REPOSITOR, Vol. 2, No. 6, Juni 2020, Pp. 787-794
ISSN : 2714-7975
E-ISSN : 2716-1382

Implementasi Asymmetric Encryption RSA Pada Port Knocking Ubuntu Server Menggunakan Knockd Dan Python

Mukhammad Zainal Amir Mahmud*[1], Syaifuddin[2], Diah Risqiwati[3]
[1,2,3]Teknik Informatika/Universitas Muhammadiyah Malang
Zainal0909@gmail.com*[1], saifuddin@umm.ac.id[2],risqiwati@umm.ac.id[3]

Abstrak

Penelitian dalam mekanisme otentikasi telah mengarah pada rancangan dan pengembangan skema baru. Keamanan yang disediakan oleh metode ini harus ditinjau dan dianalisis sebelum dapat digunakan secara luas. Dalam penelitian ini, kami menganalisis beberapa kelemahan dari metode autententikasi port knocking yang membuatnya rentan terhadap banyak serangan. Kami akan menyajikan serangan Sniffing, di mana penggunaan yang tidak sah dapat memperoleh akses ke server yang dilindungi hanya dengan melakukan perekaman lalu lintas data yang ada pada jaringan lokal menggunakan tools wireshark. Untuk itu metode port knocking akan dikembangkan, port knocking ini menggunakan knockd yang ada pada ubuntu server, port knocking merupakan metode melindungi port yang penting agar tidak dapat terlihat segai port yang terbuka. Namun metode port knocking ini masih memiliki kelemahan terhadap serangan seperti TCP replay, bruteforce, dan lain-lain. Penelitian ini mengusulkan metode autentikasi dan end to end connection untuk menambah keamanan pada metode port knocking. user harus melakukan koneksi end to end terlebih dahulu sebelum melakukan knocking, setelah melakukan knocking user juga harus melakukan autentikasi pada server untuk dapat membuka port yang diinginkan.

Kata Kunci: *Knocking, Authentication, Knockd, End to End*

Abstract

Research in authentication mechanisms has led to the design and development of a new scheme. Security provided by these methods must be reviewed and analyzed before it can be widely used. In this study, we analyze some of the weaknesses of the port knocking method autententikasi which makes it vulnerable to many attacks. We will present Sniffing attacks, where unauthorized use can obtain access to the protected server just by doing the recording of data traffic on the local network using wireshark's tools. For that port knocking method will be developed, port knocking uses knockd on ubuntu server, port knocking is a method of protecting an important port in order not to be seen as an open port. But the port knocking method still has a weakness against attacks such as TCP replay, bruteforce, and others. This research proposes a method of authentication and end to end connection to increase security at the port knocking method. the user must connect end to end first before doing the knocking, after knocking the user must also perform authentication on the server to be able to open the port that you want.

Keywords: *Knocking, Authentication, Knockd, End to End*

1. Pendahuluan

Autentikasi telah menjadi masalah dalam kemanan komunikasi dalam dekade terakhir , para pelaku melakukan identifikasi sebelum koneksi terjadi antara satu sama lain, sehingga pelaku dapat mendapatkan akses dengan menggunakan teknik teknik penyerangan tertentu. Seperti yang kita ketahui , "jalur akses" ke komunikasi, jadi target utama pelaku potensial untuk penyerang. Ini berarti proses autentifikasi harus diamankan dengan menggunakan protokol atau mekanisme yang memungkinkan masing-masing prinsip melakukan verifikasi identitas sebelum melakukan komunikasi, hal dasar dalam melakukan autentikasi adalah dengan memasukkan username dan password [1].

Port knocking merupakan mekanisme autentikasi pada *server* yang hanya diketahui oleh orang yang menggunakan metode ini [2]. Dalam hal ini, kata sandi bukanlah urutan karakter tetapi urutan *port*: *Server* membuat semua *port* jaringannya tertutup dan *port* ini harus "diketuk" dalam urutan yang benar agar *server* dapat membuka *port* komunikasi yang diinginkan. Prosedur untuk

"mengetuk" *port* terdiri atas pengiriman paket ke *port* itu, sehingga *server* akan melihat upaya koneksi terhadap *port* tertutup dan mencatatnya [3].

