JIRE (Jurnal Informatika & Rekayasa Elektronika) Volume 2, No 1, April 2019
http://e-journal.stmiklombok.ac.id/index.php/jire

ANALISIS PERBANDINGAN SISTEM AUTENTIKASI PORT KNOCKING DAN SINGLE PACKET AUTHORIZATION PADA SERVER RASPBIAN

Hasbi Muhammad[1], I Wayan Agus Arimbawa[2], Andi Hidayat Jatmika[3]
[123]Program Studi Teknik Informatika, Fakultas Teknik, Universitas Mataram
Jl. Majapahit 62, Mataram, Lombok NTB, Indonesia

[1]hasbimuhammadf1d013034@gmail.com, [2]arimbawa@unram.ac.id, [3]andyj@unram.ac.id

Abstract

Network security is the most important aspect of the system in maintaining the health and integrity of the data and ensures the availability of services to be used by user. The computer network security system must be protected against all types of attacks and infiltration or survey attempts by unqualified parties. To solve this problem is the first step taken is to take advantage of using the system of firewall. But Firewall is unable to protect the attack misuse error in application level software. To deal with network security and data privacy problems that often occur, a network security authentication system technique is developed called port knocking and single packet authorization, this authentication system will be implemented on raspbian server.

Keywords: Firewall, Port knocking, Single Packet Authorization, Raspbian

Abstrak

Keamanan jaringan adalah aspek terpenting dari sistem dalam menjaga kesehatan dan integritas data dan memastikan ketersediaan layanan yang akan digunakan oleh pengguna. Sistem keamanan jaringan komputer harus dilindungi terhadap semua jenis serangan dan infiltrasi atau upaya survei oleh pihak yang tidak berkualifikasi. Untuk mengatasi masalah ini, langkah pertama yang diambil adalah memanfaatkan sistem firewall. Tetapi Firewall tidak dapat melindungi kesalahan penyalahgunaan serangan dalam perangkat lunak tingkat aplikasi. Untuk menangani keamanan jaringan dan masalah privasi data yang sering terjadi, teknik sistem otentikasi keamanan jaringan dikembangkan yang disebut port knocking dan otorisasi paket tunggal, sistem otentikasi ini akan diterapkan pada server raspbian.

Kata kunci : Firewall, Port knocking, Single Packet Authorization, Raspbian

1. PENDAHULUAN

Keamanan jaringan merupakan aspek terpenting sebuah sistem dalam menjaga validitas dan integritas data serta menjamin ketersediaan layanan bagi penggunaannya. Sistem keamanan jaringan komputer harus dilindungi dari segala macam serangan dan usaha-usaha penyusupan atau pemindaian oleh pihak yang tidak berhak.Untuk menangani masalah ini langkah pertama yang diambil adalah dengan memanfaatkan firewall. Menggunakan firewall merupakan solusi yang pertama untuk diambil, dikarenakan firewall melindungi sistem berdasarkan alamat IP dan beberapa karakteristik lainnya. Untuk itu penggunaan firewall hanya dianggap keamanan lapisan tingkat pertama dalam upaya pencegahan terhadap sistem

Firewall tidak mampu melindungi serangan yang dilakukan untuk mengeksploitasi bug di tingkat aplikasi perangkat lunak. Pada umumnya serangan terhadap suatu komputer

dalam suatu jaringan ialah melalui aplikasi yang masuk pada port-port terbuka dalam server. Port-port yang terbuka ini rawan terhadap eksploitasi dari akses yang tidak diinginkan, untuk itu dibutuhkan suatu system yang dapat menangkal masalah tersebut.

Untuk menangani masalah kemanan jaringan dan privasi data yang sering terjadi dikembangkan suatu teknik autentikasi keamanan jaringan yang dinamakan port knocking. Sistem autentikasi ini menggunakan kombinasi lapisan-lapisan kunci untuk dapat mengamankan sebuah port komunikasi. Teknik ini mempertahankan satu atau lebih port yang dikonfigurasikan sebelumnya yang telah ditutup dan hanya akan terbuka menggunakan sequence of requests untuk nomor port yang telah ditentukan.

Selain menggunakan sistem autentikasi port knocking untuk melindungi port server, terdapat metode lain yang hampir serupa akan tetapi beda metode dalam mengamankan port server, sistem autentikasi ini disebut dengan single packet authorization (SPA). SPA merupakan sistem otentikasi komunikasi dan infromasi yang dilakukan melaui port firewall yang tertutup. Metode ini menggunakan payload dalam satu paket dengan menggunakan payload yang terdiri dari cryptography, perlindungan terhadap replay attack dan minimal network footprint

Dari berbagai uraian permasalahan keamanan serta cara penanganannya dengan menggunakan kedua metode sistem autentikasi tersebut. Maka pada penelitian ini akan dilakukan pengujian perbandingan kehandalan dari sistem autentikasi port knocking dan SPA. Sistem autentikasi port knocking dan SPA akan diimplementasikan dan diujikan pada ruang lingkup server sistem operasi raspbian.

