Terbit *online* pada laman : http://teknosi.fti.unand.ac.id/

Jurnal Nasional Teknologi dan Sistem Informasi

| ISSN (Print) 2460-3465 | ISSN (Online) 2476-8812 |

Implementasi *Low Interaction Honeypot* Untuk Peningkatan Keamanan Server dan Analisa Serangan Pada Protokol SSH

Naufal Arkaan[a], Dolly Virgian Shaka Yudha Sakti[b]

[a,b]*Teknik Informatika, Universitas Budi Luhur, Jl. Ciledug Raya, Petukangan Utara, Jakarta Selatan, 12260. DKI Jakarta, Indonesia*

INFORMASI ARTIKEL

Sejarah Artikel:
Diterima Redaksi: 02 Juli 2019
Revisi Akhir: 17 September 2019
Diterbitkan *Online*: 19 September 2019

KATA KUNCI

honeypot,
brute force,
low interaction,
Secure Shell,
network programming

KORESPONDENSI

E-mail: 1511502054@student.budiluhur.ac.id

ABSTRACT

Faktor keamanan pada teknologi informasi saat ini sangatlah penting, dikarenakan pada zaman yang semakin berkembang data merupakan segalanya. Ancaman serangan terhadap jaringan dan server juga ikut berkembang, maka diperlukan adanya sebuah penanganan terhadap ancaman yang dapat memantau dan menganalisis ancaman serangan yang sedang berlangsung pada server. *Honeypot* merupakan salah satu solusi yang dapat diberikan karena merupakan sebuah sistem umpan atau aplikasi simulasi yang dapat digunakan untuk memikat penyerang dengan menyamarkan diri sebagai sistem yang rentan. *Honeypot* dapat digunakan untuk memantau dan menganalisis kegiatan penyerang yang tertangkap di *honeypot*. *Honeypot* ini berjenis *low interaction* yang dibuat menggunakan bahasa pemrograman *python* yang memanfaatkan konsep *network programming* sebagai dasarnya dan juga *library paramiko* untuk mengimplementasi protokol SSH yang digunakan. *Honeypot* berjalan di server nantinya mengakses port pada protokol SSH asli yang biasa diakses dan diserang oleh penyerang dan juga dapat menganalisis maupun memantau penyerang yang mengancam pada server. Tujuan penelitian ini adalah untuk menganalisa perilaku apa yang dilakukan penyerang di dalam server dan juga kemungkinan kemungkinan kredensial yang digunakan oleh penyerang, dengan begitu hasil dari serangan sebagai pembelajaran untuk administrator server agar server yang dikelola lebih aman.

1. PENDAHULUAN

Kebutuhan akan jaringan komputer semakin bertambah penting, baik dalam pendidikan, pekerjaan maupun dalam sebuah permainan, salah satu hal penting dalam mengelola jaringan komputer yaitu keamanan dari jaringan itu sendiri, dengan banyaknya akses ke jaringan tersebut maka akan banyak pula peluang kejahatan yang terjadi didalam jaringan tersebut, misalkan adanya pencurian data yang terjadi di jaringan tersebut ataupun adanya peretas yang mematikan sumber daya jaringan tersebut[1]. Kombinasi dan perpaduan keamanan software dan perangkat hardware merupakan solusi keamanan yang komprehensif. Keamanan pada jaringan komputer sebagai bagian dari sebuah sistem yang sangat penting untuk menjaga validitas dan integritas data serta menjamin ketersediaan layanan bagi penggunanya. Keamanan pada jaringan komputer harus dilindungi dari segala macam serangan dan usaha-usaha penyusupan atau pemindaian oleh pihak yang tidak berhak.

Maka dari itu perlu adanya penanganan terhadap ancaman yang dapat memantau dan menganalisis ancaman serangan yang sedang berlangsung tanpa menyentuh dan merusak server. *Honeypot* merupakan salah satu solusi yang dapat diberikan karena honeypot merupakan sebuah sistem umpan atau aplikasi simulasi yang mensimulasikan seluruh jaringan untuk memikat penyerang dengan menyamarkan diri sebagai sistem yang rentan[2][3], [4]. Pada beberapa penelitian, implementasi *honeypot low interaction* memanfaatkan dua aplikasi yang berbeda, yaitu *Dionaea* dan *Honeyd* berhasil membuat layanan palsu sebagai target serangan dan mencatat aktivitas yang dianggap dapat membahayakan sistem dan jaringan, namun tidak adanya interaksi lebih lanjut ketika penyerang berhasil mengeksploitasi dan masuk dalam *honeypot*[5]–[8]. Penelitian lain membahas tentang pembuatan *honeypot* menggunakan

https://doi.org/10.25077/ TEKNOSI.v5i2.2019.112-120

bahasa pemrograman *python* yang memanfaatkan package *Scapy* dan aplikasi *T-Shark* yang bertujuan untuk menangkap serangan pada port SMB (445) mensimulasikan pada kerentanan serangan SAMBA, memanfaatkan port 22 untuk menangkap serangan SSH, dan memonitornya tetapi hanya sebatas interaksi terhadap port saja tanpa ada sistem yang menampung penyerang masuk kedalam SSH[9]. Berdasarkan penelitian yang sudah dilakukan tentang bagaimana *honeypot* berjenis *low interaction* diimplementasikan dan juga dibuat, maka akan dilakukan penelitian aplikasi *honeypot* pada server.

Aplikasi *honeypot* berjalan di server yang nantinya dapat menyembunyikan *service* port SSH asli yang biasa diakses dan diserang oleh penyerang dan juga membuat *service* port SSH palsu yang mampu menipu dan memantau penyerang yang mengancam pada server. Sehingga dengan honeypot, server dapat terhindar dan aman dari serangan yang dilakukan oleh penyerang. *Honeypot* juga meningkatkan keamanan sistem operasi. *Honeypot* dapat mendeteksi serangan dan mencatat serangan yang dilakukan penyerang sehingga catatan tersebut dapat dianalisa. Penyerang terhubung ke dalam port SSH palsu yang ada di *honeypot*. Ketika penyerang berhasil masuk, maka honeypot akan segera mengirimkan email pemberitahuan kepada administrator server.. *Honeypot* akan mencatat ip, port, dan aksi yang dilakukan penyerang. Setelah ada banyak penyerang yang masuk ke dalam SSH palsu dan melakukan aksinya, maka analisa dapat dilakukan dengan melihat informasi dari catatan yang dibuat oleh *honeypot*. *Honeypot* adalah sebuah sistem layanan palsu yang berfungsi untuk menjebak penyerang. Umumnya seorang penyerang di dunia jaringan mempunyai tujuan buruk yaitu melakukan pencurian atau perusakan data. *Honeypot* digunakan untuk menangkal usaha-usaha yang dapat merugikan sistem atau layanan[6]. *Honeypot* ini mengandung kerentanan sistem yang membuatnya menjadi target menarik bagi penyerang[3]. *Honeypot* berguna untuk pengalihan agar penyerang masuk ke server palsu dan dapat melihat log/aktivitas penyerang terhadap server[10]. Kegunaan *honeypot* lainnya adalah untuk mengetahui metodologi yang digunakan penyerang dalam menguasai sistem dan untuk mengumpulkan informasi sebagai bukti forensik[11].

