Penerapan Keamanan Server dengan Teknik Hardening pada Sistem Operasi Ubuntu Server

Disusun sebagai salah satu syarat memperoleh Gelar Strata I
pada Jurusan Informatika Fakultas Komunikasi dan Informatika

Oleh:

REZA RIVALDO FAKHRY
L200170162

PROGRAM STUDI INFORMATIKA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
UNIVERSITAS MUHAMMADIYAH SURAKARTA

2020

HALAMAN PERSETUJUAN

Penerapan Keamanan Server dengan Teknik Hardening pada Sistem Operasi Ubuntu Server

PUBLIKASI ILMIAH

oleh:
REZA RIVALDO FAKHRY
L200170162

Telah diperiksa dan disetujui untuk diuji oleh:

Dosen Pembimbing

Ir. Bana Handaga, MT, Ph.D.
NIK.793

HALAMAN PENGESAHAN

Penerapan Keamanan Server dengan Teknik Hardening pada Sistem Operasi Ubuntu Server

OLEH
REZA RIVALDO FAKHRY
L200170162

Telah dipertahankan di depan Dewan Penguji
Pada hari Rabu, 6 April 2020
dan dinyatakan telah memenuhi syarat

Dewan Penguji:

1. **Ir. Bana Handaga, MT, Ph.D.** (……..)
   (Ketua Dewan Penguji)
2. **Dr. Endah Sudarmilah, ST, M.Eng.** (……..)
   (Anggota I Dewan Penguji)
3. **Maryam, S.Kom, M.Eng.** (……..)
   (Anggota II Dewan Penguji)

Dekan
Fakultas Komunikasi dan Informatika

Nurgiyatna, S.T., M.Sc., Ph.D.
NIK. 881

PERNYATAAN

Dengan ini saya menyatakan bahwa dalam publikasi ilmiah ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu perguruan tinggi dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkanorang lain, kecuali secara tertulis diacu dalam naskah dan disebutkan dalam daftar pustaka.

Apabila kelak terbukti ada ketidakbenaran dalam pernyataan saya di atas, maka akan saya pertanggungjawabkan sepenuhnya.

.

Surakarta, 29 Maret 2021
Penulis

REZA RIVALDO FAKHRY
L200170162

Penerapan Keamanan Server dengan Teknik Hardening pada Sistem Operasi Ubuntu Server

Abstrak

Sejak era industri 4.0, kemajuan IT menuntut segalanya menjadi praktis. Seiring dengan praktisnya mengakses suatu konten pada *server*, diharapkan kegiatan bisnis ataupun transaksi yang terdapat di dunia industri tidak terganggu oleh apapun. Salah satunya adalah adanya ancaman serangan dari peretas (*hacker*). Permasalahan ini cukup merepotkan bagi instansi-instansi yang melakukan sistem *new-normal* pada pandemik saat ini, yang mengakibatkan turunnya performansi dari instansi tersebut. Sehingga pada kesempatan ini penulis melakukan penelitian "Penerapan Keamanan Server dengan Teknik Hardening pada Sistem Operasi Ubuntu Server" dengan tujuan menambah tingkat keamanan pada *server* dengan mengurangi tingkat kerawanan di dalamnya terhadap serangan peretas (*hacker*). Penerapan ini dilakukan dengan konfigurasi sistem operasi Ubuntu dengan *tool* bawaan atau *built-in* dari sistem. Penelitian ini diharapkan dapat menghasilkan aspek keamanan yang meningkat terhadap server pada instansi-instansi untuk memperlancar transaksi bisnis maupun non bisnis yang dilakukan. Adapun metode pengujian yang digunakan ialah *blackbox*. Pengujian yang dilakukan dengan ujicoba via *virtual* serangan dari *client*, *server* dapat mendeteksi alamat penyerang, membatasi *client* asing masuk, membatasi koneksi, dan dapat memantau aktivitas yang dilakuka penyerang. Hal tersebut menandakan bahwa sistem penerapan *hardening server* berjalan dengan baik tanpa ada kerusakan sistem.

Kata Kunci: keamanan, server, ubuntu, cyber crime.

Abstract

Since the industrial era 4.0, the advancement of IT has become a practical architecture. By practically accessing content on a server, it is hoped that business activities or transactions in the industrial world will not be disturbed by anything. One of them is the threat of attack from hackers (hackers). This problem is troublesome for agencies that have implemented the new-normal system in the current pandemic, which saw the decline in performances from these agencies. So that on this occasion the authors conducted research "Application of Server Security with Hardening Techniques on the Ubuntu Server Operating System" with the aim of increasing the level of security on the server by reducing the level of vulnerability in it to hackers' attacks. This implementation is done by configuring the Ubuntu operating system with the default tools or built-in from the system. This research is expected to produce increased aspects of servers in agencies to facilitate business and non- business transactions. The testing method used is blackbox. Testing is done by testing through a virtual attack from a client, the server can answer the attacker's address, limit incoming foreign clients, limit connections, and can perform activities that the attacker does. This indicates that the implementation of server hardening went well without system damage.

Keywords: security, server, ubuntu, cyber crime.

1. PENDAHULUAN

Perkembangan teknologi di era industri 4.0 telah menunjukkan peningkatan trafik jaringan internet yang digunakan oleh masyarakat secara signifikan (Putri & Rachmawati, 2019). Pertumbuhan internet dan jaringan komputer yang terjadi pada zaman sekarang ini memberikan keuntungan dan kemudahan kepada pengguna komputer untuk dapat berbagi sumber daya dan informasi (Hakim et al., 2015). Pelanggaran Keamanan dapat berdampak sedang hingga parah pada organisasi bisnis tergantung pada sifat bisnis dan cara sistem informasi digunakan (Amit, 2014). Dengan peristiwa pandemi *covid-19* tidak hanya berdampak pada kesehatan manusia saja. Tetapi berdampak ke berbagai sektor *fundamental*. Salah satunya ialah teknologi khususnya teknologi yang digunakan instansi. Banyak instansi yang memaksakan suatu sistem hingga *overload* guna untuk tetap menjalankan aktivitas *new-normal* bisnis maupun non bisnis. Hal tersebut membuat rawan sistem keamanan jaringan terkait. Sistem keamanan jaringan menjadi hal yang sangat penting dalam menjaga sebuah jaringan, serangan yang bisa mengganggu bahkan merusak sistem koneksi antar perangkat yang terhubung akan sangat merugikan (Husain Asadullah, 2019).