2. TINJAUAN PUSTAKA DAN TEORI

A. Tinjauan Pustaka

Muhammad Saleh Hafiz Fajri, dkk dalam [5] melakukan penelitian terhadap sistem autentikasi port knocking yang di implementasikan pada sistem operasi Linux Ubuntu server 12.04 LTS. Pada penilitian tersebut port knocking yang diterapkan pada layanan ssh server, ftp server dan my sql server dapat meningkatkan keamanan akses port server walaupun waktu untuk mengakses akan menjadi lebih lama dibandingkan dengan server tanpa port knocking.

Awan, dalam [6] melakukan penelitian tentang pemberian hak akses terhadap port tertentu yang ditutup oleh firewall yang menerapkan metode port knocking, didapatkan bahwa metode port knocking dapat menjadi sebuah security layer tambahan pada suatu firewall. Selain itu dengan metode port knocking dapat memungkinkan administrator melakukan koneksi kepada server meskipun firewall menutup semua port yang ada. Pemeriksaan IP yang telah ditentukan sebelumnya dan dibandingkan kembali dengan IP hasil dari proses port knocking, apabila sama dapat diberikan hak akses lebih lanjut.

Melihat kelebihan port knocking dibandingkan dengan firewall maka dalam penelitian ini akan di implementasikan sistem autentikasi port knocking untuk meningkatkan keamanan port server pada sistem operasi raspbian. Sistem autentikasi port knocking akan diujikan dengan berbagai macam tindak serangan untuk melihat bagaimana efektifitas keamanan port server dibandingkan hanya menggunakan firewall.

Haythem Zorkta dalam [7] penelitian ini mengembangkan metode yang sudah ada sebelumnya yaitu metode sistem autentikasi single packet authorization. Hasil yang diperoleh dari penelitian ini yaitu mampu menutupi kekurangan dari SPA dari segi kurangnya Asosiasi antara otentikasi dan koneksi TCP yang sedang berlangsung. Selain itu HSPA lebih kuat terhadap serangan yang dilakukan oleh hacker jika dibandingkan dengan teknik yang diterapkan pada SPA , dan memiliki marjinal pengolahan, komunikasi, dan buffering overhead.

Michael Rash dalam [9] pada penelitian ini dijelaskan bagaimana cara mengkombinasikan port knocking dan passive OS fingerprinting dengan menggunakan daemon dari fwknop. Hasil yang didapat dari penelitian ini yaitu tingkat keamanan menjadi lebih kuat dan fleksibel. Kemampuan transmisi data, ditambah dengan tehnik yang lebih baik untuk mencegah replay attck, membuat metode ini menjadi sangat ideal untuk memperluas konfigurasi paket filter untuk meneruskan semua koneksi ke beberapa layanan penting secara default. Hal ini membuat eksploitasi kerentanan dalam layanan tersebut

jauh lebih sulit, karena alamat IP yang tidak terdaftar tidak dapat berinteraksi dengan layanan ini sampai pesan SPA yang valid dikirimkan.

Dari pemaparan tentang tingkat keamanan server yang menjadi lebih kuat jika mengimplementasikan SPA, maka dalampenelitian ini juga akan menerapkan sistem autentikasi single packet authorization pada sistem raspbian. Sehingga terdapat dua buah sistem autentikasi yang akan diterapkan yaitu port knocking dan single packet authorization. Sistem autentikasi PK dan SPA akan dibandingkan dan di ujikan dengan perlakuan yang sama untuk mengetahui tingkat kemanan dari sistem mana yang lebih baik.

B. Keamanan Jaringan

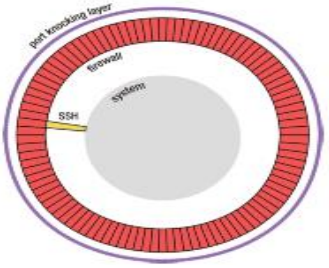
Keamanan jaringan yaitu proses pencegahan yang dilakukan oleh penyerang untuk terhubung ke dalam jaringan komputer melalui akses yang tidak sah, atau penggunaan secara ilegal dari komputer dan jaringan. Keamanan jaringan adalah suatu cara atau suatu sistem yang digunakan untuk memberikan proteksi atau perlindungan pada suatu jaringan agar terhindar dari berbagai ancaman luar yang mampu merusak jaringan.

Konsep keamanan jaringan dapat meliputi beberapa faktor antara lain: Confidentiality atau kerahasiaan adalah pencegahan bagi mereka yang tidak berkepentingan dapat mencapai informasi. Secara umum dapat disebutkan bahwa kerahasiaan mengandung makna bahwa informasi yang tepat terakses oleh mereka yang berhak (dan bukan orang lain), sama analoginya dengan e-mail maupun data-data perdagangan dari perusahaan.