Keuntungan menggunakan *port knocking* yaitu dapat menutup semua *port* pada jaringan tersebut, sehingga jika dilakukan scanning maka penyerang tidak menemukan *port* yang terbuka, klien harus melakukan ketukan dengan benar untuk membuka *port* yang tertutup dapat terbuka [4]. *Port knocking* adalah sebuah metode yang menutup service dari penyerang dengan cara data ditransmisikan melalui port yang tertentu. Pada dasarnya *Port knocking* merupakan sebuah mekanisme keamanan jaringan yang tertanam dalam Firewall pada Secure Computer System [5]. Pada referensi lain mengatakan bahwa *Port knocking* adalah sebuah metode otorisasi user berdasarkan firewall untuk melakukan komunikasi melalui port yang tertutup [2]. Serangan yang umum terjadi antara lain adalah DDoS dan NAT *knocking*, DDoS pada *port knocking* menyebabkan *server* terlalu sibuk memproses paket dari client yang melakukan serangan sehingga *server* menjadi sangat terbebani dan akhirnya layanan pada *server* tersebut menjadi terganggu, DDoS ini menekankan pada pengiriman paket SYN secara terus menerus dari client dengan pemalsuan IP Address dari client tersebut sehingga *server* akan sibuk mengirimkan paket SYN,ACK dan pada client yang menyerang *server* tersebut paket ACK yang seharusnya dikirim kembali ke *server* tidak dikirimkan sebagaimana mestinya [6]. NAT *knocking* adalah serangan yang memungkinkan client yang tidak valid dapat memiliki akses ke suatu sistem, konsep serangan ini adalah saat ada client yang berada dibalik NAT menggunakan *port knocking* untuk masuk ke suatu sistem maka si penyerang akan menunggu sampai client tersebut menyelesaikan *sequence port knocking*nya sampai port yang menjadi tujuan client tersebut terbuka, dari sini penyerang akan mencoba masuk ke port tersebut tanpa melakukan *knocking* karena client yang sebelumnya melakukan *knocking* telah membuka akses ke port, bila si penyerang dapat masuk ke port tersebut maka pada *server* yang menjalankan *port knocking* dipastikan tidak memvalidasi client yang terhubung dengannya [5].

Pada penelitian yang dilakukan oleh Leleh Boroumand yang berjudul "Virtualization Technique for *Port knocking* in Mobile Cloud Computing" memberikan saran agar timeout *sequence* yang digunakan memiliki banyak variasi dikarenakan celah ini memberikan kesempatan bagi seorang penyerang untuk menemukan *sequence* yang tepat [7]. Pada penelitian yang dilakukan oleh Mehran Pourvahab yang berjudul"Secure *Port knocking*-Tunneling" menyatakan bahwa pemanfaatan tunneling pada penelitiannya memberikan pengamanan dalam komunikasi dan mencegah NAT-*knocking* dan untuk pengembangannya kedepan diperlukan *sequence* dan model enkripsi yang lebih baik guna mengurangi kinerja berlebih pada *server* [8]. pada dasarnya metode *port knocking* ini masih rentang terhadap serangan *sniffing*, namun yang membedakan dari penelitian sebelumnya yaitu tingkat kesulitan untuk memecahkan payload yang didapat dari hasil *sniffing* [9].

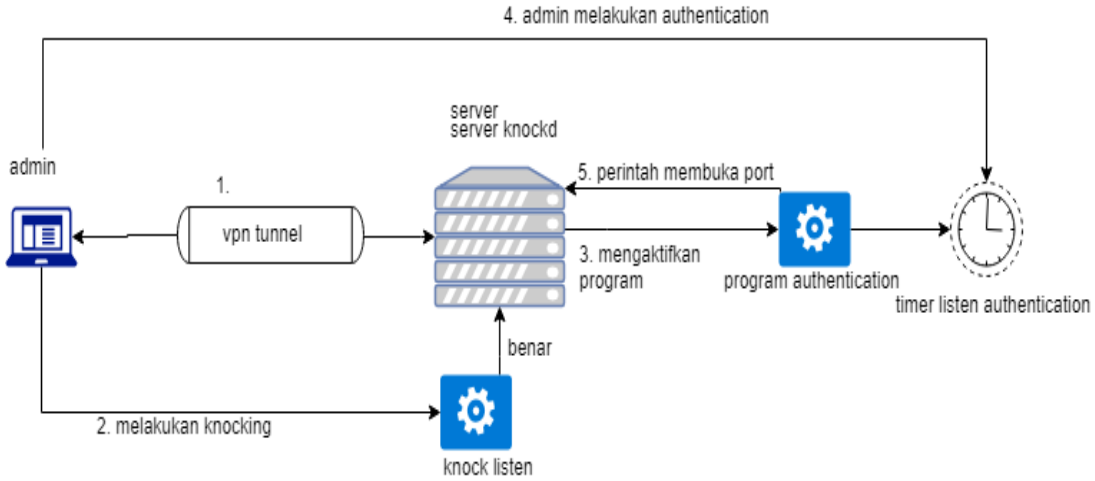
Pada penelitian, ini peneliti akan mencoba mengimplementasikan autentikasi *client port knocking* pada ubuntu *server* versi 16. Authentication ini dilakukan *client* pada *server* yang bertujuan untuk memvalidasi *client* yang melakukan *knocking* ke *server*, guna menghindari serangan *bruteforce*. dan menambahkan end to end koneksi yang bertujuan meningkatkan keamanan *port knocking*. Dan menambahkan tunneling antara client dan *server*, sehingga lebih memungkinkan menambah untuk keamanan *port knocking* itu sendiri.