Honeypot juga memiliki banyak manfaat seperti : (1) mitigasi risiko, (2) berfungsi seperti IDS, (3) untuk mencari tahu strategi serangan yang dilakukan oleh penyerang, (4) mengidentifikasi pelaku penyerangan serta melakukan klasifikasi, (5) sebagai bukti hukum untuk menuntut penyerang, (6) bermanfaat untuk riset yang dapat mengetahui teknik dan eksploitasi terbaru.

Honeypot terbagi dua jenis berdasarkan kegunaannya yaitu :

a. *Honeypot* produksi

Honeypot produksi berjenis low interaction biasanya digunakan pada server produksi oleh suatu organisasi atau perusahaan[12]

b. *Honeypot* penelitian

Honeypot penelitian biasanya digunakan untuk penelitian serangan yang dilakukan komunitas *black hat* di berbagai jaringan. *Honeypot* penelitian lebih rumit untuk digunakan daripada *honeypot* produksi karena menangkap informasi lebih banyak dan biasanya digunakan oleh penelitian, militer, dan organisasi pemerintahan[13].

Honeypot memiliki klasifikasi berdasarkan tingkat interaksinya. Tingkat interaksi dapat didefinisikan sebagai tingkat aktivitas penyerang ke dalam sistem *honeypot*, semakin tinggi tingkat aktivitas penyerang maka semakin tinggi pula tingkat interaksi *honeypot* yang harus disiapkan[14].

Berdasarkan interaksinya, terdapat dua jenis *honeypot* yaitu:

a. *Honeypot low interaction*

Honeypot low interaction menggunakan sistem operasi emulasi yang terpasang pada *honeypot* ketika berinteraksi dengan penyerang. *Honeypot low interaction* memiliki interaksi yang terbatas kepada penyerang. Serangan yang dihadapi biasanya berupa port *scanning* dan juga *digital signature attack*[13].Interaksi pada *honeypot low interaction* dengan host lain terbatas sehingga kemampuan yang dimiliki terbatas dan penyerang dapat dengan mudah mengenalinya tetapi dibalik terbatasnya *honeypot low interaction* memiliki resiko yang rendah[15].

Honeypot low interaction juga didesain untuk mensimulasikan layanan layaknya server asli dengan layanan tertentu seperti SSH. Layanan tersebut bukan sistem operasi secara keseluruhan, layanan yang berjalan tidak bisa dieksploitasi untuk mendapatkan hak akses penuh terhadap *honeypot[16]*. *Honeypot low interaction* lebih mudah diimplementasikan dan memiliki dampak resiko yang rendah pada jaringan maupun sistem[17]. *Honeypot low interaction* berfungsi seperti IDS pasif tanpa mengubah jalur lalu lintas jaringan yang ada[18].

Kelebihan *honeypot low interaction* diantaranya adalah memberikan pengalaman yang baik bagi yang belum berpengalaman dan masih dalam tahap pembelajaran membangun *honeypot[19]*. Kelebihan lainnya adalah untuk *install*, *deploy*, dan *maintenance* sangat mudah. Begitu juga dengan analisa pada log yang dihasilkan juga lebih mudah[19].

Honeypot low interaction memiliki beberapa kekurangan diantaranya: (1) log yang dihasilkan sangat terbatas, (2) kemampuan untuk menangkap serangan sudah diketahui sebelumnya, (3) dan *honeypot low interaction* mudah terdeteksi oleh penyerang yang sudah profesional. Tujuan dibuatnya *honeypot low interaction* diantaranya (1) untuk mengidentifikasi dan mendeteksi serangan yang dilakukan oleh *tools* otomatis, (2) menipu penyerang yang masih *script kiddies*, (3) mengalihkan serangan yang dilakukan oleh penyerang dari sistem asli, (4) dan mendapatkan modus serangan yang dilakukan oleh penyerang[20].

b. Honeypot high interaction

Honeypot high interaction menggunakan sistem operasi asli untuk lebih memotivasi penyerang dalam menyerang sistem sehingga strategi maupun modus serangan dapat dicatat dan dianalisis lebih detail. *Honeypot high interaction* mampu

 https://doi.org/10.25077/TEKNOSI.v5i2.2019.112-120

memproses dan membedakan antara paket yang bersih dengan paket yang dikirim oleh penyerang sehingga paket tersebut tidak dapat merusak server asli[13]. Kelebihan *honeypot high interaction* diantaranya adalah serangan yang diterima *honeypot high interaction* bisa jadi serangan yang asli dan belum dikenal, ini membuat serangan tersebut bermanfaat untuk dipelajari[19]. Serangan yang diterima mempermudah pengguna *honeypot* untuk mempelajari metode yang digunakan penyerang, dan mencegah serangan pada masa mendatang dan mendapatkan pengetahuan tentang ancaman tersebut[19].

Honeypot high interaction memiliki kekurangan diantaranya untuk membuat, konfigurasi, *deploy*, dan *maintenance* memakan waktu yang lama karena harus menyesuaikan teknologi yang digunakan seperti IDS, firewall, dan lain sebagainya, Analisa serangan memakan waktu yang lama, dan Resiko yang dihasilkan oleh *honeypot high interaction* sangat tinggi, jika tidak ada tindakan pencegahan maupun perlindungan tambahan maka dapat merugikan organisasi dari serangan yang dilakukan oleh penyerang[19].

Tabel 1. Perbandingan Honeypot

Parameter	**Low Interaction Honeypot**	**High Interaction Honeypot**
Tingkat Perkembangan	Rendah	Tinggi
Sistem Operasi Asli	Tidak	Ya
Resiko	Rendah	Tinggi
Pengumpulan Informasi	Koneksi	Semua
Emulasi Layanan	Ya	Tidak
Pengetahuan tentang menjalankan dan mengembangkan	Rendah	Tinggi
Waktu perawatan	Rendah	Tinggi

(Sumber : Mitchell A, 2018)

Honeypot memiliki perbedaan dengan konsep keamanan lainnya seperti *firewall, intrusion detection system* (IDS), *anti-virus* (AV), *intrusion prevention system* (IPS), log *monitoring*, dan *cyber security standard*. *Honeypot* lebih mengedepankan kepada deteksi dan reaksi tetapi kurang untuk melindungi ketika terjadi serangan[21].

Tabel 2. Perbedaan Honeypot dengan Konsep Keamanan Lainnya.

