STMIK Amik Riau

SATIN – Sains dan Teknologi Informasi

journal homepage : http://jurnal.stmik-amik-riau.ac.id

Aplikasi Pengendalian Port dengan Utilitas Port Knocking untuk Optimalisasi Sistem Keamanan Jaringan Komputer

Rometdo Muzawi
Jurusan Manajemen Informatika, STMIK Amik Riau
rometdomuzawi@stmik-amik-riau.ac.id

Abstrak

Port Knocking adalah salah satu sistem keamanan yang dapat melakukan fungsi yaitu mem-blok akses yang tidak diinginkan. Pada prinsipnya, port knocking bekerja menutup seluruh port yang ada di server. Bila user menginginkan akses ke server, user melakukan "ketukan" untuk menggunakan layanan, kemudian bila user telah selesai melakukan akses maka port ditutup kembali. Tujuan yang ingin dicapai dari proyek akhir ini ialah server berhasil melindungi layanan yang ada (disini berupa file server) dengan mengintegrasikan aturan firewall dengan program port knocking yang digunakan, dan menunjukkan bahwa tanpa mengirimkan ketukan yang tepat, user (client) tidak dapat menggunakan layanan pada server.

Kata Kunci : Server, Client, Firewall, Program Port Knocking, Port Knocking

1. Pendahuluan

Seiring dengan pesatnya perkembangan dunia teknologi di segala aspek dan bidang kehidupan. Salah satu bidang pengembangannya adalah pada bidang jaringan komputer (*networking*). Berbagai riset telah dilakukan untuk dapat mengkomunikasikan antara satu perangkat dengan perangkat yang lain melalui perantaraan jaringan baik yang secara fisik kabel maupun tanpa kabel (*wireless*). Hal ini menjadikan keamanan suatu informasi sangatlah penting, Banyak nya serangan yang dilakukan melalui *port* yang dalam keadaan terbuka secara bebas juga menjadi salah satu ancaman bagi keamanan data dalam sistem jaringan komputer. sehingga nantinya akan membuat orang–orang yang tidak mempunyai hak akses maupun yang tidak berkepentingan dapat mengendalikan *port-port* yang telah ia masuki. Untuk itulah, *firewall* sangat dibutuhkan di dalam jaringan tersebut. *Firewall* memiliki tugas untuk melakukan pemblokiran terhadap *port-port* komunikasi yang terbuka bebas dalam sebuah jaringan komputer.

Dalam *firewall* semua komunikasi yang keluar dan masuk dikontrol. *Port* yang tidak penting dapat diblokir (ditutup) dan *port* yang penting dan berbahaya juga dapat diblokir, sehingga hanya pihak yang diijinkan saja yang boleh masuk melalui *port* tersebut. Cara ini merupakan sistem pengamanan jaringan komputer yang paling efektif dan banyak digunakan. Akan tetapi terkadang pemblokiran yang dilakukan sering menjadi tidak fleksibel, ketika dibutuhkan untuk menjalin komunikasi dengan apa yang ada di dalam jaringan, *firewall* tidak mengijinkannya karena mungkin memang berada pada area yang tidak diijinkan. Padahal komunikasi yang ingin dilakukan sangatlah penting untuk kelancaran kerja. Misalkan Saja melakukan koneksi dengan internet dan butuh mengakses *web server* melalui SSH untuk memperbaiki konfigurasinya, sementara *port* SSH pada *server* tersebut dilarang untuk diakses dari internet oleh *firewall*, tentu hal ini akan sangat merepotkan. Untuk menghindari hal semacam ini, ada suatu metode yang sangat efektif yaitu dengan menggunakan metode *port knocking*.

Port Knocking adalah suatu metode komunikasi 2 arah yaitu (*client* dan *server*) dimana metode ini diterapkan dalam suatu sistem dengan port tertutup (Krzywinski, 2003). Pada dasarnya cara kerja dari *port knocking* adalah menutup semua *port* yang ada, dan

hanya *user* tertentu saja yang dapat mengakses *port* yang telah ditentukan, yaitu dengan cara mengetuk terlebih dahulu. Berbeda dengan *Firewall*, cara kerja dari *Firewall* adalah menutup semua *port* tanpa memperdulikan apapun meskipun *user* tersebut memiliki hak untuk mengakses *port* tersebut. Sehingga *user* yang memiliki hak akses tersebut juga tidak bisa untuk mengaksesnya. Kelebihan dari *Port Knocking* dengan *Firewall* adalah meskipun semua *port* yang ada telah ditutup, tetapi *user* yang memiliki hak akses dan mengetahui *Knocking* untuk membuka suatu port maka *user* tersebut tetap dapat menggunakan *port* yang telah ia buka.