2. Metode Penelitian

Port knocking memungkinkan terjadinya komunikasi antar host melalui *port* yang tertutup secara umum terdapat berbagai metode yang bisa diterapakan dalam *port knocking*, informasi yang dikirmkan dapat berupa *sequence* ataupun payload yang berisi data, dalam prakteknya data yang dikirm dalam bentuk payload akan diterima oleh *daemon* yang bertugas menunggu dan mengobservasi *sequence* yang dikirm oleh *client* ataupun isi payload yang dikirm oleh *client*. Dalam *port knocking*, *server* yang bertindak sebagai penyedia layanan *port knocking* tidak membuka *port* yang akan digunakan untuk komunikasi, melainkan menyediakan listener terhadap paket SYN pada *port* tertentu sehingga ketika *port* yang diberi listener tersebut menerima paket SYN dari *client* dapat digunakan sebagai acuan suatu autentikasi berdasarkan sequnce. Sebagai contoh disediakan *port* (1145,1087,1172) yang bertugas menerima SYN dari *client*, ketika *client* berhasil mengirim SYN ke *port-port* tersebut dalam interval waktu tertentu maka dapat diasumsikan *client* tersebut valid karena informasi knock *sequence* ini bersifat sangat rahasia dan hanya diketahui oleh *admin* dan *client* saja[10].

2.1 Skenario Program

Tahapan ini merupakan perancangan sistem terhadap solusi dari permasalahan yang ada dengan menggunakan perangkat pemodelan sistem seperti alur sistem dan flowchart, pada alur akan digambarkan runtutan kinerja program pada penelitian ini, pada bagian flowchart akan menggambarkan alur program yang jalankan seperti pada Gambar 1.



Gambar 1. Alur Sistem

Pada Gambar 1 merupakan gambaran dari urutan yang harus dilalui oleh *client* dalam melakukan *port knocking* autentikasi.

1. Nomor 1 *admin* diharuskan melakukan *tunnel* ke *server*.
2. *admin* melakukan *knocking* dengan menginsialisasi *port* yang telah ditentutakan contoh 1000 2000 3000.
3. Secara otamatis *server* mengaktikan program autentikasi, jika tidak ada yang melakukan autetikasi selama waktu yang telah ditentukan program akan dinonaktifkan. Untuk membuka harus melakakuan nomor 2 terlebih dahulu.
4. *admin* melakukan autentikasi dengan mengaktifkan program yang hanya ada pada *admin* valid. Jika autentikasi melewati batas waktu maka akan diulang ke nomor 2.
5. Program akan mengirimkan perintah untuk membuka *port* yang telah di tutup.

Penggunaan algoritma RSA dalam pembuatan public dan *private* key digunakan dengan tujuan untuk mengamankan *payload* yang dikirmkan menggunakan komunikasi *socket* menggunakan bahasa pemrograman python, payload tersebut harus dienkripsi terlebih dahulu oleh *firewall* yang menjalankan *daemon* knockd kemudian bila ada *client* yang melakukan *knocking* maka payload tersebut akan dikirim ke *client* kemudian tugas *client* adalah mendekrip payload tersebut untuk dikirim kembali ke firewall sebagai autentikasi dan validasi, proses enkripsi dari firewall ke *client* akan menggunakan kunci public yang dimiliki oleh firewall sementara pada *client* akan menggunakan kunci *private* untuk membuka payload tersebut.

