IMPLEMENTASI HONEYPOT DENGAN METODE HONEYTRAP

Reyzal Hildha Hassan [1], Setia Juli Irzal Ismail, S.T., M.T. [2], Periyadi, S.T., M.T. [3]
[1, 2, 3]Prodi D3 Teknologi Komputer, Fakultas Ilmu Terapan, Universitas Telkom
[1]reyzalhildha@student.telkomuniversity.ac.id, [2]julismail@telkomuniversity.ac.id, [3]periyadi@telkomunniversity.ac.id

Abstrak- Banyaknya celah keamanan yang rentan diretas dan dimanfaatkan oleh orang yang tidak bertanggung jawab untuk mencuri data dan informasi dapat diakibatkan karena pihak yang diserang tidak menyadari betapa pentingnya keamanan jaringan untuk diterapkan terhadap sistem yang dimiliki.
Honeypot merupakan keamanan jaringan yang bertujuan untuk membangun suatu server yang menyerupai server asli dan memberikan kemudahan kepada penyerang untuk mengakses server jebakan.
Oleh karena itu, untuk mencapai suatu keamanan jaringan yang optimal diperlukan pengujian terhadap Honeypot dan menjadi sebuah tolak ukur kinerja pengembangan Honeypot.
Kata kunci: *Honeypot, Honeytrap, Secure Shell, Web Server, Docker.*

Abstract*- The many vulnerabilities that are vulnerable to being hacked and exploited by people who are not responsible for stealing data and information can be caused because the parties attacked do not realize how important network security is to be applied to the system they have.*
Honeypot is a network security that aims to build a server that resembles the original server and makes it easy for an attacker to access a trap server.
Therefore, to achieve an optimal network security testing is needed on Honeypot and becomes a benchmark for the performance of Honeypot development.
Keywords: Honeypot, Honeytrap, Secure Shell, Web Server, Docker.

1. Pendahuluan

1.1 Latar Belakang

Web Server rentan diretas oleh orang yang melakukan pencurian informasi dan data untuk disalahgunakan, bahkan protokol Secure Shell (SSH) yang memiliki keamanan yang kuat dapat diretas.
Pentingnya suatu keamanan agar tidak disalahgunakan oleh seseorang yang tidak bertanggung jawab, sehingga mengakibatkan banyak informasi dan data yang di sabotase. Upaya untuk menangani hal ini dibutuhkan Honeypot. “Honeypot” merupakan metode yang dapat membuat server palsu, baik itu informasi maupun data yang terdapat dalam web server, sehingga serangan yang ditunjukan untuk web server tersebut bukan data yang sebenarnya bahkan data terlindungi dari serangan hacking.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diutarakan, maka rumusan masalah dalam proyek akhir ini ialah sebagai berikut.
1.Bagaimana mengimplementasikan keamanan Web Server menggunakan
Honeypot ?
2.Bagaimana melakukan pengujian keamanan Web Server menggunakan
Honeypot ?

1.3 Tujuan

Adapun tujuan masalah dalam penelitian ini yaitu.
1.Memperkuat keamanan *Web Server* mengimplementasikan *Honeypot* dengan metode *Honeytrap*.
2.Melakukan pengujian *Honeytrap* dari serangan *SSH* dan melihat aktifitas yang dilakukan penyerang.

1.4 Batasan Masalah

Agar penelitian ini dapat lebih sempurna dan mendalami. Maka perlu membatasi permasalahan dalam penelitiannya. Permasalahan yang dibatasi yaitu.
1.Pengujian dilakukan dengan serangan Port Scanning, Metasploit, Hydra, dan SSH-Console.
2.Mendeskripsikan proses tahapan-tahapan dalam mengimplementasi Honeypot dengan menggunakan metode *Honeytrap* terhadap *Web Server*.

