Available *online* at : http://bit.ly/InfoTekJar

InfoTekJar : Jurnal Nasional Informatika dan Teknologi Jaringan

ISSN (Print) 2540-7597 | ISSN (Online) 2540-7600

Implementasi *Honeypot* Sebagai Sistem Keamanan Jaringan Pada *Virtual Private Server*

Wahyu Adi Sulaksono, Cosmas Eko Suharyanto

Universitas Putera Batam, Jalan R. Soeprapto Muka Kuning, Kepulauan Riau, 29434, Indonesia

KEYWORDS

Network Security, Honeypot, Cowrie, Virtual Private Server, Zenmap

CORRESPONDENCE

Phone: 082285868003

E-mail: mywahyuadi@gmail.com

A B S T R A K

Media internet digunakan sebagai media mengirim dan menerima data. Sebagai jaringan yang luas dan dapat diakses oleh siapapun, tidak semua data boleh diakses oleh siapa saja, sehingga manajemen dan keamanan jaringan sangat diperlukan. Akan tetapi, masih sering ditemui jaringan yang telah memiliki sistem keamanan didalamya seperti di SMKN 5 Kota Batam namun masih kurang optimal sebab kurangnya pengetahuan pihak pengelola serta mahalnya *firewall* yang dapat diterapkan pada *server* jaringan tersebut. *Honeypot* merupakan salah satu alternatif yang dapat diimplementasikan sebagai proteksi *server* karena efisien dan ekonomis tanpa mengorbankan kualitas pengamanan yang ditawarkan pada jaringan yang memakainya. *Tool* yang digunakan untuk membantu implementasi *honeypot* pada penelitian ini menggunakan aplikasi *cowrie* yang telah dikembangkan dan di instal pada *ubuntu* 16.04 *desktop* (pengujian berbasis *scan* dengan *zenmap*). Hasil dari penelitian menemukan bahwa implementasi dari *cowrie* dalam pengamanan *virtual private server* berhasil dan efektif diterapkan sebagai mekanisme pertahanan atas penyerangan yang dilakukan oleh *attacker*.

PENDAHULUAN

Ilmu pengetahuan dan perkembangan teknologi informasi saat ini menunjukkan peningkatan yang sangat signifikan, pemanfaatannya hampir meliputi banyak bidang dan lingkup penerapannya. Hal ini ditandai dengan mudahnya mengakses informasi yang dibutuhkan dari internet karena dapat diakses secara bebas. Internet menjadi media untuk mengirim dan menerima informasi yang paling populer, serta bentuk informasi yang akan dikirim dan diterima harus menggunakan perangkat yang menganut protokol sama. Internet sendiri dapat dipahami sebagai keterhubungan dari komputer-komputer dalam suatu jaringan global agar dapat saling berinteraksi dengan yang lainnya [1].

Internet sebagai jaringan yang luas dan dapat diakses oleh siapapun, menyimpan banyak manfaat yang dapat diambil, akan tetapi hal ini berbanding lurus dengan kerentanan yang juga dikandungnya. Beberapa aspek ancaman seperti privasi, integritas, serta otentikasi, menjadi bagian yang sangat hangat dipilih oleh penyerang sistem untuk melakukan kejahatan pada lingkup informasi yang ditukarkan dalam media internet [2]. Kerentanan terjadi karena internet sendiri merupakan jaringan yang bersifat publik (umum dan terbuka). Pada media yang sangat terbuka, beberapa informasi memiliki sifat privasi (rahasia), sehingga lahirlah kebutuhan untuk memproteksi informasi dan data ketika diakses melalui Internet.

Kota Batam merupakan kota yang terletak di Provinsi Kepulauan Riau memiliki beberapa sekolah menengah kejuruan, salah satunya adalah SMKN 5 Batam. Pada SMKN 5 Batam terdapat sebuah *server* yang divirtualisasikan menjadi *virtual private server* dan digunakan untuk keperluan *web server* dan *cloud storage*. *Virtual private server* sendiri merupakan teknik untuk menggandakan sebuah komputer *server* menjadi dua atau lebih *server* secara non-fisik (*virtual*) [3]. Terkait dengan sistem keamanan *server* yang telah dimiliki oleh SMKN 5 Batam, masih terdapat beberapa peluang dalam peningkatan sistem keamanannya, yang disebabkan oleh keterbatasan dari pengetahuan yang dimiliki oleh pengelola jaringan.

Salah satu metode untuk mengamankan jaringan (*server*) adalah dengan menggunakan perangkat keamanan jaringan berupa *firewall*. *Firewall* merupakan suatu sistem keamanan jaringan yang dibuat untuk melindungi komputer dari beberapa jenis ancaman yang ada. Mekanisme pengamanan *firewall* dapat di analogikan sebagai dinding (*wall*) yang mencegah serangan dapat masuk kedalam, akan tetapi belum serta-merta mengamankan jaringan secara keseluruhan (*firewall* hanya sebatas pagar penghalang) [4]. Meski demikian, *Firewall* sering dipilih untuk mencegah *attacker* (penyerang jaringan) untuk bisa melakukan kerusakan pada targetnya. Jika diamati lebih dalam, penggunaan *firewall* sendiri memerlukan biaya

https://doi.org/10.30743/infotekjar.v5i1.2783

(pembelian/berlangganan) yang mahal karena penyedia dari pengamanan ini memerlukan perankat maupun lisensi yang diperbarui secara berkala, sehingga berlangganan lisensi maupun pembelian perangkat dengan harga mahal menjadi salah satu kendala dalam penggunaannya untuk mengamankan suatu jaringan. Dari kondisi ini dibutuhkan alternatif lain untuk mengamankan jaringan yang ada tanpa terkendala permasalahan yang ditemui.