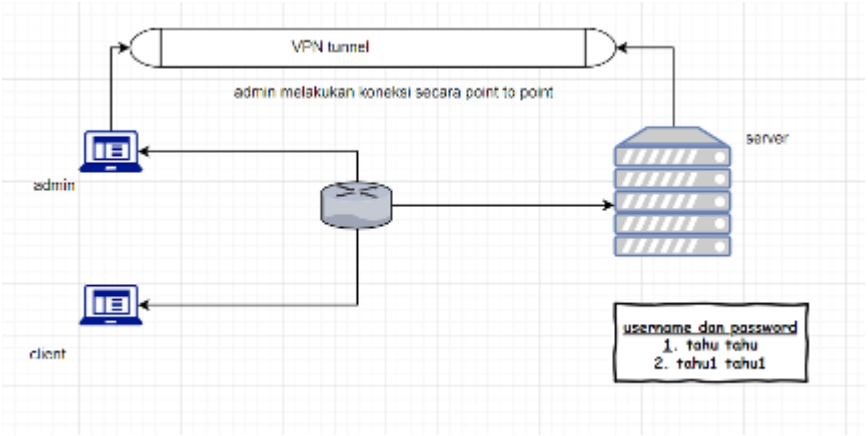
Penggunaan *tunneling* digunakan untuk menambah keamanan pada autentikasi, *admin* harus melakukan *tunneling* sebelum melakukan *knocking* dan autentikasi. program autentikasi akan berjalan jika *admin* dan *server* terhubung dengan VPN, jika tidak *admin* tidak bisa melakukan autentikasi. *Admin* tidak dapat membuka port yang diinginkan walaupun sudah melakukan *knocking*, dikarenakan program autentikasi menggunakan IP address VPN untuk melakukan autentikasi.

Penggunaan timeout untuk mebatasi program melakukan waiting autentikasi, sehingga *server* tidak akan menunggu lama untuk balasan autentikasi dari *admin*. Jika program melewati batas waktu yang ditentukan, maka program akan dimatikan atau dihentikan. Sehingga *server* akan melakukan listen port kembali, jika ada permintaan dengan mengetuk port tertentu, maka program akan bisa digunakan kembali.

Perancangan alur sistem ini dapat menambah keamanan pada metode *port knocking*, sehingga metode ini akan membantu mengamankan port yang sering terjadi eksploitasi. Penamabahan keamanan tunneling dan autentikasi hampir tidak mungkin untuk dapat mengksploitasi pada port yang dilindungi oleh metode ini.

2.2 Perancangan Sistem VPN

Pada perancangan sistem VPN akan dijelaskan pada Gambar 2.



Gambar 2. Alur Tunnel VPN

Pada Gambar 2 dijelaskan *admin* berada satu network dengan *client*, untuk masuk jaringan *private admin* harus mengetahui username dan password.

1. *Server* mengaktifkan layanan ptpp.
2. *Admin* mengaskes VPN menggunakan username dan password.
3. *Admin* dapat masuk pada jaringan *private*.

Perancangan VPN tunnel ini menggunakan PPTPD pada *server*. *Server* menggunakan ubuntu *server*, PPTPD merupakan sebuah fitur pada *server* untuk digunakan sebagai VPN *server*. *Admin* menggunakan username dan password untuk dapat melakukan koneksi private terhadap *server* ataupun dengan client yang lain. VPN dapat menambah keamanan pada metode *port knocking* .

2.3 Tahapan Sistem

Metode *port knocking* pada penelitian ini memiliki 3 tahapan yang digunakan untuk melakukan otorasi *client*, *client* harus melewati tahapan untuk membuka *port* yang tertutup tersebut, yang pertama *client* melakukan tunneling ke *server* sehingga didalam satu jaringan *private*, kedua melakukan ketukan yang telah ditentukan, dan yang terakhir melakukan autentifikasi pada *server*.

Proses pertama adalah melakukan koneksi tunnel antara *client* dan *server* sehingga berada dalam satu jaringan *private*, untuk melakukan jaringan *private* user harus menggunakan username dan password yang sudah ditentukan pada *server*. Username dan password dibuat pada saat konfigurasi VPN pada *server*, hanya username dan password yang terdaftar yang bisa login VPN.

Proses kedua adalah melakukan ketukan untuk menjalankan sebuah program autentifikasi yang otomatis berjalan setelah melakukan ketukan dengan benar. Tahapan ini merupakan awal untuk membuka port yang diinginkan pada metode *port knocking*.