Server adalah sebuah sistem komputer yang menyediakan jenis layanan *(service)* tertentu dalam sebuah jaringan komputer. *Server* didukung dengan prosesor yang bersifat scalable dan RAM yang besar, juga dilengkapi dengan sistem operasi khusus, yang disebut sebagai sistem operasi jaringan *(network operating system).*

Keamanan jaringan sangat penting untuk diperhatikan terutama di era teknologi sekarang ini. Banyak institusi atau organisasi tidak peduli dengan masalah keamanan. Namun, ketika jaringan diserang dan sistem gagal, biaya perbaikan sistem akan menjadi tinggi. Oleh karena itu, lebih banyak perhatian harus diberikan pada investasi dalam keamanan jaringan untuk mencegah kerusakan dari ancaman serangan, yang semakin beragam. Poin terpenting dalam layanan jaringan adalah keamanan akses di port (Idhom et al., 2020). Selain itu, ketika komputer *server* terkoneksi dengan internet, serangan akan meningkat, dan berbagai teknik serangan terus berkembang, sehingga tidak dapat diabaikan. Sebagai lanjutan, perlu disiapkan keamanan untuk melindungi dan meminimalkan ancaman terhadap jaringan dan *server*

Pemecahan permasalahan tersebut ada beberapa teknik pengamanan yang dapat diterapkan. Salah satunya adalah teknik *hardening*. Teknik *hardening* bertujuan untuk

menambah tingkat keamanan pada server dengan mengurangi tingkat kerawanan di dalamnya. Proses *hardening* diterapkan pada tiap *fundamental server* seperti *firewall,ssh,*dll.

Teknik *Hardening* adalah proses pengerasan suatu lapisan yang lembut sehingga lapisan tersebut menjadi lebih kuat dan lebih tahan terhadap kerusakan. Prinsip itu juga yang digunakan untuk menerapkan *hardening server* yang berpengaruh terhadap keamanan *server*. Sekumpulan disiplin ilmu dan teknik yang meningkatkan keamanan *server* yang siap pakai.

Keamanan adalah kemampuan sistem untuk melindungi informasi dan sumber daya sistem sehubungan dengan kerahasiaan dan integritas." Perhatikan bahwa ruang lingkup definisi kedua ini mencakup sumber daya sistem, yang meliputi *CPU, disk*, dan *program*, selain informasi. (Seth T. Ross ,1999).

Terdefinisikan pada buku *Unix System Security Tools by* Seth T. Ross (1999), bahwa keamanan sistem sebagai pelaksanaan perlindungan yang terus menerus dan berlebihan untuk kerahasiaan dan integritas informasi dan sumber daya sistem sehingga pengguna yang tidak sah harus menghabiskan waktu atau uang yang tidak dapat diterima atau menyerap terlalu banyak risiko untuk mengalahkannya, dengan tujuan akhir bahwa sistem dapat dipercaya dengan informasi sensitif. Keamanan komputer sering kali dikaitkan dengan tiga area inti, yang dapat dengan mudah diringkas dengan singkatan "CIA": *Confidentiality* - Memastikan bahwa informasi tidak diakses oleh orang yang tidak berwenang. *Integrity* - Memastikan bahwa informasi tidak diubah oleh orang yang tidak berwenang dengan cara yang tidak dapat dideteksi oleh pengguna yang berwenang. *Authentication* - Memastikan bahwa pengguna adalah orang yang mereka klaim.

Beberapa penelitian yang pernah dikerjakan sebelumnya kebanyakan membahas pengamanan jaringan yang mana tidak detail dengan instalasi proses konfigurasinya dan hanya menanggulangi pada satu sisi layanan. Seperti penelitian terdahulu yang pernah dilakukan dengan hasil yang menampilkan analisis data dan sampling terhadap server. (Sirait et al., 2018).

Penelitian dengan hasil menggunakan *snort backdoor*, dapat dideteksi *backdoor* yang mencurigakan. Dalam kasus ini, peneliti menemukan dua *backdoor*, yaitu *type c99.php* dan *r57.php* karena *backdoor* tersebut terdapat *script* yang mencurigakan yang

bertujuan untuk mengakses hak penuh server dan merusak sistem nya. Tetapi, *snort* belum dapat mendeteksi *backdoor* secara baik, hanya *backdoor* yang tidak terenkripsi seperti *r57.php* dan *c99.php* yang dapat terdeteksi didalam *snort*. Guna sistem pendeteksian dan pencegahan yang lebih *real*, peneliti menggunakan *backdoor scanner* yang berisi *script.php* karena *backdoor* terdeteksi dengan baik. Akan tetapi, untuk pencegahannya, perlu dideteksi dulu dengan cara di *scan*, kemudian hapus *manual script.php* yang terdapat *backdoor* di *folder* tersebut. (Kurniawan, 2013)

Penelitian dengan hasil *The rule-set firewall* yang telah diimplementasikan pada *FTP server* membuat penyerang tidak dapat mengetahui port mana saja yang sedang *open*. Metode *port knocking* dapat melindugi akses *FTP server* walaupun *client* mengetahui *username* dan *password* namun tetap dapat mendapatkan akses untuk *transfer file* atau *data*, menggunakan urutan nomor *port knocking* baik untuk membukak akses maupun menutup akses layanan *FTP* menambah keamanan dalam *transfer file*. (Khadafi et al., 2019)

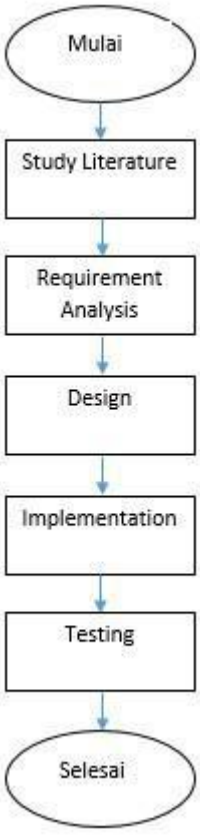
Berdasarkan studi pustaka yang telah diuraikan di atas, perbedaan penelitian ini dengan penelitian sebelumnya adalah studi kasus yang dikerjakan menitikberatkan pada pengamanan *fundamental* di *layer datalink, layer network, layer transport, layer session* layanan *server ubuntu* menggunakan teknik *hardening tools built-in* dengan instalasi konfigurasinya. Layer OSI terdiri atas tujuh lapisan yaitu Layer Application, Layer Presentation, Layer Session, Layer Transport, Layer Network, Layer Data Link, Layer Physical (Sirait et al., 2018). Pengambilan judul juga diperkuat dengan adanya kuisioner minimnya kesadaran tentang keamanan siber. Selain itu, penelitian ini dilakukan karena melihat begitu besar penggunaan server disaat pandemi *covid* ini yang mana akan rawan serangan siber.