2. Tinjauan Pustaka

2.1 Penelitian Sebelumnya

Berdasarkan dari penelitian sebelumnya *Modern Honey Network (MHN)* merupakan server terpusat untuk manajemen dan pengumpulan data atau *logs*. MHN memerlukan dua atau lebih sensor *Honeypot* diantaranya *Sensor Honeypot Kippo* dan *Sensor Honeypot Dionae.* Sebagaimana mestinya sistem kerja MHN dapat digunakan untuk menjebak penyerang dengan memberi kemudahan untuk menyerang ke server palsu, dan server asli aman dari segala serangan [8].

Hasil dari penelitian yang dilakukan oleh penulis jurnal tersebut dapat memperkuat sistem keamanan terhadap *Honeypot* yang dapat memonitoring dan menganalisa peretas yang masuk ke sistem komputer yang asli. Tools tersebut dapat di kembangkan dengan metode baru yaitu *Honeytrap* yang bertujuan untuk memonitoring dan menganalisis yang dilakukan penyerang. *Honeytrap* bekerja dengan membuat pengalihan pada pelemeahan port SSH dan memberikan kemudahan penyerang untuk meretas, akan tetapi server yang diretas merupakan server jebakan yang sudah dialihkan oleh *Honeytrap*.

2.2 Teori

2.2.1 Linux

Linux merupakan salah satu pengembangan sebuah perangkat lunak bebas dan sumber terbuka utama. Yang memiliki kode sumber yang dapat dimodifikasi, digunakan dan didistribusikan kembali secara bebas oleh siapa saja.

Linux diperkenalkan pada tahun 1991 oleh Linus Torvalds dan juga nama *Linux* itu sendiri berasal dari sebuah nama pembuatnya “Linus”. *Linux* diumumkan pada tahun 1983 oleh Richard Stallman. Sistemnya, peralatan dan pustakanya umumnya berasal dari sistem operasi GNU, kontribusi GNU merupakan dasar dari munculnya nama alternatif *GNU/Linux* [4].

2.2.2 Honeytrap

Honeytrap merupakan *opensource framework* untuk menjalankan, memantau dan mengelola *Honeypots*. Dengan dukungan untuk berbagai jenis mode, *Honeytrap* dapat digunakan untuk menggunakan arsitektur *Honeypot* yang kompleks atau hanya untuk menggunakan satu server. Bergantung pada mode yang dipilih dan dapat memeriksa semua port untuk mendeteksi ancaman dan mengumpulkan informasi, atau memeriksa port tertentu dan memberikan jawaban yang telah ditentukan. Beberapa sistem operasi didukung, seperti *Linux*, MacOS dan Windows. Beberapa fungsi yang tersedia tergantung pada sistem operasi host-direktur LXC misalnya hanya tersedia di *Linux*, tetapi kebanyakan yang tersedia untuk setiap OS[6].

2.2.3 Sistem Operasi

Sistem Operasi merupakan sebuah program yang mengontrol eksekusi program-program aplikasi dan berfungsi sebagai penghubung antara pengguna dengan komputer dan perangkat keras komputer. Terdapat dua fungsi utama dari sistem operasi, yaitu:

a. Sistem Operasi sebagai *interface* pengguna / komputer.

b. Sistem Operasi menyembunyikan kerumitan *hardware* dari pengguna dan meyediakan *interface* yang nyaman untuk menggunkaan sistem bagi pengguna komputer [8].

2.2.4 Golang

Golang (atau biasa disebut dengan Go) adalah bahasa pemrograman baru yang dikembangkan di Google oleh Robert Griesemer, Rob Pike, dan Ken Thompson pada tahun 2007 dan mulai diperkenalkan di publik tahun 2009. *Golang* memiliki banyak kelebihan, terbukti dengan banyaknya perusahaan besar yang menggunakan bahasa ini dalam pengembangan produk-produknya, hingga level production tentunya[5].