Proses ketiga adalah autentifikasi dari *client*, *client* harus dapat mengautentikasikan diri sehingga proses dapat berlanjut, setelah *client* dapat mengautentikasikan dirinya, *client* dapat menggunakan layanan yang dibutuhkan, proses ini digunakan untuk mencegah serangan replay attack dan *bruteforce*. Untuk authentication ini menggunakan program yang diletakan di *server* dan *client*, *client* harus mengaktifkan program sehingga dapat melakukan authentication pada *server*, sebaliknya pada *server* program automatic ketika *knocking* yang dilakukan *client* berhasil.

3. Hasil Dan Pembahasan

Pengujian dilakukan dengan sistem pada penelitian ini menggunakan serangan *sniffing* dan *bruteforce*. Pada pengujian ini akan dialakukan tiga sistem uji yaitu *port knocking* biasa, *port knocking* ditambahkan dengan autentikasi program, serta *port knocking* ditambahkan autentikasi dan VPN. Dari percobaan serangan pada tiga sistem dapat diketahui perbedaan dari tiga sistem tersebut.

3.1 *Port knocking* Biasa

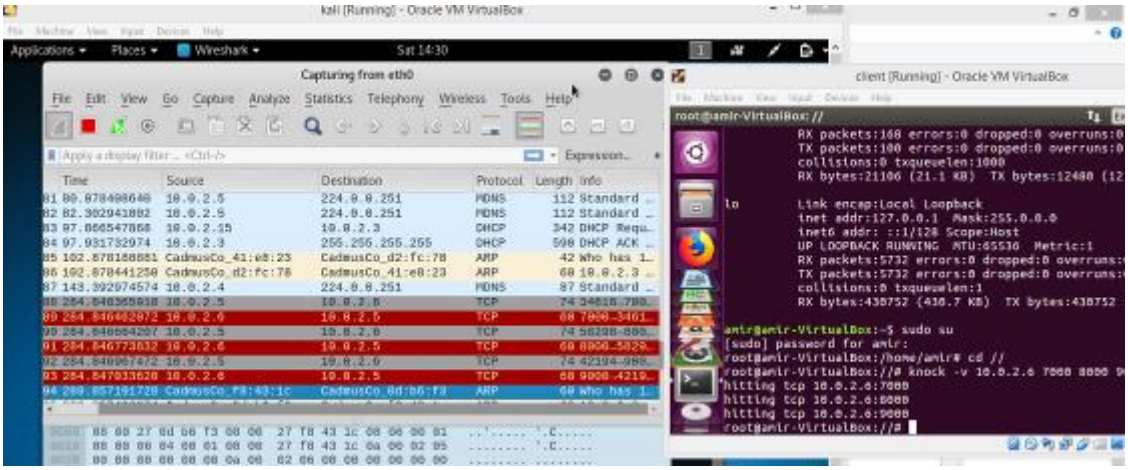
Langkah-langkah yang digunakan pada port kncoking biasa agar dapat membuka port.

1. *Server* melakaukaan listening port.
2. *Admin* melakukan *knocking* pada *server* knockd.
3. Menerima Ketukan dengan benar dari *admin*.
4. *Server* membuka akses port yang dilindungi

Tabel 1. Hasil Port Knocking Biasa

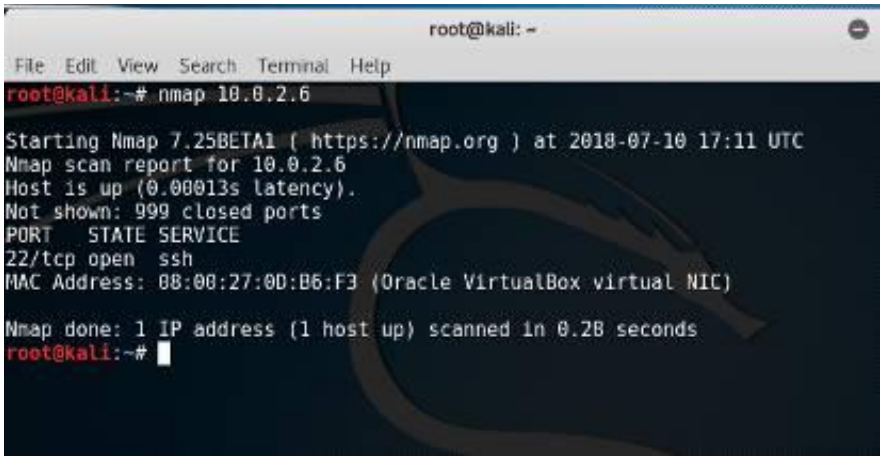
Serangan	Port Dapat Terbuka
Bruteforce	ya
Sniffing	ya