Contoh perusahaan atau instansi yang peneliti ambil adalah PT.Airnav Indonesia cabang Solo. Perusahaan ini bergerak pada bidang navigasi. Adanya *server*, dapat membuat perusahaan ini melakukan transaksi data dengan cabang lain serta membantu proses navigasi yang dilakukan menjadi praktis. Namun dikarenakan pandemi ini, banyak perusahaan menggunakan *server* secara maksimal hingga membuat celah keamanan pada server tersebut rawan. *Status server* PT.Airnav Indonesia saat ini sedang dalam keadaan tahap perancangan. Sehingga peneliti juga akan mencantumkan beberapa rekomendasi hardware yang cocok untuk suatu server menghadapi era *new*

normal serta yang utama yaitu menambahkan beberapa sistem keamanan seperti *IDS* sebagai dasaran keamanan untuk mengurangi tingkat kerawanan *server*. Karena PT Airnav Indonesia cabang Solo hanya sebagai kantor cabang maka perancangan *server* tersebut belum dilaksanakan seperti kantor pusat di Tangerang,Banten. Peneliti juga berharap agar konsep pengembangan *server hardening* ini disampaikan ke kantor pusat.

2. METODE

Metode yang digunakan untuk menyusun penelitian "Penerapan Keamanan Server dengan Teknik Hardening pada Sistem Operasi Ubuntu Server" ini yaitu menggunakan metode eksperimental. Metode eksperimen merupakan penelitian yang digunakan untuk mencari pengaruh perlakuan tertentu terhadap dampaknya dalam kondisi yang terkendalikan (Jaedun, 2011). Adapun diagram alir dalam percobaan ini sebagai berikut.



Gambar 1. Diagram alir pengerjaan

2.1 Requirement Analysis

Langkah awal dalam penelitian ini adalah analisis kebutuhan yang berguna untuk menentukan kebutuhan dalam penelitian. *Server* tersebut direncanakan mempunyai

penyimpanan data yang berkaitan dengan navigasi udara seperti lalu lintas udara, telekomunikasi penerbangan, informasi aeronautika, informasi meteorologi penerbangan, informasi pencarian dan pertolongan. Dikarenakan data tersebut *vital* bagi navigasi maka peneliti memprioritaskan pengamanan pada sisi *ssh server* tersebut agar tidak di *overtake* oleh pihak asing. Peneliti juga membatasi koneksi untuk akses ke *server* agar lebih terorganisir. Hanya *host* yang didaftarkan saja yang diberikan izin akses. Selain *host* yang terdaftar, host akan ditolak. Berikut beberapa perangkat lunak dan keras yang diperlukan.

a. Laptop (ASUS Notebook K46CM, Processor Core i7, RAM 8GB, Nvidia GT630M, Windows 10 dan Linux Kayon OS)
b. PC (Processor Intel G5400 Gold, RAM 12GB, Nvidia GT730, Windows 10)
c. Virtualbox (Linux Client dan Server)
d. Tools Built-in di Linux Ubuntu Server
e. Nmap, knockd, portsentry, cowrie
f. Putty
g. Modul python,git

Pengumpulan data dilakukan di perusahaan Airnav Indonesia. Adapun data yang terkumpul adalah sebagai berikut :

a. Spesifikasi pada *server* yang diinginkan
b. Perkiraan kendala *server* yang akan terjadi

Tabel 1. Spesifikasi server yang direncanakan

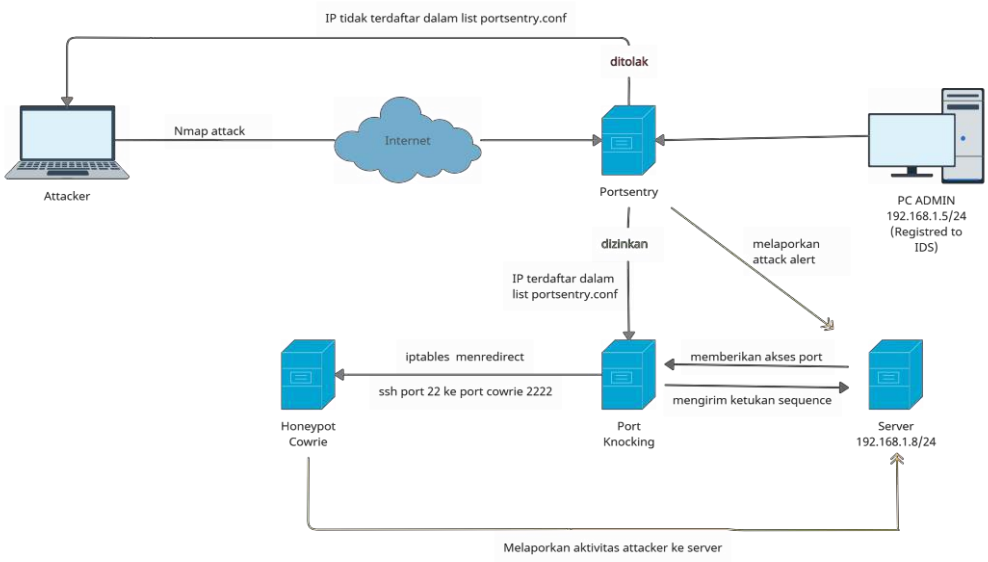
Hardware	Merk
Processor	Intel - Core i9 7980XE
Motherboard	Gigabyte - X299 Designare EX
VGA	2 x Gigabyte - GeForce GTX 1080 Ti Gaming OC 11G
Memory	Gskill Trident 64GB Quad Channel
Harddrive	8TB Western Digital Gold HDD & 500GB Samsung NvMe SSD
Cooler	Corsair H150i 360mm Watercooling
Case	Crystal Series 570X RGB ATX Mid-Tower Case White
PSU	Corsair HX1000i

Tabel 2. Perkiraan kendala server umumnya

Kendala	Lokasi
Spesifikasi server yang sudah using	Hardware
Belum ada pengamanan port tertentu	Port
sistem firewall belum dimaksimalkan	Firewall
Sistem kernel lawas mengakibatkan proses task lambat	Sistem kernel
Belum adanya intrusion detection sistem	Sistem kernel
Belum adanya honeypot	Sistem kernel

2.2 Design

Setelah menyelesaikan langkah analisis kebutuhan, desain dibutuhkan untuk memperjelas cara kerja sistem hardening server. Adapun skema sistem perancangan seperti berikut.