Honeypot merupakan sistem tiruan yang dibuat untuk menirukan keaslian layanan layaknya seperti *server yang* sebenarnya sehingga dapat mengelabui *attacker* yang mencoba menyerang [5]. *Honeypot* merupakan alternatif pengamanan *server* yang gratis karena *tool* ini dapat diberdayakan tanpa dipungut biaya. Dalam penerapan sistem *honeypot* memerlukan alat pendukung, salah satunya adalah *cowrie*. *Cowrie* merupakan sebuah *software* tambahan yang dibuat untuk mempermudah penggunaan (antarmuka inisialisasi) pada *honeypot* dan digunakan untuk melakukan penyamaran layanan pada *openssh server*. *Cowrie* termasuk tipe *honeypot* interaksi sedang, yang terbukti handal dalam mendeteksi dan mencatat serangan *brute force* pada ssh, telnet dan openssh *server* [6].

TINJAUAN PUSTAKA

A. Honeypot

Honeypot sebagai suatu mekanisme pertahanan yang bekerja dengan menjadi duplikasi layanan palsu dari *server* yang dijaga, telah dikembangkan secara *open source* dan dapat diunduh oleh calon pemakainya tanpa dipungut biaya apapun. *Honeypot* menggeser *firewall* sebagai proteksi terluar apabila biaya merupakan aspek yang dipertimbangkan dalam mengamankan *server* jaringan. Terdapat tiga jenis layanan *honeypot* yang dapat disesuaikan dengan potensi ancaman yang diterima *server*, dan pemakai diberikan fleksibilitas untuk memilih salah satu dari ketiga layanan ini untuk dipergunakan didalam sistem jaringannya [7]. Adapun ketiga layanan tersebut sebagai berikut.

a) *Low Interaction Honeypot*
   Low Interaction Honeypot merupakan layanan pertama dalam *honeypot* dimana *Honeypot* akan menciptakan *server* tiruan dan pengelola jaringan selaku pemilik *server* masih memiliki kendali penuh untuk mengawasi kegiatan penyusupan yang terjadi.
b) *Medium Interaction Honeypot*
   Medium Interaction Honeypot merupakan layanan kedua dalam *honeypot* dimana sebuah sistem operasi palsu dibuat untuk menjebak *attacker*. Pada layanan ini beberapa perintah *honeypot* akan dilewatkan oleh sistem, sebagai gantinya setiap informasi dari *attacker* akan direkam dan dapat dievaluasi oleh pihak pengelola jaringan. Salah satu yang menyediakan layanan ini adalah *Cowrie*.
c) *High Interaction Honeypot*
   High Interaction Honeypot merupakan layanan ketiga dalam *honeypot* dimana pengelola jaringan tidak lagi perlu mengawasi kegiatan penyusupan karena *server* asli telah direplikasi secara keseluruhan, sehingga *attacker* dipersilakan menyerang *server* replikasi yang diisikan informasi palsu, sehingga *attacker* merasa puas telah mendapatkan informasi secara ilegal, padahal *server* yang sebenarnya masih aman tanpa tersentuh sedikitpun.

B. Virtual Private Server

Teknologi yang terus berkembang memungkinkan terjadinya penerapan dari virtualisasi pada media penyimpanan, jaringan, hingga *server*. Pada komputer *server* yang diintegrasikan dengan teknologi virtualisasi disebut dengan *virtual private server*. *Virtual private server* merupakan suatu metode untuk menggandakan sumber daya yang ada. Adapun sumber daya yang dimaksud ialah sebuah komputer yang dapat dijadikan menjadi dua atau lebih komputer tidak nyata secara fisik (virtual) dalam satu komputer fisik [3]. *Virtual Private Server* membuka kesempatan untuk membangun sebuah infrastruktur *server* yang memiliki fungsi beragam (lebih dari satu) namun dengan satu sumber daya fisik yang ada (komputer tunggal).

C. Cowrie

Cowrie merupakan sebuah *software* pendukung yang berguna untuk mempermudah inisialisasi pada *honeypot* dan dimanfaatkan untuk melakukan penyamaran layanan pada *openssh server*. *Cowrie* masuk pada kategori tipe *honeypot* interaksi sedang, yang dipakai untuk mendeteksi dan mencatat serangan dari *brute force* yang menyerang ssh, telnet dan openssh *server* [6]. Konsep yang dipakai pada *cowrie* adalah pengalihan, yaitu setelah *openssh* berhasil diserang, *cowrie* akan mengarahkan *attacker* untuk masuk pada layanan palsu *honeypot*. Sehingga *attacker* akan mengira bahwa penyerangan tersebut telah berhasil, padahal *attacker* hanya masuk dalam perangkap *honeypot*. Pada fitur *cowrie* terdapat *log* atau *loging*. *Logging* adalah suatu proses untuk mencatat semua kegiatan yang dilakukan oleh *attacker* yang terjadi pada sistem palsu (*honeypot*). Sehingga pengelola jaringan dapat mengetahui kegiatan apa saja yang dilakukan *attacker* pada sistem palsu tersebut.

STATE OF THE ART

Implementasi dari Honeypot sebagai salah satu alternatif untuk mengamankan jaringan telah dipakai dan dibuktikan oleh beberapa penelitian yang telah berhasil dilakukan sebelum-sebelumnya. Fitriana dan Khasanah (2018) menemui permasalahan jaringan nikabel sebagai infrastruktur publik yang dapat diakses oleh banyak pemakai memiliki potensi ancaman yang nyata. Untuk menyiasati ini, *honeypot* dipiih sebagai mekanisme proteksi jaringan nirkabel yang ada. Berkat kekayaan yang dimiliki oleh *honeypot*, salah satu layanan tambahan seperti *Honeyd* berhasil diinisialisasi dan diketahui dalam hasil penelitian bahwa bentuk serangan daring seperti *Denial of Service*, FTP *Attack*, dan *Scan Attack* berhasil diminimalisir pada jaringan yang diteliti [5].