Kesimpulan yang didapat dari Tabel 1 adalah serangan *bruteforce* dapat membuka port yang tertutup dengan cara melakukan ketukan secara urut sampai menemukan kombinasi kunci yang telah ditentukan oleh *admin*. Dan serangan *sniffing* juga dapat menemukan kombinasi kunci dengan melakukan perekaman lalu lintas data yang ada pada jaringan lokal, kombinasi kunci akan diketahui jika *admin* melakukan *knocking* dan terekam oleh wireshark. Sehingga penyerang dapat menggunakan kunci untuk memperoleh akses port yang tertutup.



Gambar 3 Hasil Sniffing Port Knocking Biasa

Pada Gambar 3 dapat disimpulkan aktifitas client ke *server* terkam oleh wireshark, dan penyerang mendapatkan kombinasi kunci yang benar.



Gambar 4. Hasil Bruteforce

Pada Gambar 4 dijelaskan port akan terbuka jika serangan *bruteforce* berhasil menemukan kombinasi kunci yang benar.

3.2 *Port knocking* Autentikasi

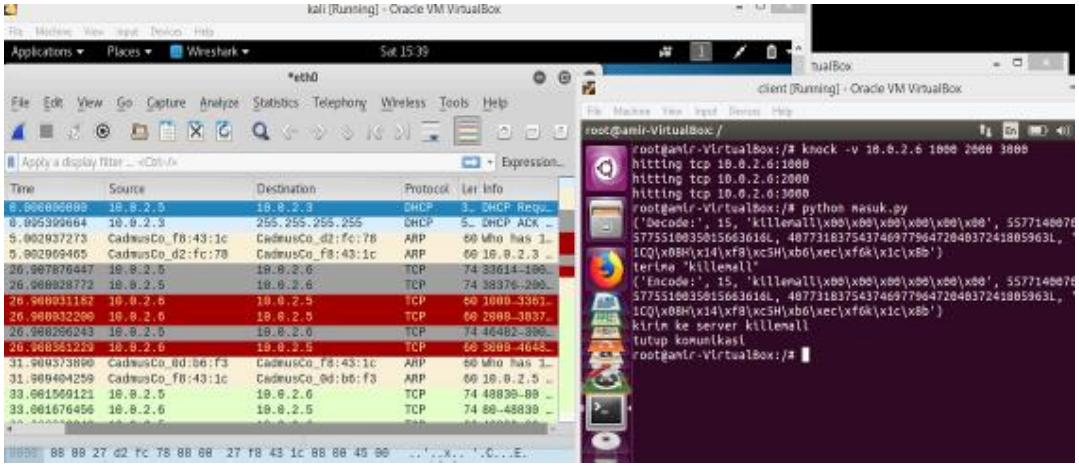
Langkah-langkah yang digunakan pada port kncoking biasa agar dapat membuka port.

1. *Server* melakaukaan listening port.
2. *Admin* melakukan *knocking* pada *server* knockd.
3. Menerima Ketukan dengan benar dari *admin*.
4. *Server* menjalankan program autentikasi.
5. *Admin* melakukan autentikasi.
6. *Server* membuka akses port yang dilindungi

Tabel 2. Hasil Port knocking Autentikasi

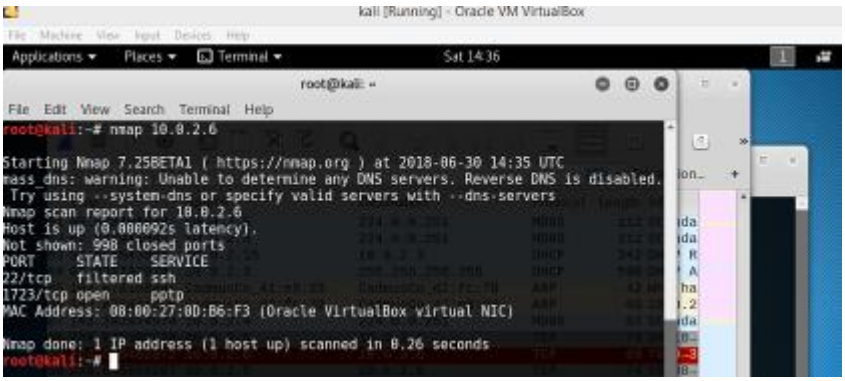
Serangan	Port Dapat Terbuka
Bruteforce	tidak
Sniffing	tidak