Gambar 2. Skema desain sistem perancangan

Keamanan pada skema desain sistem perancangan ini yaitu melindungi *server* yang mencakup layanan file *server* dimana menyimpan dari serangan *attacker* yang memanfaatkan celah pada *port ssh*. Dalam skema ini terdapat tiga lapisan keamanan yang diterapkan yaitu *IDS Portsentry*, *Port Knocking dan Honeypot Cowrie*. Ketiga lapisan tersebut memberikan layanan keamanan berupa pendeteksi *port scanning*, pemblokiran *IP*, sistem buka-tutup port dengan *sequence*, mengetahui *IP* penyerang dan pemantauan aktivitas penyerang melalui *server* jebakan. Jadi untuk mengakses

server diperlukan akses khusus yang dieksekusi oleh ketiga lapisan tersebut, yang dimana akses khusus itu diberikan kepada PC admin (192.168.1.8).

Berdasarkan gambar 2.2, terlihat bahwa penyerang berusaha melakukan serangan *nmap* melalui internet di *port server SSH* untuk men*takeover server*. *Server* berisi *IDS Portsentry* yang merupakan aplikasi yangdirancang untuk mendeteksi port scanning dan merespon secara aktif dengan cara melakukan blocking terhadap IP yang melakukan scanning ke server. *Intrusion Detection System* atau *IDS Portsentry* pada dasarnya memerlukan konfigurasi di bagian *etc portsentry portsentry.conf, portsentry.igrone.static* dan *etc default portsentry.conf*. Yang dimana ketiga file tersebut berfungsi untuk membuat daftar list *IP* yang di ijinkan dapat dieksekusi. Apabila *portsentry* mendeteksi *IP address* yang tidak didaftarkan maka *portsentry* akan menolak koneksi yang dilakukan sedangkan jika *portsentry* mengenali *IP address* tersebut maka *portsentry* akan meneruskannya ke *port knocking*. *Portsentry* juga akan mencatat *log* penyerangan dari *attacker* dan melaporkannya ke *server*. Apabila penyerang dapat membobol sistem *portsentry* maka *port knocking* akan bekerja menutup port yang ada dengan *sequence* tertentu yang sudah ditentukan oleh *administrator*. Untuk membukanya kembali membutuhkan ketukan atau bisa disebut dengan *knockd* pada *IP* tujuan serta *sequencenya*. Sebagai pengamanan tambahan, maka dibuatlah server bayangan *atau honeypot cowrie.*

Adanya *honeypot cowrie*, *administrator* dapat menjebak penyerang masuk ke *server* bayangan tersebut tanpa penyerang sadari. Setelah itu, *cowrie* mencatat semua penyerang aktivitas selama mereka tetap dalam sistem *cowrie*. Adapun *port* 1999 yang valid untuk masuk koneksi ke layanan *server ssh*. Apabila penyerang melakukan koneksi langsung pada server asli yang menggunakan port 1999, IP penyerang akan diblokir oleh *IDS portsentry*. sebagai pencegahan dari serangan *insider* maupun luar berupa *ddos* atau *denial of service* maka peneliti menkonfigurasikan *icmp* pada *sysctl.conf* agar semua request *icmp* ditolak.

2.3 Implementation

Bagian ini membahas tentang langkah konfigurasi penerapan teknik hardening menggunakan *iptables, managament user* (membuat dan menghapus user) *port knocking, portsentry* dan *honeypot cowrie di ubuntu server 20.04 LTS*. Proses pembangunan keamanan *server* terdiri dari lima langkah. Langkah pertama adalah menerapkan

patching yang berguna untuk menambal dan mengupdate aplikasi, system, maupun distribusi sistem operasi *ubuntu server* serta menjaga *kernel linux* dan *software* tetap *up to date*. Hal Ini akan mencegah penyerang menggunakan kerentanan yang diketahui untuk masuk ke sistem. Setelah proses *patching* selesai, langkah berikutnya yaitu menerapkan *management user* dengan membuat atau menghapus user yang tidak perlu. Langkah ketiga yaitu menginstal *iptables* sebagai dasaran untuk menerima *rule-rule* keamanan, konfig yang diterapkan berupa pembatasan koneksi ke *port ssh,* paket yang dibutuhkan adalah *iptables-persistent*. Langkah keempat yaitu instalasi *port knocking* yang akan diterapkan pada *port ssh*. *Port knocking* membutuhkan paket yang bernama *knockd* untuk instalasi awal. Konfigurasi pada *port knocking* bertitik pada *etc default knockd* dan *etc knockd.conf*. Langkah keempat yaitu menginstal *IDS portsentry* yang akan bekerja pada sistem *kernel, port*, dan *iptables*. IDS memantau dan mendeteksi jaringan dari berbagai aktivitas anomali yang mengindikasikan ancaman serangan hacker, malware atau kerentanan pada sistem jaringan(Ernawati et al., 2019). *Portsentry* membutuhkan paket yang bernama *portsentry* untuk instalasi awal. Langkah kelima yaitu menginstal *honeypot cowrie* yang akan ditempatkan pada sistem *kernel*. *Honeypot cowrie* membutuhkan paket yang bernama *git python3-virtualenv libssl-dev build-essential libpython3-dev python3-minimal authbind* untuk instalasi awal. Honeypot merupakan sebuah sistem yang di bangun menyerupai atau persis dengan sistem yang sesungguhnya, dengan tujuan agar para attacker teralih perhatiannya dari sistem utama yang akan di serang, dan beralih menyerang ke sistem palsu tersebut (Handoyo, Tri; Triawan, Adi; Oktavian, n.d.). Langkah terakhir yaitu *management icmp request* pada sistem *kernel* guna untuk pencegahan serangan *ddos*. *Management icmp* ini bekerja pada *sysctl.conf*. Hasil konfigurasi dan langkah penerapan *hardening* akan dibahas pada bab 3.

2.4 Testing

Pengujian dalam penelitian ini berbentuk client yang mencoba masuk via ssh ke sistem server hardening yang telah diamankan. Fungsi server yang sudah diamankan adalah dapat mengenali IP address penyerang lalu menyerang balik apabila memungkinkan yang berarti apabila *administrator* memiliki sertifikasi CEH atau sertifikasi yang memuat tentang *ethical hacking* dimana *skill administrator* tersebut setara dengan *hacker-*

hacker, meminimalir adanya gangguan *remote control*, dan memantau aktivitas yang dilakukan penyerang melalui jebakan.