2.2.5 Secure Shell

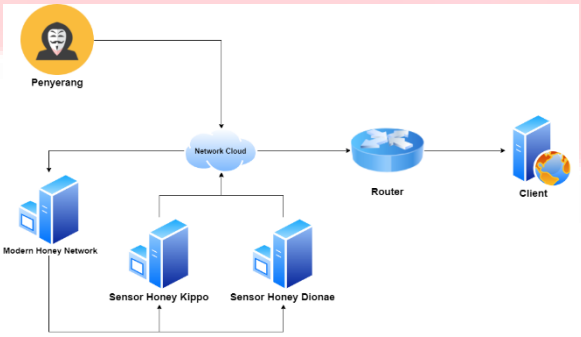
Secure Shell adalah protokol jaringan yang berada di lapisan aplikasi pada protokol TCP/IP, memfasilitasi sistem komunikasi yang aman diantara dua sistem yang menggunakan arsitektur klien server dengan menyediakan kerahasiaan dan integritas data melalui teknik enkripsi dan dekripsi yang dilakukan secara otomatis di dalam koneksinya, untuk menggunakan SSH dibutuhkan otentifikasi user berupa kunci umum dan *password* yang terenkripsi. SSH digunakan untuk mengendalikan komputer jarak jauh (*remote*), mengirim file, membuat terowongan yang terenkripsi (tunneling/port forwarding) dan lain-lain. *Port forwarding* menyediakan kemampuan untuk mengkonversi koneksi TCP tidak aman ke koneksi SSH aman untuk pengalihan koneksi dari suatu IP ke IP lain sehingga seolah-olah klien menghubungi IP tujuan secara langsung, port forwarding melalui SSH akan membentuk sambungan yang aman antara komputer lokal

dengan komputer *remote* melalui layanan yang disampaikan[7].

3. Analisis dan Perancangan

3.1 Analisis

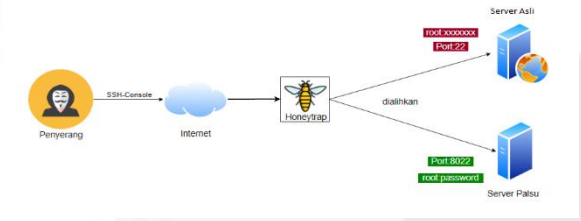
3.1.1 Gambaran Sistem Saat ini



Gambar 3.1 Gambaran Sistem Saat Ini

Berdasarkan gambaran sistem saat ini keamanan jaringan Web Server menggunakan *Modern Honey Network* yang dibantu dengan dua sensor yaitu *Sensor Honey Kippo* dan *Sensor Honey Dionae.* Ketika penyerang melakukan serangan terhadap *Web Server,* kedua sensor akan mendeteksi adanya serangan dari luar dan mengirimkan hasil aktifitas penyerang ke *Modern Honey Network* yang berfungsi untuk memanajemen dan pengelolaan data atau *logs*[8].
Modern Honey Network akan berfungsi dengan adanya jaringan internet atau *Network Cloud* yang dapat mengakses *Web Server* yang ditargetkan penyerang. Maka, *Modern Honey Network* akan mengalihkan penyerang ke dalam server jebakan dan mengelola *logs* yang dilakukan penyerang[8].

