlah waktu yang diperlukan untuk mengetahui nama pengguna dan kata sandi tergantung pada seberapa rumit nama pengguna dan kata sandi yang digunakan administrator server. Adapun diagram alir dari *brute force attack* dapat dilihat pada Gambar 6.

Gambar 6 merupakan alur dari pengujian *bruteforce attack*, adapun alurnya sebagai berikut :

1. Memasukan perintah *hydra usertarget wordlist ipservertarget ssh* pada *command line terminal linux.*
2. Tunggu beberapa saat untuk proses dari *bruteforce attack server* target.
3. Setelah itu maka akan tampil daftar *list user* yang berisikan *username* dan *password* dari *server* target.

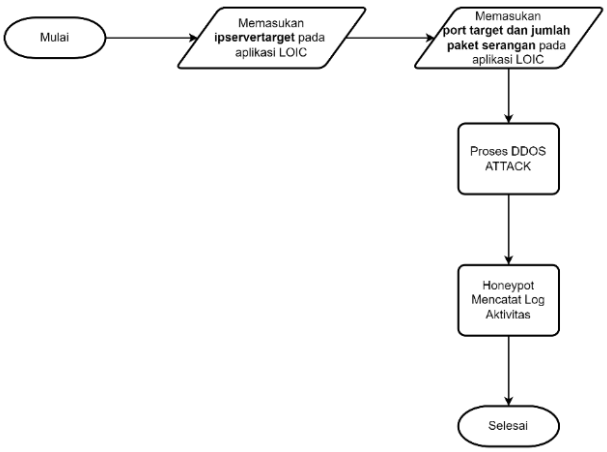
Cowrie mendeteksi aktivitas serangan yang terjadi, lalu mencatat ke dalam *log file.*



Gambar 6. Diagram Alir Pengujian BruteForce Attack

Skenario Pengujian *Distributed Denial of Service Attack*

Teknik serangan *Distributed Denial of Service* atau *DDoS* dilakukan dengan cara menyerang *resource* dari komputer / *server* target sampai target tersebut tidak dapat menjalankan layanan yang dimiliki dengan baik. Adapun *tools* yang digunakan untuk melakukan serangan *DDoS* bernama *LOIC*. *Server* yang telah memiliki *honeypot* akan menangkap serta mencatat serangan *DDoS* yang dilakukan oleh penyerang ke *server* target. Adapun tujuan dari serangan *DDoS* untuk mengetahui performa dari *server* dalam melayani *service* yang ada walaupun sedang mengalami serangan *DDoS*. Adapun diagram alir dari *DDoS Attack* dapat dilihat pada Gambar 7.



Gambar 7. Diagram Alir Pengujian Distributed Denial of Service Attack

Gambar 7 merupakan alur dari pengujian *DDoS* adapun alurnya sebagai berikut :

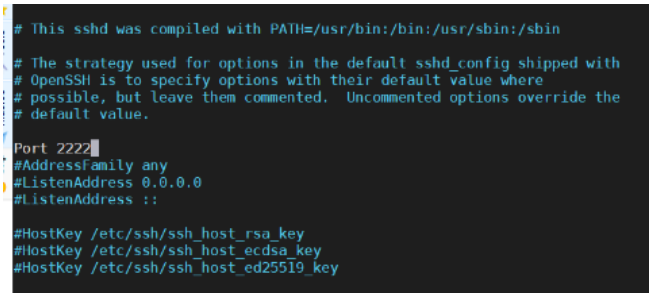
1. Memasukan *ipaddress server* target pada aplikasi *LOIC*.
2. Memasukan *port server* target dan jumlah paket serangan.
3. Tunggu beberapa saat untuk proses dari *DDoS Attack* ke *server* target.
4. *Cowrie* mendeteksi aktivitas serangan yang terjadi, lalu mencatat ke dalam *log file*

3. Hasil

Tahapan ini akan membahas implementasi atau konfigurasi *cowrie* pada *VPS* seperti pada Gambar 3 topologi *honeypot*, lalu setelah itu maka akan dilanjutkan tahapan pengujian mulai dari pengujian sistem sampai dengan pengujian serangan pada *VPS*