Konsep Keamanan	**Pencegahan**	**Deteksi**	**Reaksi**
Honeypot	+	++	+++
Firewall	+++	++	+
IDS	+	+++	+
IPS	++	+++	++
Antivirus	++	++	++
Log Monitoring	+	++	+
Cyber Security Standard	+++	+	+

(Sumber : Nawrocki M, 2016)

Tujuan penelitian ini adalah untuk menganalisa perilaku apa yang dilakukan penyerang di dalam server dan juga kemungkinan kredensial masuk yang digunakan oleh penyerang, dengan begitu hasil dari serangan sebagai pembelajaran bagi administrator sistem untuk membuat server lebih aman.

2. METODE

Metode yang dilakukan memiliki beberapa tahapan yaitu studi literatur untuk pengumpulan data lalu setelah terkumpulnya data yang dibutuhkan yaitu dilakukannya analisa data untuk menganalisa apa saja yang dibutuhkan dari data yang telah didapat kemudian digambarkan dengan rancangan yang jelas untuk memenuhi kebutuhan serta pembuatan *honeypot* dan agar *honeypot* berjalan dengan semestinya dilakukanlah pengujian.

2.1 Studi Literatur

Metode ini menggunakan pembelajaran dengan cara mengumpulkan, membaca dan memahami jurnal ilmiah, skripsi, artikel, dan juga mencari informasi melalui forum-forum dan juga komunitas terpercaya serta referensi lain secara online guna mendapatkan informasi yang dibutuhkan dalam menunjang penelitian.

2.2 Analisa Data

Pada analisa data dilakukannya proses penganalisaan terhadap konsep yaitu *network programming* dan juga jenis *honeypot* yang akan diterapkan yaitu *low interaction* untuk mendapatkan analisa yang dibutuhkan *honeypot*.

2.3 Perancangan dan Pembuatan *Honeypot*

Perancangan dilakukan untuk memudahkan dalam mengimplementasikan rancangan *honeypot* dan menentukan model perangkat lunak yang bertujuan agar memberikan gambaran yang jelas *honeypot* yang ingin dibuat dan memenuhi kebutuhan dalam menganalisa serangan kemudian dibuatnya *honeypot* dari rancangan yang telah dibuat.

2.4 Pengujian *Honeypot*

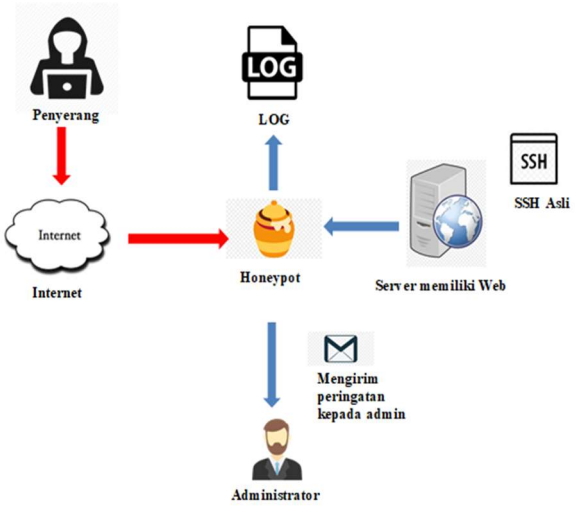
Diperlukan pengujian *honeypot* untuk mengetahui dan menguji aplikasi agar berjalan dengan semestinya. Adapun tools yang digunakan untuk menguji aplikasi, yaitu menggunakan *nmap* dan *hydra.*

3. HASIL DAN PEMBAHASAN

3.1 Rancangan *Honeypot*

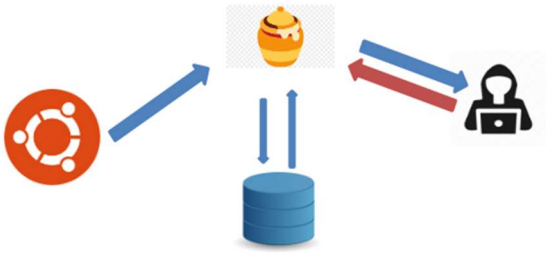
Pada penelitian ini protokol SSH asli pada sistem operasi dipindahkan ke port protokol lain yang diketahui oleh administrator server saja, sedangkan *honeypot* menggunakan port standar SSH. Penyerang yang menggunakan protokol SSH akan masuk ke dalam *honeypot* sehingga penyerang mengira masuk ke protokol SSH asli. *Honeypot* menghasilkan log yang berisi aktivitas penyerang ketika berusaha *login* ke dalam SSH yang

disediakan oleh *honeypot*. Selain itu log akan mencatat ip, port, negara asal, kota, maupun kodepos milik penyerang. Ketika penyerang berhasil *login*, *honeypot* akan mengirimkan email berupa pemberitahuan kepada administrator server akan adanya serangan pada servernya. Seluruh kegiatan yang dilakukan oleh penyerang setelah berhasil *login* akan direspon oleh *honeypot* dan sistem operasi di dalamnya. Dibuat *database* yang berisi perintah-perintah yang kemungkinan akan digunakan oleh penyerang ketika berhasil *login*.



Gambar 1. Rancangan Honeypot

Gambar 2 menjelaskan ketika sistem operasi linux server menjalankan aplikasi *honeypot* dan penyerang terkoneksi dan sudah masuk kedalam *honeypot*. *Honeypot* memiliki database yang berfungsi sebagai kumpulan list reaksi palsu yang dibuat untuk penyerang ketika menginput sesuatu kedalam *honeypot* maka *honeypot* akan berkomunikasi dengan database lalu mengeluarkan reaksi yang ada di database sesuai input yang dilakukan oleh penyerang. Reaksi ditampilkan untuk penyerang sesuai dengan list di dalam database. Penyerang ketika menginput sesuatu tidak mengganggu sistem operasi asli, karena semua input masuk ke dalam aplikasi *honeypot*.



Gambar 2. Rancangan Deteksi Honeypot

3.2 List Basis Data

Berikut ini merupakan contoh perintah dan reaksi yang akan digunakan dan dihasilkan aplikasi *honeypot* yang ada pada database ketika penyerang melakukan kegiatan didalam honeypot yang telah dioperasikan, perintah ini merupakan perintah dasar pada sistem operasi Linux server sebagai berikut :

Tabel 3. List Basis Data Honeypot

Input	Reaksi
whoami	root
pwd	/
rm	was some error in remove the files try again later
uname	Linux server 4.11.0-32-generic #35~20.04.1-Ubuntu SMP Thu Jan 25 10:13:43 UTC 2018 x86_64 GNU/Linux
id	uid=0(root) gid=0(root) groups=0(root)
chmod	: There was some error in changing access the files..try again later
chown	: There was some error in changing access the files..try again later
mv	: There was some error in changing access the files..try again later
cat /etc/issue	Ubuntu 16.04
cat /etc/resolv.conf	nameserver 208.67.222.222 nameserver 208.67.220.220

(Sumber : Matotek D, 2017)

