MENINGKATKAN KEAMANAN PORT KNOCKING DENGAN KOMBINASI SPECIAL FEATURES ICMP, SOURCE PORT, DAN TUNNELING

Edy Haryanto[1)], Widyawan[2)], dan Dani Adhipta [3)]
[1, 2,3)]Jurusan Teknik Elektro dan Teknologi Informasi
Universitas Gadjah Mada
Jl. Grafika No.2 Yogyakarta - 55281
e-mail: edy.haryanto.mti13@mail.ugm.ac.id[1)], widyawan@ugm.ac.id[2)], dhani@te.ugm.ac.id[3)]

ABSTRAK

Jaringan komputer selalu rentan terhadap berbagai macam serangan. Secara umum serangan-serangan tersebut terdiri dari identification attacks, acquire attacks dan disabling services attacks. Pada Identification attack, penyerang mengumpulkan informasi tentang service-service yang bejalan guna menemukan kelemahan yang belum di patch bahkan zero-day. Port knocking merupakan teknik pertahanan yang digunakan untuk mencegah penyerang melakukan scanning guna mendapatkan informasi tentang kelemahan service yang berpotensi dieksploitasi. Port knocking merupakan sebuah metode otorisasi user berdasarkan firewall untuk melakukan komunikasi melalui port yang tertutup. Akan tetapi port knocking masih memiliki beberapa kelemahan seperti NAT Knocking, Dos Knocking dan paket out of delivery. penelitian ini akan mengkombinasikan beberapa metode untuk membangun metode Port Knocking yang lebih ringan, sederhana namun tetap aman. Menggunakan special features pada ICMP dan source port guna mencegah serangan DOS Knocking dan juga menggunakan tunneling VPN guna mencegah serangan NAT Knocking.

***Kata Kunci**: ICMP, Port Knocking, Source Port, Tunneling*

ABSTRACT

Computer network always vulnerable with any kind of attack, these attacks are typically include identification attacks, acquire attacks and disabling services attacks. At identification attacks, the attackers attempt to gather information and identify running services for discovering the vulnerable, unpatch service, even the zero day. Port-Knocking is a unique method to prevent detection and exploiting vulnerable services. Port knocking is a method to authorized user base on firewall to transmite the communication throught close port. Ufortunately port knocking still have any vulnerabilities such as NAT Knocking, DOS Knocking and paket out of delivery. This research will combine any method to make a new port knocking method that more light, simple, but still secure. Using special feature of ICMP and source port to prevent DOS Knocking and using VPN tunneling to prevent NAT knocking.

***Keywords**: ICMP, Port Knocking, Source Port, Tunneling*

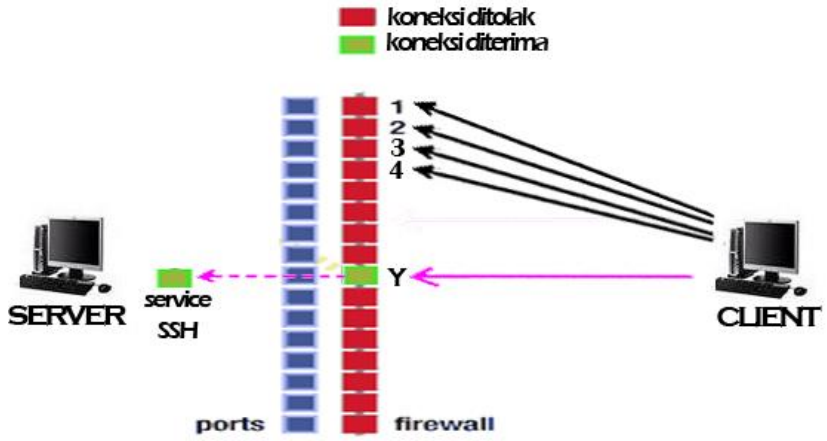
I. PENDAHULUAN

Proses otorisasi sangatlah penting pada jaringan komputer, khususnya untuk komunikasi secara *remote* pada *public network*. Memberikan layanan aman yang berjalan pada *public network* bukahlah hal yang mudah. Membiarkan *port service* terbuka untuk *publik* akan mengundang perhatian untuk terjadinya penyerangan. Akan tetapi beberapa *service* harus dapat diakses oleh *public* seperti *HTTP* dan *SMTP*.