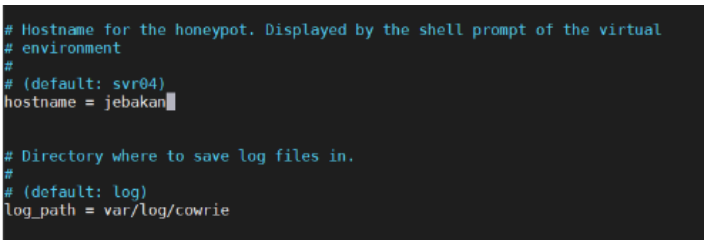
Implementasi *Cowrie*

Yang pertama dilakukan yaitu melakukan konfigurasi pada *port ssh* dengan cara mengganti *default port ssh* dengan cara ketikan perintah *nano /etc/ssh/sshd_config* lalu cari *port 22*, dan hapus tanda # didepannya lalu ganti menjadi 2222 setelah itu simpan *file config* atau bisa dilihat pada Gambar 8.



Gambar 8. Konfigurasi Port SSH

Selanjutnya melakukan konfigurasi pada *file config cowrie*, untuk membuka *file config cowrie* bisa dengan mengetikan perintah *nano /etc/cowrie.cfg* maka akan tampil *file config cowrie*, yang pertama diganti ialah *hostname* yang mana *default svr04* bisa diubah menjadi jebakan ataupun yang lainnya karena *hostname* akan menjadi identitas dari *honeypot cowrie* nantinya atau bisa dilihat pada Gambar 9.

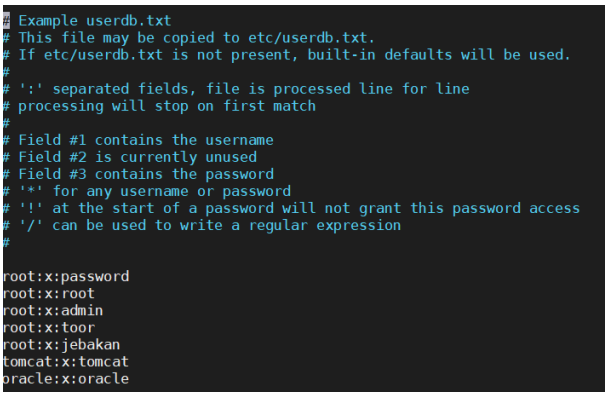


Gambar 9. Mengubah Hostname Honeypot

Selanjutnya mengubah *default listens port ssh*, dengan cara mencari kalimat *listen_endpoint* didalam *file config cowrie*, apabila sudah ketemu ganti dari *port 2222* menjadi 22 karena *port 22* nantinya akan menjadi *port* atau *server* jebakan *honeypot* atau bisa dilihat pada Gambar 10

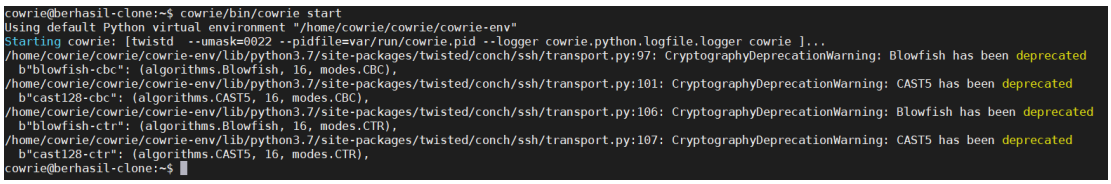
Gambar 10. Mengubah Listen Endpoints

Lalu selanjutnya membuat list *username* dan *password* pada *file userdb.txt*, yang mana *username* dan *password* tersebutlah yang diizinkan masuk atau mengakses *server* jebakan nantinya, untuk list *username* dan *password* bisa dilihat pada Gambar 11.