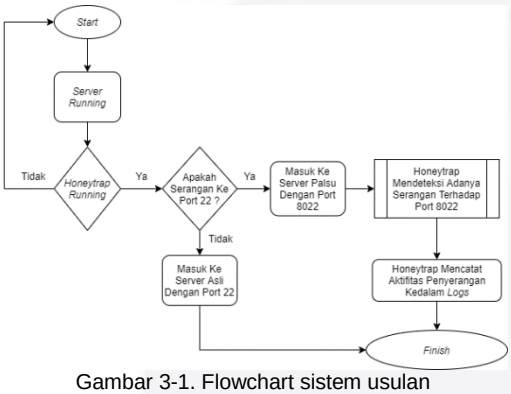
3.1.2 Gambaran Sistem Usulan



Gambar 3.2 Gambar Sistem Usulan

Dalam penelitian yang dikembangkan berdasarkan dari penelitian sebelumnya yaitu dengan cara mengoptimalkan cara kerja *Honeypot* di dalam *Web Server.* Dalam topologi yang diusulkan tersebut penyerang atau biasa dikenal dengan sebutan *"Hacker"* akan melakukan penyerangan terhadap *Web Server* dengan menggunakan metode SSH-*Console* dan menghubungkan perangkat yang digunakan penyerang dalam jaringan komputer, yaitu melalui jalur *internet.*
Web Server yang digunakan telah dilengkapi bahasa pemrograman golang untuk menjalankan *Honeytrap*, yang bertujuan untuk menjebak penyerang yang melakukan penyerangan terhadap port SSH. Setelah itu, *Honeytrap* akan bekerja dengan cara mengalihkan IP Target ke sebuah server yang dibuat untuk menjebak penyerang. Selain itu, jika *Honeytrap* telah berjalan sebelum penyerang melakukan akan penyerangan maka port SSH yang sudah dilengkapi *Honeytrap* akan terbuka dan mudah untuk diretas.

3.1.3 Flowchart Sistem Usulan



Gambar 3-1. Flowchart sistem usulan

3.1.4 Analisis Kebutuhan Fungsional dan Non Fungsional

Kebutuhan sistem dibagi menjadi 2, yaitu kebutuhan fungsional dan kebutuhan non fungsional. Sebagai berikut:

Kebutuhan Fungsional
1. Membutuhkan IP Public untuk menjebak penyerang dan menjalankan *Honeytrap.*
2. Membutuhkan *Web Server* untuk menjadi wadah dalam menjalankan *Honeytrap* dan menjebak penyerang.

Kebutuhan Non Fungsional
1. Dibutuhkan laptop dengan sistem operasi ubuntu 18.04 LTS.
2. Dibutuhkan modem untuk mendapatkan akses internet yang dapat menghubungkan antara komputer satu dengan komputer lain.

3. Dibutuhkan *Web Server* untuk menyediakan layanan HTTPS.
4. Dibuthkan *Honeytrap*, program yang berfungsi untuk mengamankan *Web Server*

4. Implementasi dan Pengujian

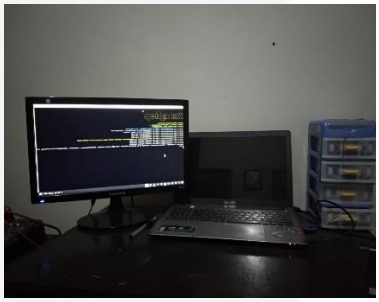
4.1 Implementasi

Dalam tahap implementasi terdapat beberapa hal yang harus disiapkan, diantaranya instalasi sistem operasi Ubuntu 18.04 LTS, instalasi *Web Server* menggunakan VPS (*Virtual Private Server*), dan instalasi *Honeytrap*. Berikut tahapan-tahapan yang harus dilakukan:

4.2 Pengujian

Setelah tahap instalasi selesai dilakukan, tahap selanjutnya adalah tahap pengujian untuk mengetahui cara kerja dari implementasi *Honeypot* dengan metode *Honeytrap*. Pengujian dilakukan dengan teknik Port Scanner, Metasploit, dan SSH-Console.

4.2.1 Server

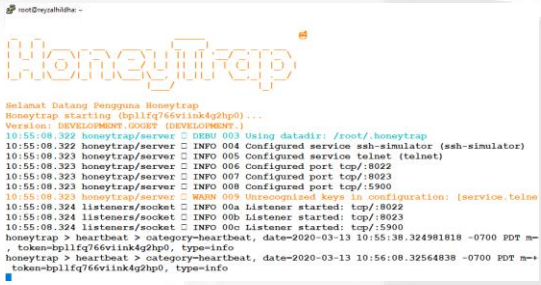


Gambar 4.1 Laptop Pertama Sebagai Server

Laptop pertama yang menggunakan monitor digunakan untuk menjalankan server dan *Honeytrap.*

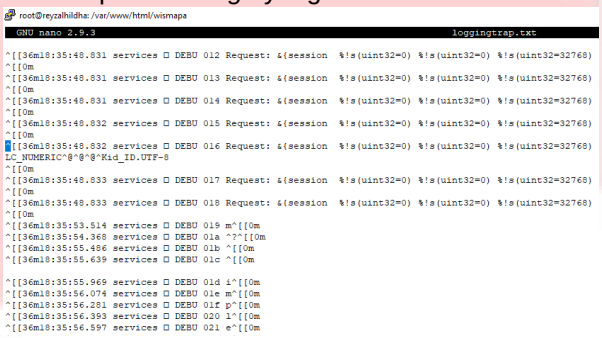
Honeytrap dapat berjalan dengan menggunakan perintah $ ./[nama_tools]. Contoh menggunakan perintah untuk menjalankan *Honeytrap* yaitu:

$ ./Honeytrap.go



Gambar 4.2 Dashboard Honeytrap

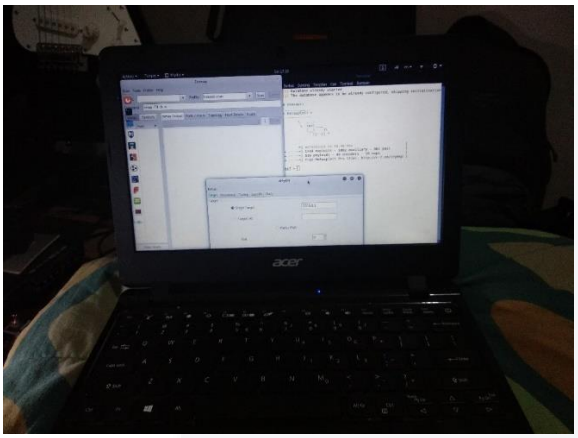
Honeytrap akan tetap berjalan disaat SSH-Console ditutup, ketika penyerang melakukan serangan kepada server jebakan maka akan tercatat pada *file logs* yang sudah disediakan.



Gambar 4.2 Logs Aktifitas Penyerang

4.2.2 Penyerang

Laptop kedua digunakan untuk melakukan serangan terhadap laptop pertama yang digunakan sebagai server.



Gambar 4.3 Laptop Kedua Sebagai Penyerang

Teknik penyerangan *Port Scanner, Metasploit, Hydra,* dan SSH-*Console*. Bertujuan untuk mengetahui celah dan mengambil informasi untuk *login* dengan mengandalkan *username, password*, serta port yang terbuka. Untuk melakukan 3 teknik penyerangan ini diharuskan untuk menyiapkan *Dictionary Attack* atau *Brute Force (*membuat daftar *username*,*password*, *IP Address*, dan masih banyak lagi).

1. Port Scanner

Teknik penyerangan ini dapat dilakukan menggunakan aplikasi Nmap atau Zenmap. Teknik ini berfungsi untuk mencari celah port, bahkan mengetahui sistem operasi yang digunakan.

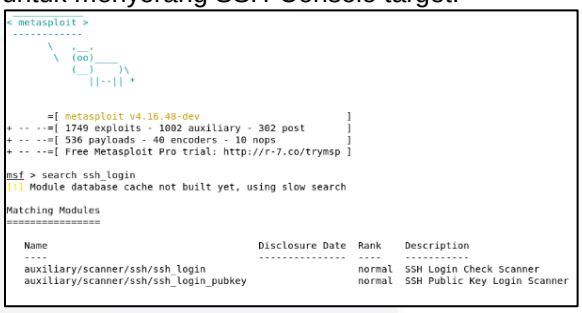
a. Langkah pertama, jalankan aplikasi Zenmap. Kemudian setting target IP target dan gunakan pilihan metode *scanning.* Untuk melihat semua celah bisa menggunakan *Intense Scan.*

Gambar 4.4 Scanning IP Address

2. Metasploit

Teknik ini bertujuan untuk mencari *username* dan *password* yang mengandalkan celah keamanan pada server target.