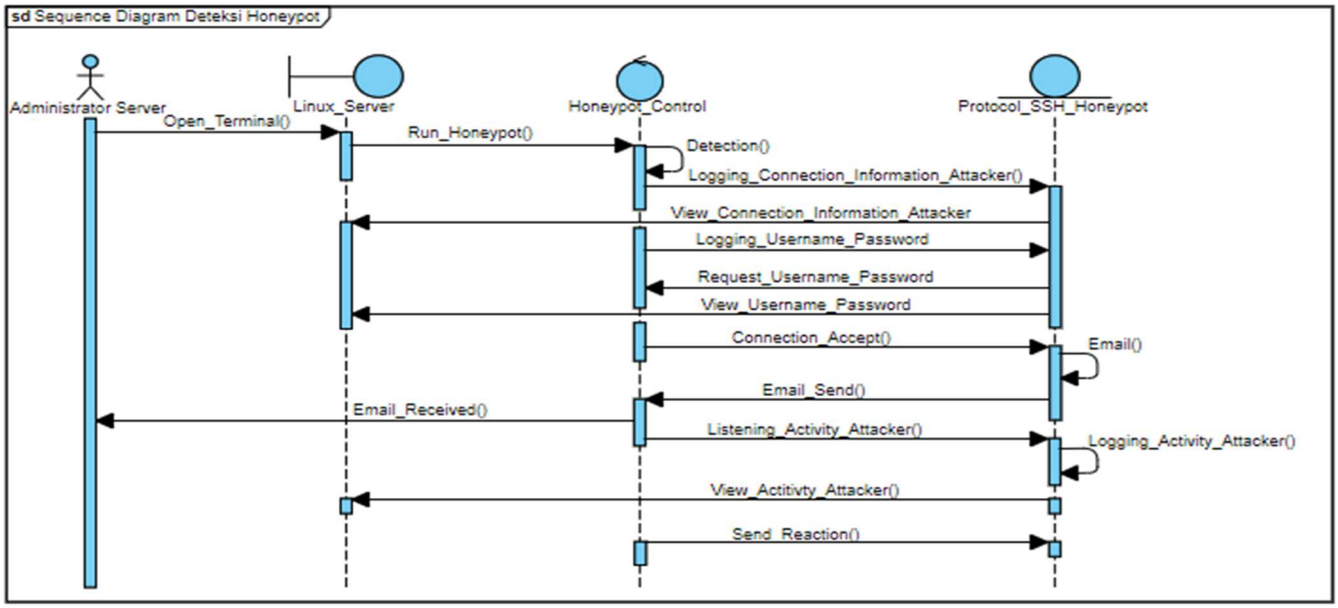
List basis data bersumber dari referensi yang membahas tentang administrasi sistem operasi linux[25]. Sebagai contoh ketika seorang penyerang menggunakan *command "cat /etc/issue"* maka aplikasi *honeypot* akan berkomunikasi dengan *database* dan *database* mencocokan didalam *list*, sehingga aplikasi honeypot akan mengeluarkan reaksi yaitu "Ubuntu 16.04". Karena jenis aplikasi honeypot bersifat *low-interaction*, reaksi yang dibuat tidak seperti pada sistem operasi sesungguhnya yang dapat melakukan reaksi dengan semestinya.

3.3 Sequence Diagram

Gambar 3 menjelaskan *sequence diagram* yang merepresentasikan fungsi dan aksi ketika *honeypot* mendeteksi adanya serangan. *Sequence diagram* deteksi *honeypot* memiliki satu aktor yaitu administrator server yang berperan untuk mengatur dan memonitoring *honeypot* yang berada didalam sistem operasi Linux server dan juga memiliki 16 *method* yaitu : (1) *Open_Terminal* merupakan *method* yang menjelaskan ketika seorang aktor yaitu administrator server mengakses terminal pada server sistem operasi Linux, (2) *Run_Honeypot* merupakan *method* yang menjelaskan ketika seorang aktor yaitu administrator server menjalankan *honeypot* pada server sistem operasi Linux melalui terminal, (3) *Detection* adalah *method* yang menjelaskan ketika honeypot mendeteksi adanya serangan pada server melalui protokol SSH yang dibuat oleh *honeypot*. (4) *Logging_connection_information_attacker* merupakan *method* yang menjelaskan ketika *honeypot* merekam informasi profil penyerang yang terhubung pada protokol SSH yang dibuat oleh honeypot, (5) *View_connection_information_attacker* adalah

https://doi.org/10.25077/TEKNOSI.v5i2.2019.112-120

method untuk menampilkan informasi profil penyerang yang terhubung pada protokol SSH yang dibuat oleh *honeypot* pada terminal sistem operasi Linux server sehingga administrator server dapat melihat langsung penyerang yang masuk kedalam honeypot yang telah dijalankan pada servernya, (6) *Logging_username_password* merupakan *method* yang menjelaskan ketika honeypot merekam *username* dan *password* yang dilakukan penyerang untuk mencoba login kedalam protokol SSH yang dibuat oleh honeypot, (7) *Request_username_password* merupakan *method* yang menjelaskan ketika penyerang mencoba login pada protokol SSH yang dibuat oleh *honeypot* menggunakan *username* dan *password,* (8) *View_username_password* merupakan *method* yang menampilkan *username* dan *password* yang telah digunakan oleh penyerang kedalam protokol SSH yang dibuat oleh honeypot sehingga administrator server dapat melihat langsung pada terminal sistem operasi Linux server miliknya, (9) *Connection_accept* merupakan *method* ketika *username* dan *password* yang digunakan penyerang untuk masuk kedalam protokol SSH yang telah disediakan oleh *honeypot* benar, sehingga penyerang dapat masuk kedalam server palsu *honeypot*, (10) *Email* merupakan *method* ketika *honeypot* menyiapkan email untuk mengirim kepada administrator server sebagai peringatan adanya penyerang yang berhasil masuk kedalam *honeypot*, (11) *Email_send* merupakan *method* untuk mengirim email kepada administrator server sebagai peringatan bahwa adanya penyerang yang berhasil masuk kedalam honeypot, (12) *Email_received* merupakan *method* ketika email peringatan diterima oleh administrator server yang dikirim oleh *honeypot*.