Memantau dan mengontrol aksesibilitas *port* dapat meningkatkan keamanan jaringan komputer dan *port knocking* adalah salah satu solusinya. *Port knocking* adalah sebuat metode yang dapat menyembunyikan *service* dari penyerang dengan cara mentrasmisikan data melalui *port* yang tertutup[1]. *Port Knocking* adalah sebuah strategi untuk mendapatkan akses secara *remote* tanpa harus membiarkan *Port* terbuka secara konstan dimana *port* tersebut dikontrol oleh *Firewall* [5]. *Firewall* berfungsi untuk mengontrol aliran *traffic packet*, ketika paket menuju *Firewall*, maka *Action* yang akan dipilih adalah salah satu dari "*Allow*" "*Reject* dan "*Drop*".
Allow action akan mengizinkan paket melintasi *firewall* untuk mengakses *service* tertentu. *Reject* dan *Drop* paket akan mencegah paket melintasi *firewall*.

Metode otentikasi *port knocking* bertujuan untuk memberikan lapisan keamanan tambahan yang ringan pada jaringan komputer yang berjalan malalui *port* yang tertutup[6]. Klien akan mengirim beberapa *non reply synchronized* (*SYN*) *packet* secara spesifik menuju *port* yang tertutup pada *firewall server*, hal ini disebut Proses Ketukan. jumlah dan interval ketukan tersebut dinamakan *knock*

sequence atau ketukan rahsia dan ketukan rahasia inilah yang akan digunakan sebagai otentikasi. Ketukan rahasia ini bisa didefinisikan secara *static* maupun *dynamic*.
Server akan mengalokasikan *memory* untuk menganalisa setiap paket yang datang guna mendeteksi ketukan rahasia dan bila ketukan rahasia valid, maka *port* yang dimaksud akan terbuka untuk klien tersebut.



Gambar 1. Mekasime otorisasi port pada port knocking
Submber : Meningkatkan Keamanan Port Ssh Dengan Metode Port Knocking Menggunakan Shorewall Pada Sistem Operasi Linux [25]

Port knocking dapat diilustrasikan pada gambar 1. Bila klien ingin mengakses *service SSH* pada *server* melui *port* Y, maka klien harus terlebih dahulu mengirim paket *SYN* menujut *port* 1,2,3,4 agar dapat mengakses *port* Y. Namun metode *port knocking* masih memiliki beberapa kelemahan sebagi berikut

A. *Plain Text Port Sequence :* ketukan rahasia yang digunakan untuk validasi *user* mengadung *port number* dalam format *plain text*, sehingga penyerang akan sangat mudah mengetahui ketukan rahasia yang valid dengan melakukan *scanning* maupun *sniffing* [7].

B. *Network Address Translate (NAT) Knocking : Port Knocking* akan membuka *port* untuk klien yang telah melakukan *Knock Sequnece* yang valid. *Port Knocking* mengidentifikasi *Knock Sequnece* klien berdasarkan *network address client*. Masalah muncul ketika beberapa klien memiliki *network address* yang sama, contohnya pada kasus *Network Address Translation (NAT)*. ketika beberapa klien mengirimkan paket keluar, dimana klien-klien tersebut berada dalam *NAT* yang sama, maka semua paket tersebut akan menggunakan *source address* yang sama yaitu *Public address* dari *NAT* tesebut [8] [9].

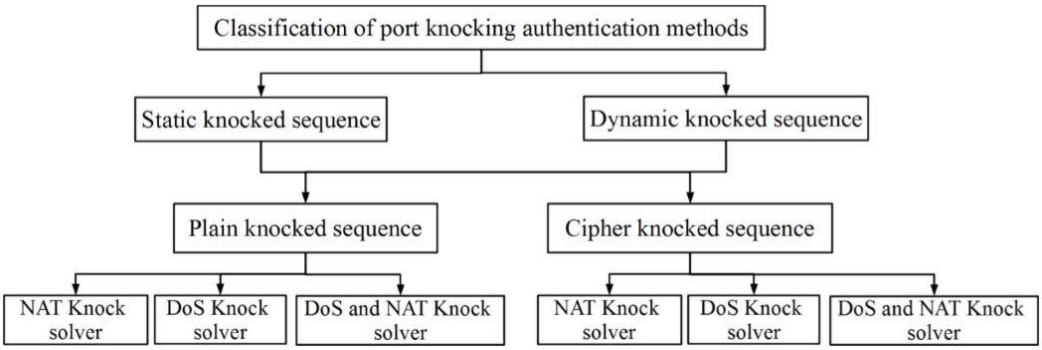
C. *Denial of Service (DoS) knocking : DOS-Knocking* terjadi ketika penyerang mengirim paket secara trus-menerus dengan *random fake network address* kepada *server*. *Server* harus mengalokasikan memori untuk mencatat *log* pengiriman paket untuk setiap *fake network address tersebut*. Hal ini menyebabkan meningkatnya penggunaan memori secara signifikan dan dapat mengakibatkan server *Overload* [10] [11].