1. Integrity atau Integritas adalah pencegahan terhadap kemungkinan amandemen atau penghapusan informasi oleh mereka yang tidak berhak.
2. Availability atau ketersediaan adalah upaya pencegahan ditahannya informasi atau sumber daya terkait oleh mereka yang tidak berhak.
3. Non-repudiation adalah aspek ini menjaga agar seseorang tidak dapat menyangkal telah melakukan sebuah transaksi.
4. Autentikasi adalah suatu langkah untuk menentukan atau mengonfirmasi bahwa seseorang (atau sesuatu) adalah autentik atau asli.

C. Port knocking

Port knocking adalah metode yang dilakukan untuk membuka akses ke port tertentu yang telah diblock oleh Firewall pada perangkat jaringan dengan cara mengirimkan paket atau koneksi tertentu. Koneksi bisa berupa protocol TCP, UDP maupuan ICMP. Menurut Luberti, Port knocking adalah sebuah metode sederhana untuk memberikan akses remote tanpa meninggalkan port dalam keadaan selalu terbuka [10].



Gambar 2.1 Port knocking

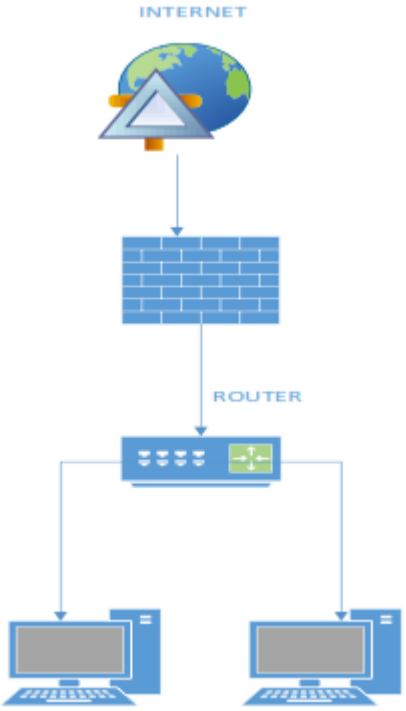
Single packet authorization merupakan metode autentikasi yang hampir serupa dengan port knocking akan tetapi yang membedakannya yaitu penggunaan packet header. SPA mengkomunikasikan informasi menggunakan payload dalam satu paket, paket payload yang digunakan pada SPA menawarkan banyak peningkatan dibandingkan port knocking dalam tingkat keamanan seperti penggunaan cryptography, perlindungan terhadap replay attack dan minimal network footprint.

SPA melakukan pengamanan komunikasi otentikasi dan otorisasi informasi melalui port firewall yang tertutup. Hal ini dilakukan untuk membuka port tertentu yang diakses secara sementara. Sebagian atau bahkan keseluruhan port akan terlihat seperti tertutup hal ini bertujuan agar server tidak terlihat terbuka dari pihak luar guna melindungi setiap layanan yang sedang berjalan.

Gambar 2.2 SPA dalam tingkatan lapisan sistem

D. Firewall

Firewall dapat digambarkan sebagai suatu sistem yang digunakan untuk membatasi ataupun mengatur hak akses dari suatu segmen jaringan ke segmen jaringan yang lain, biasanya firewall digunkan untuk membatasi antara jaringan Wide Area Network (WAN) dan Local Area Network (LAN), firewall umumnya digunakan untuk menentukan kebijakan apa saja yang boleh di akses dari jaringan luar (WAN) kedalam (LAN) dan apa yang tidak boleh dan juga sebaliknya yang bertujuan untuk meningkatkan sistem keamanan.



Gambar 2.3 Sistem kerja firewall

Pada gambar 2.3 dapat dilihat bahwa Fungsi dari firewall adalah sebagai pengontrol, mengawasi arus paket data yang mengalir dijaringan. Fungsi firewal mengatur, mem-filter dan mengontrol lalu lintas data yang diizinkan untuk mengakses jaringan privat yang dilindungi, beberapa kriteria yang dilakukan firewall apakah memperbolehkan paket data lewati atau tidak, antara lain:

1. Alamat IP dari komputer sumber
2. Port TCP/UDP sumber dari sumber.
3. Alamat IP dari komputer tujuan.
4. Port TCP/UDP tujuan data pada komputer tujuan
5. Informasi dari header yang disimpan dalam paket data.

E. IPtables

Iptables adalah suatu tools dalam sistem operasi linux yang berfungsi sebagai alat untuk melakukan filter (penyaringan) terhadap (trafic) lalulintas data. Secara sederhana digambarkan sebagai pengatur lalulintas data. Sebuah policy pada iptables dibuat berdasarkan sekumpulan peraturan yang diberikan pada kernel untuk mengatur setiap paket yang datang. Pada iptables ada istilah yang disebut dengan ipchain yang merupakan daftar aturan bawaan dalam iptables. Fitur yang dimiliki Iptables:

Connection Tracking Capability yaitu kemampuan unutk inspeksi paket serta bekerja dengan icmp dan udp sebagaimana koneksi TCP. Menyederhanakan perilaku paket-pake dalam melakukan negosiasi built in chain (input, output, dan forward). Rate-Limited connection dan logging capability. Kita dapat membatasi usaha-usaha koneksi sebagai tindakan preventif serangan Syn flooding denial of services (DOS). Kemampuan untuk mem-filter flag-flag dan opsi tcp dan address-address MAC.