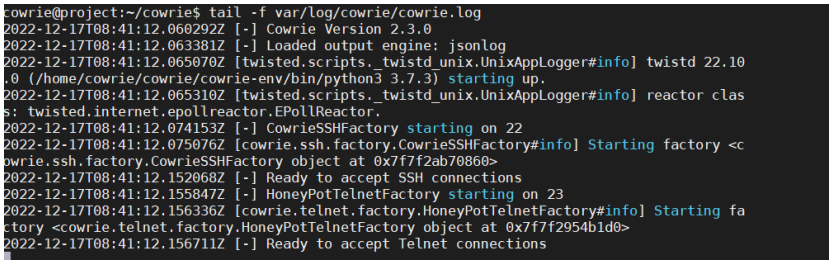
Gambar 11. File userdb.txt

Lalu lakukan *service restart* pada *cowrie* dengan perintah *cowrie/bin/cowrie restart* untuk menjalankan *cowrie* atau bisa dilihat pada Gambar 12.



Gambar 12. Service Restart Cowrie

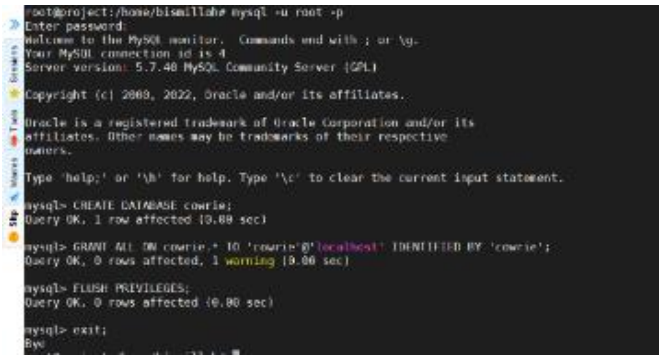
Terakhir bisa dilihat juga pada *file log cowrie* untuk melihat apakah *cowrie* telah berjalan dengan benar atau tidak, dapat dilakukan dengan ketik perintah *tail -f /home/cowrie/cowrie/var/log/cowrie/cowrie.log* atau bisa dilihat pada Gambar 13.



Gambar 13. Hasil File Log Cowrie

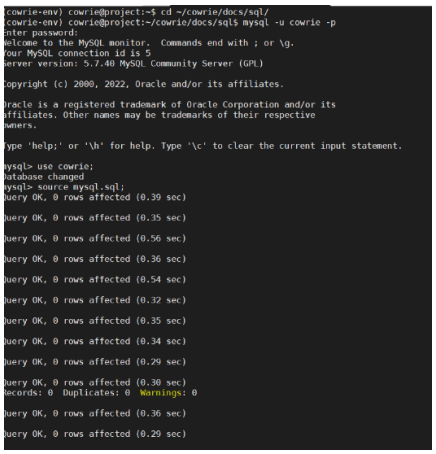
Implementasi Kippo-Graph

Yang pertama dilakukan untuk implementasi *kippo-graph* ialah membuat database untuk menampung aktivitas serangan yang tersimpan di dalam *file log cowrie* atau bisa dilihat pada Gambar 14.



Gambar 14 Membuat Database Kippo-Graph

Lalu selanjutnya melakukan sinkronisasi antar *file log cowrie* dengan *database* yang telah dibuat sebelumnya untuk caranya bisa dilihat pada Gambar 15.



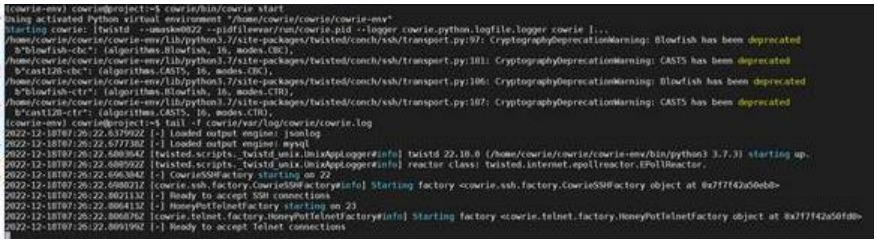
Gambar 15. Sinkronisasi File Config Cowrie

Selanjutnya mengubah *file config cowrie* kembali pada bagian *output mysql* dengan database yang telah dibuat sebelumnya, atau bisa dilihat pada Gambar 16.

```
# MySQL logging module
# Database structure for this module is supplied in docs/sql/mysql.sql
#
# MySQL logging requires extra software: sudo apt-get install libmysqlclient-dev
# MySQL logging requires an extra Python module: pip install mysql-python
#
[output_mysql]
enabled = true
host = localhost
database = cowrie
username = cowrie
password = cowrie
port = 3306
debug = false
```

Gambar 16. File Config Cowrie

Selanjutnya dapat lakukan *start* kembali pada *file config cowrie* untuk melihat apakah *loaded output mysql* telah berhasil terhubung atau bisa dilihat pada Gambar 17.