D. *Appropriate port range : port range* akan mempengaruhi performa bila tidak ditentukan secara tepat. Jika *range* yang ditentukan terlalu pendek, maka *sniffer* akan mudah menemukan *knock sequence* dengan cara *mencapture traffic*. Sedangkan bila *range* yang dipilih terlalu panjang makan akan meningkatkan kemungkinan serangan *DOS Knocking*. penentuan *range* yang seimbang dapat mengurangi kemungkinan terjadinya serangan [12]

E. *Out of order packet delivery :* ketukan rahasia yang menuju *port* tertentu pada *server* digunakan sebagai validasi *user* yang sah. Masalah terjadi ketika berada pada *internet backbone router* yang memiliki *traffic* yang tinggi, kemungkinan paket tidak sampai pada tujuan sangat tinggi [13]

Metode autentikasi pada *port knocking* dibagi menjadi dua kategori yaitu *static* dan *dynamic knock sequence*. Metode *static knock sequence* menggunakan *port* yang telah didefinisikan atau ditetapkan sehingga klien akan melakukan ketukan sesuai dengan *port* yang telah didefinisikan tersebut. Sedangkan *dynamic knock sequence* menggunakan *randon generator modul*, baik pada klien maupun pada *server* untuk melakukan otentikasi [14]. Paket dan *key* dari algoritma *random generator* tersebut dikirim ke *server* lalu *server* akan melakukan validasi berdasarkan *key* tersebut dan akan mem-

bandingkanya dengan *sequence* yang telah *dibuffer* sebelumnya. Oleh karena itu *dynamic knock sequence* memberikan tingkat keamanan yang lebih tinggi dengan kerumitan yang lebih tinggi pula. Pengklasifikasian *port knocking* berdasarkan metode otentikasinya dapat dilihat pada gambar berikut :



Gambar 2. Pengklasifikasian port knocking berdasarkan metode otentikasi [7]

Metode *Static plain knock*

Metode ini menggunakan metode otentikasi *basic port knocking* dimana klien mengirimkan paket menuju *port* yang telah didefinisikan secara sekuensial. *Knock* module pada *server* akan mencatat *knock sequence* tersebut lalu membandingkannya dengan *knock sequence* yang telah ditetapkan, bila sama maka otentikasi sukses. Tidak ada proses enkripsi pada metode ini sehingga metode ini masuk dalam kategori *static plain knock*.

Metode *Static Cipher Knock*

Silent knock merupakan sebuah metode otentikasi *port knocking* yang mengkombinasikan antara *cryptography*, *steganography* dan *mutual authentication*. Teknik ini sangat bagus untuk mencegah *port scanning* dan *TCP Replay Attack*. Metode ini menyediakan otentikasi yang ringan dan meminimalisir terjadinya *overhead* saat komputasi[15].

Extension of the port-knocking client-server architecture with NTP, menggunakan *one time knocking* berdasarkan dua *improvement*, pertama *Network time protocol* (NTP) yang akan mensingkronasikan antara *knocker* (klien) dan *knock demon* menggunakan jam. Kedua menggunakan *one-way hash function* untuk memastikan adanya *secret knock sequence* (ketukan rahasia) [16].

Secure Port-knock-Tunneling (SPKT), menggunakan dua otentikasi, pertama menggunakan *text passphrases* pada setiap ketukannya. Penggunaan *text passphrases* ini dapat mencegah serangan *DOS Knocking*. Kedua mengunakan *tunneling VPN* untuk mengakses *service* yang dituju setelah proses *port knocking* selesai. Penggunaan *tunneling* ini dapat mencegah dari serangan *NAT Knock Attack* [1].