F. Raspbian

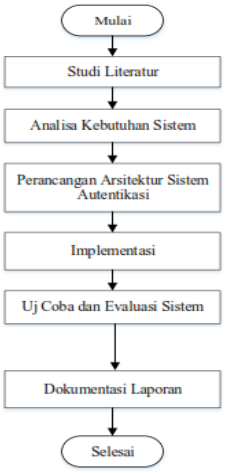
Raspbian adalah sistem operasi gratis yang berdasarkan pada Debian dan dioptimisasi untuk perangkat keras Raspberry Pi. Sistem operasi Raspbian adalah satu set program dasar dan program kegunaan (utility) yang membuat Raspberry Pi dapat bekerja. Namun, Raspbian menyediakan lebih dari sekedar sistem operasi murni: Raspbian datang dengan lebih dari 35.000 paket program, bundel perangkat lunak yang telah di pra-compile.

Build pertama dari Raspbian melebihi 35.000 paket Raspbian, dioptimisasi untuk performa terbaik pada Raspberry Pi. Sekarang Raspbian masih dalam pengembangan aktif dengan perhatian pada meningkatkan stabilitas dan performa dari sebanyak-banyaknya paket Debian. Sebagai catatan, Raspbian tidaklah berafiliasi dengan Raspberry Pi Foundation. Raspbian diciptakan oleh tim kecil yang

berdedikasi yang merupakan penggemar dari perangkat keras Raspberry Pi, yang merupakan tujuan dari pendidikan Raspberry Pi Foundation, dan tentunya juga dari Debian Project. Raspbian merupakan sistem operasi umum yang paling banyak orang gunakan pada Raspberry Pi.

3. METODE PENELITIAN

Secara garis besar penelitian ini dibagi menjadi beberapa tahapan di mulai dari studi literatur, pengumpulan data, analisa kebutuhan sistem, perancangan arsitektur sistem, implementasi, uji coba sistem, dokumentasi hingga pembuatan laporan. Diagram alir penelitian dapat dilihat pada Gambar 3.1



Gambar 3.1 Diagram alir penelitian

Analisa Kebutuhan Sistem Pada tahapan ini dilakukan analisa terhadap sistem yang akan dibangun, mulai dari kebutuhan-kebutuhan sistem yang akan menunjang dalam keberhasilan arsitektur sistem yang akan dibangun. Kebutuhan perangkat keras untuk kebutuhan sistem autentikasi port knocking dan SPA yang akan dibangun sebagai berikut:

1. Server, Processor Server : 4× ARM Cortex-A53, 1.2GHz, Memory Server : RAM 1 GB, SD Cardserver : 8 GB
2. Client, Processor Client : Intel® Core i5, Memory Client : RAM 2 GB, Harrdisk client : 500 GB

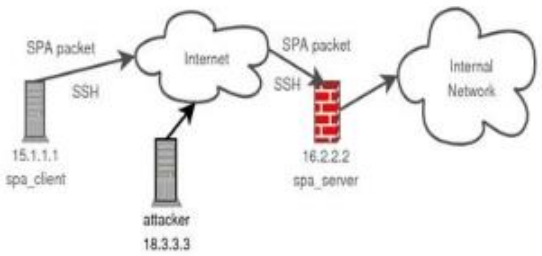
Kebutuhan perangkat lunak kebutuhan sistem autentikasi port knocking yang akan dibangun yaitu Nmap-7.7 merupakan tool yang digunakan untuk mengecek port yang terbuka dari sebuah server atau komputer. Ketika sebuah port jaringan terbuka maka pasti ada layanan dibelakangnya, bisa berupa web server, FTP dan layanan lainnya. Nmap sendiri adalah toolhacking yang sangat canggih dan komplek. Nmap tersedia baik di Linux maupun windows.

Knockd-0.7 adalah aplikasi pengetukan port pada server. Aplikasi ini mampu melacak semua lalu lintas pada sebuah jaringan antarmuka Ethernet, mencari urutan "ketukan" khusus pada lubang port. Komputer klien membuat lubang port ini dengan cara mengirimkan sebuah paket TCP (Transmission Control Protocol) ke port pada sebuah server. Port ini harus pada posisi yang terbuka, karena knockd memperhatikan batas lapisan pada suatu link, aplikasi ini memantau seluruh jalur bahkan untuk port yang tertutup sekalipun. Ketika komputer server mendeteksi urutan ketukan port tertentu, aplikasi ini menjalankan perintah yang ditentukan dalam file konfigurasinya. Aplikasi ini bisa digunakan untuk menjebol firewall dengan akses yang cepat.

