

VARIOUS WAY OF PROTECTING YOUR CLOUD SERVER PORT

Abdullah a.k.a Mr.Doel
(@aabdullahfath)

Malang Cyber Crew, email: doel@mc-crew.or.id

ABSTRAK

Dunia teknologi informasi khususnya di bidang jaringan komputer pasti tidak bisa lepas dari yang namanya server. Server merupakan pusat dari segalanya dalam jaringan komputer, server banyak digunakan sebagai mail server, remote server, web server, penyimpanan file secara bersama dan sebagainya. Kini server juga semakin mudah untuk didapatkan, di internet kita melihat banyak penyedia server cloud baik windows maupun Linux, sehingga siapapun yang ingin punya server sekarang dapat memilikinya dengan harga terjangkau. Namun permasalahan dari mudahnya mendapatkan server cloud ini adalah di tingkat keamanannya, terkadang ada orang menawarkan server yang murah namun tidak berkualitas dalam tingkat keamanannya, seorang yang membeli server jika dia berpengalaman dalam keamanan jaringan pasti bisa mengatur server nya keamanan yang baik, namun bagaimana dengan orang yang baru belajar mengelola server?, pasti kalau masalah keamanan akan di nomor duakan. Biasanya seorang hacker akan mendapatkan informasi dari port yang terbuka pada server khususnya pada server cloud dan dari port ini pula, seorang hacker bisa masuk ke dalam server dan melakukan eksploitasi hingga mendapatkan akses root. Paper ini akan membahas tentang berbagai cara yang dapat dilakukan untuk melindungi port pada server cloud agar melindungi server dari serangan hacker.

Kata kunci : *mengamankan server cloud, hardening port server.*

I Pendahuluan

Dalam jaringan komputer kita mengenal server dan klien. Dimana server adalah komputer yang menyediakan berbagai sumber daya untuk memenuhi kebutuhan klien seperti penyimpanan file bersama, mail server, web server dan lain-lain. Server dapat digunakan baik untuk lokal maupun publik, maksud dari server lokal adalah server yang biasanya berada di jaringan perumahan, kantor,

universitas dan sebagainya, sedangkan server publik adalah server yang memiliki IP Address untuk membedakan antara server-server yang berada di Internet sehingga server ini bisa diakses dari mana saja.

Para pengguna server sekarang lebih banyak memilih server cloud daripada harus membuat jaringan sendiri dengan peralatan yang mahal dan alat yang dibutuhkan juga sangat banyak. Tapi memang kebutuhan orang berbeda-beda, ada yang mengharuskan seseorang untuk membangun server sendiri dan ada juga yang menyewa server cloud. Server cloud adalah server pihak ketiga yang kita manfaatkan sumber dayanya, kita hanya perlu melakukan remote atau kontrol dari jarak jauh untuk mengendalikan server tersebut walaupun server itu berada jauh di negara lain.

Seorang administrator jaringan harus selalu siap siaga melakukan monitoring pada server yang dikelolanya, karena bisa jadi ada orang yang tidak bertanggung jawab berbuat jahil pada server tersebut. Hal utama yang harus diwaspadai adalah serangan masif dari hacker yang ingin menyerang server, hacker mempunyai 1001 cara untuk melakukan teknik hacking pada server cloud, namun jangan khawatir, karena tidak ada masalah yang tidak bisa diatasi, banyak cara juga yang dapat dilakukan untuk melindungi server cloud kita dari serangan hacker asalkan seorang administrator harus selalu meng-update informasi tentang kemungkinan-kemungkinan serangan yang akan terjadi pada servernya, ibaratnya kita duluan yang melakukan hacking pada server kita sebelum ada hacker lain melakukan aksinya pada server kita.

Server memiliki port-port dimana port-port tersebut dapat menjadi jalan masuk bagi para hacker, port bukan hanya jalan masuk bagi hacker tapi juga berisi informasi header dari aplikasi yang berada pada port tersebut. Sebelumnya kita harus mengetahui apa arti dari port. Port adalah suatu nomor aplikasi yang terdapat pada jaringan komputer, nomor ini berfungsi untuk membedakan antara satu aplikasi dengan aplikasi yang lainnya dalam jaringan komputer, contohnya aplikasi SMTP yang digunakan untuk pengiriman e-mail. Aplikasi atau protokol SMTP memiliki nomor port 25, sehingga jika seseorang ingin mengirim e-mail maka pengiriman tersebut dilakukan melalui port 25. Sama seperti IP Address yang berfungsi untuk membedakan antara satu komputer dengan komputer lain yang terhubung ke jaringan, port juga seperti itu, yaitu untuk membedakan antara satu aplikasi dengan aplikasi yang lainnya pada jaringan komputer.

Setelah mengetahui pengertian dari port, kita mendapatkan kesimpulan bahwa semua aplikasi memiliki port masing-masing dan dari port inilah bisa menjadi jalan masuk bagi para hacker, misalnya

seorang hacker menembus port 25 dimana port ini digunakan untuk mengirim e-mail, bisa jadi hacker tersebut memanfaatkan server tersebut untuk mengirimkan email palsu kepada targetnya. Bagaimana jika server yang dimanfaatkan oleh hacker ini adalah sebuah server perusahaan yang terkenal? Pasti orang yang menerima email kemungkinan memiliki kepercayaan pada email palsu tersebut. Inilah bahaya yang harus diwaspadai oleh administrator jaringan, biasanya sebuah server cloud membuka banyak port untuk keperluan khusus seperti kontrol server jarak jauh dengan menggunakan SSH di port 22, akses web di port 80, database mysql di port 3306 dan lain-lain.

II Pembahasan

Banyak cara yang dapat dilakukan untuk melindungi port-port pada server cloud, tapi sebelumnya kita harus mengetahui terlebih dahulu kemungkinan-kemungkinan serangan yang terjadi pada server. Ada beberapa serangan yang mungkin terjadi pada port di sebuah server :