Hariyanto dan Surateno (2016) melakukan penelitian yang berkaitan dengan konsep kompleksitas dari sistem keamanan yang dipakai berdampak pada kemungkinan tingkat keberhasilan diterobosnya suatu jaringan. Penanganan umum pada permasalahan sistem kemanan jaringan meliputi pada *Antivirus*, *Firewall*, dan *Network Intrusion Detection System*. Mengerucut pada implementasi dan pengamatan penelitian, telah dilakukan simulasi skenario penyerangan (menggunakan *nmap*) atas penerapan dari *honeypot* sebagai *server* tiruan yang ditujukan sebagai pengalihan serangan berupa *malware*. Hasil ditemui bahwa penerapan dari *honeypot* ini berhasil mengecoh serangan yang menganggap sumber daya berhasil diterobos,

https://doi.org/10.30743/infotekjar.v5i1.2783

akan tetapi kenyataannya *server* utama masih dalam keadaan utuh tak tersentuh berkat pengamanan yang dilakukan oleh *Honeypot* [8].

Prasetya (2016) dalam penelitian yang dilakukannya membahas salah satu ancaman yang sering ditemui oleh *server* dan berakibat pada lumpuhnya kemampuan *server* dalam melayani kelompok client yang dilayaninya, yaitu serangan Denial of Service (DoS). Perkembangan pada keamanan jaringan menuntut *attacker* agar bisa tetap mengikuti kondisi dengan meningkatkan kemampuan metode serangnya, dalam hal DoS lahirlah DDoS (Distributed DoS). DDos lebih maju metode serangnya sebab pengembangan terjadi pada efisiensi distribusi pembanjiran layanan yang memicu kelumpuhan *server*. Dari permasalahan ini dirumuskan solusi berupa pembuatan sistem palsu untuk mengelabuhi *attacker* yang mengira sedang menyerang sistem aslinya. Hasil penelitian yang diterapkan pada *Linux Virtual Server* menggunakan sistem palsu dari *honeypot* mencapai keberhasilan keamanan 100% sebab ditemui bahwa sistem utama bersih dari serangan dan *honeypot* menanggung semua serangan masuk yang dilancarkan *attacker* [9].

METODOLOGI

Metodologi pada penelitian akan memaparkan atas langkah yang peneliti tentukan untuk mencapai hasil yang direncanakan dan cara untuk mencapainya. Proses penelitian dimulai pada September 2019 hingga Desember 2019. Lokasi penelitian dilakukan beralamat di SMKN 5 Batam, Kavling Bukit Kamboja Kelurahan Sei Pelunggut Kecamatan Sagulung Kota Batam Provinsi Kepulauan Riau.

A. Pengumpulan Data

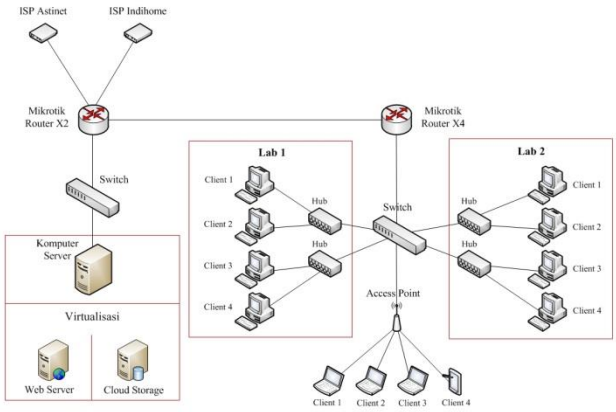
Hipotesis identifikasi permasalahan ini didapatkan dari pengumpulan data yang dilakukan secara berkala dengan memanfaatkan tiga teknik yang tersedia, yaitu

1. Observasi
   Observasi (Pengamatan) dilakukan peneliti dengan mengunjungi langsung dan mengamati secara seksama terkait keberadaan jaringan dan komponen yang dimiliki SMKN 5 Batam.
2. Wawancara
   Wawancara dilakukan kepada pihak yang bertanggung jawab untuk mengelola dan memelihara jaringan sebagai narasumber yang menyediakan informasi.
3. Studi Literatur
   Studi Literatur (Evaluasi SOTA) peneliti mencoba memahami pola, bentuk data, dan semua hal yang diperlukan untuk menyelesaikan permasalahan dari penelitian terdahulu yang relevan.

B. Analisis Jaringan

Peneliti melakukan analisis terhadap jaringan lama untuk ditemukan aspek dari sistem jaringan berjalan yang masih terdapat kesempatan pengembangan dalam keamanannya, terutama pada virtualisasi *server*. Adapun informasi yang dikumpulkan pada analisis ini untuk mengetahui terkait rancangan (geografis), topologi yang digunakan, sistem yang terpasang, dan kebijakan jaringan yang berlaku. Hal ini dilakukan guna mendapatkan informasi sebanyak mungkin dari semua data yang ada dan dijadikan bahan untuk merumuskan perancangan jaringan baru yang lebih optimal.

Hasil dari pengumpulan data yang dilakukan, ditemui bahwa jaringan lama yang dibangun berkonsep pada topologi star dan dirancang pada lokasi kelas dengan dua tingkatan geografis (lantai dasar dan lantai 2). Sistem yang berjalan mengacu pada fungsi utama dibuatnya jaringan, yaitu untuk operasional pembelajaran harian siswa dan penyedia layanan *website* resmi sekolah. Kebijakan yang diterapkan berbentuk regulasi pemakaian laboraturium komputer dan standar operasional pemeliharaan *server*.