Iptables adalah suatu tools dalam sistem operasi linux yang berfungsi sebagai alat untuk melakukan filter (penyaringan) terhadap (trafic) lalulintas data. Secara sederhana digambarkan sebagai pengatur lalulintas data.

fwknop-2.6.10 merupakan daemon tools yang digunakan untuk menjalankan sistem autentikasi SPA. Pada fwknop telah terdapat enkripsi yang digunakan untuk menyembunyikan username dan password yang digunakan untuk mengakses server dari sisi client. Ncrack-0.5 merupakan tools yang digunakan untuk melakukan penetrasi terhadap suatu jaringan dengan cara brute force.

Arsitektur Sistem Autentikasi Port knocking Melakukan perancangan dan memberikan gambaran mengenai sistem autentikasi port knocking yang akan dibangun. Perancangan Arsitektur Sistem Autentikasi *Single Packet Authorization* Melakukan perancangan dan memberikan gambaran tentang bagaimana cara kerja dan alur dari sistem autentikasi *single packet authorization.*



Gambar 3.2 Arsitektur sistem *single packet authorization*

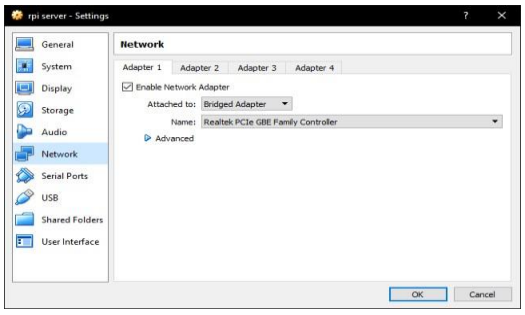
Pada gambar diatas dapat dilihat bagaimana cara kerja dari system autentikasi SPA. Terdapat *client* dan *attacker* yang hendak ingin melakukan pengaksesan terhadap jaringan *server* melalui koneksi internet, akan tetapi *request* yang dilakukan oleh *attacker* tidak dapat diteruskan karena pada lapisan *firewall* telah diimplementasikan sistem SPA. Sedangkan *request* dari *client* dapat dilakukan karena request
yang dilakukan telah dilapisi dengan paket SPA yang dimana dalam paket tersebut telah dilengkapi dengan enkripsi dari *usename* dan *password* untuk mengakses *server* yang hendak diakses.

4. HASIL DAN PEMBAHASAN

Pada tahap ini dilakukan instalasi system operasi yang dibutuhkan serta tambahan

Gambar 4.1 Virtual Machine

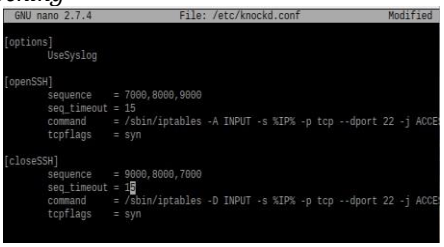
Sistem operasi yang digunakan pada penelitian ini baik dari sisi client maupun server adalah Raspbian versi desktop. Konfigurasi *Virtual Machine* Pada *Server* Pada sisi *server* menggunakan *network interface* dengan *mode Bridge Adaptor* agar *server* dapat terhubung dengan *client*, sehingga terbentuk koneksi *client* dan *server*.



Gambar 4.2 Konfigurasi Virtual Machine Pada Server

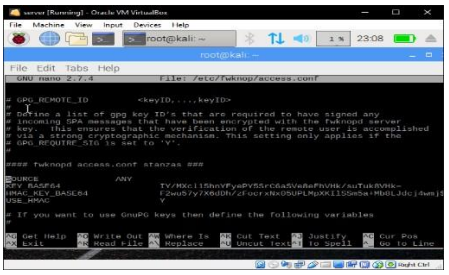
Pengujian sistem dilakukan untuk mengetahui apakah sistem sudah berjalan dengan baik, pengujian sistem pada penelitian ini dilakukan dengan melakukan *scanning port* untuk memeriksa kondisi bagaimana jika suatu *server* tertanam sistem autentikasi dan tanpa sistem autentikasi, melakukan *penetration test* dengan menggunakan Ncrack

1. Melakukan konfigurasi Program *Port knocking*



Gambar 4.3 Konfigurasi *port knocking*

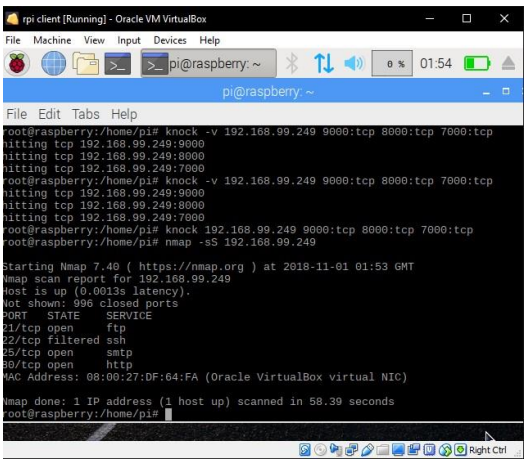
2. Melakukan Konfigurasi Program *Single Packet Authorization*