a. Hacker bisa mengetahui versi aplikasi pada suatu port

Setiap aplikasi pada jaringan komputer memiliki port dan aplikasi pada setiap port memiliki jenis dan versi. Misalnya aplikasi FTP pada port 21 menggunakan aplikasi ProFTPD versi 1.3.5. Lalu apa bahayanya jika suatu aplikasi diketahui versinya?, kemungkinan yang dapat terjadi adalah ketika ada suatu exploit yang di posting di internet dan ternyata exploit tersebut mengarah ke aplikasi yang sedang kita gunakan saat ini. Bukan hanya server kita yang dapat dieksploitas oleh hacker, server lain yang menggunakan aplikasi dengan versi yang sama juga kemungkinan dapat diserang oleh hacker. Banyak tool yang dapat digunakan untuk mengetahui header pada aplikasi di suatu port, kita dapat menggunakan Nmap atau telnet.

```
mrrwx@mrrwx:/opt$ nmap -sV scanme.nmap.org -T4
Starting Nmap 6.46 ( http://nmap.org ) at 2014-08-26 05:55 WIB
Nmap scan report for scanme.nmap.org (74.207.244.221)
Host is up (0.27s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 5.3p1 Debian 3ubuntu7.1 (Ubuntu
1 2.0)
80/tcp    open  http         Apache httpd 2.2.14 ((Ubuntu))
9929/tcp  open  nping-echo   Nping echo
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Gambar 1: Melihat Versi Aplikasi Pada Port Yang Terbuka Dengan Nmap

Pada gambar 1 kita bisa melihat ada 3 port yang terbuka, yaitu port 22,80 dan 9929 dan terlihat kita juga mendapatkan versi aplikasi pada setiap port yang terbuka, misalnya port 22 dengan versi **OpenSSH 5.3p1 Debian 3ubuntu7.1** yang berarti aplikasi ini berjalan pada sistem operasi linux. Dengan adanya informasi ini, seorang hacker akan berusaha untuk mencari exploit untuk bisa masuk melalui port yang terbuka, karena bukan hanya satu port maka seseorang dapat melakukan eksploitasi pada port lain yang terbuka.

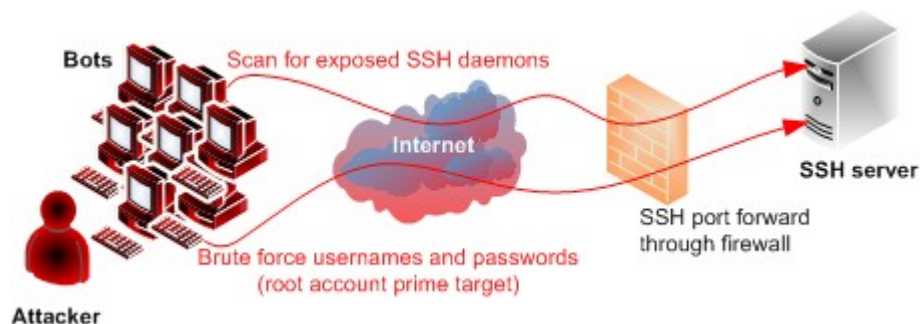
b. Brute Force Attack

Brute force attack adalah teknik untuk menebak username dan password pada suatu sistem. Walaupun teknik ini membutuhkan waktu yang tidak sedikit, namun cara ini masih banyak digunakan oleh orang-orang untuk melakukan pengetesan terhadap password yang lemah. Terkadang pada aplikasi yang berada di jaringan komputer membutuhkan login sebelum kita dapat menggunakannya. Contohnya SSH yang berada pada port 22. SSH membutuhkan username dan password sebelum kita dapat menggunakan sistem.

```
mrrwx@mrrwx:~$ ssh scanme.nmap.org
The authenticity of host 'scanme.nmap.org (74.207.244.221)'
.
RSA key fingerprint is 79:f8:09:ac:d4:e2:32:42:10:49:d3:bd:2
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'scanme.nmap.org,74.207.244.221'
known hosts.
mrrwx@scanme.nmap.org's password: 
```

Gambar 2: Prompt Login SSH Port 22

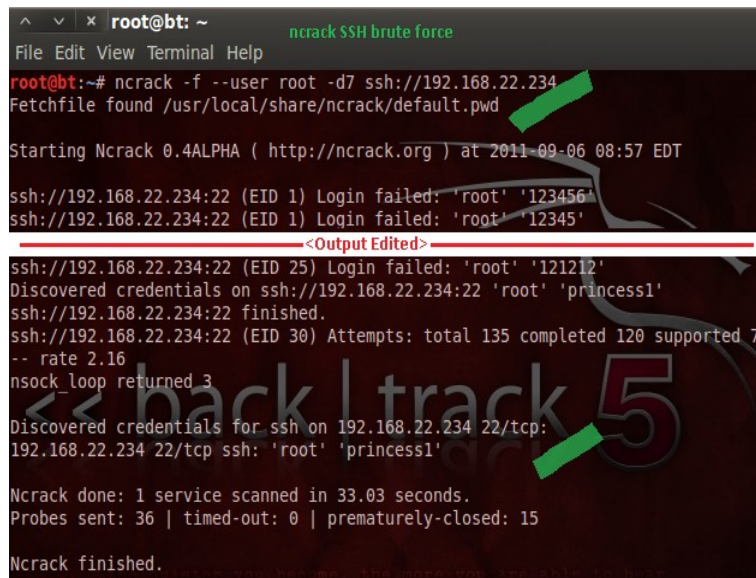
Banyak tool yang dapat digunakan untuk melakukan brute force pada SSH salah satunya adalah Ncrack. Ncrack akan mencoba untuk menebak username dan password pada SSH dengan menggunakan database yang disediakan oleh attacker itu sendiri.



Gambar 3: Metode Brute-Force

source : http://www.firewalls.com/media/wysiwyg/firewalls/blog_anat.png

Pada gambar 3, kita bisa melihat seorang attacker akan menggunakan komputer yang banyak untuk melakukan brute force, ini dilakukan karena banyak username dan password yang akan digunakan untuk dilakukan pengetesan. Biasanya akun yang terdapat pada SSH bernama **root** dan kita hanya perlu untuk mencari password dari username **root** ini. Di bawah ini adalah contoh tool Ncrack yang berhasil melakukan metode brute force pada SSH.



```
root@bt: ~ ncrack SSH brute force
File Edit View Terminal Help
root@bt:~# ncrack -f --user root -d7 ssh://192.168.22.234
Fetchfile found /usr/local/share/ncrack/default.pwd

Starting Ncrack 0.4ALPHA ( http://ncrack.org ) at 2011-09-06 08:57 EDT

ssh://192.168.22.234:22 (EID 1) Login failed: 'root' '123456'
ssh://192.168.22.234:22 (EID 1) Login failed: 'root' '12345'

<Output Edited>

ssh://192.168.22.234:22 (EID 25) Login failed: 'root' '121212'
Discovered credentials on ssh://192.168.22.234:22 'root' 'princess1'
ssh://192.168.22.234:22 finished.
ssh://192.168.22.234:22 (EID 30) Attempts: total 135 completed 120 supported 7
-- rate 2.16
nsock_loop returned 3
Discovered credentials for ssh on 192.168.22.234 22/tcp:
192.168.22.234 22/tcp ssh: 'root' 'princess1'

Ncrack done: 1 service scanned in 33.03 seconds.
Probes sent: 36 | timed-out: 0 | prematurely-closed: 15

Ncrack finished.
```

Gambar 4: Menggunakan Ncrack Untuk Brute Force

Source :

http://www.firewalls.com/media/wysiwyg/firewalls/blog_ncrack.png

3. Remote Code Injection

Code injection adalah eksploitasi bug komputer yang disebabkan oleh pengolahan data yang tidak valid. Code injection dapat digunakan oleh penyerang untuk memperkenalkan (atau "menyuntikkan") kode ke dalam program komputer untuk mengubah arah eksekusi. Hasil serangan injeksi kode dapat menjadi bencana. Misalnya, kode injeksi digunakan oleh beberapa orang untuk memasang backdoor di suatu aplikasi. Contoh dari bug remote code injection adalah bug pada aplikasi ProFTPD versi 1.3.3c (http://www.rapid7.com/db/modules/exploit/unix/ftp/proftpd_133c_backdoor) dimana seseorang dapat memasang backdoor pada aplikasi tersebut dengan menggunakan metasploit dan seseorang bisa mendapatkan hak akses root jika backdoor ini telah terpasang.

4. Memanfaatkan Kesalahan Konfigurasi Port

Hal ini sering terjadi, terkadang ada administrator yang sering melakukan konfigurasi tanpa memperhatikan keamanan dari sistemnya. Padahal banyak hal-hal yang penting yang harus dilakukan pada saat konfigurasi terutama dalam keamanan, bisa jadi konfigurasi yang kita lakukan dapat dimanfaatkan oleh orang yang tidak bertanggung jawab. Sebagai contoh adalah konfigurasi open relay pada Postfix SMTP di port 25. SMTP Open relay adalah mail server yang mengizinkan pihak ketiga untuk mengirimkan email melalui mail server tersebut. Singkatnya kita dapat memanfaatkan server yang open relay untuk mengirimkan email dengan nama server tersebut. Hal ini dapat membahayakan karena bisa saja ada orang yang memanfaatkan open relay untuk mendapatkan keuntungan dan merugikan pemilik server, bayangkan jika server yang terdeteksi open relay adalah server milik sebuah perusahaan terkenal di Indonesia. Bisa saja orang membuat email palsu untuk mendapatkan keuntungan melalui server tersebut. Banyak cara yang dapat dilakukan untuk melihat apakah suatu server terdeteksi open relay salah satunya dengan tool Nmap. Nmap akan mencoba untuk melakukan 16 tes yang berbeda pada server SMTP yang berada pada port 25.