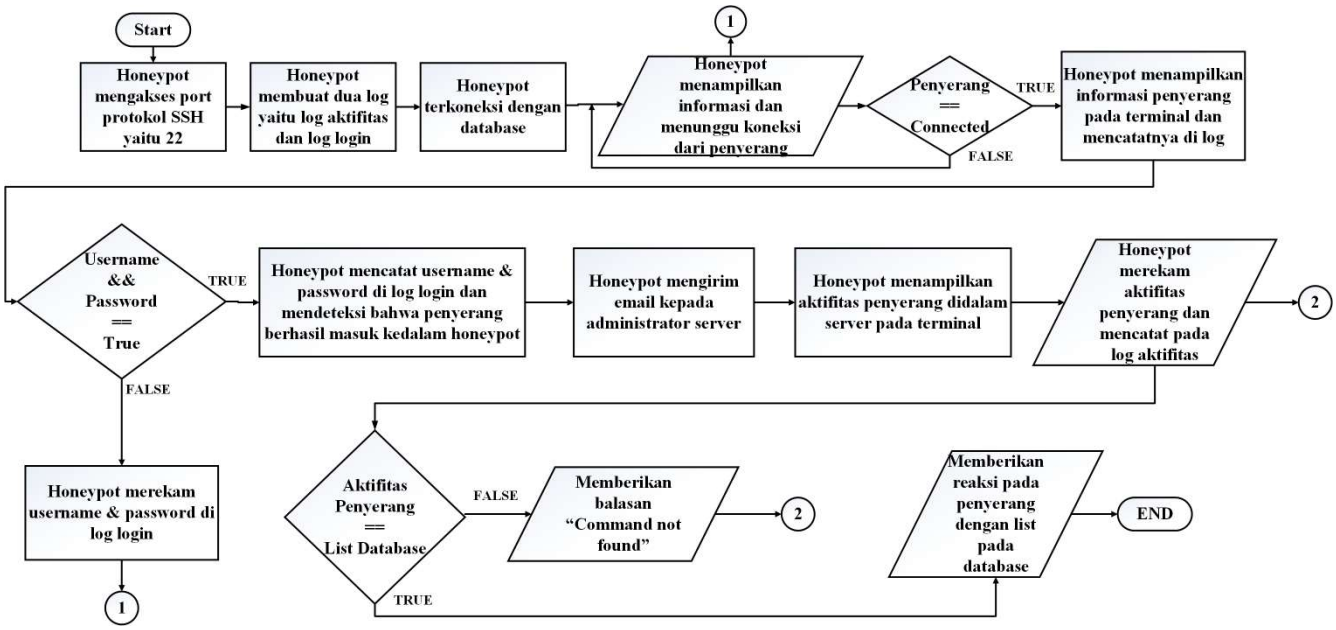


Gambar 3. *Sequence Diagram* Deteksi *Honeypot*

Listening_activity_attacker merupakan *method* ketika *honeypot* memonitoring aktifitas yang dilakukan oleh penyerang saat didalam *honeypot,* (13) *Logging_activity_attacker* merupakan *method* yang menjelaskan ketika *honeypot* mencatat aktifitas yang dilakukan oleh penyerang saat didalam *honeypot*, (14) *View_activity_attacker* merupakan *method* yang menampilkan aktifitas penyerang saat didalam *honeypot* sehingga administrator server dapat melihat langsung pada terminal sistem operasi Linux server miliknya, (15) *Send_reaction* merupakan method yang mengirim reaksi kepada penyerang dimana reaksi tersebut berdasarkan apapun yang dilakukan oleh penyerang didalam *honeypot* dan juga reaksi tersebut sudah disiapkan didalam database milik *honeypot.*

3.4 Flowchart

Pada Gambar 4 *flowchart* dibawah menjelaskan proses aplikasi *honeypot* ketika mendeteksi penyerang pada protokol SSH yang digunakan oleh *honeypot.* Pertama kali ketika dijalankan, *honeypot* mengakses port milik protokol SSH sebagai umpan untuk penyerang. Kemudian, *honeypot* menggunakan kunci SSH yang sudah dibuat sebagai autentikasi ketika penyerang mencoba masuk kedalam SSH yang dibuat *honeypot.* Lalu, *honeypot* terkoneksi kepada *database* yang telah dibuat sebagai *list* reaksi jika penyerang menginput sesuatu pada *honeypot. Honeypot* menampilkan informasi yang berarti sudah berjalan dan menunggu koneksi dari penyerang. Jika port SSH yang berarti *honeypot* mendapatkan koneksi dari penyerang, maka *honeypot* menampilkan informasi koneksi penyerang, dan jika tidak maka *honeypot* menunggu sampai penyerang terkoneksi pada port SSH. Setelah penyerang terkoneksi kedalam port SSH yang berarti *honeypot,* maka penyerang diminta *username* dan *password* untuk masuk kedalam *honeypot.*



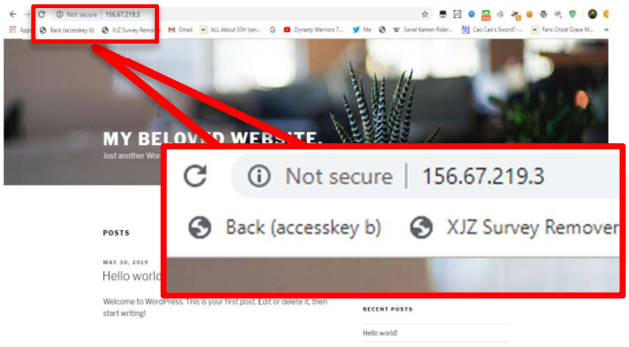
Gambar 4. *Flowchart* Deteksi *Honeypot*

Jika *username* dan *password* benar maka penyerang dapat masuk kedalam *honeypot,* lalu jika *username* dan *password* salah maka penyerang dapat mengulangi lagi memasukan *username* dan *password. Username* dan *password* yang digunakan oleh penyerang dicatat didalam log *login*. Ketika penyerang berhasil masuk kedalam *honeypot*, *honeypot* mengirim email kepada administrator server sebagai peringatan karena penyerang berhasil masuk kedalam *honeypot*nya. *Honeypot* menampilkan aktivitas yang dilakukan penyerang di dalam server palsu yaitu *honeypot* pada terminal agar dapat dilihat oleh administrator server. Setiap aktivitas yang dilakukan oleh penyerang, *honeypot* akan mencatat kedalam log aktivitas.

Aktivitas yang dilakukan oleh penyerang berupa suatu input yang dilakukan ketika didalam server. Aktivitas tersebut perlu ada reaksi dari *honeypot* agar dapat mengelabui penyerang yang seakan-akan sudah masuk ke server asli. Maka dibuatlah *database* yang isinya kumpulan *list* kemungkinan reaksi yang dilakukan ketika *honeypot* mendapatkan input dari penyerang. Jika input yang dilakukan oleh penyerang tidak ada di dalam *list* pada *database* maka *honeypot* akan menjawab "*command not found"* pada penyerang.

4.5 Hasil Pengujian Serangan

Untuk menguji coba *honeypot* ini dibuat skenario penyerangan pada *honeypot* dan juga nantinya memberikan notifikasi email jika aplikasi tersebut berjalan dengan baik dan benar.



Gambar 5. Website yang dimiliki server

Gambar 5 menjelaskan server yang dijalankan memiliki service web yang dibuat menggunakan CMS Wordpress pada IP server 156.67.219.3. Untuk menghindari penyerangan pada port protokol SSH maka dipasang aplikasi *honeypot* pada sisi servernya.