The simple port-knocking method against a TCP replay and Port Scanning. Metode ini tidak menggunakan *firewall* untuk mekanisme otentikasinya, menyedikan sebuah metode yang lebih ringan dari *port knocking* standar namun di sisi lain lebih aman. Metode ini menggunakan *source port* sebagai ketukan rahasia dan bertujuan untuk melakukan *trigger start* dan *stop service* yang dituju, bukan untuk membuka atau menutup *port service* yang dituju. Klien akan mengirim paket mengandung *source port* menuju *server*, lalu *server* akan melakukan validasi. Jika *source port* valid, *server* akan melakukan *trigger start* pada *serve SSH* dan klien akan melakukan koneksi ke *service* tersebut sesuai *port* yang telah ditentukan. Jika *source port* tidak valid, maka paket akan langsung di *drop*[17].

Table dibawah ini menunjukan *static cipher knock* terhadap pencegahan *NAT* dan *DOS Knocking attacks*

TABLE I.
KEAMANAN PORT KNOCKING BERDASARKAN SERANGAN NAT DAN DOS KNOCKING [7]

The static-cipher knock approaches in case of overcoming NAT and DoS knocking attacks		
DoS Knock solver	**NAT Knock solver**	**DoS and NAT Knock solver**
Simple port-knocking method against TCP replay attack and port scanning	Extension of a port-knocking client-server architecture with NTP	Secure Port-knock-Tunneling (SPKT) Silent knock

Metode *Dynamic Plain Knock Sequence*

Berdasarkan dari aplikasi dan tingkat keamanan yang dibutuhkan, *dynamic plain knock sequence* akan di kembangkan di masa depan. Sampai waktu itu tiba, semua *dynamic port knocking* menggunakan *Cipher knock* untuk meningkatkan keamanan mekanisme otentikasi.

Metode *Dynamic Cipher knock sequence*

Metode ini memungkinkan *server* untuk menjalankan otentikasi *port knocking* pada *port* yang berbeda-beda, dikarenakan otentikasi pada *port* yang tetap akan meningkatkan kemungkinan terjadinya serangan.

The advanced port-knocking authentication scheme with QRC using AES[18]. Penelitian ini melakukan pengacakan pada *soruce IP address* dan *port number* ketika proses *port knocking* terjadi berdasarkan *quadratic residue ciphers* (*QRC*) [19] [20]. *Knocker* akan mengirim SMS ke SMS *server* guna meminta *One Time Password* (*OTP*). *OTP* ini diciptakan berdasarkan *time factor* sehingga tidak akan terjadi duplikat *OTP* bahkan untuk *user ID* yang sama sehingga dapat melindungi sistem dari serangan *DOS Knock attack*. SMS *server* akan membalas dengan pesan (SMS) yang mengandung *timestamp* yang akan digunakan untuk otentikasi *OTP*. Bagian selanjutnya dari SMS berisi *One Time Key* dengan 256 *bits* yang akan digunakan oleh *advanced encryption standard* (*AES*) [21]. untuk melakukan enkripsi *Knock Sequence*. Dan bagian terahir adalah *Random number* "R" yang akan diciptakan oleh *pseudo random number generator* (*PRNG*) [22] yang akan digunakan sebagai *key* pada *QRC*.

The one time knocking framework using the SPA and IPsec[4]. Metode ini juga menggunakan SMS *server*. Klien mengirim SMS menuju *server*, lalu *server* akan mengidentifikasinya berdasarkan *user ID* yang sebelumnya sudah diregistrasikan pada *server*, hal ini dapat mencegah serangan *DOS Knocking*. *server* akan menciptakan *One Time Password* (*OTP*) yang akan kirim kembali ke klien menggunakan *Out of Band* (*OoB*) *channel* yang sama dengan yang digunakan klien. Pesan ini mengandung *time stamp random port* dan juga *OTP*. *Random port* dan *OTP* akan dikirim ke *SPA user* sebagai *input* untuk *Key Derivation Function* (*KDF*) [23]. Fungsi ini akan mempersiapkan empat fungsi yaitu *k1,k2,k3,k4* yang akan digunakan untuk enkripsi data, kalkulasi *MAC* untuk *SPA*, dan 2 *key* terahir digunakan untuk *IPsec VPN connection* [24]