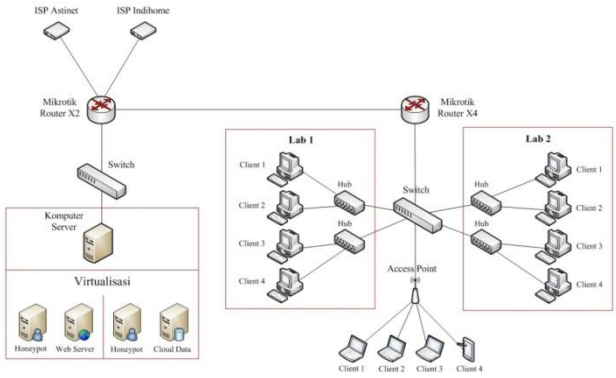


Gambar 1. Ilustrasi Jaringan Berjalan

Dapat dilihat pada gambar 1 diatas telah tercermin dari pemetaan jaringan yang ada pada lokasi penelitian. Pengembangan akan berpusat pada pemberian proteksi berupa *honeypot* pada komputer *server* yang ada agar dapat beroperasi dengan keamanan yang lebih baik.

C. Implementasi Honeypot

Tahapan Implementasi *honeypot* merupakan tahapan puncak dari penelitian yang dilakukan. Berbekal pada perumusan solusi dan perencanaan instalasi jaringan, maka proses implementasi dapat dilakukan. Hal ini mencakup pada implementasi dari solusi yang ditawarkan dan beberapa hal yang harus dilakukan agar implementasi bisa berjalan dengan baik (secara teori). Tiap *server* virtual kini dilapisi satu lapisan *honeypot* didepannya yang akan memproteksi *server* tersebut kedepannya. Sumber daya yang dimiliki *server* yang terdistribusi dalam bentuk *virtual private server* yang ada dipecah menjadi beberapa bagian (akan menjadi 4 buah) setelah implementasi dari *honeypot* dilakukan Setelah melakukan implementasi dari *honeypot* tersebut, akan dilakukan pengamatan kinerja yang berupa tahapan pengujian & pengawasan. Ilustrasi implementasi honeypot dapat dilihat pada gambar 2 berikut ini.



Gambar 2. Ilustrasi Jaringan Pasca Implementasi

Pada gambar 2 diatas telah terjadi pengembangan pada *virtual private server* dengan menggunakan honeypot yang memproteksinya. Setelah proses ini maka tahapan pengujian dan pengawasan dapat dilakukan.

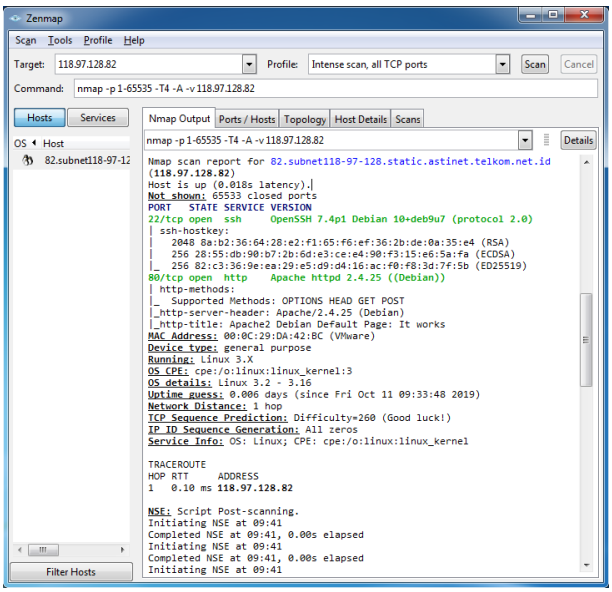
1. Pengujian
   Tahapan Pengujian merupakan tahapan lanjutan dari penelitian ini, karena pada bagian ini, peneliti mencoba mengamati dari segala aspek yang terpengaruhi dari dilakukannya implementasi *honeypot*. Tujuan dari tahapan ini berguna untuk bisa merumuskan perbaikan yang dibutuhkan (jika ada) maupun menjadi bahan pelaporan dari penelitian yang dilakukan.
2. Pengawasan
   Setelah melakukan tahapan pengujian, dilakukan tahapan terakhir dari implementasi, yaitu pengawasan (*monitoring*) terhadaap *server* yang telah diimplementasikan *honeypot* didalamnya. Hal ini merupakan bagian terakhir setelah melakukan semua implementasi yang direncanakan. Tahapan ini berguna untuk melakukan pengawasan dan mencatat semua daftar aktifitas yang terjadi pada jaringan setelah implementasi terjadi. Sehingga apabila terjadi hal yang diluar perencanaan, dapat diketahui dan ditangani agar hasil implementasi bisa berjalan dengan lancar, baik selama proses implementasi dan bahkan setelah melakukan penelitian.

HASIL DAN PEMBAHASAN

Setelah semua informasi didapatkan serta persiapan untuk implementasi telah lengkap, maka proses instalasi honeypot pada *web server* dan *cloud storage* menggunakan *cowrie* dapat dilakukan. Tahapan kegiatan yang selanjutnya yang dilakukan antara lain melakukan *scanning* sebelum instalasi *honeypot*, mengetahui bagaimana cara instalasi (menggunakan) *honeypot*, melakukan *scanning* setelah melakukan instalasi *honeypot*, pengujian penyerangan *honeypot*, pengontrolan *honeypot*, dan peninjauan keuntungan antara menggunakan *firewall* dan *honeypot*.