```
Starting Nmap 6.46 ( http://nmap.org ) at 2014-08-27 05:00
Nmap scan report for 192.168.1.4
Host is up (0.00047s latency).
PORT      STATE SERVICE
25/tcp    open  smtp
|_smtp-open-relay: Server is an open relay (16/16 tests)
```

Gambar 5: Server terdeteksi open relay dengan Nmap

Masih banyak hal-hal yang dapat dilakukan oleh seorang hacker pada server kita. Setiap hari selalu ada cara baru untuk melakukan hacking, setiap hari selalu ada exploit baru yang dapat kita lihat di website <http://exploit-db.com> atau <http://1337day.com>. Kedua website tersebut bukan hanya untuk melakukan hacking tapi menjadi bahan peringatan untuk penyedia aplikasi agar dapat melakukan perbaikan pada aplikasinya jika terdapat celah keamanan.

Setelah kita mengetahui kemungkinan-kemungkinan serangan yang terjadi pada server kita terutama pada port yang tersedia, selanjutnya akan dibahas berbagai cara yang dapat dilakukan untuk melindungi port pada server. Cara-cara yang dapat kita lakukan adalah :

1. Mengganti port standar secara manual

Teknik ini adalah teknik yang bagus untuk pemula yang ingin belajar dalam bidang keamanan jaringan. Mengganti port standar maksudnya adalah mengganti port default pada setiap aplikasi yang tersedia misalnya kita dapat mengubah port 25 menjadi port 2555 sehingga jika ada seseorang yang ingin

menggunakan port 25 maka dia tidak akan bisa mengakses tersebut kecuali dia sudah mengetahui bahwa sebenarnya port yang asli berada pada port 2555. Cara ini tidak akan menghentikan proses hacker untuk dapat menembus server kita, tapi cara ini akan memperlambat kinerja dari hacker, karena bisa saja dia menggunakan port scanning untuk melakukan pencarian pada port yang terbuka. Tool seperti Nmap dapat melakukan ini, Nmap dapat melakukan port scanning pada ribuan port di jaringan komputer sehingga besar kemungkinan port yang telah kita ganti akan ditemukan oleh attacker.

Untuk mengganti suatu port, setiap aplikasi biasanya sudah menyediakan konfigurasi yang berisi pengaturan untuk mengubah port default. Contohnya pada aplikasi ProFTPD yang secara default berada pada port 21, untuk menggantinya kita hanya perlu mengedit file **proftpd.conf** lalu cari tulisan *"#Port 21 is the standart FTP port"* dan anda dapat mengganti port 21 menjadi port yang lain dengan syarat port yang ingin diganti harus diatas port 2048, misalnya 2500, 2050 dan lain-lain karena port-port di bawah antara 0-2048 merupakan well-known port yang biasanya sering digunakan pada jaringan komputer sehingga hal ini juga bertujuan untuk menghindari tabrakan antar port. Selanjutnya untuk mengganti port default pada Postfix SMTP yang berada di port 25, edit file master.cf yang biasanya berada pada direktori **/etc/postfix** cari kode ini **smtp inet n - - - smtpd** dan ganti dengan kode ini **9768 inet n - - - smtpd** angka 9768 merupakan port yang akan digunakan. Masih banyak lagi cara yang dapat dilakukan untuk mengganti port pada masin-masing aplikasi di server anda, anda bisa googling untuk melihat file konfigurasi port default pada aplikasi yang tersedia di server anda.

2. Menggunakan Kippo Honeypot Pada Protokol SSH

Kippo adalah sebuah honeypot atau firewall yang digunakan khusus untuk protokol atau aplikasi SSH. Seperti yang kita ketahui SSH (Secure Shell) merupakan sebuah protokol jaringan untuk melakukan komunikasi remote atau untuk mengontrol server dari jarak jauh dengan tingkat keamanan yang tinggi. Dulu sebelum ada SSH orang-orang menggunakan telnet untuk menghubungi server dari jarak jauh, namun telnet memiliki kelemahan dimana setiap data yang mengalir dari klien ke server atau sebaliknya dapat dimonitoring dan datanya dapat dicuri (sniffing). SSH dikembangkan untuk mengatasi masalah ini, dengan SSH semua aliran data akan dienkripsi sehingga jika ada orang melakukan sniffing pun akan mendapatkan data yang ter-enkripsi. Walaupun SSH merupakan protokol yang aman namun dalam sistem komputer tidak ada satupun yang aman, masih banyak cara yang dapat dilakukan untuk menembus pertahanan protokol ini. SSH merupakan protokol yang penting pada server, karena dengan SSH lah seorang administrator dapat mengontrol dan mengkonfigurasi

server, biasanya SSH menggunakan akun root untuk melakukan hak akses penuh terhadap server, inilah yang harus diwaspadai jika suatu saat ada orang yang mencoba melakukan brute-force pada akun root di server. Untuk mengantisipasi terjadinya serangan pada protokol SSH kita akan menggunakan Kippo Honeypot. Kippo ditulis dalam bahasa pemrograman Python, kippo dapat mencatat setiap serangan yang mungkin terjadi pada SSH seperti brute force dan percobaan lainnya yang dilakukan attacker pada SSH, beberapa fitur menarik yang disediakan oleh Kippo adalah :

- Kemampuan membuat sistem palsu, dengan fitur ini attacker seolah-olah sudah masuk ke dalam server tapi sebenarnya itu hanya sistem palsu. Sistem palsu ini memiliki fitur untuk menambah dan menghapus file sehingga akan menambah keyakinan dari attacker bahwa dia sudah benar-benar masuk ke dalam server.
- Kemampuan menambahkan isi file pada sistem palsu. Jika seorang attacker sudah berhasil masuk pada sistem palsu dia diberi akses untuk menambah isi file sehingga attacker dapat menggunakan perintah "cat" misalnya "*cat /etc/passwd*". Namun sekali lagi itu semua hanyalah sistem palsu.
- Kippo menyimpan semua aktivitas yang dilakukan attacker pada sistem palsu.
- Attacker seolah-oleh dibuat berada pada sistem yang sebenarnya baik ketika login mau logout. Padahal itu semua tidaklah asli.

Desaster, si pembuat Kippo mengatakan "By running kippo, you're virtually mooning the attackers. Just like in real life, doing something like that, you better know really well how to defend yourself!" atau dalam bahasa Indonesia "Dengan menjalankan Kippo, anda hanya perlu membayangkan sebagai seorang attacker. Seperti dalam dunia nyata, lakukan sesuatu layaknya seorang attacker, maka kamu akan tahu bagaimana cara bertahan yang baik!"

Sebagai contoh kita akan coba menginstall Kippo dan melakukan simulasi attacker yang mencoba untuk menembus protokol SSH. Disini saya memiliki sebuah server dengan sistem operasi **Ubuntu Server 14.04** dan dengan IP Address **192.168.1.4**. Untuk menginstall kippo ikuti langkah-langkah di bawah ini :

2.1 Persiapan dan Instalasi Kippo Honeypot

- Login pada server anda, anda dapat menggunakan SSH :