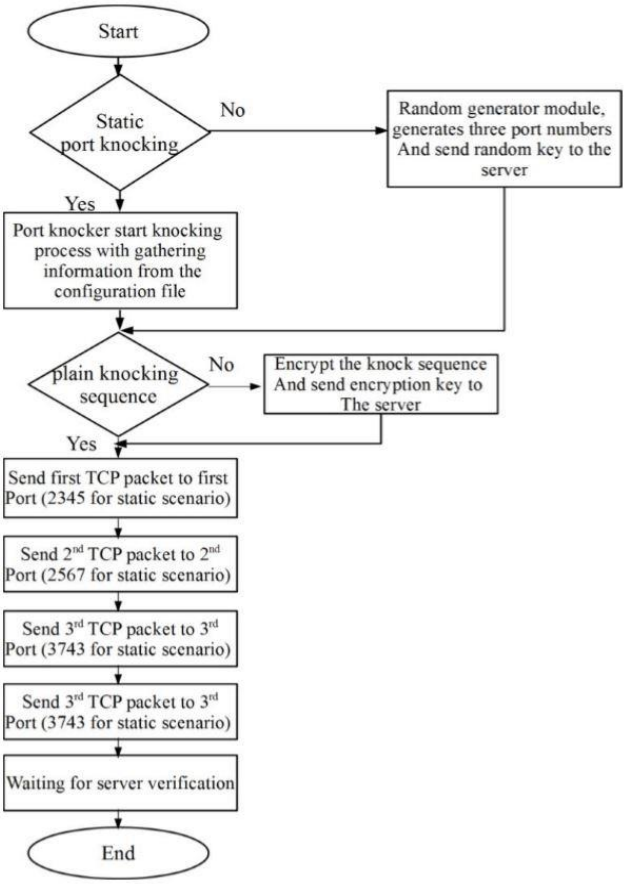
Network security using hybrid port-knocking [26]. Metode ini memperkenalkan *port knocking* menggunakan *hybrid* dari *cryptography*, *steganography* and *mutual authentication*. Teknik ini sangat bagus untuk mengatasi serangan *port scanning* dan *TCP Replay Atack*. Pada metode *hybrid knock* ini terdapat enam tahapan. *Traffic monitoring* dan *traffic capturing* merupakan dua tahap awal dan sisanya adalah tahapan dari *port knocking* itu sendiri. Tahap ketiga adalah *image processing* guna mengidentifikasi *knocker* yang berupa *payload* pada setiap paketnya mengunakan *stenography*. Tahap keempat dan kelima adalah otorisasis klien dan otentikasi *server* dimana *server* menggunakan *random encrypted number* untuk *payload* dan dikirim kembali ke klien. Yang terahir adalah menutup *port*.

Tabel dibawah ini menunjukan *Dynamic cipher knock* terhadap pencegahan *NAT* dan *DOS Knocking attacks*.

TABEL II.
KEAMANAN DYNAMIC CIPHER KNOCK TERHADAP PENCEGAHAN NAT DAN DOS KNOCKING ATTACKS [7]

The dynamic-cipher knock approaches in case of overcoming NAT and DoS knocking attacks	
DoS Knock solver	**DoS and NAT Knock solver**
Dynamic knocking and forwarding port SecureID integration in EFDA Advanced port knocking authentication scheme	Hybrid port knocking One time knocking framework using SPA and IPSec

Sedangkan *flowchart* proses otorisasi *port knocking* dapat dilihat pada gambar dibawah ini



Gambar 3. Flowchart otorisasi port knocking [7]

Knock module pada *server* akan memantau *port* yang telah dialokasikan untuk *port knocking* dan akan mengalokasikan memori untuk menganalisa semua *trafik* yang datang. Jika klien telah menyelesaikan ketukan, maka *server* akan membandingkannya dengan ketukan rahasia yang sudan ditentukan. Skenario ini untuk *static knocking*. sedangkan untuk *dynamic knocking*, ketukan rahasia diciptakan berdasarkan dari *random generator key*. Setelah itu *server* akan mengirimkan paket *Acknowledgement* kepada klien yang sah dan membuka *port* 22 untuk *service SSH*.