A. Scanning Port Sebelum Instalasi

Scanning port dilakukan agar mengetahui port berapa saja yang terbuka sebelum melakukan proses instalasi *honeypot* pada *web server*. *Scanning* port ini dilakukan menggunakan aplikasi zenmap.

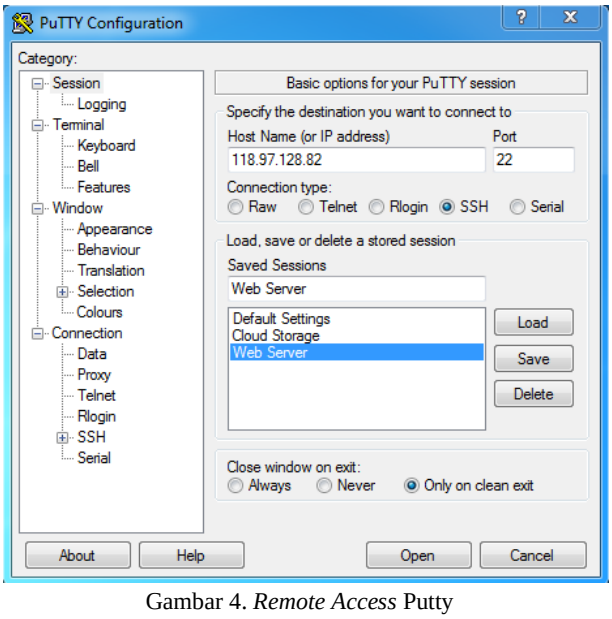


Gambar 3. *Scanning* Sebelum Instalasi

Gambar 3 diatas merupakan tampilan hasil dari *scanning* (pemindaian) menggunakan aplikasi *zenmap* sebelum instalasi *honeypot* dilakukan. Terlihat bahwa tulisan berwarna hijau hanya mendeteksi dua port saja yang terbuka, yakni port 22 yang digunakan oleh SSH dan port 80 yang digunakan http.

B. Instalasi Honeypot

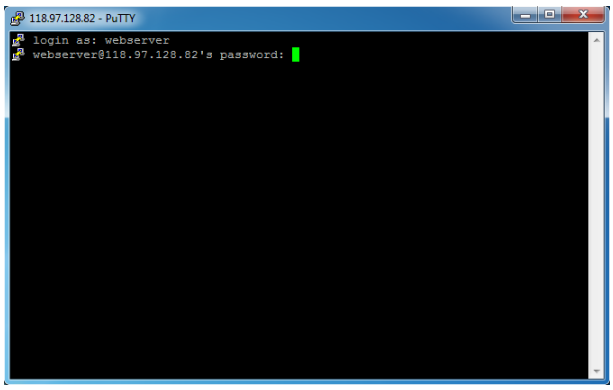
Tahap awal instalasi *honeypot* pada *web server* dengan menggunakan bantuan aplikasi *putty*. Aplikasi *putty* digunakan untuk mengontrol (*remote access*) *web server* dari jarak jauh menggunakan internet atau jaringan lokal yang berbasis CLI (*Command-Line Interface*).



Gambar 4. *Remote Access* Putty

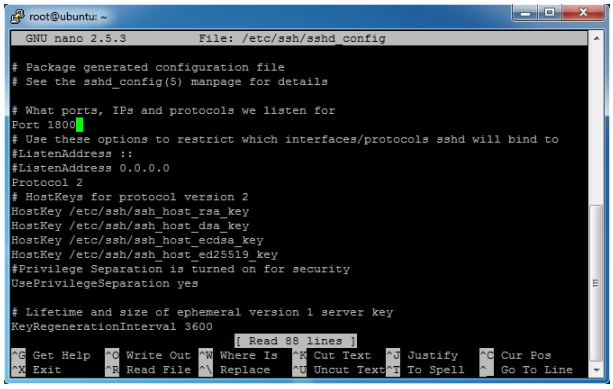
Gambar 4 diatas merupakan tampilan untuk melakukan pengontrolan terhadap *web server* menggunakan aplikasi *putty* dengan cara memasukkan IP dan port *default* (bawaan) yakni port 22 pada *web server*. Selanjutnya tampilan *putty* akan berubah dan mengarahkan pada tampilan selanjutny yaitu *login*.

https://doi.org/10.30743/infotekjar.v5i1.2783



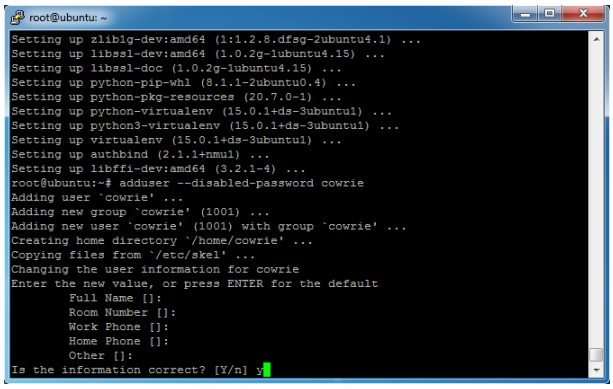
Gambar 5. *Login Web Server*

Gambar 5 diatas merupakan tampilan *login* untuk mengakses *web server* menggunakan *putty*. Setelah berhasil *login*, kemudian pembaruan (*update*) akan berjalan, maka setelahnya dapat dilakukan penggantian port bawaan (port 22) menjadi port 1800.