```
$ ssh root@192.168.1.4
```

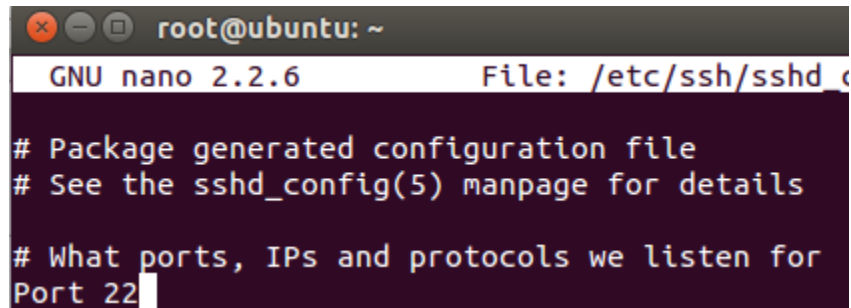

- Lakukan update dan upgrade terlebih dahulu

```
$ apt-get update
$ apt-get upgrade
```

- Ganti port default SSH, karena kita ingin membuat sistem palsu pada port 22, maka kita terlebih dahulu harus mengganti port default asli SSH ke port lain. Dalam contoh ini kita mengganti port asli dari port 22 menjadi port 3049. ketikkan perintah ini :

```
$ nano /etc/ssh/sshd_config
```

anda akan melihat kode seperti ini :



```

root@ubuntu: ~
GNU nano 2.2.6 File: /etc/ssh/sshd_config

# Package generated configuration file
# See the sshd_config(5) manpage for details

# What ports, IPs and protocols we listen for
Port 22

```

Gambar 6: Mengganti port default SSH

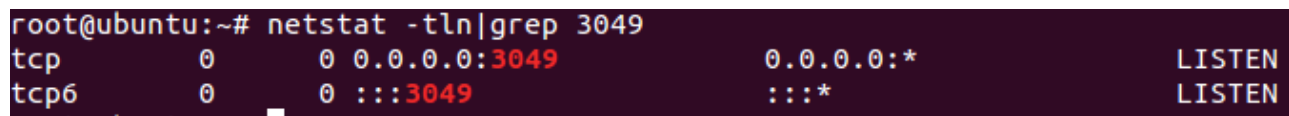
Ganti **Port 22** menjadi **Port 3049**. Tekan Ctrl + o untuk save dan Ctrl + x untuk exit. Lalu restart SSH dengan perintah :

```
$ service ssh restart
```

Sekarang lakukan pengecekan apakah port sudah terganti dengan perintah :

```
$ netstat -tln|grep 3049
```

Jika hasilnya seperti gambar di bawah ini, maka port default sudah terganti :



```

root@ubuntu:~# netstat -tln|grep 3049
tcp        0      0 0.0.0.0:3049 0.0.0.0:* LISTEN
tcp6       0      0 :::3049    :::*    LISTEN

```

Gambar 7: Melihat port SSH yang sudah terganti

- Selanjutnya install dependensi tambahan Kippo dengan perintah ini :

```
$ apt-get install python-dev openssl python-openssl python-pyasn1
python-twisted
```

Lalu install subversion yang nantinya akan digunakan untuk mendownload Kippo :

```
$ apt-get install subversion
```

- Tambahkan user baru untuk Kippo Honeypot dengan perintah :

```
$ useradd -d /home/kippo -s /bin/bash -m kippo -g sudo
```

User ini nantinya akan kita gunakan untuk mengkonfigurasi semua tentang Kippo.

- Pada sistem Linux, user root adalah satu-satunya pengguna yang dapat mengakses port di bawah 1024, karena kebanyakan konfigurasi pada port tersebut membutuhkan hak akses root, untuk alasan keamanan ini bukanlah ide yang bagus untuk menjalankan kippo sebagai user root. Cara yang terbaik adalah kita akan menggunakan user lain untuk menjalankan Kippo Honeypot dan juga layanan yang lainnya jika diperlukan. Kita akan menggunakan AuthBind untuk dapat menjalankan layanan pada server dengan status non-root user. AuthBind memungkinkan administrator server untuk mengizinkan user tertentu dalam menggunakan layanan pada port yang berada di bawah port 1024. Perintah untuk melakukannya adalah :

```
$ sudo apt-get install authbind
```

Lalu buat sebuah file sesuai dengan port yang diinginkan :

```
$ touch /etc/authbind/byport/22
```

Selanjutnya ubah kepemilikan file tersebut kepada user yang akan kita izinkan untuk mengelola port 22. Contohnya user **kippo** yang telah dibuat sebelumnya :

```
$ chown kippo /etc/authbind/byport/22
```

Ubah permission file tersebut agar bisa dibaca, ditulis dan dieksekusi :

```
$ chmod 777 /etc/authbind/byport/22
```

- Setelah melakukan konfigurasi pada authbind, saatnya men-download Kippo, dari sini kita akan menggunakan user **kippo** untuk melakukan instalasi dan konfigurasi Kippo Honeypot, untuk mengganti user ketikkan :

```
$ su kippo
```

Lalu ketikkan perintah untuk memastikan kita berada pada direktori home kippo :

```
$ cd
```

Download Kippo menggunakan subversion :

```
$ svn checkout http://kippo.googlecode.com/svn/trunk/ ./kippo
```

Setelah mendownload Kippo, kita pindah ke direktori kippo :

```
$ cd kippo
```

Secara default, Kippo menggunakan port 2222 sebagai port standar, karena kita ingin menggunakan port 22 sebagai port Kippo kita dapat mengganti port standar tersebut pada file konfigurasi, sebelumnya ketikkan perintah di bawah ini agar file dapat dibaca oleh Kippo :

```
$ mv kippo.cfg.dist kippo.cfg
```

Lalu edit file **kippo.cfg** menggunakan nano :

```
$ nano kippo.cfg
```

Cari kode ini :

```
# Port to listen for incoming SSH connections.  
#  
# (default: 2222)  
ssh_port = 2222
```

Lalu ganti **ssh_port=2222** menjadi **ssh_port=22** sehingga kodenya seperti ini :

```
# Port to listen for incoming SSH connections.  
#  
# (default: 2222)  
ssh_port = 22
```

Tekan **Ctrl + o** untuk save dan **Ctrl + x** untuk keluar.

- Sekarang kita akan menggunakan AuthBind pada Kippo Honeypot yang telah dikonfigurasi sebelumnya, edit file **start.sh** :

```
$ nano start.sh
```

Cari kode ini :

```
twistd -y kippo.tac -l log/kippo.log --pidfile kippo.pid
```

Lalu ganti dengan kode di bawah ini :

```
authbind --deep twistd -y kippo.tac -l log/kippo.log --pidfile  
kippo.pid
```

save dan exit.

Lalu jalankan Kippo Honeypot dengan perintah :

```
$ ./start.sh
```

```
Starting kippo in background...Generating RSA keypair...  
done.
```

Pada langkah ini, Kippo Honeypot telah berjalan pada server, kita dapat melakukan pengecekan dengan perintah :