2. Landasan Teori

2.1. Pengertian Keamanan Komputer

Keamanan Komputer adalah bagian dari ilmu komputer yang bertugas untuk mengontrol resiko yang berhubungan dengan penggunaan komputer (W & Sanjaya, 2008).Keamanan Komputer yang dimaksud adalah keamanan sebuah komputer yang terhubung ke dalam sebuah jaringan internet, dari akses yang tidak memiliki hak untuk mencoba masuk memperoleh informasi dan *service* tertentu yang ada di dalam sistem.

2.2 Port Knocking

Menurut (Saleh et al., 2014) , *Port Knocking* adalah sebuah metode sederhana untuk memberikan akses *remote* tanpa meninggalkan *port* dalam keadaan selalu terbuka. Hal ini akan memberikan perlindungan kepada *server* dari *port scanning* dan serangan *scripts kiddies*.

Port Knocking memiliki metode buka *port* kepada suatu klien bila klien itu meminta, dan tutup kembali bila klien telah selesai. Untuk menjalankan metode ini, sebuah *server* haruslah memiliki *firewall* dan daemon untuk menjalankan metode *port knocking* yang berjalan di *server* tersebut. Dengan metode tersebut, *user* dituntut untuk memberikan *autentikasi* ke *server* agar *firewall* menulis ulang *rulenya* sehingga *user* diberi izin untuk mengakses *port* yang dimaksud. Dan setelah selesai, *user* mengirimkan *autentikasi* kembali untuk menutup *port* agar *firewall* menghapus *rulenya* yang ditulis sebelumnya untuk membuka *port*.

Port Knocking adalah suatu metode komunikasi 2 arah yaitu (*client* dan *server*) di mana metode ini diterapkan dalam suatu sistem dengan port tertutup (Krzywinski, 2003).Pada dasarnya cara kerja dari *port knocking* adalah menutup semua *port* yang ada, dan hanya *user* tertentu saja yang dapat mengakses sebuah *port* yang telah ditentukan.

2.3 Format Ketukan Port Knocking

Format ketukan yang digunakan dalam perancangan sistem adalah format *port* tunggal dengan pemetaan tetap, dan hanya menggunakan tiga *port* ketukan sebagai tujuan pengiriman paket data untuk melakukan ketukan.

Untuk mempermudah penentuan *port* ketukan, maka dibuat aturan pemilihan *port* ketukan sesuai dengan nomor *port* tujuan, digit terakhir pada nomor *port* ketukan merujuk pada nomor *port* tujuan. Sebagai contoh, seorang *user* ingin mengakses *port* 22, dengan range *port* ketukan yang telah ditentukan yaitu antara *port* 2000 sampai dengan 4000, maka pemilihan *port* ketukan yang digunakan adalah seperti pada Gambar 1.

Nomor Port Ketukan	Port Tujuan
2000+a,3000+b,4000+c	abc

Gambar 1. Penentuan format ketukan

Nomor *port* **2000+a, 3000+b, 4000+c** merupakan nomor *port* tujuan pengiriman paket data yang berfungsi sebagai *port* ketukan. Nomor *port* ketukan menunjukkan *port* tujuan abc yang akan dibuka atau ditutup. Maka ketukan yang dilakukan oleh pengguna jika ingin membuka 22 adalah seperti pada Gambar 2.

2000,3002,4002

Gambar 2. Format ketukan membuka port 22

Sedangkan jika user ingin menutup port 22, maka ketukan yang dilakukan oleh pengguna adalah seperti pada Gambar 3.