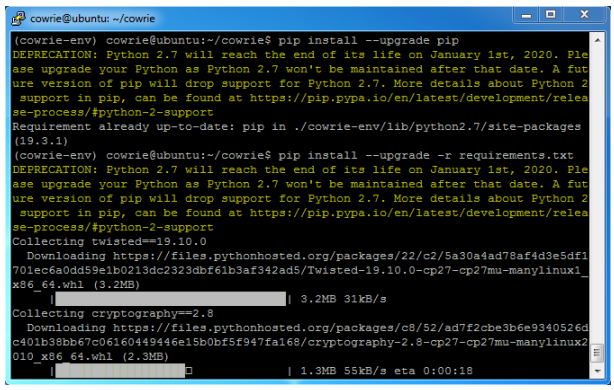
Gambar 6. Mengganti Port 1800

Gambar 6 merupakan tampilan perubahan port 22 menjadi port 1800. Setelah melakukan perubahan port, selanjutnya yakni membuat *user cowrie* yang nantinya digunakan oleh *honeypot*.



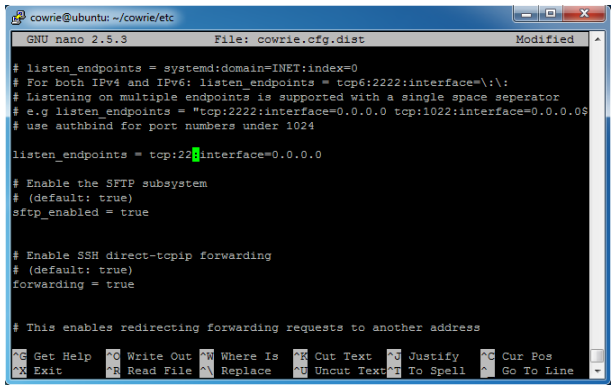
Gambar 7. Membuat *User Cowrie*

Gambar 7 diatas merupakan tampilan untuk membuat *user* pada *cowrie*. Setelah pembuatan *user* yang dapat melakukan manajerial dari *cowrie* selesai, langkah selanjutnya yakni tahap penginstalasian *honeypot*.



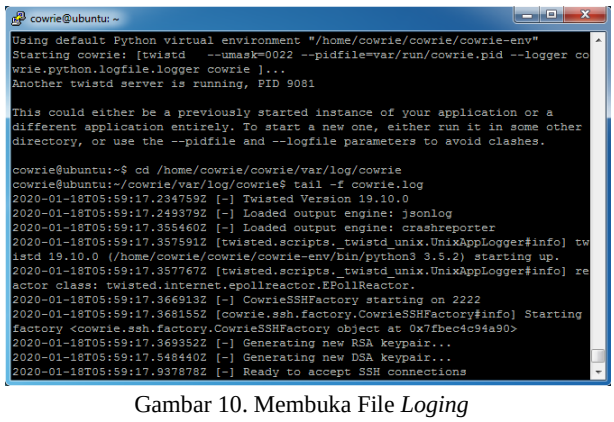
Gambar 8. Proses Instalasi *Honeypot*

Gambar 8 diatas merupakan tahapan proses instalasi *honeypot*. Setelah selesai instalasi *honeypot*, selanjutnya merubah port 2222 *honeypot* menjadi port 22.



Gambar 9. Mengganti Port 22

Gambar 9 diatas merupakan tahapan merubah port 2222 *honeypot* menjadi port 22. Sehingga port 22 yang kosong digunakan pada *honeypot*. Tahapan selanjutnya yakni membuka *file log*.

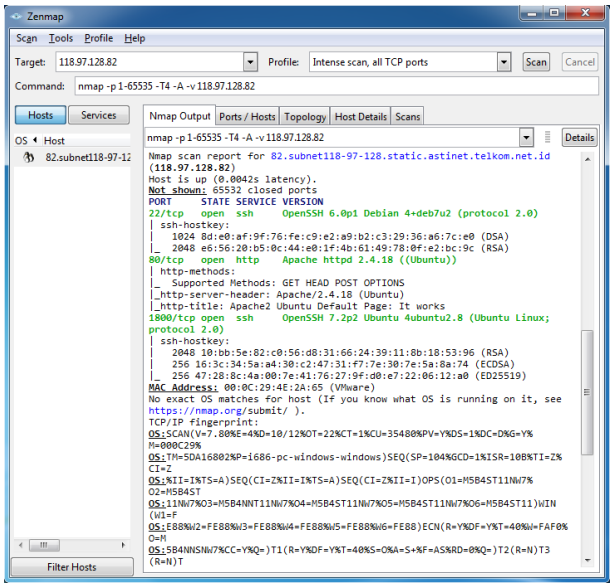


Gambar 10. Membuka File *Loging*

Gambar 10 diatas merupakan tampilan *logging* untuk melihat rekaman kejadian. Namun berhubung belum adanya penyerangan yang terjadi maka belum ada informasi berarti yang dapat dipelajari oleh pengelola jaringan ketika membuka *file log* ini.

C. *Scanning Port Setelah Instalasi*

Setelah instalasi *honeypot* dan *cowrie* selesai, maka selanjutnya dapat dilakukan *scanning* port pada *web server*.