3. HASIL DAN PEMBAHASAN

3.1 Hasil Implementasi

Tabel 3. List hasil implementasi

Patching	Sistem kernel menjadi up-to-date atau terbaru
Iptables	Filter terhadap trafik jaringan dan dasaran dari port knocking, ids, honeypot
Port Knocking	Pembatasan konektivitas pada port SSH
IDS	Sistem kernel memiliki deteksi terhadap host asing dan pemblokiran
Honeypot	Sistem kernel memiliki server tipuan sebagai pengecoh penyerang dan pemantau lalu lintas di dalamnya
ICMP	Server menolak permintaan paket ICMP

3.1.1 Patching

Langkah ini dimaksudkan untuk menambal celah-celah keamanan atau memperbaiki kekurangan system dan aplikasi yang ada. Seperti gambar dibawah, patching dapat dilakukan dengan menggunakan *apt-get update* dan *apt-get dist-upgrade*. *Sudo su* merupakan salah satu perintah dalam sistem operasi linux yang hanya dapat dilakukan jika user memiliki akses *root.* Maka *sudo* akan melakukan perintah sebagai *superuser "sudo su"* dan memberikan kewenangan agar *user* biasa dapat bertindak seperti *super user*, sehingga user biasa pun dapat leluasa "menguasai sistem"

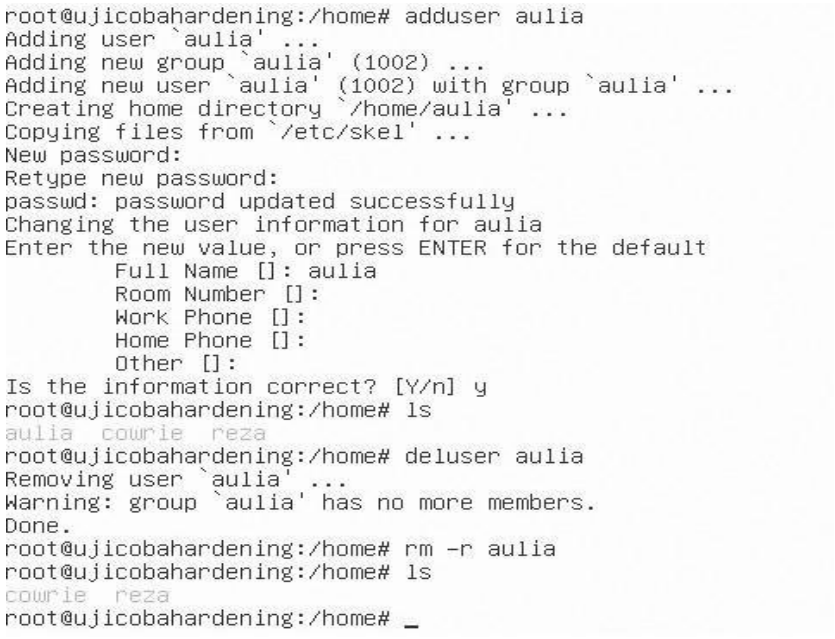
```
reza@ujicobahardening:~$ sudo su
[sudo] password for reza:
root@ujicobahardening:/home/reza# apt-get update
Hit:1 http://id.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://id.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://id.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Get:4 http://id.archive.ubuntu.com/ubuntu focal-security InRelease [109 kB]
Get:5 http://id.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [699 kB]
40% [Waiting for headers]
```

Gambar 3. Proses patching

```
root@ujicobahardening:/home/reza# apt-get dist-upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  linux-headers-5.4.0-56 linux-headers-5.4.0-56-generic linux-image-5.4.0-56-generic
  linux-modules-5.4.0-56-generic linux-modules-extra-5.4.0-56-generic
```

Gambar 4. Proses lanjutan patching

3.1.2 Management User (optional)



Gambar 5. Proses pembuatan user dan penghapusannya

Bagian ini merupakan langkah optional. Sebagai tambahan, apabila ada salah satu *user* yang sudah tidak terpakai alangkah baiknya dihilangkan untuk keefektifan *management user.* Atau apabila dibutuhkan *user* tambahan dapat dilakukan seperti pada gambar 3.3 . Perintah yang dapat digunakan untuk menambahkan *user* adalah *adduser* dimana *adduser* mencakup pembuatan *group*, direktori *home* dan informasi pribadi. Untuk menghapus *user* dapat menggunakan *deluser* dimana perintah ini akan segala yang berhubungan dengan *user* terkait kecuali direktori *user.* Dan untuk menghapus direktori yang tersisa, perintah yang digunakan ialah *rm –r.*

3.1.3 Firewall/Iptables

Mensetup 3 pilar keamanan *port knocking, portsentry dan honeypot* diperlukan *rule* baru pada *iptables* yang harus dikonfigurasikasikan agar *default* dari *rule* menjadi *filtered* ketika *discan*. Seperti yang dilakukan pada gambar 3.4. Adapun arti dari command iptables masing-masing yaitu

-A Menambahkan rule/aturan ke rantai aturan yang ada. Rantai yang valid adalah *INPUT, FORWARD, OUTPUT.*

-p Protokol yang digunakan untuk sambungan.

--dport *Port* tujuan yang digunakan oleh aturan iptables.

-j Jump ke target yang spesifik. Iptables mempunyai 4 target yaitu *accept, reject, drop, log*

Setelah menerapkan rule diwajibkan untuk menjelankan perintah *netfilter-persistent save* dan *nettfilter-persistent reload* yang berfungsi untuk merestart menginisialisasi

iptables.

```
root@ujicobahardening:/home/reza# iptables -A INPUT -p tcp --dport 22 -j REJECT
root@ujicobahardening:/home/reza# netfilter-persistent save
run-parts: executing /usr/share/netfilter-persistent/plugins.d/15-ip4tables save
run-parts: executing /usr/share/netfilter-persistent/plugins.d/25-ip6tables save
root@ujicobahardening:/home/reza# netfilter-persistent reload
run-parts: executing /usr/share/netfilter-persistent/plugins.d/15-ip4tables start
run-parts: executing /usr/share/netfilter-persistent/plugins.d/25-ip6tables start
```

Gambar 5. Iptables menerapkan rule ke port 22 dan menginisialisasi ulang rule tersebut