```
$ netstat -tln|grep 22
```

2.2 Simulasi Attacker Menyerang Server Yang Terinstall Kippo Honeypot

Setelah menjalankan kippo honeypot, kippo akan mencatat semua kegiatan yang terjadi pada port 22, cara melihat log tersebut adalah :

```
$ cat /home/kippo/kippo/log/kippo.log
```

Kita akan melakukan percobaan. Komputer attacker dengan IP **192.168.1.3** dengan sistem operasi Ubuntu akan menyerang layanan SSH pada server yang terinstall kippo honeypot. Ada dua istilah yang akan digunakan yaitu “Sisi Attacker” dan “Sisi Server”.

Sisi Attacker (Percobaan 1)

Attacker mencoba untuk melakukan scanning pada port 22 apakah terbuka

```
mrrwx@mrrwx:/opt$ nmap -p22 192.168.1.4

Starting Nmap 6.46 ( http://nmap.org )
Nmap scan report for 192.168.1.4
Host is up (0.00048s latency).
PORT      STATE SERVICE
22/tcp    open  ssh
```

Gambar 8: Attacker mencoba melakukan scanning pada port 22

Setelah melakukan scanning, attacker mencoba melakukan login pada ssh dengan perintah :

```
$ ssh root@192.168.1.4
```

Dan hasilnya, attacker tidak bisa masuk ke dalam server dengan beberapa password percobaan.

```
mrrwx@mrrwx:~$ ssh root@192.168.1.4
The authenticity of host '192.168.1.4 (192.168.1.4)' can't be established.
RSA key fingerprint is 59:98:40:c9:a4:8f:0f:68:48:bc:c1:b9:79:6d:bd:d7.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.1.4' (RSA) to the list of known hosts.
Password:
Password:
Password:
root@192.168.1.4's password:
Permission denied, please try again.
root@192.168.1.4's password:
Permission denied, please try again.
root@192.168.1.4's password:
Permission denied (password,keyboard-interactive).
```

Gambar 9: Attacker mencoba melakukan brute force

Sebenarnya perbuatan si attacker ini adalah sia-sia karena yang dilakukannya adalah mencoba melakukan brute force pada port palsu.

Sisi Server (Monitoring 1)

Selanjutnya kita akan membahas hal yang terjadi di sisi server, karena baru saja terjadi brute force pada port 22 yang sebenarnya adalah sistem palsu maka Kippo mencatat kegiatan attacker tersebut.

Dari hasil log, kippo mencatat IP address si attacker yaitu 192.168.1.2 dan juga kippo mencatat password yang digunakan si attacker untuk mencoba masuk ke server.

```
2014-08-27 13:55:58+0700 [-] root failed auth password
2014-08-27 13:55:58+0700 [-] unauthorized login:
2014-08-27 13:56:03+0700 [SSHSservice ssh-userauth on HoneyPotTransport,3,192.168
.1.2] root trying auth password
2014-08-27 13:56:03+0700 [SSHSservice ssh-userauth on HoneyPotTransport,3,192.168
.1.2] login attempt [root/indonesia] failed
2014-08-27 13:56:04+0700 [-] root failed auth password
2014-08-27 13:56:04+0700 [-] unauthorized login:
2014-08-27 13:56:04+0700 [HoneyPotTransport,3,192.168.1.2] connection lost
```

Gambar 10: Log attacker mencoba login

Pada gambar 10, kita bisa melihat si attacker mencoba menggunakan password **indonesia** dengan username **root** dan password lainnya untuk mencoba masuk ke server.

Secara default, Kippo menggunakan password 123456 untuk masuk melalui SSH palsu. Selanjutnya kita akan melakukan percobaan pada server di mana seorang attacker berhasil masuk ke server yang terinstall honeypot, hacker tersebut masuk ke dalam sistem palsu yang dibuat oleh kippo dengan username **root** dan password **123456**.

Sisi Attacker (Percobaan 2)

Di sini seorang attacker sudah masuk ke server melalui layanan SSH pada port 22 yang sebenarnya adalah sistem palsu.

```
mrrwx@mrrwx:~$ ssh root@192.168.1.4
Password:
nas3:~#
```

Gambar 11: Attacker masuk ke dalam sistem palsu

Disini seolah-olah attacker berada pada sistem yang sebenarnya, dia dapat menjalankan perintah linux seperti **wget**, **cat**, **ls** dan lain-lain bahkan attacker dapat melihat versi kernel dari server menggunakan perintah **uname -a**.