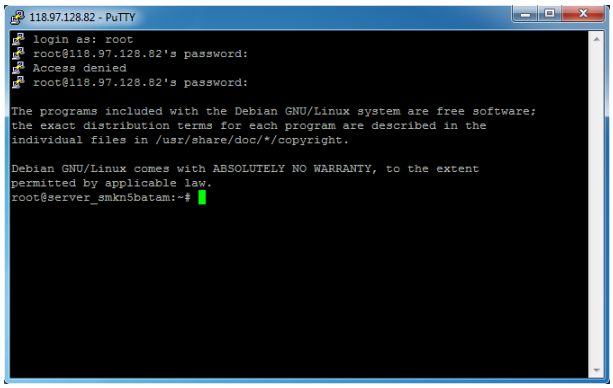


Gambar 11. *Scanning* Setelah Instalasi

Gambar 11 diatas merupakan hasil dari proses *scanning* terhadap *web server* setelah melakukan instalasi *honeypot* pada *web server*. Terlihat bahwa tulisan berwarna hijau mendeteksi adanya tiga port yang terbuka yakni port 22 digunakan pada SSH oleh *honeypot*, port 80 digunakan pada http dan port 1800 digunakan pada SSH oleh sistem asli pada *web server* untuk melakukan pengontrolan (*remote access*).

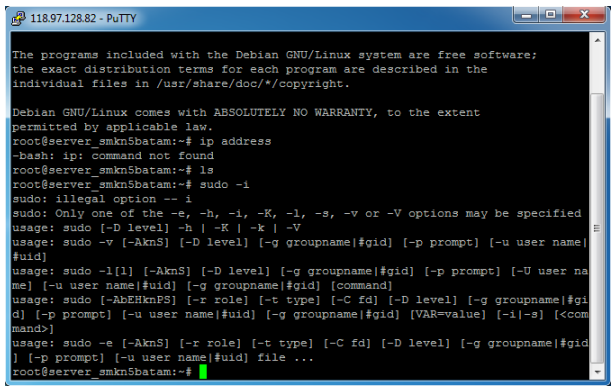
D. *Pengujian Penyerangan Honeypot*

Pengujian penyerangan *honeypot* dilakukan agar dapat mengetahui apakah sistem *honeypot* dapat bekerja dengan baik atau tidak. Selanjutnya tahap penyerangan dilakukan menggunakan port 22. Apabila pengelola jaringan melakukan pengontrolan pada *web server*, administrator jaringan tidak akan menggunakan port 22 pada pengontrolan *web server*. Langkah awal pengujian penyerangan *honeypot* yakni melakukan pengontrolan *web server* menggunakan port 22 dengan menggunakan aplikasi putty. Setelah melakukan pengontrolan *web server* menggunakan aplikasi putty dengan port 22, *cowrie* akan mengarahkan ke sistem *honeypot* pada tampilan *login*.



Gambar 12. Pengujian Penyerangan

Gambar 12 diatas merupakan tampilan penyerangan berhasil masuk ke dalam sistem *honeypot* menggunakan port 22. Terlihat bahwa saat pertama kali akan *login*, tetapi ditolak oleh sistem *honeypot* karena mamasukkan *username* atau *password* yang salah. Akan tetapi pada percobaan *login* yang kedua telah berhasil masuk kedalam sistem *honeypot*, dikarenakan memasukkan *username* dan *password* dengen benar. Selanjutnya yakni percobaan memasukkan perintah pada sistem *honeypot*.

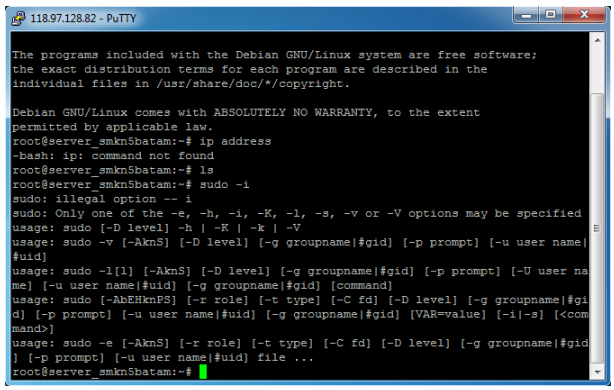


Gambar 13. Penyerang *Input* Perintah

Gambar 13 diatas merupakan tampilan percobaan memasukkan (*input*) perintah pada sistem *honeypot*. Perintah tersebut diantaranya, “ip address, ls, sudo –i”. Perintah tersebut dimasukkan dengan alih-alih agar sistem *honeypot* mengeksekusi perintah tersebut. Namun, tidak semua perintah yang dimasukkan dieksekusi oleh sistem *honeypot* dan penyerang terjebak pada sistem palsu *honeypot*.

E. *Pengontrolan/Pengawasan Honeypot*

Pengontrolan (*monitoring*) *honeypot* dilakukan agar pengelola jaringan dapat mengetahui perkembangan pada sistem *honeypot* yang telah diinstal apakah sedang diserang atau tidak. Administrator jaringan dapat melakukan *remote access* pada *web server* (asli) melalui port 1800. Karena port 22 digunakan oleh sistem palsu (*honeypot*) untuk mengecoh *attacker*. Berikut ini tampilan penyerangan yang terjadi pada port 22 *honeypot*.



Gambar 14. Pengontrolan Honeypot

Gambar 14 diatas merupakan tampilan pengontrolan *loging* oleh administrator jaringan. Terlihat bahwa telah terjadinya penyerangan dan terdapat beberapa perintah yang dimasukkan yakni “ip address, ls, sudo –i”. Sehingga semua kegiatan yang dilakukan oleh penyerang akan tersimpan pada *loging* dan administrator dapat melihatnya. Bentuk serangan yang diterima

https://doi.org/10.30743/infotekjar.v5i1.2783

honeypot terekam pada *logging file* sebagai acuan pengelola jaringan untuk mengevaluasi keamanan jaringan ataupun melakukan forensik atas serangan yang telah terjadi. Dikarenakan pengujian ini berbentuk simulasi, maka serangan berdasarkan bentuk serangan yang telah direncanakan oleh peneliti. Sedangkan bentuk skenario sebenarnya atas penyerangan yang mungkin terjadi akan lebih beragam hasil log nya, sesuai dengan bentuk serangan yang dilancarkan oleh *attacker* pada *server*.