Perbandingan beberapa metode otentikasi *port knocking* dapat dilihat pada tabel dibawah ini

TABEL III.
PERBANDINGAN BEBERAPA METODE OTENTIKASI PORT KNOCKING

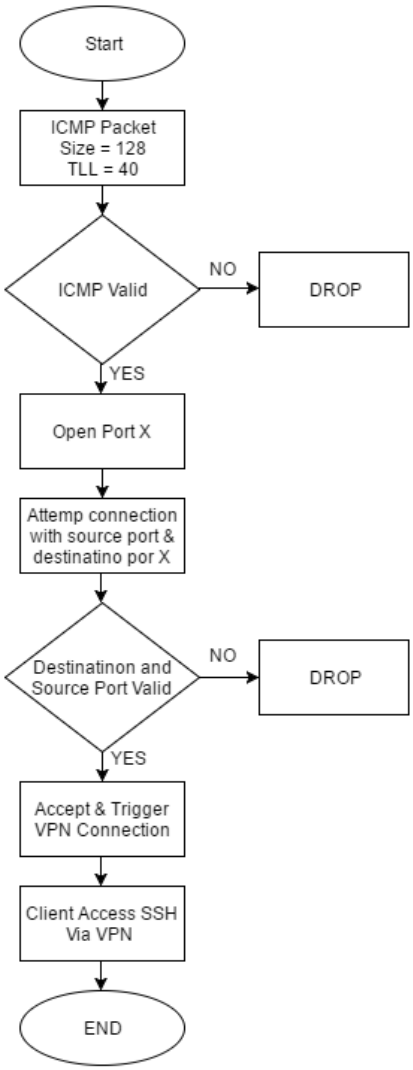
Judul	Platform OS yang Didukung	Protokol	Kelemahan Otentikasi Port knocking				Dependency Ketiga
			Plain Text Ports	NAT Attack	DoS Attack	Out of Order Packet Delivery	
Remote Server Management using Dynamic Port-knocking and Forwarding	Linux & Win32	UDP & TCP	Yes	No	Yes	No	No
Silent knock: Practical, provably undetectable authentication	Linux	TCP	Yes	Yes	Yes	No	No
Network security using hybrid portknocking	Linux	TCP	Yes	Yes	Yes	No	No
One-time knocking framework using SPA and IPSec	NA	NA	Yes	Yes	Yes	No	Yes menggunakan SMS Server
Securing remote services integrating secure ID strong Authentication technology in EFDAfederation Infrastructure	Linux	NA	Yes	No	Yes	No	Yes menggunakan PoA

Tabel diatas memaparkan perbandingan beberapa metode *Port Knocking* dengan beberapa parameter seperti sistem operasi yang didukung, Protokol yang digunakan, *plain text ports*, *NAT Attack*, *DOS Attack*, *out of order paket delivery* dan juga *dependency* ketiga.

II. METODE PENELITIAN

Metode *port knocking* pada penelitian ini akan menggabungkan beberapa metode. Guna mencipatakn sebuah metode *port knocking* yang lebih ringan sederhana namun tetap aman. Metode *port knocking* pada penelitian ini terdiri dari tiga tahap dalam proses otorisasi *user*.

Tahap pertama adalah menggunakan *special features* pada *ICMP* untuk membuka *port* tertentu yang telah didefinisikan sebelumya, contohnya *port* X. Tahap kedua menggunakan *source port* sebagai ketukan rahasia, *source port* ini dikirim menuju *server* melalui *port* X yang telah terbuka pada tahap pertama. Dan tahap ketiga adalah *server* akan mealkukan validasi terhadap *soruce port* yang diterima, jika valid maka *server* akan melakukan *trigger* koneksi *VPN* untuk mengakses *service* yang dinginkan misalnya *service SSH*. Kinerja *port knocking* pada penelitian ini dapat dilihat pada *flowchart* di bawah ini



Gambar 4. Flowchart metode port knocking yang akan dikembangakan

Penelitian ini dilakukan dengan cara merancang metode *port knocking* seperti yang telah dijelaskan sebelumnya. setelah itu akan melakukan tes ketahanan dari beberapa serangan seperti *port scanning, replay attack, NAT knocking, dan DOS Knocking*. Setelah itu akan membandingkan kinerja dengan beberapa metode sebelumnya.