Kesimpulan yang didapat dari Tabel 2 adalah serangan *bruteforce* tidak dapat membuka port yang tertutup, walaupun menemukan kombinasi kunci yang telah ditentukan oleh *admin* dikarenakan adanya step autentikasi yang harus dilakukan setelah *knocking*. Dan serangan *sniffing* juga dapat menemukan kombinasi kunci namun juga tidak dapat mengakses dikarenakan harus melakukan autentikasi terlebih dahulu untuk membuka port yang tertutup tersebut.



Gambar 5. Hasil Sniffing Port knocking Autentikasi

Pada Gambar 5 dapat dijelaskan serangan *sniffing* juga dapat menemukan kombinasi kunci, Namun tidak dapat mengakses layanan diakrenakan adanya proses autentikasi setelah proses *knocking*.



Gambar 6. Hasil Bruteforce Port knocking Autentikasi

Pada Gambar 6 dapat dijelaskan penyerangan tidak dapat mengetahui adanya autentikasi setelah *knocking*, sehingga penyerang tidak dapat mengakses layanan dikarenakan status port masih filtered.

3.3 Port Knoking Autentikasi VPN

Langkah-langkah yang digunakan pada port kncoking biasa agar dapat membuka port.

1. *Server* membuka layanan pptp
2. *Admin* melakukan jaringan private
3. *Server* melakaukaan listening port.
4. *Admin* melakukan *knocking* pada *server* knockd.
5. Menerima Ketukan dengan benar dari *admin*.
6. *Server* menjalankan program autentikasi.
7. *Admin* melakukan autentikasi.
8. *Server* membuka akses port yang dilindungi

Table 3. Hasil Port knocking Autentikasi VPN

Serangan	Port Dapat Terbuka
Bruteforce	tidak
Sniffing	tidak

Kesimpulan yang didapat dari Tabel 3 adalah serangan *bruteforce* tidak dapat membuka port yang tertutup, walaupun menemukan kombinasi kunci yang telah ditentukan oleh *admin* dikarenakan adanya step autentikasi yang harus dilakukan setelah *knocking* dan adanya juga proses tunnel terlebih dahulu sebelum melakukan *knocking*. Dan serangan *sniffing* juga dapat menemukan kombinasi kunci namun juga tidak dapat mengakses dikarenakan harus melakukan autentikasi terlebih dahulu untuk membuka port yang tertutup tersebut dan adanya juga proses tunnel terlebih dahulu sebelum melakukan *knocking*. Autentikasi pada sistem ini juga menggunakan IP address setelah tunnel. jika *admin* ingin melakukan melakukan autentikasi harus terlebih dahulu masuk dalam jaringan private.

3.4 Analisa Pengujian Sistem

Setelah melakukan pengujian pada tiga sistem diatas terdapat perbedaan hasil dari pengujian tersebut. Perbedaan yang paling terlihat dari tiga sistem yaitu hasil sistem *port knocking* biasa, hasil dari pengujian sistem 2 dan yang ketiga terlihat tidak jauh berbeda, seperti pada Tabel 4.

Tabel 4. Hasil Analisa Pengujian Tiga Sistem

Sistem	Percobaan serangan		Dapat masuk SSH
	Sniffing	*Bruteforce*	
Sistem 1	Ya	Ya	Ya
Sistem 2	Ya	Tidak	Tidak
Sistem 3	Ya	Tidak	Tidak

Setelah melakukan serangan *sniffing* dan *bruteforce* dapat dengan media *server* yang menggunakan knockd standard, knockd +autentikasi dan knockd +autentikasi+VPN. Pada knockd standard dilakukan *sniffing* dan mendapatkan *sequence* yang digunakan untuk mengakses *port knocking* pada *server*, pada saat client melakukan *knocking* ke *server*. Jika penyerang menggunakan tersebut maka akan dapat mengakses SSH pada *server* tersebut. Namun pada *server* knockd + autentikasi, jika dilakukan *sniffing* juga akan mendapatkan *sequence* untuk mengakses SSH, Namun penyerang tidak dapat akses SSH dikarenakan setelah *knocking* harus melakukan autentikasi yang dimiliki oleh client yang sah. Dan sistem yang menggunakan knockd +autentikasi+VPN, Hanya dapat merekam kombinasi kunci namun tidak dapat merakam autentikasinya.