F. Perbandingan Keuntungan

Perbandingan keuntungan dilihat dari komparasi atas harga tiap alternatif solusi terhadap pemecahan permasalahan yang ditemui pada lokasi penelitian, yaitu pemilihan solusi yang menjunjung tingkat ekonomis dari solusi tersebut untuk bisa diterapkan.

Tabel 1. Perbandingan keuntungan proteksi server

	Firewall		***Cowrie***
Model	Cisco ASA 5506-X	Avast! Anti Virus Firewall	Medium Honeypot
Gambar		Avast!	
Harga	Rp. 27.000.000	Rp. 500.000/bulan	Rp. 0

Dapat dilihat tabel 3. diatas telah dibandingkan keuntungan dari pemakaian *Honeypot*, *software firewall* (Anti Virus dengan *bundling firewall*), dan perangkat *firewall* (referensi harga dari toko *online* "bhineka.com"). Diketahui bahwa apabila menggunakan perangkat keamanan jaringan *firewall* harus mengeluarkan biaya yang besar. Hal tersebut merupakan masalah yang terjadi, dikarenakan keterbatasan biaya. Kemudian pada penggunaan firewall berbentuk software, masih dikenakan biaya. Meskipun relatif lebih murah, namun pembayaran dilakukan secara berlangganan (bulanan). Untuk itu peneliti mengusulkan penggunaan sistem *honeypot* yang tidak memerlukan biaya karena bersifat *open source*. Sehingga sistem *honeypot* dapat diputuskan menjadi alternatif yang dipilih dalam mengamankan suatu jaringan pada *virtual private server* tepatnya pada *web server* dan *cloud storage*.

KESIMPULAN

Hasil yang didapatkan dari implementasi *honeypot* menggunakan *cowrie* pada *web server* dan *cloud storage* dinyatakan berhasil. Hal ini didasarkan pada suksesnya implementasi *honeypot* pada lokasi penelitian diadakan tanpa munculnya kendala pasca implementasi. Kemudian pada sisi keamanan juga disimpulkan bahwa *honeypot* yang dipasangkan berhasil mencegah serangan masuk dan memberikan bahan evaluasi bagi pengelola jaringan dalam pemeliharaan dan peningkatan keamanan server jika dibutuhkan. Berkat kehadiran *cowrie*, proses instalasi dapat dilakukan dengan mudah, begitu juga dengan proses pengawasan lanjutan yang telah dipahami oleh pengelola server di SMKN 5 berkat honeypot yang ramah pengguna (mudah dioperasikan). *Honeypot* dipilih juga karena keunggulannya yang menekan biaya pengeluaran hingga 0 rupiah karena bersifat *open source*.

DAFTAR PUSTAKA

[1] D. Winarso, S. Syahril, A. Aryanto, E. Arribe, dan R. Diansyah, "Pemanfaatan Internet Sehat Menuju Kehidupan Berkemajuan," *Jurnal Pengabdi UntukMu NegeRI*, vol. 1, no. 1, hal. 19–23, 2017.

[2] H. Pranata, L. A. Abdillah, dan U. Ependi, "Analisis Keamanan Protokol Secure Socket Layer (SSL) Terhadap Proses Sniffing di Jaringan," hal. 21–22, 2015. *SC-SITI*.

[3] A. S. Nugraha, H. R. Andrian, dan I. Puncuna, "Penerapan Teknologi Virtualisasi Menggunakan Virtual Private Server Pada Seal Online Guardian Forest," *e-Proceeding Appl. Sci.*, vol. 1, 2015.

[4] F. A. Purwaningrum, A. Purwanto, dan E. A. Darmadi, "Optimasi Jaringan Menggunakan Firewall," *IKRA-ITH Inform.*, vol. 2, 2018.

[5] N. Fitriana dan F. N. Khasanah, "Honeypot Menggunakan Honeyd Sebagai Solusi Keamanan Jaringan Dari Aktivitas Serangan," *BINA Insani. ICT J.*, vol. 5, 2018.

[6] I. Y. M. AL-Mahbashi, P. Chauhan, S. Shukla, dan M. B. Potdar, "Review on efficient log analysis to evaluate multiple honeypots using ELK," *IJARIIE*, vol. 2, 2016.

[7] L. P. Aidin, S. M. Nasution, dan F. Azmi, "Implementasi High Interaction Honeypot Pada Implementation of High Interaction Honeypot," in *e-Proceeding of Engineering*, 2016, vol. 3, no. 2, hal. 2172–2178.

[8] A. Hariyanto dan Surateno, "Peningkatan Keamanan Jaringan Terhadap Serangan Malware Menggunakan Teknik Honeypot Dionaea," *J-TIT*, vol. 3, no. 1, 2016.

[9] N. I. Prasetya, "Mereduksi Serangan Denial Of Services Terdistribusi Pada Linux Virtual Server Menggunakan Honeypot" *SCAN*, vol. XI, no. 3, hal. 33–47, 2016.

BIOGRAFI PENULIS

Wahyu Adi Sulaksono
Lulusan Universitas Putera Batam tahun 2020. Aktif sebagai aktivis dan pembelajar yang antusias mendalami pada bidang teknologi dan informasi.

Cosmas Eko Suharyanto
Bekerja sebagai dosen senior di Perguruan Tinggi Swasta Universitas Putera Batam. Aktif sebagai tenaga pengajar sekaligus peneliti dalam bidang keamanan jaringan yang telah menerbitkan beberapa artikel ilmiah yang dimuat berskala nasional.