Gambar 6. Honeypot Dijalankan

Gambar 6 menjelaskan ketika menjalankan *honeypot* pada server. Pengguna atau administrator server menjalankan *honeypot* pada terminal menggunakan *command "sudo python3 honeyta.py"* dan

https://doi.org/10.25077/TEKNOSI.v5i2.2019.112-120

hasilnya *honeypot* berjalan lalu mengeluarkan informasi dan menunggu penyerang yang masuk pada port SSH yang dibuat oleh *honeypot*.

```
> nmap 156.67.219.3
Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-23 16:44 SE Asia Standard Time
Nmap scan report for 156.67.219.3
Host is up (0.054s latency).
Not shown: 997 closed ports
PORT   STATE    SERVICE
22/tcp open     ssh
25/tcp filtered smtp
80/tcp open     http

Nmap done: 1 IP address (1 host up) scanned in 21.28 seconds

C:\Users\neofall-T440P\Desktop
>
```

Gambar 7. Protokol SSH *Honeypot* ketika di *Scan*

Gambar 7 menjelaskan bahwa ketika IP server di scan menggunakan port *scanner* yaitu aplikasi *nmap.* Hasil scan port terdapat protokol SSH yang menggunakan nomor port 22. Protokol dengan port 22 ini adalah SSH palsu yang diakses oleh *honeypot* sesuai dengan source code pada honeypot pada gambar 8,gambar 9 SSH yang asli di letakkan pada port 18000 untuk menghindari serangan langsung jika seseorang masuk melalui port 22 maka bisa dipastikan adalah serangan.

honeypot_ta.py

```
import smtplib
import pymysql
import geoip2.database
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart



logging.basicConfig(level=logging.DEBUG,
                    format='%(asctime)s %(levelname)-8s %(message)s',
                    datefmt='%Y-%m-%d %H:%M:%s',
                    filename='log/honeyta.log',
                    filemode='a')

LOGINLOG = open('log/login.log','a')

SSH_KEY = paramiko.RSAKey(filename='sshkey/honeyta.key')
PORT = 22
loglock_thread = threading.Lock()
```

Gambar 8. Source Code Honeypot mengakses port 22

```
taserver1@vps615259: ~
 GNU nano 2.9.3                  /etc/ssh/sshd_config

#       $OpenBSD: sshd_config,v 1.101 2017/03/14 07:19:07 djm Exp $

# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/bin:/bin:/usr/sbin:/sbin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

Port 18000
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::
```

Gambar 9. Config SSH pada port 18000

Ini membuktikan bahwa pemasangan port yang dilakukan berhasil dan juga dapat membedakan serangan.

```
taserver1@vps615259: ~/testinghoneypot
[+] ------------------------------------------------------ [+]
[!] Semua aksi attacker terekam di honeyta.log dan login.log [!]
[+] ------------------------------------------------------ [+]
[-] Attacker IP : 185.203.211.174
[-] Attacker PORT : 27796
Attacker Country Code: ES
Attacker Country Name : Spain
Attacker City : Bellpuig
Attacker Postal Code : 25250
[+] ---------------------------------------- [+]
[#] Attacker memasukan username & password: admin:admin

SSH channel dan attacker tidak terhubung
Tidak ada request pada server
Attacker gagal terkoneksi dengan honeypot
Attacker gagal masuk kedalam honeypot
[+] ------------------------------------------------------ [+]
[!] Semua aksi attacker terekam di honeyta.log dan login.log [!]
[+] ------------------------------------------------------ [+]
```

Gambar 10. Honeypot Mendeteksi Penyerang

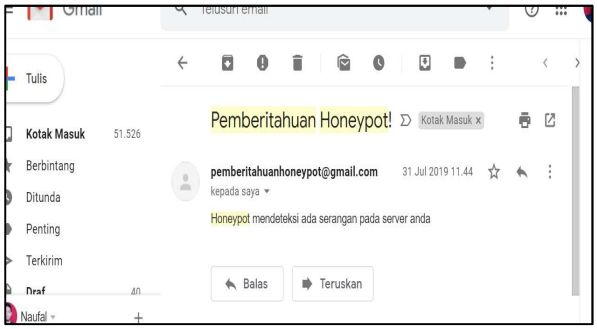
Gambar 10 menjelaskan bahwa ketika *honeypot* mendeteksi adanya penyerangan terhadap port protokol SSH yang dimiliki aplikasi *honeypot*. Hanya saja penyerang tidak berhasil masuk kedalam aplikasi *honeypot* karena *username* dan *password* yang digunakan salah.

```
taserver1@vps615259: ~/testinghoneypot
[!] Semua aksi attacker terekam di honeyta.log dan login.log [!]
[+] ------------------------------------------------------ [+]
[-] Attacker IP : 110.136.18.75
[-] Attacker PORT : 53289
Attacker Country Code: ID
Attacker Country Name : Indonesia
Attacker City : Tangerang
Attacker Postal Code : None
[+] ---------------------------------------- [+]
[#] Attacker memasukan username & password: root:password123

uname -a
uname -a
uname a
uname
cat /etc/passwd
ls
apt
```

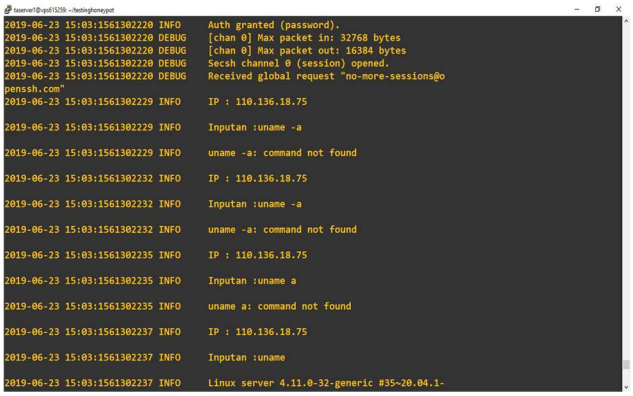
Gambar 11. Penyerang Berhasil Masuk *Honeypot*

Gambar 11 menjelaskan ketika penyerang berhasil masuk kedalam *honeypot*. Seluruh aktivitas penyerang terlihat pada layar terminal server.



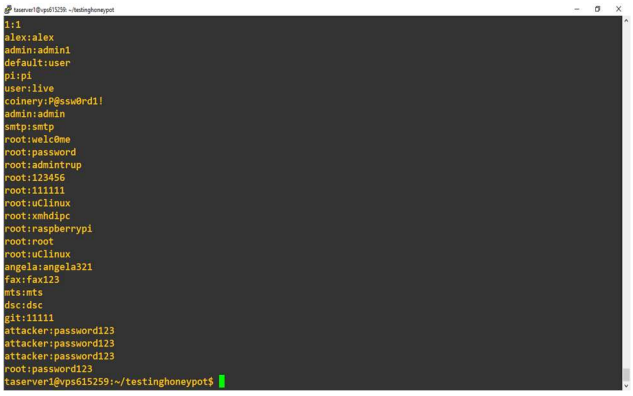
Gambar 12. Notifikasi Email *Honeypot*

Gambar 12 menjelaskan menjelaskan ketika penyerang berhasil masuk kedalam *honeypot*, *honeypot* mengirimkan notifikasi email kepada email administrator server untuk memberitahukan bahwa penyerang telah masuk kedalam *honeypot*.