3.1.4 Port Knocking

Pada gambar 3.5, skrip *START_KNOCKD=1* berfungsi untuk mengaktivasi skrip *knockd* yang diterapkan. Sedangkan skrip *KNOCKD_OPTS=”-i enp0s3”* berfungsi untuk menentukan *interface* yang akan diterapkan yang ditandai dengan command option *–i interface* dan *enp0se3* meruapakan *interface.*

```
# PLEASE EDIT /etc/knockd.conf BEFORE ENABLING
START_KNOCKD=1

# command line options
KNOCKD_OPTS="-i enp0s3"
```

Gambar 6. Konfigurasi port knocking

```
[openSSH]
        sequence    = 10001,10002,10003
        seq_timeout = 3
        command     = /sbin/iptables -I INPUT -s %IP% -p tcp --dport 22 -j ACCEPT
        tcpflags    = syn

[closeSSH]
        sequence    = 10003,10002,10001
        seq_timeout = 3
        command     = /sbin/iptables -D INPUT -s %IP% -p tcp --dport 22 -j ACCEPT
        tcpflags    = syn
```

Gambar 7. Konfigurasi port knocking lanjutan

Penjelasan gambar 7, hal utama yang harus dikonfigurasikan adalah pada skrip *[openSSH]* dan *[closeSSH]*. *sequence* = 10001,10002,10003: *Knock* akan membuka *port SSH* saat urutan selesai dari mesin klien. *seq_timeout* = Opsi ini menentukan berapa lama waktu yang dimiliki untuk menyelesaikan urutan ketukan. *command = / sbin / iptables -I INPUT -s% IP% -p tcp –dport 22 -j ACCEPT*: Perintah ini akan membuka *port* 22. *sequence* = 10003,10002,10001: Knock akan menutup *port* SSH saat urutan selesai dari mesin klien. *command = / sbin / iptables -D INPUT -s% IP% -p tcp –dport 22 -j ACCEPT*: Perintah ini akan menutup *port* 22. *tcpflags* = Jenis paket yang harus diterima setiap port dalam secret sequence. Paket *SYN (synchronize*) adalah yang pertama dalam permintaan koneksi *TCP*, yang disebut *three-way handshake*.

3.1.5 Portsentry

Fungsi dari mengganti angka 0 menjadi 1 pada skrip *BLOCK_UDP* dan *BLOCK_TCP* yaitu untuk memblokir *scanning* pada *port udp* dan *tcp*.

```
# 0 = Do not block UDP/TCP scans.
# 1 = Block UDP/TCP scans.
# 2 = Run external command only (KILL_RUN_CMD)

BLOCK_UDP="1"
BLOCK_TCP="1"
```

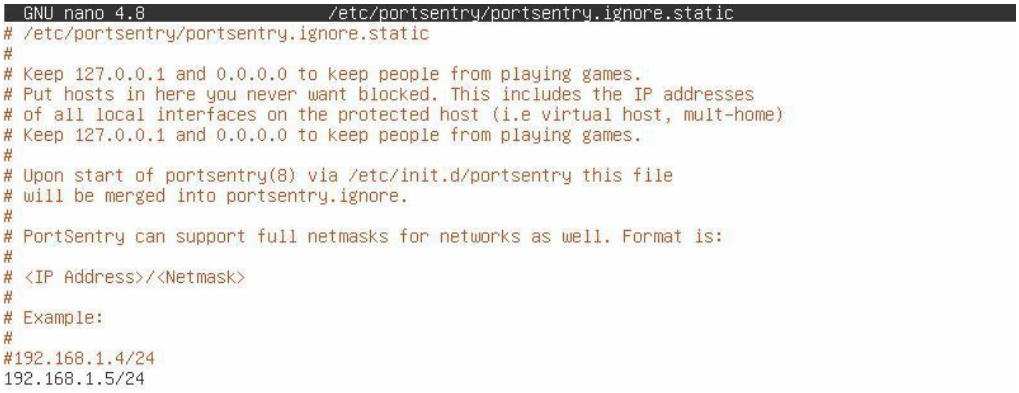
Gambar 8. Konfig IDS portsentry

```
# iptables support for Linux
KILL_ROUTE="/sbin/iptables -I INPUT -s $TARGET$ -j DROP"
#
```

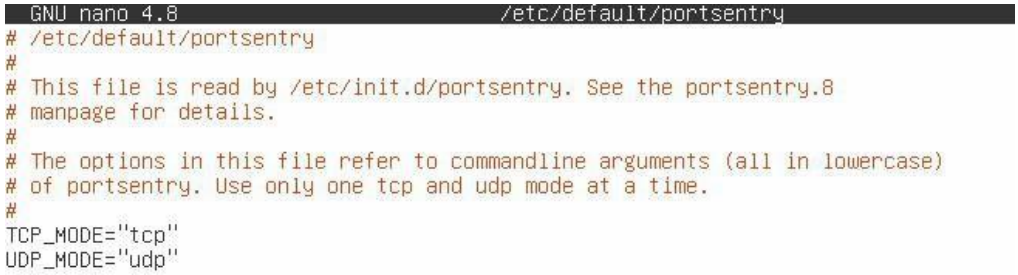
Gambar 9. Konfig IDS blokir
dengan iptables

Perintah pada gambar 8. berikut merupakan untuk memaksimalkan fungsi *IDS* untuk melakukan pemblokiran dengan *iptables. sbin/iptables* = lokasi penyimpanan *iptables -I* = aksi chain yang akan dilakukan, biasanya ada *INPUT*.

Penjelasan gambar 10 dan 11 merupakan skrip yang berfungsi sebagai daftar *IP address* yang diberikan akses oleh *IDS* dan memastikan bahwa *mode tcp* dan *udp* telah terinisialisasi.