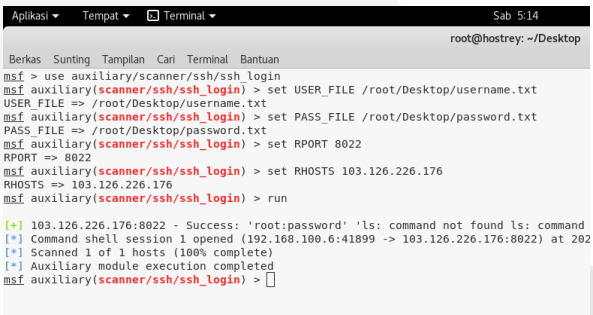
a. Langkah pertama, jalankan aplikasi metasploit. Setelah itu, gunakan perintah $search ssh_login. Perintah tersebut digunakan untuk mencari metode serangan metasploit yang bertujuan untuk menyerang SSH-Console target.



```
< metasploit >
 ------------
      \   ,__,
       \  (oo)____
          (__)    )\
             ||--|| *


       =[ metasploit v4.16.48-dev                         ]
+ -- --=[ 1749 exploits - 1002 auxiliary - 302 post       ]
+ -- --=[ 536 payloads - 40 encoders - 10 nops            ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > search ssh_login
[!] Module database cache not built yet, using slow search

Matching Modules
================

   Name                                      Disclosure Date  Rank    Description
   ----                                      ---------------  ----    -----------
   auxiliary/scanner/ssh/ssh_login                            normal  SSH Login Check Scanner
   auxiliary/scanner/ssh/ssh_login_pubkey                     normal  SSH Public Key Login Scanner
```

Gambar 4.5 Mencari Serangan Metode SSH-Console

b. Selanjutnya atur IP target, Port Target, bahkan *Username* dan *Password* target dengan menggunakan perintah berikut.



```
Aplikasi  Tempat  Terminal                                   Sab 5:14
                                                  root@hostrey: ~/Desktop
Berkas  Sunting  Tampilan  Cari  Terminal  Bantuan
msf > use auxiliary/scanner/ssh/ssh_login
msf auxiliary(scanner/ssh/ssh_login) > set USER_FILE /root/Desktop/username.txt
USER_FILE => /root/Desktop/username.txt
msf auxiliary(scanner/ssh/ssh_login) > set PASS_FILE /root/Desktop/password.txt
PASS_FILE => /root/Desktop/password.txt
msf auxiliary(scanner/ssh/ssh_login) > set RPORT 8022
RPORT => 8022
msf auxiliary(scanner/ssh/ssh_login) > set RHOSTS 103.126.226.176
RHOSTS => 103.126.226.176
msf auxiliary(scanner/ssh/ssh_login) > run

[+] 103.126.226.176:8022 - Success: 'root:password' 'ls: command not found ls: command
[*] Command shell session 1 opened (192.168.100.6:41899 -> 103.126.226.176:8022) at 202
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/ssh/ssh_login) >
```

Gambar 4.6 Teknik Metasploit

a. $ use auxilary/scanner/SSH/SSH_login

b. $ set USE_FILE /root/Desktop/username.txt

c. $ set PASS_FILE /root/Desktop/Password.txt

d. $ set RPORT 8022

e. $ set RHOSTS 103.126.226.176

f. $ run

Keterangan:

a) Bertujuan untuk metode metasploit tersebut menggunakan *auxilary* yang dapat memodifikasi serangan dan membuat sistem tertembus.
b) Teknik ini dinamakan Teknik *Brute Force* yang bertujuan untuk mencari *username* dengan daftar yang sudah disiapkan.
c) Teknik ini dinamakan Teknik *Brute Force* yang bertujuan untuk mencari *password* dengan daftar yang sudah disiapkan.
d) Digunakan untuk menentukan *PORT target* yang akan diserang.
e) Digunakan untuk menentukan *IP Address target* yang akan diserang.
f) Menjalankan Metasploit dengan metode *auxilary*. Dan *username* serta *password* berhasil ditemukan dengan “root:password”.