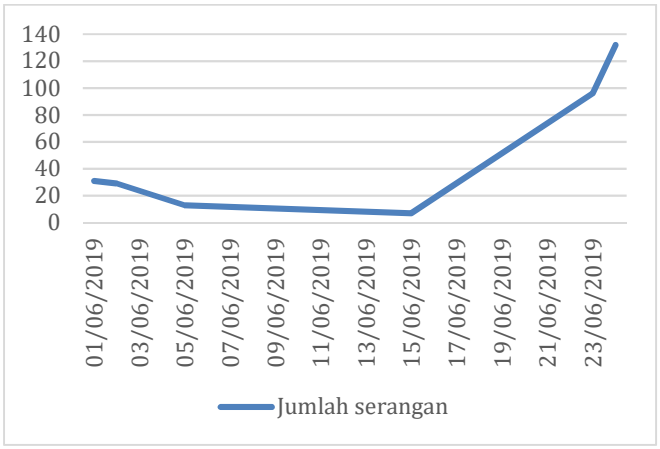
Gambar 13 : Log *Honeypot*

Gambar 13 menjelaskan bahwa seluruh aktivitas yang diterima dan keluaran dari *honeypot* terekam pada log yang dihasilkan oleh *honeypot* tersebut.



Gambar 14. Log *Login Honeypot*

Gambar 14 menjelaskan bahwa seluruh *username* dan *password* yang digunakan oleh penyerang terekam pada log *login* yang telah dihasilkan oleh aplikasi *honeypot*.



Gambar 15. Jumlah Serangan Perhari

Pengujian yang dihasilkan membahas tentang tentang seberapa banyak *command* yang dieksekusi penyerangan , data *username,* data *password*, kombinasi data *username* dan *password*, dan juga negara mana saja yang telah melakukan penyerangan terhadap aplikasi *honeypot*. Pada gambar 15, Waktu pengujian dilakukan dari tanggal 1 sampai dengan tanggal 24 Juni 2019. Hasil keseluruhan pada data *username* dan *password* yang tercatat oleh aplikasi *honeypot* yaitu 327 data serangan. Hasil keseluruhan data penyerang berdasarkan negara yaitu 311 data serangan dengan jumlah 27 negara.

Tabel 4. Top 5 Username

Data Username	Jumlah	Persentase
root	142	43%
admin	102	31%
user	27	8%
1234	9	3%
test	4	1%
Username Lainnya	43	14%

Tabel 4 merupakan hasil data *username* terbanyak yang digunakan oleh penyerang berdasarkan hasil tangkapan dari aplikasi *honeypot*. Table tersebut terdiri dari data *username* yang selalu digunakan terus menerus dan juga data *username* yang berbeda. Data *username* yang berbeda terdiri dari nama depan, nama *service*, nama sistem operasi, dan juga nama-nama lainnya. Data *username* yang berbeda terdiri dari nama depan orang diantaranya yaitu Alex, Allan, Bernard, Eloise, Linda, Seth, dan Steve. Data *username* yang berbeda juga terdiri dari nama-nama *service* diantaranya *tomcat, zimbra, nagios, weblogic, webmaster, wwwmp2s,* dan *wwwrun*.

Tabel 5 merupakan Top 5 hasil data *password* yang digunakan oleh penyerang berdasarkan hasil tangkapan dari aplikasi *honeypot*. Berbeda dari data *username*, data *password* yang digunakan oleh penyerang lebih kepada *password* yang berbeda.

https://doi.org/10.25077/TEKNOSI.v5i2.2019.112-120

Tabel 5. Top 5 Password

Data Username	Jumlah	Persentase
password123	11	3%
admin	7	2%
123	6	2%
password	6	2%
1234	5	2%
Password Lainnya	294	89%

Tabel 6. Top 5 Kombinasi Username dan Password

Data Username	Jumlah	Persentase
root/password123	10	3%
root/0	2	1%
admin/1234	2	1%
root/111111	2	1%
root/987654321	2	1%
Username/Password Lainnya	310	93%

Tabel 6 merupakan lima hasil pengujian data dari kombinasi *username* dan *password* yang digunakan oleh penyerang berdasarkan hasil tangkapan dari aplikasi *honeypot*. Kombinasi *username* dan *password* yang digunakan oleh penyerang sebagian besar menggunakan *root*. Kombinasi *username* dan *password* paling banyak digunakan yaitu *root/password123* yang juga merupakan kombinasi yang sah untuk masuk kedalam aplikasi *honeypot*.

Tabel 7. Top 5 penyerangan berdasarkan negara

Data Username	Jumlah	Persentase
Indonesia	101	33%
Cina	42	13%
Amerika	31	10%
Panama	19	6%
Rusia	17	5%
Negara Lainnya	101	33%

Tabel 7 merupakan lima negara asal penyerang dengan jumlah serangan terbanyak. Negara yang tercatat pada tabel paling banyak dan juga konsisten melakukan serangan yang tercatat pada aplikasi *honeypot* adalah Indonesia. Negara yang tercatat pada aplikasi *honeypot* setelah Indonesia yaitu Cina, Amerika, Panama, dan Rusia. Negara yang tercatat pada aplikasi *honeypot* lainnya seperti Prancis, Spanyol, Brazil, Kanada, Belanda, Jerman, Romania, Singapura, Argentina, India, Korea Selatan, Vietnam, Meksiko, Colombia, Ukraina, Mozambique, Polandia, Ekuador, Denmark, Swedia, Irlandia, dan Thailand.

Tabel 8. Command yang Dieksekusi Penyerang

Data Command	Jumlah
ls	9
sudo su	3
clear	3
uname -a	2
uname	2
uname a	1
ls -a	1
apt	1
wget https://github.com/FireFart/dirtycow	1
wget	1
cat /etc/passwd	1
cls	1
nmap	1
-h	1
-help	1
--help	1
root	1
whoami	1
fix	1
ls /usr/bin	1
sudo	1

Tabel 8 merupakan hasil *command* yang dieksekusi oleh penyerang ketika didalam aplikasi *honeypot*. Hasil *command* yang dieksekusi yang digunakan oleh penyerang paling banyak adalah *command "ls"* yang berguna untuk melihat daftar direktori yang ada pada server. Hasil data yang menarik disini adalah ketika penyerang mengeksekusi *command "wget https://github.com/FireFart/dirtycow"* yang berguna untuk men*download* suatu *malicious code* atau kode jahat yang digunakan untuk memperoleh akses *root* dan menguasai server. Hasil *command* yang dieksekusi lainnya kebanyakan hanya untuk mengidentifikasi informasi yang ada di dalam server.