Gambar 4.4 Konfigurasi *key* fwknop

```
fwknop-server.service - Firewall Knock Operator Daemon
   Loaded: loaded (/etc/systemd/system/fwknop-server.service;
enabled)
   Active: active (running) since Wed 2017-07-26 19:59:18 CEST; 3h
14min ago
  Process: 422 ExecStart=/usr/sbin/fwknopd (code=exited,
status=0/SUCCESS)
 Main PID: 516 (fwknopd)
   CGroup: /system.slice/fwknop-server.service
           └─516 /usr/sbin/fwknopd
```

Gambar 4.5 Fwknop aktif

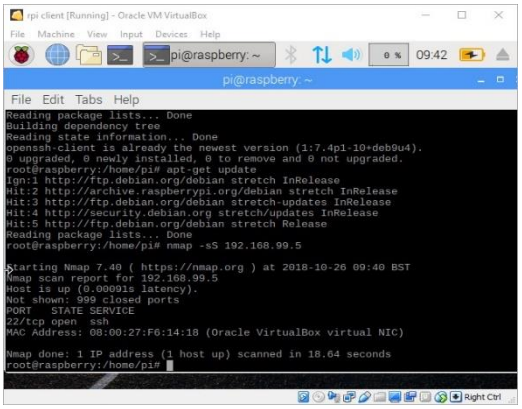


Gambar 4.6 Hasil *Scan Port kocking*

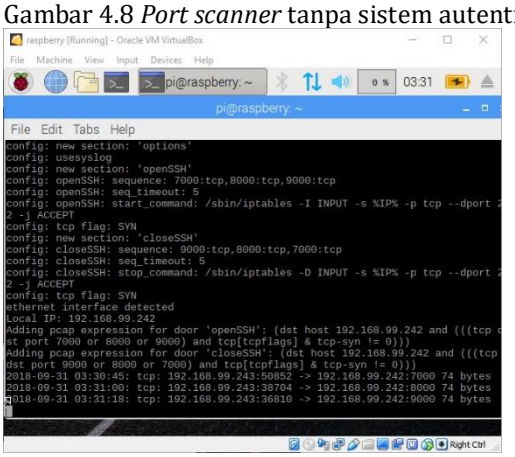
Gambar diatas menunjukan bahwa program fwknop yang telah berhasil dijalankan. Untuk menjlankan program *port knocking* dan SPA terlebih dahulu harus mengkonfigurasikan *iptables firewall* dengan menjalankan sebuah *command* untuk memastikan *port* 22 yaitu *port* SSH tertutup dan tidak bisa diakses sebelum melalui sistem autentikasi. *Script command* yang dijalankan adalah sebagai berikut :

```
system("iptables –A INPUT –p tcp –
dport 22 –j REJECT")
```

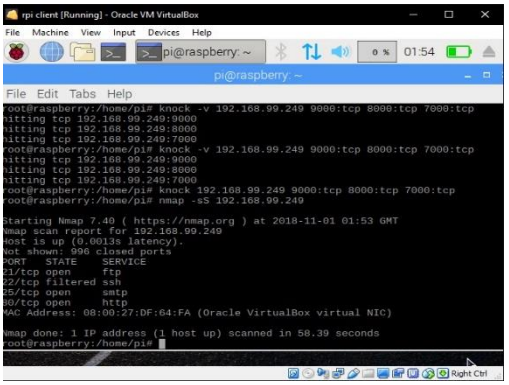
Perintah diatas digunakan untuk membatasi/menutup *port* 22 dari koneksi luar, jika ada koneksi yang masuk, maka firewall akan melakukan penolakan. Dengan berjalannya *program server* ini maka *port* 22 tidak akan bisa diakses atau tertutup dari segala koneksi yang masuk, sehingga saat dilakukan pengujian *scanning port* menggunakan Nmap maka status dari *port* 22 adalah *filtered*



Gambar 4.8 *Port scanner* tanpa sistem autentikasi



Gambar 4.9 Tampilan *server* saat *client* sukses melakukan ketukan

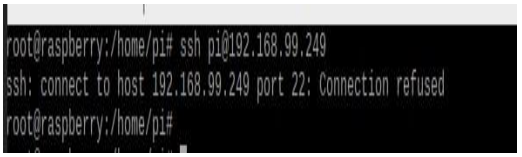


Gambar 4.7 Hasil *Scan Port* 22

Pada hasil pengujian *scanning port* pada gambar 4.13 menggunakan *tool* Nmap menunjukan *port* 22 berada dalam kondisi *filtered* sehingga segala jenis koneksi menuju *port* ini akan ditolak. Hasil penguijan *scanning port* akan berbeda pada *server* yang tidak tertanam program *server* dimana status dari *port* 22 adalah *open*.