Gambar 17. Hasil File Log Cowrie

Selanjutnya melakukan konfigurasi juga pada *file config kippo-graph* untuk menyesuaikan dengan database yang telah dibuat sebelumnya atau bisa dilihat pada Gambar 4.18.

```
# MySQL server configuration: you will have to change the following
# four definitions from the default values to the correct ones,
# according to your MySQL server instance. When you installed Kippo/Cowrie
# and configured MySQL logging, you should have created a new
# MySQL user just for this job, otherwise use root (not recommended!)
define('DB_HOST', 'localhost');
define('DB_USER', 'cowrie');
define('DB_PASS', 'cowrie');
define('DB_NAME', 'cowrie');
define('DB_PORT', '3306');

# Which geolocation method should be used -- Default: LOCAL (MaxMind)
# Note: LOCAL (MaxMind) enables additional fields in various components.
#  When using LOCAL you might want to periodically update (monthly) the
#  kippo-graph/include/maxmind/GeoLite2-City.mmdb MaxMind database file
#  with a new one from: http://dev.maxmind.com/geoip/geoip2/geolite2/
# Available options:
#  LOCAL: fastest, uses a local MaxMind GeoLite2 database
#  GEOPLUGIN: uses the geoplugin.com web service (online)
define('GEO_METHOD', 'LOCAL');
```

Gambar 18. File Config Kippo-Graph

Terakhir dapat melihat *kippo-graph* berhasil atau tidak dengan mengakses *web interface kippo-graph* dengan cara membuka *web browser* lalu ketikan *ip-addressserver/kippo-graph* atau bisa dilihat pada Gambar 19.

Gambar 19. Tampilan Kippo-Graph

VPS yang telah terimplementasi *honeypot cowrie* dan *kippo-graph* mampu mendeteksi serangan yang terjadi pada *VPS*. Setelah dilakukan pengujian sebelumnya dengan menggunakan

3 jenis serangan yaitu *port scanning*, *bruteforce password*, dan *denial of service attack*, untuk mengukur tingkat akurasi dalam mendeteksi serangan dapat diukur dengan menggunakan rumus :

$$akurasi = \frac{\text{jumlah data pengujian yang } terdeteksi}{\text{jumlah data pengujian}} \; x \; 100\% \quad (1)$$

Adapun hasil pengujian dapat dilihat pada Tabel 1.

Table 1. Hasil Pengujian Serangan

No	Serangan	Nama Pengujian	Awal	Akhir	Hasil
1	Individual	*PS*	2%	2%	Terdeteksi
2		*BFA*		48%	Terdeteksi
3		*DDoS*		25%	Terdeteksi
4	Double	*PSc & BFAttack*		56%	Terdeteksi
5		*BFAttack & DDoS*		58%	Terdeteksi
6		*DDoS & PS*		51%	Terdeteksi
7	Multiple	*PS, BFA, & DDoS*		97%	Terdeteksi

Keterangan :
PS : Port Scanning
BFA : Bruteforce Attack
DDoS : Distributed Denial of Service Attack
Awal : Kondisi awal *server* sebelum terjadi serangan
Akhir : Kondisi ketika *server* dilakukan serangan
Hasil pengujian menggunakan jenis serangan *port scanning*, adapun hasil yang didapatkan adalah :

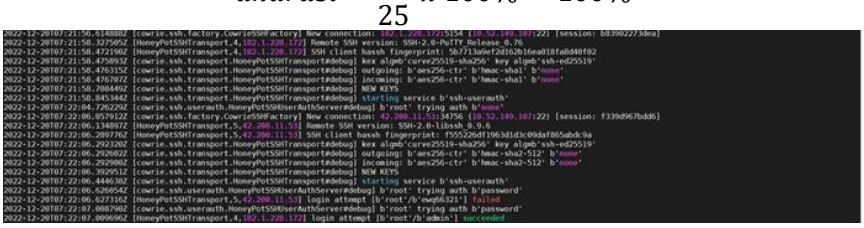
$$akurasi = \frac{25}{25} \; x \; 100\% = 100\%$$