III. HASIL

Penelitian ini bertujuan untuk merancang sebuah metode *port knocking* yang lebih sederhana dan riangan namun tetap aman dari beberpa serangan yang belakangan ini dapat menyerang *port knocking*

seperti yang telah dipaparkan pada penelitian penelitian sebelunya. Hasil dari penelitian ini sebagai berikut

A. Mencegah DOS Knocking

Diharapkan dengan menggunakan *special feature* pada *ICMP* dan juga penggunaan *source port* akan dapat mencegah dari serangan *DOS Knocing*. *Server* akan melakukan validasi pada paket *ICMP* yang dikirimkan, bila valid maka *port* akan terbuka, bila tidak maka paket akan di "*drop*" sehingga *server* tidak perlu mengalokasikan memori untuk mencatat semua paket yang datang.

B. Mencegah NAT Knocking

Penggunaan *tunneling VPN* akan mencegah dari serang *NAT Knocking*. *NAT Knocking* terjadi ketika beberapa klien memiliki *network address* yang sama, contohnya pada kasus *Network Address Translation* (*NAT*). ketika beberapa klien mengirimkan paket keluar, dimana klien-klien tersebut berada dalam *NAT* yang sama, maka semua paket tersebut akan menggunakan *source address* yang sama yaitu *Public address* dari *NAT* tesebut sehingga *server* tidak bisa membedakan *user* yang valid. Masalah ini dapat dicegah dengan menggunakan *tunneling VPN*, karena *service* tersebut hanya dapat diakses melalui *tunneling VPN*.

C. Lebih Ringan

Penelitian yang berjudul "*SPKT: Secure Port Knock-Tunneling, an enhanced port security authentication mechanism*" menggunakan *text-passphrases* pada setiap ketukannya, sedangakan pada penelitian ini akan menggunakan *soruce port* sebagai ketukan rahasianya. Diharapkan dengan menggunakan *source port* ini dapat membuat proses otorisasi menjadi lebih ringan.

IV. PEMBAHASAN

Penelitian yang berjudul "*Simple port knocking method: Against TCP replay attack and port scanning*" menggunakan *source port* sebagai pengganti ketukan rahasia, tapi metode ini masih rentah terhadap serangan *NAT Knocking*. sedangakan penelitian yang berjudul "*Utilizing Port-Knocking as first defensive layer at defense-in-depth strategies using hybrid of the Internet Control Message Protocol features, Internet Addresses and Tunneling*" menggunakan *special feature* pada ICMP dan juga penambahan *text-passphrases* pada setiap ketukannya. Di samping itu juga penelitian ini menggunakan *tunneling VPN* untuk mengakses *service* yang dituju.

Pada penelitian ini akan mengkombinasikan penggunaan *special feature* pada *ICMP* dengan penggunaan *source port* sehingga terdapat dua lapis *port knocking* guna mencegah serangan DOS *Knocking*. bukan hanya itu, metode *port knocking* pada penelitian ini juga menggunakan *tunneling VPN* untuk mengakses *service* yang dituju guna mencegah serangan *NAT Knocking*.

V. SIMPULAN DAN SARAN

Hal pertama yang dilakukan penyerang dalam melakukan aksi penyerangan adalah mengumpulkan informasi tentang korban sebanyak banyaknya, Guna mengetahui celah yang terbuka, *service* yang berjalan, dan kelemahan pada sistem. Karena itulah dibutuhkan *port knocking* guna mencegah penyerang melakukan observasi terhadap *server*. Namun *port knocking* itu sendiri masih rentan terhadap beberapa serangan. Karena itulah masih dibutuhkan inovasi baru pada *port knocking* agar lebih aman akan tetapi di lain sisi lebih ringan dan sederhana.

REFERENSI

[1] Mehran, P., E.A. Reza, and B. Laleh. SPKT: Secure Port Knock-Tunneling, an enhanced port security authentication mechanism. in Computers & Informatics (ISCI), 2012

[2] A. Narayanan, "A critique of port knocking", Linux Journal , 2004

[3] M. Rash, "Protecting SSH Servers with Single Packet Authorization", The Linux Journal, vol. 2007 issue no. 157, 2007.