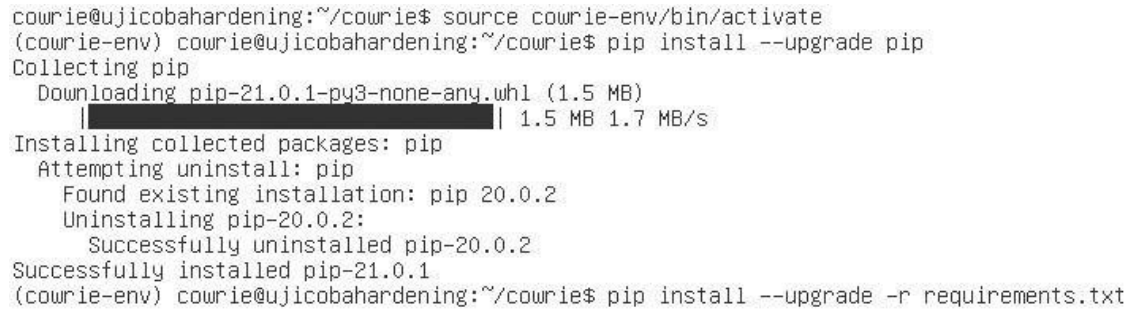
Gambar 10 Konfig IDS
portsentry lanjutan



Gambar 11. Pengecekan protokol
pada IDS portsentry

3.1.6 Honeypot

Pada gambar 12. dijelaskan bahwa untuk *menginstall* keperluan *cowrie* dibutuhkan aktivasi *virtual environment* dengan cara *source cowrie-env-/bin/activate* . Setelah *virtualenv* telah aktif maka instalasi *pip* yang berkaitan dengan *cowrie* dapat dilanjutkan.



Gambar 12. Mengaktifkan virtualenv dan menginstall keperluan pip cowrie

Penjelasan dari gambar 3.12 merupakan agar *port* 22 menjadi *port* yang digunakan masuk *server cowrie* yang dimana penyerang masuk ke *server* tipuan maka dilakukan perintah rule dibawah ini.

```
cowrie@ujicobahardening:~$ sudo iptables -t nat -A PREROUTING -p tcp --dport 22 -j REDIRECT --to-por
t 2222
[sudo] password for cowrie:
cowrie@ujicobahardening:~$
```

Gambar 13. menerapkan rule
firewall untuk cowrie

-t = *table chain* yang digunakan.Bisa berupa *nat,filter*.

-A = menambahkan *rule* PRE-ROUTING= Digunakan untuk mentranslasikan address sebelum proses *routing* terjadi, yaitu merubah *IP* tujuan dari paket data biasanya disebut dengan *Destination NAT* atau *DNAT*

-p = Parameter ini untuk menentukan perlakuan terhadap protokol.

--dport = *Port* tujuan yang digunakan oleh aturan *iptables.*

-j = Memberikan keputusan setelah paket data cocok dengan aturan. REDIRECT=*Chain* paket di *redirect* ke suatu *address* dan *port* tertentu

3.1.7 ICMP Request

```
 GNU nano 4.8                                /etc/sysctl.conf
#
# /etc/sysctl.conf - Configuration file for setting system variables
# See /etc/sysctl.d/ for additional system variables.
# See sysctl.conf (5) for information.
#

#kernel.domainname = example.com

# Uncomment the following to stop low-level messages on console
#kernel.printk = 3 4 1 3

##############################################################3
# Functions previously found in netbase
#

# Uncomment the next two lines to enable Spoof protection (reverse-path filter)
# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks
#net.ipv4.conf.default.rp_filter=1
#net.ipv4.conf.all.rp_filter=1
net.ipv4.icmp_echo_ignore_all=1
# Uncomment the next line to enable TCP/IP SYN cookies
# See http://lwn.net/Articles/277146/
# Note: This may impact IPv6 TCP sessions too
#net.ipv4.tcp_syncookies=1
```

Gambar 14. Konfigurasi ICMP
pada sysctl.conf

Dilakukan penambahan *command net.ipv4.icmp_echo_ignore_all=1* yang digunakan untuk pemblokiran permintaan terlalu besar pada *single system* dapat berpotensi menimbulkan *denial of service* yang dapat menghambat kinerja maupun performa sistem.

3.2 Pengujian

Tabel 4. List pengujian

Kelas Uji	Skenario Uji	Input	Response Server	Hasil	Keterangan
Patching	Melakukan pengecakan pembaruan sistem	lsb-release -a	menampilkan versi sistem	valid	sistem telah berhasil terupdate dengan versi terbaru
Iptables	Melakukan inisialiasi untuk 3 pilar keamanan dengan tes scan ke localhost	nmap 192.168.1.8	memfilter port 22	valid	sistem telah berhasil mengubah status port 22 menjadi filtered
Port Knocking	Mencoba knocking	knock -v 192.168.1.8 10001 10002 10003 -d 500	membuka knock	valid	sistem telah berhasil membuka port knock
Port Knocking	Mencoba Knocking	knock -v 192.168.1.8 10003 10002 10001 -d 500	menutup knock	valid	sistem telah berhasil menutup port knock
IDS	Melakukan scanning terhadap server	nmap 192.168.1.8	memberikan alert attack & memblokir IP attacker	valid	sistem telah berhasil mencatat serangan & memblokir ip penyerang
Honeypot	Melakukan akses ke server tipuan	ssh root@192.168.1.8	mendirect ke server tipuan	valid	sistem telah berhasil menjebak penyerang masuk ke dalam server tipuan
ICMP	Melakukan tes ping	ping 192.168.1.8	memblokir permintaan icmp	valid	sistem telah berhasil menolak permintaan icmp

Pengujian dilakukan dengan menyajikan percobaan dengan konsep *blackbox testing* dimana terdapat beberapa aspek yang diuji yaitu kelas, skenario, *input*, *response server*, hasil dan keterangan. Hal tersebut telah dijabarkan pada tabel 3.2 dengan tujuan menampilkan data informatif yang dapat dimengerti oleh pembaca.

Setelah beberapa percobaan dilakukan, ditemukan beberapa malfungsi sistem atau *bug* yang dijabarkan pada tabel dibawah ini.

Tabel 5. List Malfungsi

Kelas Uji	Malfungsi
Port Knocking	Port Knocking tidak bisa diterapkan secara bersamaan dengan honeypot. Dikarenakan menggunakan port yang sama. Yang dimana port pada port knocking telah di redirect oleh iptables pada konfigurasi sebelumnya. Apabila port knocking dijalankan secara individu maka hasil yang didapat adalah valid yaitu dapat membuka dan menutup knock.
Honeypot	Honeypot memerlukan rule khusus terhadap port knocking. Dikarenakan kedua pilar tersebut sama-sama menggunakan port 22. Apabila honeypot dijalankan secara individu maka hasil yang didapat adalah valid yaitu sistem dapat menjebak penyerang ke dalam server tipuan

3.3 Pembahasan

3.3.1 Iptables

Penggunaan *iptables* disini adalah sebagai penerapan *rule* yang akan diterapak pada aplikasi- aplikasi terkait. Pertama peneliti menggunakan *iptables persistent* dikarenakan apabila terjadi *server hang* atau membutuhkan *reboot server* agar *rule* didalamnya tidak hilang. Setelah diterapkan *rule* di implementasi diatas, maka *iptables* akan menyebabkan *port* 22 berstatus *filtered* yang berarti tidak semua koneksi dapat diterima.