3. Hydra

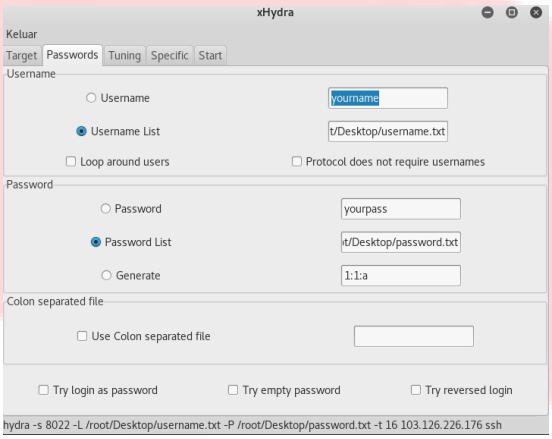
Teknik ini digunakan untuk mencari atau mencocokkan *username* dan *password* seperti halnya dengan teknik penyerangan metasploit yang menggunakan *Brute Force*.

a. Langkah pertama, tentukan *IP Address, Username*, dan *Password* target menggunakan *Dictionary Attack.*

b. Setelah itu, jalankan aplikasi *Hydra.* Lalu pada sub menu Target digunakan metode *Target List* yang dapat menggunakan daftar IP Address target yang sudah disiapkan. Kemudian gunakan port 22 dengan protokol SSH.

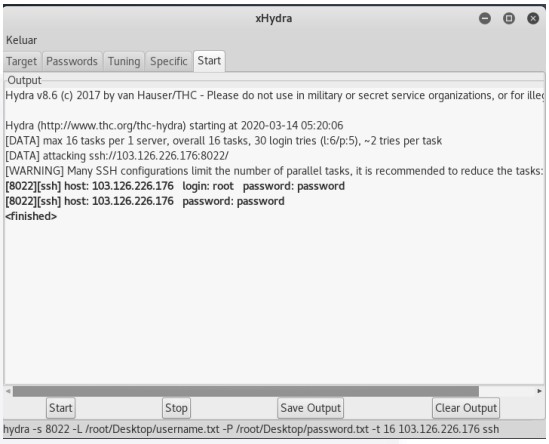
Gambar 4.7 Menentukan *IP Address* Target

c. Selanjutnya, tentukan *username* dan *password* dengan menggunakan Dictionary Attack yang sudah disiapkan.



Gambar 4.8 Menentukan *Username* dan *Password*

d. Setelah itu, pada menu Start jalankan proses scanning pencocokkan *username* dan *password* yang sudah ditentukan.



Gambar 4.9 Proses *Scanning* Hydra

4. SSH-Console

Dalam penelitian ini ketika penyerang mencoba meretas *Web Server* targetnya dengan ip 103.126.226.176 melalui port 8022 dengan menggunakan perintah berikut:

$ SSH –p 8022 root@103.126.226.176

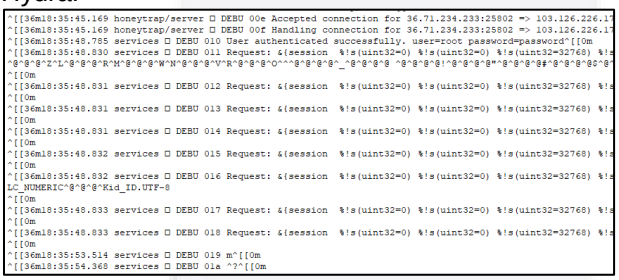
Gambar 4.10 Teknik Penyerangan SSH-Console

Tabel 4.1 Tabel Pengujian *Honeytrap*

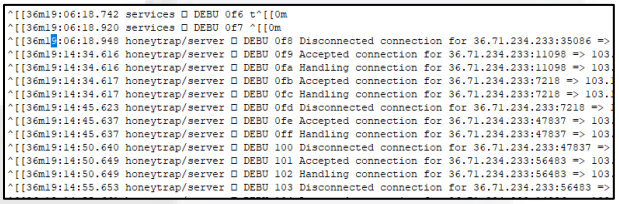
Pengujian	Terdeteksi *Honeytrap*	IP Address Target	Tanggal, dan Waktu
SSH	Ya	36.71.234.233	13 Maret 2020, 18:35:45.169 (PDT)
Nmap	Ya	36.71.234.233	13 Maret 2020, 19:06:18.948 (PDT)
Metasploit	Ya	36.71.234.233	13 Maret 2020, 19:21:24.567 (PDT)
Hydra	Ya	36.71.234.233	13 Maret 2020, 19:18:10.795 (PDT)

Dari tabel di atas dapat dibuktikan bahwa dalam pengujian menggunakan SSH, Nmap, Metasploit, dan Hydra berhasil mendeteksi IP Address Target dan juga tanggal waktu penyerangan. Pada tanggal dan waktu di tabel disebutkan bahwa waktu yang digunakan adalah waktu Los Angels (PDT), dan dengan format waktu Jam:Menit:Detik:MiliDetik.