[4] Jiun-Hau, L., et al. One-Time Knocking framework using SPA and IPsec. in Education Technology and Computer (ICETC), 2010.

[5] Nurika, O., et al. Review of various firewall deployment models. 2012

[6] Support, R. Port Knocking. 2013 January 11, 2013; Available from: http://www.rackspace.com/knowledge_center/article/port-knocking.

[7] Laleh Boroumand, Muhammad shiraz, Abdulla gani, Suleman Khan, Syed Adeel ali shah, "A Review on Port knocking Authentication Methods for Mobile cloud computing", Journal of Centre for mobile cloud computing research (C4MCCR), Vol.4, Issue.2, October 2013

[8] T. Popeea, V. Olteanu, L. Gheorghe, R. Rughinis, "Extension of a port knocking klien-server architecture with NTP synchronization," 10th Roedunet International Conference (RoEduNet), 2011, pp. 1 - 5.

[9] S. Jeanquier, "An Analysis of Port Knocking and Single Packet," MSc Thesis, Information Security Group, Royal Holloway College, University of London, 2006.

[10] A. I. Manzanares, J. T. Marquez, J. M. Estevez-Tapiador, J. Cesar Hern´andez Castro, "Attacks on port knocking authentication mechanism,", Computational Science and Its Application, ICCSA 2005, pp. 1292-1300.

[11] Prowell, S., R. Kraus, and M. Borkin, Chapter 1 - Denial of Service, in Seven Deadliest Network Attacks 2010, Syngress: Boston. pp. 1-21.

[12] Doyle, M., IMPLEMENTING A PORT KNOCKING SYSTEM IN C, in J. William Fulbright College of Arts and Sciences 2004, Arkansas.

[13] Jiun-Hau, L., et al. One-Time Knocking framework using SPA and IPsec. in Education Technology and Computer (ICETC), 2010.

[14] D., I. Port Knocking: Beyond the Basics. 2005

[15] Vasserman, E., N. Hopper, and J. Tyra, SilentKnock: practical, provably undetectable authentication. International Journal of Information Security, 2009. 8(2): pp. 121-135.

[16] Popeea, T., et al. Extension of a port knocking klien-server architecture with NTP synchronization. in Roedunet International Conference (RoEduNet), 10th. 2011.

[17] Ali, F.H.M., R. Yunos, and M.A.M. Alias. Simple port knocking method: Against TCP replay attack and port scanning. in Cyber Security, Cyber Warfare and Digital Forensic (CyberSec), 2012.

[18] Srivastava, V., et al. Advanced port knocking authentication scheme with QRC using AES. in Emerging Trends in Networks and Computer Communications (ETNCC), 2011.

[19] Zhu, H., et al. A new chaos-based image encryption scheme using quadratic residue. 2012.

[20] Chapter 2 Stream ciphers, in North-Holland Mathematical Library, Thomas W. Cusick, Cunsheng Ding, and R. Ari, Editors. 2004, Elsevier. pp. 11-43.

[21] Kulikowski, K.J., M.G. Karpovsky, and A. Taubin, Robust codes and robust, fault-tolerant architectures of the Advanced Encryption Standard. Journal of Systems Architecture, 2007. 53(2– 3): pp. 139-149.

[22] Karimi, H., S. Morteza Hosseini, and M. Vafaei Jahan, On the combination of self-organized systems to generate pseudo-random numbers. Information Sciences, 2013. 221(0): pp. 371-388.

[23] Smits, R., et al., BridgeSPA: improving Tor bridges with single packet authorization, in Proceedings of the 10th annual ACM workshop on Privacy in the electronic society2011, ACM: Chicago, Illinois, USA. pp. 93-102.

[24] Khan, Z.A., et al. Performance Evaluation of Widely Used Portknoking Algorithms. in High Performance Computing and Communication & IEEE 9th International Conference on Embedded Software and Systems (HPCC-ICESS), 2012.

[25] Haryanto. Edy, 2013 "Meningkatkan keamanan port ssh dengan metode port knocking menggunakan shorewall pada sistem operasi linux" Yogyakarta, STMIK Amikom Yogyakarta

[26] Al-Bahadili H., H.A., Network security using hybrid port knocking. International Journal Of Computer Science and Network Security (IJCSNS), 2010.