Gambar diatas menunjukan tampilan dari sisi *server* ketika menerima ketukan yang telah sesuai dengan *sequence* dari program *server*. Maka *server* akan menjalankan perintah *script* yaitu : Pengujian Percobaan *Login* SSH Setelah client sukses melakukan *port knocking* dengan benar maka *server* akan mengeksesui *command* untuk membuka *port* 22 agar dapat melakukan koneksi SSH. Setelah *client* sukses melakukan *port knocking* maka status dari *port* 22 pada *server* jika dilakukan *scanning port* akan berubah dari yang sebelumnya *filtered* berubah menjadi *open*.

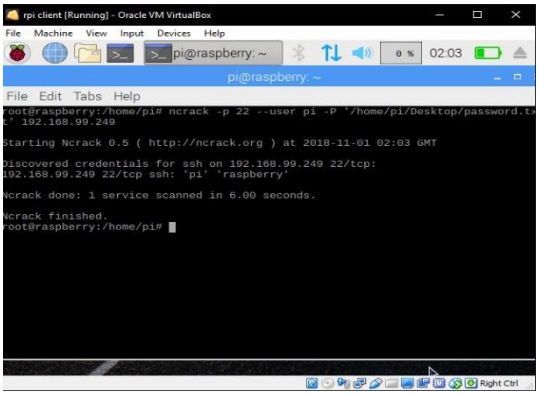
karena *port* 22 di *server* sudah terbuka karena *client* telah sukses melakukan *port knocking*. Berikut adalah tampilan hasil *scanning port* pada *server* setelah *client* sukses melakukan *port knocking*.



Gambar 4.10 Koneksi SSH Ditolak

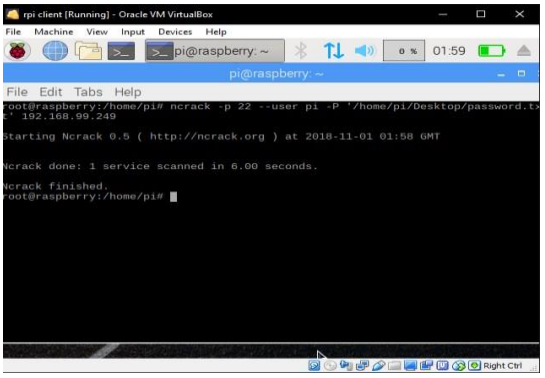
Pada gambar 4.19 menunjukan bagaimana koneksi SSH dari *client* menuju *server* tidak bisa dilakukan karena *client* belum melakukan *port knocking* dan *state* dari *port* 22 di *server* saat itu dalah masih *filtered*, sehingga koneksi SSH tidak bisa dilakukan.

Pengujian selanjutnya adalah pengujian menggunakan *tool* Ncrack, dimana Ncrack merupakan sebuah *tool* untuk melakukan *brute force* yang bertujuan untuk menguji *password* pada seluruh *host* ataupun perangkat jaringan. Ncrack melakukan pencocokan *username* dan *password* berdasarkan data *username* dan *password* yang digunakan saat pengujian. Pada pengujian pertama dilakukan *penetration test* pada *port* 22 di *server* yang tidak menggunakan sistem autentikasi, *state* dari *port* 22 saat dilakukan *scanning* tanpa sistem *port knocking* adalah *open* atau siap menerima koneksi dari mana saja, sehingga saat Ncrack melakukan *penetration test*, *password* dan *username* dari *server* dapat ditebak oleh *tool* Ncrack.



Gambar 4.11 Ncrack Sukses Menebak *Password* dan *Username*

Pada gambar 4.20 menunjukan bagaimana Ncrack sukses menebak *password* dan *username* dari *server* yang tidak menggunakan sistem *port knocking*. Hasil yang berbeda ditunjukan saat *server* menggunakan sistem *port knocking* dimana Ncrack tidak bisa mencocokan *username* dan *password* yang ada di databasenya ke *server* karena *state* dari *port* 22 saat sudah menggunakan sistem *port knocking* adalah *filtered* sehingga tidak ada koneksi yang bisa masuk.



Gambar 4.12 Ncrack Tidak Bisa Mencocokan *Password* dan *Username*

Setelah dilakukan pengujian *port knocking* dari *client* menuju *server* berikut adalah tabel hasil pengujian yang telah dilakukan

1. Tabel Hasil Pengujian Sistem

Tabel 4.1 Hasil perbandingan pengujian system

Pada tabel 1 menunjukan hasil dari perbandingan saat *server* menggunakan *port knocking, single packet authorization* dan tanpa sistem autentikasi, dimana pada tabel menunjukan bagaimana *port knocking* dan *single packet authorization* mampu untuk melindungi *port* 22 atau *port* SSH karena *port knocking* dan SPA sukses untuk menutup *port* 22 dari segala jenis koneksi dan hanya bisa terbuka saat proses *client* dan *server*

2. Tabel Hasil Pengujian *Bruteforce*

Berikut adalah tabel hasil pengujian metode penyerangan *brute force* menggunakan *tool* Ncrack