```
nas3:~# ls
nas3:~# touch tes.txt
nas3:~# echo "Hallo" > tes.txt
Hallo > /root/tes.txt
nas3:~# uname -a
Linux nas3 2.6.26-2-686 #1 SMP Wed Nov
```

Gambar 12: Attacker menjalankan perintah linux

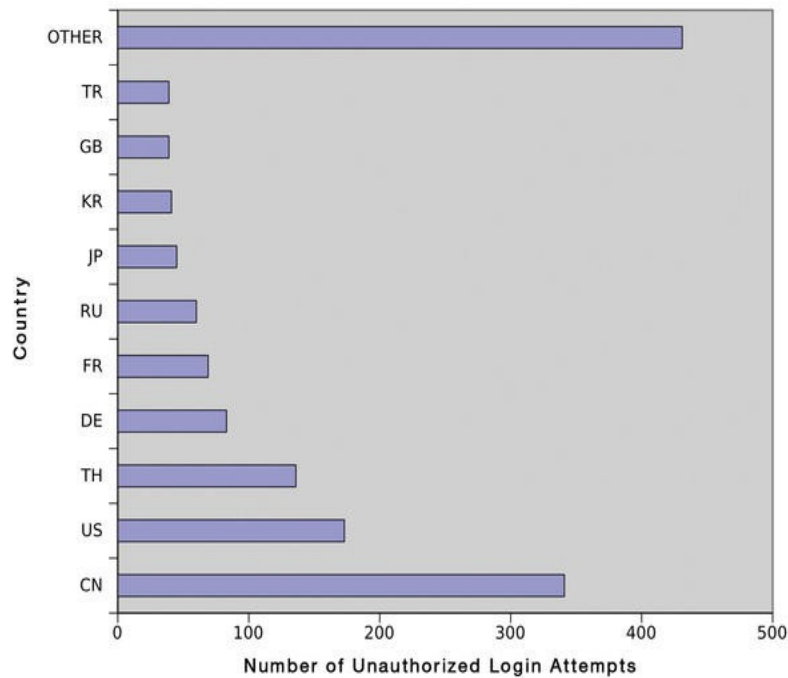
Versi kernel yang ditampilkan pun bukanlah yang asli, semuanya palsu. Kippo akan berusaha membuat seorang attacker seolah-olah sudah masuk ke dalam server. Namun ada satu hal lagi yang biasanya menjadi incaran attacker, yaitu melakukan rooting. Pada contoh diatas user sudah menggunakan akses root, namun sebenarnya bisa digunakan username apapun. Misalnya tes@192.168.1.4 atau indo@192.168.1.4, semua bisa, tapi password yang diterima hanya username root yaitu 123456, ini bisa diganti pada konfigurasi kippo. Oke kembali ke pembahasan rooting, biasanya sebelum melakukan rooting, seorang attacker akan mendownload localroot exploit. Kippo menyediakan fitur **wget** pada sistem palsu yang dapat digunakan untuk mendownload file. Namun cara inipun tetap tidak bisa dilakukan oleh attacker. Segala sesuatu yang dilakukan attacker disinipun telah dicatat pada log kippo.

Itulah fitur-fitur yang disediakan oleh Kippo honeypot, intinya kippo akan membuat sistem palsu seolah-olah attacker sudah masuk ke dalam server.

3. Menggunakan Portspooft

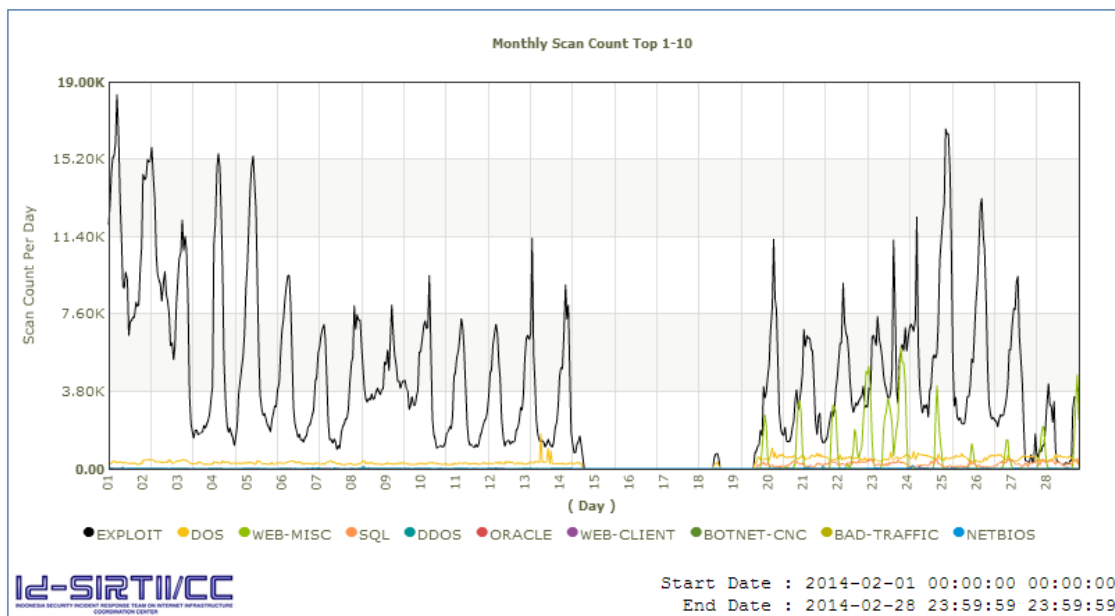
Kippo Honeypot hanya digunakan untuk protokol SSH. Namun bagaimana dengan port yang lainnya?. Kita dapat menggunakan **portspooft** yang berfungsi untuk menyamarkan port asli di suatu server. Tujuan umum dari portspooft adalah untuk memperlambat dan mengelabui proses scanning dengan tool Nmap. Seperti yang kita ketahui, Nmap sangat terkenal dengan port scanning dimana Nmap dapat melakukan scanning hingga ribuan port di jaringan komputer. Portspooft dirilis pada tahun 2012 oleh Piotr Duszynski yang diimplementasikan dalam pemrograman C++.

Attacker berpengalaman dan bahkan beberapa pengacau server lain biasanya sering melakukan port scanning dan membajak server tersebut dalam waktu yang sama, menurut survey Falko Benthin di situs <http://linux-magazine.com>, dalam beberapa negara para attacker lebih tertarik pada percobaan login di suatu sistem, attacker akan melakukan berbagai cara agar dapat login di sistem.



Gambar 13: Jumlah usaha login tanpa izin pada jaringan Falko Benthin dalam waktu dua bulan (berdasarkan negara atau asal).

Berdasarkan pemantauan trafik internet ID-SIRTII, di Indonesia dalam satu bulan tepatnya pada bulan Februari tahun 2014, telah terjadi banyak eksploitasi pada suatu sistem dan inipun meningkat setiap tahunnya, eksploitasi bisa dilakukan dalam berbagai cara sehingga hal ini harus menjadi perhatian bagi para administrator sistem untuk terus meningkatkan keamanan server mereka.



Gambar 14: Pemantauan internet ID-SIRTII februari 2014

Portspooof merupakan program kecil yang sangat berguna, bisa dibilang “Kecil-kecil cabe rawit”. Karena portspooof memiliki kemampuan untuk melindungi port-port yang terdapat pada server dengan cara meniru port asli dan kemudian membuat port palsu dari port yang asli tersebut dimana jika kita melakukan scanning dengan Nmap pada suatu server, maka akan banyak port-port yang terbuka yang sebenarnya itu hanyalah port palsu alias port tersebut sudah disamarkan oleh portspooof. Sekali lagi, dengan menggunakan portspooof kita tidak dapat menghentikan seorang attacker tapi kita dapat memperlambat kinerja dari attacker atau orang-orang yang ingin mengganggu server kita. Portspooof akan membuat ribuan port seolah-olah terbuka yang akan membuat attacker bingung sehingga ada dua kemungkinan, pertama dia terus berusaha melakukan scanning atau kedua, dia menyerah. Dan yang terpenting adalah Portspooof juga akan mencatat (log) setiap kegiatan scanning yang dilakukan oleh attacker.

Secara default portspooof berjalan pada port 4444, nantinya semua port akan dialihkan ke port ini menggunakan iptables.

3.1 Persiapan dan instalasi

Disini saya memiliki server dengan sistem operasi Ubuntu 14.04. Langkah pertama kita download dulu portspooof dengan perintah ini :

```
$ apt-get install git
```



```
$ git clone https://github.com/drklwi/portspooof.git
```

Lalu pindah ke direktori portspooof :

```
$ cd portspooof
```

Lalu install portspooof dengan cara compile :

```
$ ./configure
```

```
$ make
```

```
$ make install
```

Jika sudah terinstall cek dengan perintah “portspooof -h” :

```
root@ubuntu:~/portspooof# portspooof -h
```

```
Usage: portspooof [OPTION]...
```

```
Portspooof - service emulator / frontend exploitation framework.
```

```
-i          ip : Bind to a particular  IP address
-p          port : Bind to a particular PORT number
-s          file_path : Portspooof service signature regex. file
-c          file_path : Portspooof configuration file
-l          file_path : Log port scanning alerts to a file
-f          file_path : FUZZER_MODE - fuzzing payload file list
-n          file_path : FUZZER_MODE - wrapping signatures file list
-1          FUZZER_MODE - generate fuzzing payloads internally
-2          switch to simple reply mode (doesn't work for Nmap)!
-D          run as daemon process
-d          disable syslog
-v          be verbose
-h          display this help and exit
```

Selanjutnya kita atur semua port agar dialihkan ke port 4444 dengan menggunakan iptables. Namun ada beberapa pengecualian pada port tertentu. Misalnya saat ini kita menggunakan port 22,25 dan 80. Maka ketiga port tersebut tidak dialihkan ke port 4444 karena port tersebut terinstall aplikasi asli. Sedangkan port lain yang dialihkan ke port 4444 hanya untuk port yang tidak ada aplikasi apapun, karena semua port yang dialihkan akan dibuat seolah-olah memiliki aplikasi oleh portspooof. Sebagai

contoh server saya menjalankan port 22,25,dan port 80 maka perintah untuk iptables nya adalah :