Hasil pengujian menggunakan jenis serangan *brute force attack*, adapun hasil yang didapatkan adalah :

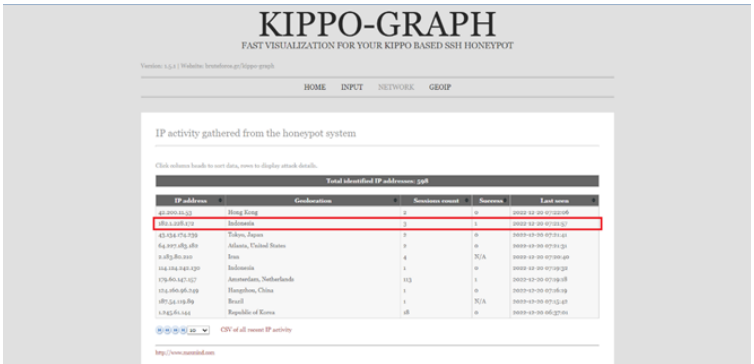
$$akurasi = \frac{25}{25} \; x \; 100\% = 100\%$$

Hasil pengujian menggunakan jenis serangan *denial of service attack*, adapun hasil yang didapatkan adalah :

$$akurasi = \frac{25}{25} \; x \; 100\% = 100\%$$



Gambar 20. Hasil Serangan Yang Tercatat di File Log Cowrie



Gambar 21. Hasil Serangan Yang Tercatat di Website Kippo-Graph

Berdasarkan hasil pengujian serangan yang dilakukan 25 kali setiap pengujian serangan yang ada, yang dimulai dari *port scanning, bruteforce attack,* maupun *DDoS attack,* baik dilakukan pengujian secara individual, *double* maupun *multiple, cowrie* dapat mendeteksi serta mencatat semua serangan terjadi dan *cowrie* dapat dinilai maupun bekerja dengan sangat baik dan akurat, dengan data yang ditunjukan pada *file log cowrie* pada Gambar 20, *web kippo-graph* pada Gambar 21, serta tabel pengujian serangan tabel 1, maka dapat disimpulkan juga berdasarkan 3 pola jenis pengujian yaitu :

1. Pengujian penyerangan secara individual atau berurutan, *cowrie* dapat mendeteksi dan mencatat semua serangan yang terjadi dan untuk *server* masih bisa bekerja dan digunakan dengan baik tidak mengalami *down* ataupun *lag* ketika diakses
2. Lalu selanjutnya pengujian penyerangan secara *double* atau melakukan 2 jenis serangan sekaligus ke *server, cowrie* pun tetap dapat berhasil mendeteksi dan mencatat semua serangan yang terjadi, namun terjadi peningkatan kinerja pada *cpu* yang tidak signifikan akibat banyaknya *traffic* masuk secara bersamaan, namun *VPS* masih dapat digunakan dan diakses dengan baik walaupun sedikit mengalami *down* ketika diakses.
3. Lalu yang yang terakhir pengujian penyerangan secara *multiple* atau melakukan 3 jenis serangan sekaligus, sama seperti sebelumnya *cowrie* juga dapat mendeteksi dan mencatat semua serangan yang terjadi, namun terdapat peningkatan yang signifikan pada kinerja *cpu* yang menyentuh angka 94% yang mengakibatkan ketika mengakses *server* terjadi *down* dan *lag* hal tersebut terjadi karena *server* banyak dibanjiri *traffic* oleh 3 jenis serangan tersebut.

Dengan terimplementasinya *cowrie* dan *kippo-graph* dapat membantu *administrator server* dalam melakukan pengambilan tindakan apabila terjadinya serangan atau penyusupan oleh *attacker* kedalam *server*, serta untuk mengatasi *down* atau *lag* pada *server* ketika terjadi banyak serangan pada *server* dapat meningkatkan spesifikasi *server* agar mampu menahan *traffic* yang tinggi.