Tabel 4.1 Hasil pengujian *brute force*

Pengujian ke-	Tanpa Sistem Autentikasi	Dengan Port knocking	Dengan SPA
1	Ncrack sukses mencocokan password dan username	Ncrack tidak bisa mencocokan password dan username	Ncrack tidak bisa mencocokan password dan username
2	Ncrack sukses mencocokan password dan username	Ncrack tidak bisa mencocokan password dan username	Ncrack tidak bisa mencocokan password dan username
3	Ncrack sukses mencocokan password dan username	Ncrack tidak bisa mencocokan password dan username	Ncrack tidak bisa mencocokan password dan username

Pada tabel 1 merupakan hasil pengujian dari *tools* Ncrack untuk melakukan *brute foce* pada *server*, dimana hasil pengujian menunjukan dari 3 kali percobaan pada *server* yang menggunakan *port knocking*, *tool Ncrack* tidak mampu untuk melakukan pencocokan *password* dan *username* dimana saat pada *server* dengan *port knocking* status dari *port* 22 adalah *filtered*. Hasil yang berbeda ditunjukan pada *server* yang tidak menggunakan sistem autentikasi dimana status dari *port* 22 adalah *open*, sehingga dari 3 kali pengujian pada *server* tidak menggunakan sistem autentikasi adalah *tools* Ncrack sukses mencocokan *password* dan *username* sebanyak 3 kali.

Tabel 4.2 Perbandingan kecepatan akses *port server*

Metode	Sequence 1 (detik)	Sequence 2 (detik)	Sequence 3 (detik)	Total (detik)
PK	6	6	6	18
SPA	5	-	-	5

5. SIMPULAN DAN SARAN

A. Kesimpulan

`Dari penelitian yang dilakukan dan hasil pengujian yang didapatkan setelah penelitian, dapat ditarik kesimpulan yaitu, Dengan menggunakan metode *port knocking*, *port server* tidak dapat langsung diakses oleh *client* tanpa melakukan *knocking* dengan *port sequence* yang telah ditentukan.

Dengan menggkonfigurasikan *iptables firewall* akan meningkatkan keamanan metode *port knocking.* Mengimplementasikan *Port knocking* pada sisi *server* menambah waktu pengaksesan terhadap *server* karena harus melakukan *knocking* terlebih dahulu. Dengan menggunakan metode *single packet authorization* mampu meningkatkan keamanan *server* dari berbagai macam serangan yang dilakukan dari pihak dalam maupun luar.

Metode *single packet authorization* menggunakan kombinasi *username*, *password* dan menggunakan enkripsi melalui *payload* data.Perbedaan antara *port knocking* dengan *single packet authorization* terdapat diantara waktu yang dibutuhkan oleh kedua metode yang dimana metode *port knocking* memakan waktu lebih lama dibandingkan *single packet authorization. Port knocking* dan *single packet authorization* memiliki metode yang berbeda, dimana pada metode *port knocking* menggunakan *sequence port* yang terdiri dari beberapa *sequence* sedangkan SPA hanya menggunakan *single packet* untuk melakukan pengaksesan *server*

B. Saran

Dapat ditambahkan peningkatan untuk keamanan data yang dikirim, seperti misalnya adalah kode request saat client pertama kali ingin melakukan port knocking dan keamanan data yang dikirim ke port – port saat proses port knocking. Dapat ditambahkan filterisasi untuk ip address/mac address, hal ini bertujuan agar setiap ip address/mac address yang terdaftar saja yang dapat melakukan port knocking menuju server.

Daftar Pustaka

[1] Pribadi, Harijanto. (2008). Firewall melindungi jaringan dari DdoS menggunakan Linux+Mikrotik. Yogyakarta: ANDI, 2008.
[2] Jelena, Mirkovic. (2005). "A Taxonomy of DDoS Attacks and DDoS Defense Mechanisms". University of California, Los Angeles.
[3] Alex, Manzanares. (2005). Attacks on port knocking authentication mechanism. ICCSA LNCS-Springer Berlin/Heidelberg, 2005, pp 1292-1300.
[4] Sam, Jeanquier. (2006). An analysis of port knocking and single packet authorization. MSc Thesis, Royal Holloway College, University Of London September 9, 2006.
[5] Fajri, Muhammad Saleh Hafiz. dkk (2014). Analisa Port knocking Pada Sistem Operas Linux Ubuntu Server. Vol.2, No.1, April 2014.
[6] Awan. (2014). Memberikan Akses Legal Terhadap Port Tertentu Yang Telah Ditutup oleh Firewall dengan Metode Port knocking. Jurnal Ilmiah Core ItVol. 2 No. 1.
[7] Zorkta, Haythem. (2012). Harden Single Packet Authentication. Vol. 4, No. 5, October 2012.
[8] Al-Bahadili, Hussein. (2013). A Secure Block Permutation Image Steganography Algorithm. Faculty of Information Technology, University of Petra.
[9] Kalaena, L. S., & Bagye, W. (2018). Implementasi Network Attached Storage (NAS) Menggunakan Freenas Pada STMIK Lombok. Jurnal Manajemen Informatika dan Sistem Informasi, 1(1), 6-10.