4. KESIMPULAN

Dari hasil pengujian dapat disimpulkan bahwa honeypot yang dibuat menggunakan SSH server palsu atau SSH tiruan yang untuk mengelabui penyerang sehingga penyerang tidak menyerang SSH server yang asli. *Honeypot* yang dibuat membantu meningkatkan keamanan pada server. *Honeypot* yang dibuat menangkap *username* dan *password* yang digunakan penyerang untuk masuk ke dalam server agar administrator server lebih memperkuat *login credentials* pada SSH lalu mendeteksi adanya serangan *brute force* berupa *login credentials* yang dikirim terus menerus yang dilakukan oleh penyerang. *Honeypot* mencatat data *username* dan *password* secara keseluruhan yaitu 327 data serangan. Hasil keseluruhan data penyerang berdasarkan negara yaitu 311 data serangan dengan jumlah 27 negara yang tercatat dan ditangkap oleh *honeypot*. Penyerang yang masuk

kedalam *honeypot* rata-rata mencari tahu informasi direktori yang ada didalam sistem operasi. Ditemukan pada log *honeypot* penyerang berusaha mengunduh *code exploit Dirtycow* yang berfungsi untuk menguasai server.

DAFTAR PUSTAKA

[1] O. K. Sulaiman, "Analisis Sistem Keamanan Jaringan dengan Menggunakan Switch Port Security," *CESS (Journal Of Computer Engineering, System And Science)*, Jan. 2016.

[2] C. K. Ng, L. Pan, and Y. Xiang, *Honeypot Frameworks and Their Applications: A New Framework*. Springer, 2018.

[3] M. M. Mustofa and E. Aribowo, "Penerapan Sistem Keamanan Honeypot Dan IDS Pada Jaringan Nirkabel (Hotspot)," *Jurnal Sarjana Teknik Informatika*, vol. 1, no. 1, pp. 111–118, 2013.

[4] A. Zainuddin, L. Affandi, and A. D. Susilo, "Analisis Sistem Keamanan Hotspot Dengan Menggunakan Honeypot Dan IDS Di Kampus STMIK PPKIA PRADNYA PARAMITA Malang," *JURNAL TEKNOLOGI INFORMASI: Teori, Konsep, dan Implementasi*, vol. 5, no. 2, pp. 107–112, 2014.

[5] A. F. Nurrahman, "Implementasi Virtual Low-Interaction Honeypot Dengan Dionea Untuk Mendukung Keamanan Jaringan," *Journal of Informatics and Technology*, vol. 2, no. 4, pp. 28–37, Sep. 2014.

[6] A. S. Nugroho, S. Raharjo, and J. Triyono, "Analisis dan Implementasi Honeypot Menggunakan Honeyd Sebagai Alat Bantu Pengumpulan Informasi Aktivitas Serangan pada Jaringan," *Jurnal JARKOM*, vol. 1, no. 1, pp. 40–48, Dec. 2013.

[7] T. A. Cahyanto, H. Oktavianto, and A. W. Royan, "Analisis dan Implementasi Honeypot Menggunakan Dionaea Sebagai Penunjang Keamanan Jaringan," *JUSTINDO (Jurnal Sistem dan Teknologi Informasi Indonesia)*, vol. 1, no. 2, May 2017.

[8] N. Fitriana and F. N. Khasanah, "Honeypot Menggunakan Honeyd Sebagai Solusi Keamanan Jaringan Dari Aktivitas Serangan," *Bina Insani ICT Journal*, vol. 5, no. 2, pp. 143–152, Dec. 2018.

[9] A. Mitchell, "An Intelligent Honeypot," Cork Institute of Technology, 2018.

[10] W. Wilman, I. Fitri, and N. D. Nathasia, "Port Knocking Dan Honeypot Sebagai Keamanan Jaringan Pada Server Ubuntu Virtual," *J I M P - Jurnal Informatika Merdeka Pasuruan*, vol. 3, no. 1, Mar. 2018.

[11] V. Malik, M. Jhawar, Harleen, A. Khanijau, and N. Chawla, "LAN Based Intrusion Detection And Alerts," *IJSTR*, May 2014.

[12] D. X. Gkoutzelis and M. S. Sardis, "Web Server Security on Open Source Environments," in *Next Generation Society. Technological and Legal Issues*, 2010, pp. 236–247.

[13] R. Upadhayay, T. K. Mandal, S. Joshi, and M. Kala, "Data Security Using Honeypot," *IJIRT*, Apr. 2017.

[14] R. K. Singh and T. Ramanujam, "Intrusion Detection System Using Advanced Honeypots," *International Journal of Computer Science and Information Security*, vol. 2, no. 1, 2009.

[15] A. Jain and D. B. Buksh, "Advance Trends in Network Security with Honeypot and its Comparative Study with other Techniques," *International Journal of Engineering Trends and Technology*, vol. 29, pp. 304–312, 2015.

[16] B. Tambunan, W. S. Raharjo, and J. Purwadi, "Desain dan Implementasi Honeypot dengan Fwsnort dan PSAD Sebagai Intrusion Prevention System," *ULTIMA Computing*, Sep. 2013.

[17] S. Z. Melese and P. S. Avadhani, "Honeypot System for Attacks on SSH Protocol," *I. J. Computer Network and Information Security*, Sep. 2016.

[18] N. Kambow and L. K. Passi, "Honeypots: The Need of Network Security," *IJCSIT*, 2014.

[19] S. Gupta and V. Singhal, "Honeypot a Trap for Hackers," *INDIACom*, 2011.

[20] R. C. Joshi and A. Sardana, *Honeypots A New Paradigm to Information Security*. New York: Science Publishers, 2011.

[21] M. Nawrocki, M. Wählisch, T. C. Schmidt, C. Keil, and J. Schönfelder, "A Survey on Honeypot Software and Data Analysis," *arXiv [cs.CR]*, 22-Aug-2016.

[22] D. Matotek, J. Turnbull, and P. Lieverdink, *Pro Linux System Administration: Learn to Build Systems for Your Business Using Free and Open Source Software*. New York: Apress, Berkeley, CA, 2017.

BIODATA PENULIS

Naufal Arkaan

Penulis merupakan mahasiswa sarjana Universitas Budi Luhur Fakultas Teknologi Informasi. Jurusan Teknik Informatika dengan peminatan Network & Web Security.

Dolly Virgian Shaka Yudha Sakti

Kontributor merupakan dosen pada Program Studi Teknik Informatika Universitas Budi Luhur. Tertarik pada penelitian bidang kriptografi, keamanan komputer, dan *computer science*.

 https://doi.org/10.25077/TEKNOSI.v5i2.2019.112-120