Percobaan kedua yaitu melakukan serangan buteforce, pada *server* knockd standard serangan buterforce dapat menemukan kombinasi *sequence* namun membutuhkan waktu dan serangan dapat melakukan akses SSH setelah menemukan kombinasi *sequence* dengan cara mengetuk secara looping. Pada *server* knockd + autentikasi dicoba juga di serang menggunakan *bruteforce* namun setelah menemukan kombinasi *sequence* dengan cara melooping port, penyerang juga tidak dapat mengakses SSH dikarenakan harus melakukan autentikasi terlebih dahulu. Tidak jauh beda pada *server* knockd + autentikasi+VPN dilakukan serangan menggunakan *bruteforce* namun setelah menemukan kombinasi *sequence* dengan cara

melooping port, penyerang juga tidak dapat mengakses SSH dikarenakan harus melakukan autentikasi terlebih dahulu.

4. Kesimpulan

Port knocking memiliki potensi pengembangan yang cukup baik terutama dibagian pengamanan *sequence* dan timeout serta autentikasi yang terjadi setelah melakukan *knocking*, dalam penelitian ini metode asymmetric enctyption RSA digunakan guna autentikasi dari *server* ke *client* dan sebaliknya, tidak hanya itu proses enkrip dan dekrip dilakukan dengan waktu yang paling cepat, sehingga proses *knocking* tidak membutuhkan waktu yang lama. Berdasarkan penelitian tentang metode *port knocking* yang telah dilakukan oleh peneliti. Berdasarkan hasil uji sistem *port knocking* yang dilakukan oleh peneliti, maka dapat menyimpulkan saran kepada peneliti selanjutnya, yaitu memperbaiki bug yang ada pada sistem knockd itu sendiri, knockd tidak mampu mengontrol jika user melakukan *knocking* lebih dari satu kali , knockd akan menuliskan rule setiap *knocking* user benar. Untuk penelitian selanjutnya agar dapat memperbaiki bug yang terjadi pada knockd itu sendiri.

Referensi

[1] S. Ibbi, “Memberikan Akses Legal Terhadap Port Tertentu Yang Telah Ditutup oleh Firewall dengan Metode Port Knocking,” pp. 1–9.
[2] M. Rash, *linux Firewalls: Attack Detection and Response with iptables, psad, and fwsnort*. No Starch Press, 2007.
[3] A. I. Manzanares, J. T. Márquez, J. M. Estevez-Tapiador, and J. C. H. Castro, “Attacks on port knocking authentication mechanism,” *Lect. Notes Comput. Sci.*, vol. 3483, no. April, pp. 1292–1300, 2005.
[4] dkk M. Fahru Rozi, “Implementasi Remote Server Menggunakan Metode Port Knocking dengan Asymetric Encryption,” *Semin. TA*, vol. 2, no. Jaringan Komputer, pp. 1–5, 2010.
[5] Z. A. Khan, N. Javaid, M. H. Arshad, A. Bibi, and B. Qasim, “Performance evaluation of widely used portknoking algorithms,” *Proc. 14th IEEE Int. Conf. High Perform. Comput. Commun. HPCC-2012 - 9th IEEE Int. Conf. Embed. Softw. Syst. ICESS-2012*, pp. 903–907, 2012.
[6] D. Isabel, “InfoSec Reading Room Port Knocking : Beyond the Basics Port Knocking : Beyond the Basics In tu ho f rig,” *Inf. Secur.*
[7] L. Boroumand, M. Shiraz, A. Gani, and R. Khokhar, “Virtualization Technique for Port Knocking in Mobile Cloud Computing,” vol. 6, no. 1, pp. 1–25, 2014.
[8] P. Mehran, E. A. Reza, and B. Laleh, “SPKT: Secure Port Knock-Tunneling, an enhanced port security authentication mechanism,” *2012 IEEE Symp. Comput. Informatics, Isc. 2012*, no. March 2015, pp. 145–149, 2012.
[9] C. Indore, “A Result Analysis of Modified Hybrid Port Knocking (MHPK) With Strong Authentication,” vol. 14, no. 3, pp. 107–114, 2014.
[10] M. Krzywinski, “Port knocking from the inside out,” *SysAdmin Mag.*, vol. 12, no. 6, pp. 12–17, 2003.