4. Kesimpulan

Berikut kesimpulan penulis mengenai implementasi honeypot sebagai pendeteksi serangan pada virtual private server (VPS) berdasarkan temuan penelitian yang telah dijelaskan pada bab sebelumnya:

1. Pemindaian port, serangan denial-of-service, dan serangan kata sandi brute force semuanya dapat dideteksi oleh sistem cowrie honeypot yang telah diimplementasikan pada VPS, baik pengujian secara individual, ganda, atau berkali-kali.
2. Dengan antarmuka web kippo-graph, administrator akan dapat memantau serangan yang terjadi pada VPS dan mengambil tindakan pengamanan jika terjadi serangan lalu lintas tinggi pada VPS. Memiliki cowrie dapat membantu mengamankan VPS dengan mendeteksi semua aktivitas serangan yang terjadi pada server honeypot trap. Aktivitas yang terjadi di server trap dicatat di log cowrie.

Daftar Pustaka

[1] M. Syani, "Implementasi Intrusion Detection System (Ids) Menggunakan Suricata Pada Linux Debian 9 Berbasis Cloud Virtual Private Servers (Vps)," *J. Inkofar*, vol. 1, no. 1, pp. 13–20, 2020, doi: 10.46846/jurnalinkofar.v1i1.155.

[2] W. A. Sulaksono and C. E. Suharyanto, "Implementasi Honeypot Sebagai Sistem Keamanan Jaringan Pada Virtual Private Server," *InfoTekJar J. Nas. Inform. dan Teknol. Jar.*, vol. 5, no. 1, pp. 90–95, 2020.

[3] A. R. Gunawan, N. P. Sastra, and D. M. Wiharta, "Penerapan Keamanan Jaringan Menggunakan Sistem Snort dan Honeypot Sebagai Pendeteksi dan Pencegah Malware," *Maj. Ilm. Teknol. Elektro*, vol. 20, no. 1, p. 81, 2021, doi: 10.24843/mite.2021.v20i01.p09.

[4] R. E. Susanti, A. W. Muhammad, and W. A. Prabowo, "Implementasi Intrusion Prevention System (IPS) OSSEC dan Honeypot Cowrie," *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 11, no. 1, pp. 73–78, 2022, doi: 10.32736/sisfokom.v11i1.1246.

[5] Y. M. Abdussyakur, A. Z. Mardiansyah, and A. H. Jatmika, *Optimasi Port Knocking dan Honeypot Menggunakan IPTables Sebagai Keamanan Jaringan pada Server*, vol. 3, no. 2. 2021.

[6] A. P. Sujana, "Analisis PVS Cloud pada Database Server," *Komputika J. Sist. Komput.*, vol. 6, no. 2, pp. 75–82, 2019, doi: 10.34010/komputika.v6i2.1710.

[7] P. D. Oktaviansyah, "Penerapan Sistem Pengamanan Port pada Mikrotik Menggunakan Metode Port Knocking," *NetPLG J. Netw. Comput. Appl.*, vol. 1, no. 2, pp. 13–24, 2022.

[8] Wiwarno, "RANCANG BANGUN WEB HOSTING MENGGUNAKAN DOCKER CONTAINER DAN CLUSTERING PADA COREOS : CLUSTERING," Universitas Muhammadiyah Malang, 2017.

[9] D. D. Rahmadani, "ANALISIS PERBANDINGAN PERFORMANSI INTRUSION DETECTION SYSTEM (IDS) SNORT DAN SURICATA DALAM MENDETEKSI SERANGAN JARINGAN PADA KOMPUTER," Purwokerto, 2021.

[10] A. S. Fadhlillah, N. Bogi, and A. I. Irawan, "Analisis Performansi Ids Menggunakan Metode Deteksi Anomaly-Based Terhadap Serangan Dos," *e-Proceeding Eng.*, vol. 6, no. 2, pp. 3398–3405, 2019.

[11] D. M. J. Putra, I. N. N. Y. Anantra, P. A. Kusuma, P. D. J. Pratama, G. A. J. Saskara, and I. M. E. Listarrtha, "ANALISIS PERBANDINGAN SERANGAN HYDRA, MEDUSA DAN NCRACK PADA PASSWORD ATTACK," *(Jurnal Inform. Teknol. dan Sains*, vol. 4, no. 4, pp. 461–466, 2022.