```
$ iptables -t nat -A PREROUTING -i eth0 -p tcp -m tcp -m multiport --dports 1:21,23:24,26:79,81:65535 -j REDIRECT --to-ports 4444
```

Selanjutnya kita dapat langsung menjalankan portspooft dengan perintah :

```
$portspooft -D
```

Perintah diatas akan menjalankan portspooft pada background proses.

Sekarang kita akan mencoba melakukan scanning pada server tersebut dengan Nmap dan hasilnya banyak port yang terbuka bahkan hingga ribuan port :

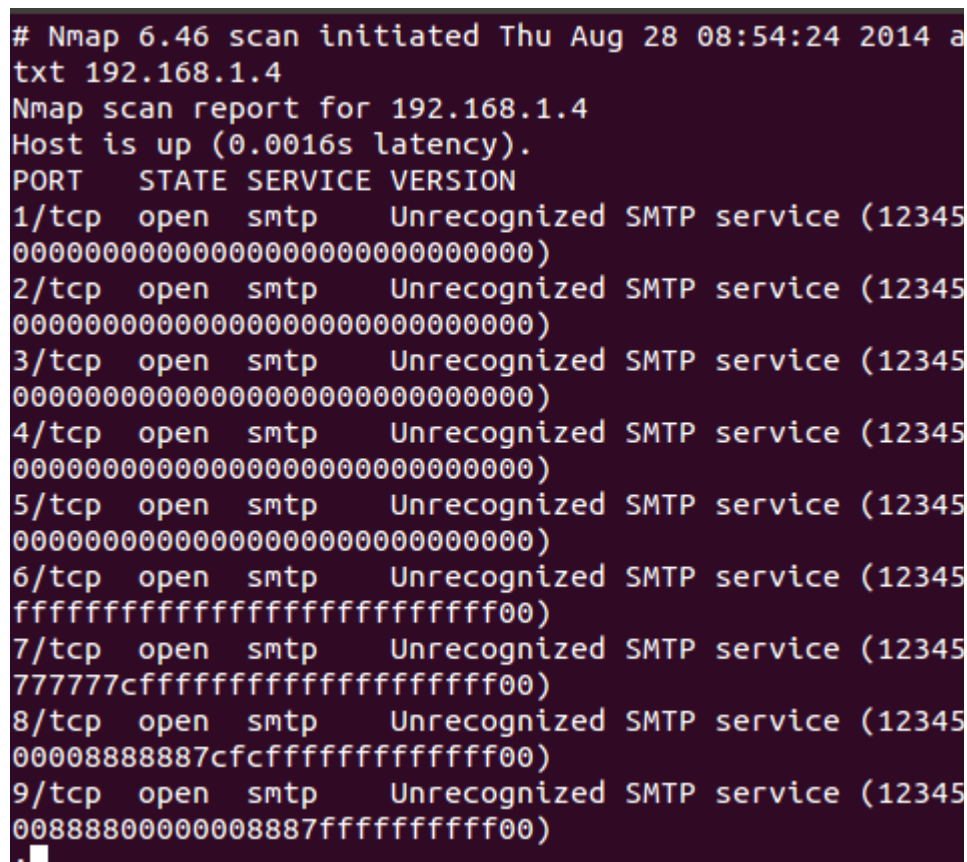
```
# Nmap 6.46 scan initiated Thu Aug 28 08:12:01 2014 as: i
Nmap scan report for 192.168.1.4
Host is up (0.0034s latency).
PORT      STATE SERVICE      VERSION
21/tcp    open  tcpwrapped
23/tcp    open  tcpwrapped
24/tcp    open  tcpwrapped
25/tcp    open  tcpwrapped
26/tcp    open  tcpwrapped
27/tcp    open  tcpwrapped
29/tcp    open  tcpwrapped
33/tcp    open  tcpwrapped
37/tcp    open  tcpwrapped
43/tcp    open  tcpwrapped
49/tcp    open  tcpwrapped
53/tcp    open  tcpwrapped
70/tcp    open  tcpwrapped
79/tcp    open  tcpwrapped
80/tcp    open  http      Apache httpd 2.4.7 ((Ubuntu))
22/tcp    open  ssh        (protocol 2.0)
23/tcp    open  tcpwrapped
24/tcp    open  tcpwrapped
25/tcp    open  smtp      Postfix smtpd
26/tcp    open  tcpwrapped
30/tcp    open  tcpwrapped
32/tcp    open  tcpwrapped
33/tcp    open  dsp?
37/tcp    open  time?
42/tcp    open  tcpwrapped
43/tcp    open  tcpwrapped
49/tcp    open  tcpwrapped
53/tcp    open  tcpwrapped
70/tcp    open  tcpwrapped
79/tcp    open  tcpwrapped
80/tcp    open  http      Apache httpd 2.4.7 ((Ubuntu))
```

Gambar 15: Hasil scanning Nmap

Cara diatas belum efektif, karena kita bisa masih bisa melihat pada Gambar 15 bahwa port 22,25,80 masih terlihat jelas bahwa itu adalah port asli pada kolom **VERSION**. Portspooft menyediakan fitur untuk menambahkan nama aplikasi di setiap port agar menyusahkan si attacker, perintahnya adalah :

```
$ portspooft -c /usr/local/etc/portspooft.conf -s  
/usr/local/etc/portspooft_signatures -D
```

Hasil scanning Nmap akan terlihat lebih rumit :



```
# Nmap 6.46 scan initiated Thu Aug 28 08:54:24 2014 a  
txt 192.168.1.4  
Nmap scan report for 192.168.1.4  
Host is up (0.0016s latency).  
PORT      STATE SERVICE VERSION  
1/tcp    open  smtp      Unrecognized SMTP service (12345  
00000000000000000000000000000000)  
2/tcp    open  smtp      Unrecognized SMTP service (12345  
00000000000000000000000000000000)  
3/tcp    open  smtp      Unrecognized SMTP service (12345  
00000000000000000000000000000000)  
4/tcp    open  smtp      Unrecognized SMTP service (12345  
00000000000000000000000000000000)  
5/tcp    open  smtp      Unrecognized SMTP service (12345  
00000000000000000000000000000000)  
6/tcp    open  smtp      Unrecognized SMTP service (12345  
ffffffffffffffffffffffffffff00)  
7/tcp    open  smtp      Unrecognized SMTP service (12345  
777777cffffffffffffffffffff00)  
8/tcp    open  smtp      Unrecognized SMTP service (12345  
00008888887cfcffffffffffff00)  
9/tcp    open  smtp      Unrecognized SMTP service (12345  
0088880000008887ffffffffffff00)  
.
```

Gambar 16: Hasil scanning kedua dengan Nmap

Kita juga dapat menjalankan portspooft dalam mode verbose :

```
$ portspooft -v
```

Maka dari sisi server akan tampil log hasil scanning dari attacker :

```
Thread nr.8 for port 8193
signature sent ->
---
new conn - thread choosen: 8 - nr. of
new conn - thread choosen: 7 - nr. of
new conn - thread choosen: 4 - nr. of
new conn - thread choosen: 3 - nr. of
new conn - thread choosen: 2 - nr. of
new conn - thread choosen: 1 - nr. of
---
Thread nr.6 for port 443
signature sent ->
```

Gambar 17: Log Portspooft