3.3.2 Port Knocking

Konfigurasi iptables tadi dilakukan untuk men*setup port knocking*. Yang mana *port knocking* ini akan dieksekusi pada *port* 22. Setelah menkonfigurasi konfig, knock menghasilkan beberapa hasil :

a. Hasil *knocking open* dari *client* 192.168.1.5 yang diijinkan *server IDS* menunjukkan berjalan dengan baik.
b. Hasil *knocking close* dari client 192.168.1.5 yang diijinkan *server IDS* menunjukkan berjalan dengan baik.
c. Hasil *knocking open* dari client 192.168.1.13 yang tidak diijinkan *server IDS* mengalami penolakan koneksi. Jadi setiap ada ujicoba dari *ip address* 192.168.1.13 akan dicatat oleh *IDS portsentry*.

3.3.3 Portsentry

Pada *port knocking* terjadi penolakan *ssh* di *client* 192.168.1.13 yang disebabkan oleh *portsentry*. *Portsentry* melaporkan kejadian dan langsung mengeksekusi pemblokiran *IP* melalui *TCPwrapper* dan *iptables.* Adapun list *ip address* yang diizinkan aksesnya yang dikonfigurasi pada *portsentry.ignore.static*.

3.3.4 Honeypot

Penggunaan *honeypot* tidak hanya sebagai tambahan pengamanan, tetapi dapat digunakan sebagai *decoy* sehingga mengetahui IP address dari attacker dan kemudian menyerang balik apabila *administrator* mempunyai kemampuan selayaknya *hacker* atau sertifikasi *CEH.*

3.3.5 ICMP

Hasil dari konfigurasi *sysctl.conf* sebanyak 338 paket permintaan mengalami paket loss. Untuk percobaan dalam *ICMP* apabila tidak dihentikan paksa melalui *ctrl+c* maka akan terjadi paket *loss looping* yang tidak ada hentinya.

4. PENUTUP

4.1 Kesimpulan

Penerapan keamanan server yang dilakukan dapat dikatakan bahwa teknik *hardening* dapat meminimalisir atau mengurangi tingkat kerawanan terhadap *attacker*. Penelitian yang sudah diimplementasikan di *ubuntu server* juga bisa menghasilkan berupa *output file images* atau *iso* yang dapat digunakan perusahaan sebagai dasaran keamanan terhadap *server*. Pengujian yang dilakukan juga berjalan lancar ditandai dengan *server* dapat mendeteksi alamat penyerang, membatasi *client* asing masuk, membatasi koneksi, penolakan ping dan dapat memantau aktivitas yang dilakukan penyerang.

4.2 Saran

Diharapkan untuk penelitian kedepannya dapat menggunakan teknik pengujian yang lebih variatif dan aspek *hardening* yang diharapkan lebih *solid* lagi seperti menambahkan pengamanan dalam bidang enkripsi.

PERSANTUNAN

Karya ini saya persembahkan untuk Papa dan Mama atas cinta, kasih sayang, doa, nasehat, dan pengorbanan yang setulusnya tercurah untukku. Bapak Dosen pembimbing yang telah membimbing saya dengan sabar dan tekun. Teman-teman Logic Error yang selalu menghibur dari awal masuk kuliah. Dan saudari Aulia yang senantiasa menyemangati dan menemani disaat pandemi.

DAFTAR PUSTAKA

Amit, N. (2014). *Linux Server & Hardening Security Amit K Nepal*. *August*, 0–65.

Ernawati, T., Fachrozi, M. F., & Syaputri, D. D. (2019). Analysis of Intrusion Detection System Performance for the Port Scan Attack Detector, Portsentry, and Suricata. *IOP Conference Series: Materials Science and Engineering*, *662*(5). https://doi.org/10.1088/1757- 899X/662/5/052013

Hakim, L. N., Murtiyasa, B., & Handaga, B. (2015). Analisis Perbandingan Intrusion Detection System Snort Dan Suricata. *Universitas Muhammadiyah Surakarta*, 6–14.

Handoyo, Tri; Triawan, Adi; Oktavian, H. (n.d.). *ANALYSIS AND IMPLEMENTATION OF HONEYPOT USING KIPPO AS A SUPPORTING NETWORK SECURITY 1 Tri Handoyo Saputro, 2 Triawan Adi Cahyanto, 3 Hardian Oktavianto*. 1–6.

Husain Asadullah, M. (2019). SISTEM KEAMANAN SERVER DENGAN HONEYPOT DAN INTRUSION DETECTION SYSTEM (IDS) (STUDI KASUS PERUSAHAAN PRINTING SOMATEX). *Αγαη*, *8*(5), 55.

Idhom, M., Wahanani, H. E., & Fauzi, A. (2020). Network Security Applications Using the Port Knocking Method. *Journal of Physics: Conference Series*, *1569*(2). https://doi.org/10.1088/1742-6596/1569/2/022046

Jaedun, A. (2011). Metodologi Penelitian Eksperimen. *Metodologi Penelitian Eksperimen*, 0–12.

Khadafi, S., Nurmuslimah, S., & Anggakusuma, F. K. (2019). Implementasi Firewall Dan Port Knocking Sebagai Keamanan Data Transfer Pada Ftp Server Berbasiskan Linux Ubuntu Server. *Nero*, *4*(3), 181–188.

Kurniawan, B. (2013). *Analisis Pendeteksian dan Pencegahan Serangan Backdoor Pada Layanan Server*. *12*, 1–10.

Putri, D. A. P., & Rachmawati, A. (2019). Honeypot cowrie implementation to protect ssh protocol in ubuntu server with visualisation using kippo-graph. *International Journal of Advanced Trends in Computer Science and Engineering*, *8*(6), 3200–3207. https://doi.org/10.30534/ijatcse/2019/86862019

Sirait, F., Studi, P., Elektro, T., Teknik, F., Buana, U. M., Studi, P., Elektro, T., Teknik, F., & Buana, U. M. (2018). *Jurnal Teknologi Elektro , Universitas Mercu Buana ISSN : 2086 -9479 Implementasi Metode Vulnerability Dan Hardening Pada Sistem Keamanan Jaringan Fadli Sirait Program Studi Teknik Elektro , Fakultas Teknik ISSN : 2086 -9479*. *9*(1), 16–22.