Berikut ini beberapa *Logs* Penyerangan dalam tahap pengujian SSH, Nmap, Metasploit, dan Hydra:

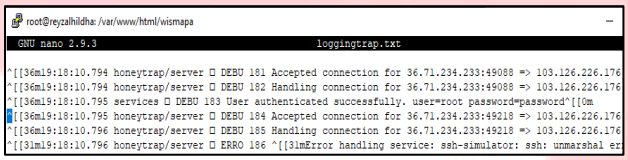


Gambar 4.11 *Logs* Serangan SSH



Gambar 4.12 *Logs* Serangan Nmap

Gambar 4.13 *Logs* Serangan Metasploit



Gambar 4.14 *Logs* Serangan Hydra

5. Kesimpulan dan Saran

5.1 Kesimpulan

Setelah melakukan pengujian terhadap implementasi *Honeypot* dengan metode *Honeytrap*, maka dapat disimpulkan bahwa:

1. *Honeytrap* yang dibangun berhasil memperkuat keamanan Web Server yaitu dengan cara mengalihkan penyerang ke server palsu.
2. *Honeytrap* yang dibangun dapat mendeteksi dari serangan port SSH, IP Address Target, dan Tanggal Waktu.

5.2 Saran

Adapun Saran dari penulis terhadap penelitian implementasi *Honeypot* dengan metode *Honeytrap*, yaitu:

1. *Honeytrap* dapat terintegrasi dengan *Mail Server*.
2. *Honeytrap* dapat dikembangakan dengan metode *Medium Interaction Honeypot* dan *High Interaction Honeypot.*

6. Daftar Pustaka

[1] R. Wahyudi, “Malware Curi Password di Server Linux,” *Kompas*, 2019. [Online]. Available: https://tekno.kompas.com/read/2013/01/28/1547210/Malware.Curi.Password.di.Server.Linux. [Accessed: 20-Dec-2019].

[2] RI, “Uu-2008-11 Informasi Dan Transaksi Elektronik,” *Undang-undang*, vol. 11, pp. 1–18, 2008.

[3] H. Awi, “DESAIN DAN IMPLEMENTASI BASIS DATA PADA IOT WEB SERVER UNTUK PRAKTIKUM IOT DAN MIKROKONTROLER,” *Libarary Telkom University*, 2017. [Online]. Available: https://openlibrary.telkomuniversity.ac.id/home/catalog/id/139987/slug/desain-dan-implementasi-basis-data-pada-iot-web-server-untuk-praktikum-iot-dan-mikrokontroler.html. [Accessed: 25-Dec-2019].

[4] Linus Torvalds, “Linux,” *Wikipedia*, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Linux. [Accessed: 20-Apr-2018].

[5] M. Edy Susanto, “Golang,” *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1689–1699, 2019, doi: 10.1017/CBO9781107415324.004.

[6] R. Verhoef, “Honeytrap,” *Duthsec*, 2019. [Online]. Available: https://docs.honeytrap.io/. [Accessed: 18-Dec-2019].

[7] H. Jusuf, “Penggunaan Secure Shell (SSH) Sebagai Sistem Komunikasi Aman Pada Web Ujian Online,” *Bina Insa. Ict J.*, vol. 2, no. 2, pp. 75–84, 2015.

[8] D. D. Laksana, S. J. I. Ismail, and N. Hendrarini, “Implementasi Honeypot Dengan Modern Honey Network,” *e-Proceeding Appl. Sci.*, vol. 3, no. 3, pp. 1815–1821, 2017.

[9] A. Budiyanto, “Pengantar Linux,” pp. 1–12, 2005.