```
mrrwx@mrrwx:~$ nmap -sV 192.168.1.5

Starting Nmap 6.47SVN ( http://nmap.org ) at 2014-08-30 07:48 WIB
Nmap scan report for 192.168.1.5
Host is up (0.0017s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      (protocol 2.0)
25/tcp    open  smtp     Postfix smtpd
80/tcp    open  http     Apache httpd 2.4.7 ((Ubuntu))
1 service unrecognized despite returning data. If you know the service/version,
please submit the following fingerprint at http://www.insecure.org/cgi-bin/servi
cefp-submit.cgi :
SF-Port22-TCP:V=6.47SVN%I=7%D=8/30%Time=54011F51%P=i686-pc-linux-gnu%r(NUL
SF:L,29,"SSH-2\0-OpenSSH_6\0.6\0.1p1\0x20Ubuntu-2ubuntu2\r\n");
Service Info: Host: ubuntu
```

Gambar 18: Attacker mencoba melakukan port scanning pada server

4. Mendeteksi port scanning dengan Portsentry

Portsentry merupakan tool yang berfungsi untuk melindungi server dari port scanning dan mendeteksi aktivitas yang mencurigakan pada server. Sama seperti portspooft, portsentry akan membuat port-port palsu pada server, bukan hanya itu, portsentry juga secara otomatis akan mem-block IP address yang

melakukan port scanning pada server sehingga si attacker tidak akan bisa lagi mengakses server tersebut kecuali mendapat izin dari administrator.

Untuk menginstall portspooft pada linux debian gunakan perintah :

```
$ sudo apt-get install portsentry
```

Untuk RedHat :

```
$ wget
```

```
ftp://ftp.pbone.net/mirror/ftp.falsehope.net/home/tengel/centos/4/te/i386/RPMS/portsentry-1.2-1.te.i386.rpm
```

```
$ sudo rpm -Uvh portsentry-1.2-1.te.i386.rpm
```

Konfigurasi portsentry pada **/etc/portsentry/portsentry.conf**, cari kode ini :

```
BLOCK_UDP="0"
```

```
BLOCK_TCP="0"
```

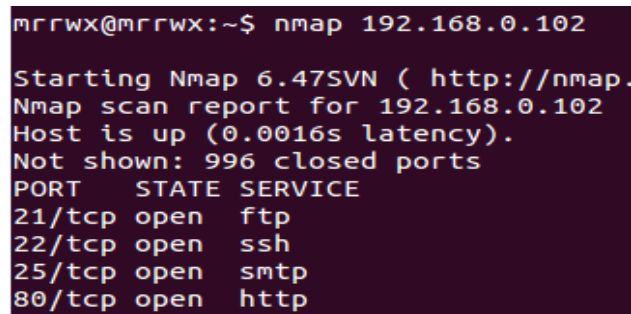
ganti menjadi :

```
BLOCK_UDP="1"
```

```
BLOCK_TCP="1"
```

cara diatas dilakukan agar server mem-block scanning TCP dan UDP melalui portsentry dan secara otomatis IP address attacker akan di block juga.

Kondisi server sebelum diinstall portsentry :



```
mrrwx@mrrwx:~$ nmap 192.168.0.102

Starting Nmap 6.47SVN ( http://nmap.org )
Nmap scan report for 192.168.0.102
Host is up (0.0016s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
25/tcp    open  smtp
80/tcp    open  http
```

Gambar 19: Kondisi server normal

Server masih terlihat normal ketika dilakukan scanning, lalu kita lihat kondisi server setelah diinstall portsentry :

```
mrrwx@mrrwx:~$ nmap 192.168.0.102

Starting Nmap 6.47SVN ( http://nmap.org )
Nmap scan report for 192.168.0.102
Host is up (0.0014s latency).
Not shown: 981 closed ports
PORT      STATE SERVICE
1/tcp     open  tcpmux
21/tcp    open  ftp
22/tcp    open  ssh
25/tcp    open  smtp
79/tcp    open  finger
80/tcp    open  http
111/tcp   open  rpcbind
119/tcp   open  nntp
143/tcp   open  imap
1080/tcp  open  socks
1524/tcp  open  ingreslock
2000/tcp  open  cisco-sccp
```

Gambar 20: Kondisi server setelah menggunakan portsentry

Terlihat banyak port palsu yang dibuat oleh portsentry, sampai langkah ini, server akan mem-block IP Address attacker, sehingga ketika attacker melakukan scanning lagi, maka hasilnya akan nihil :

```
mrrwx@mrrwx:/opt/pnmap$ nmap 192.168.0.102

Starting Nmap 6.47SVN ( http://nmap.org ) at 2014-09-09 07:15 WIB
Note: Host seems down. If it is really up, but blocking our ping
n
Nmap done: 1 IP address (0 hosts up) scanned in 3.04 seconds
```

Gambar 21: IP address attacker di block server

Untuk menghapus IP yang di-block, gunakan perintah :

```
$ route del -host [IP Address]
```

Log portsentry dapat dilihat pada file portsentry.history atau anda dapat melihat di /var/log/syslog

III Kesimpulan

Banyak cara yang dapat dilakukan untuk melakukan serangan pada server cloud karena kemudahan aksesnya terutama melalui port yang terbuka atau port yang terbuka dengan versi layanan yang memiliki celah kemanan. Karena banyaknya cara melakukan penyerangan, maka akan muncul berbagai macam cara juga untuk melindungi server cloud ini dari serangan orang yang tidak bertanggung jawab. Port merupakan suatu hal yang rentan dan harus diwaspadai oleh seorang administrator server, solusi terbaik untuk melindungi port atau layanan port dari serangan hacker adalah dengan cara selalu mencari informasi terbaru seputar bug atau celah keamanan yang terdapat pada server, terkadang ada aplikasi terbaru yang muncul untuk jaringan komputer namun memiliki bug, sehingga kita memiliki kesimpulan bahwa tidak semua aplikasi baru atau update terbaru dari suatu aplikasi aman dari serangan, sehingga sekali lagi penting untuk seorang administrator harus memiliki pengetahuan yang luas tentang kemanan jaringan komputer. Kesalahan terbesar yang biasanya dilakukan administrator adalah jarang mengupdate informasi tentang celah kemanan terbaru, aplikasi terbaru dan lain sebagainya sehingga servernya sering dimanfaatkan oleh orang-orang yang tidak bertanggung jawab tanpa sepengetahuan administrator.

Terkait dengan teknik-teknik pengamanan yang dijelaskan pada paper ini dapat diterapkan pada server cloud anda.

Referensi

- [1] “*Code Injection*,” http://en.wikipedia.org/wiki/Code_injection
- [2] “*Secure Shell*,” http://en.wikipedia.org/wiki/Secure_Shell
- [3] “*Authbind*,”” <http://en.wikipedia.org/wiki/Authbind>
- [4] “*How To Install Kippo, an SSH Honeypot, on an Ubuntu Cloud Server*,”
<https://www.digitalocean.com/community/tutorials/how-to-install-kippo-an-ssh-honeypot-on-an-ubuntu-cloud-server>
- [5] Falko Benthin, “*Trick Attackers with Portspooof*,” <http://www.linux-magazine.com/Online/Features/Trick-Attackers-with-Portspooof>
- [6] C_Coffie, “*How To Tetup Portspooof*,” <http://calebcoffie.com/how-to-setup-portspooof/>
- [7] Michael Rash, “*psad: Intrusion Detection and Log Analysis with iptables*,”
<http://cipherdyne.org/psad/>
- [8] Dony Ramansyah, “*Mengamankan Server Linux Dengan Portsentry*”, <http://dony-ramansyah.blogspot.com/2011/10/mengamankan-server-linux-dengan.html>