3002,4002,2000

Gambar 3. Format ketukan menutup port 22

2.4 Mekanisme Port Knocking

Menurut (Krzywinski, 2003), pola kerja atau mekanisme metode *port knocking* ini memiliki beberapa tahap, sebagai berikut:

1. Pada tahap pertama, klien melakukan koneksi ke komputer *server* ke salah satu *port* di komputer *server*, misal *port* 22, namun koneksi tersebut di blok oleh *firewall* komputer *server*
2. Ditahap kedua, klien melakukan koneksi ke *port – port* sequences yang telah didefinisikan dalam *file* konfigurasi daemon *port knocking* ke komputer *server* dengan mengirimkan paket SYN didalamnya. Selama fase ini, klien tidak akan mendapatkan respon apa – apa.
3. Pada tahap ketiga, Daemon *Port Knocking* mencatat adanya percobaan koneksi dan kemudian

melakukan *autentikasi* terhadap percobaan tersebut. Apabila *autentikasi* sesuai dengan yang didefinisikan pada daemon *port knocking* dalam hal ini adalah *port sequences* yang didefinisikan, maka daemon *port knocking* akan melakukan *overwrite* terhadap *rule* yang telah didefinisikan didalam *firewall* agar membuka *port* yang ingin dituju oleh klien.

4. Ditahap keempat, setelah melakukan *autentikasi* klien telah bisa melakukan koneksi ke *port* yang dituju menggunakan aplikasi seperti pada umumnya.
5. Setelah selesai, klien memutuskan koneksi dengan *port* dan kemudian mengirimkan paket SYN kembali agar daemon *port knocking* menulis ulang *rule* pada *firewall* agar tidak bisa dilakukan koneksi kembali ke *port* 22.

3. Metodologi Penelitian

Berdasarkan permasalahan yang terjadi, yaitu lemahnya sistem keamanan jaringan pada STMIK-Amik-Riau dan *firewall* yang tidak mampu membedakan *user* yang dapat dipercaya dengan menutup *port* yang ada tanpa membedakan *user* menjadi kelemahan dalam layanan akses. Sehingga dicari sebuah metode yang efektif dan mampu untuk menyelesaikan masalah yang terjadi STMIK-Amik-Riau menggunakan metode *Port Knocking*.

Uraian kerja akan dijelaskan di dalam kerangka penelitian yang nantinya akan menjelaskan prosedur maupun langkah-langkah yang akan dihadapi dalam membangun *Server*.

Kerangka Kerja

Kerangka penelitian yang dipakai dalam mengerjakan suatu penelitian dengan membuat tahapan metodologi penelitian untuk mengerjakan tesis sehingga tidak terjadi kerancuan selama pengerjaan dan hasil yang akan dicapai menjadi lebih maksimal, sesuai kerangka kerja pada gambar 4.

Gambar 4. Kerangka kerja

4. Analisa Dan Perancangan

4.1 Analisis Metode Port Knocking

Untuk mendapatkan keamanan yang baik dan kemampuan untuk mengizinkan *user* yang berhak untuk mengakses *server* maka diperlukan suatu metode yang memenuhi kedua kriteria tersebut. Salah satu metoda baru yang memiliki kemampuan untuk memenuhi kedua kriteria tersebut adalah *port knocking*.

Berdasarkan analisis masalah yang didapat, beberapa serangan terhadap *server* seperti *sniffing acttack*, *Port Scanning* merupakan penyalah gunaan *port-port* komunikasi yang terbuka secara bebas pada *server* sehingga keamanan *server* rentan terjadinya serangan oleh *hacker* maupun *cracker*. Pengimplementasian *port knocking* bertujuan untuk memberikan keamanan berlapis pada *server* dan memfilter *ip address* mana saja yang diperkenankan terkoneki terhadap *server*.

Untuk memenuhi kondisi yang baik seperti yang telah tersebut di atas, dilakukan beberapa pendekatan:

1. Perbaikan metode deteksi
   Metode memeriksa serangan berdasarkan *logs* untuk mendeteksi beberapa jenis serangan. Khususnya dengan serangan yang disebabkan *Port Scanning*.
2. Mengubah response
   Salah satu fitur yang dimiliki oleh *Port Knocking* adalah memberikan limit akses pada *user* saat melakukan koneksi dengan *server* dalam mengakses port yang diinginkan oleh user. Oleh karena itu, di samping memberikan *limit* akses kepada sistem *client*, penanganan secara otomatis masalah yang diakibatkan oleh serangan menjadi sangat penting. Integrasi perangkat keamanan seperti *firewall* kedalam sistem yang baru ini menjadi hal yang mutlak dilakukan. Dengan begitu, setiap gerakan yang dicurigai sebagai serangan akan dengan cepat ditanggulangi dengan cara mengubah *rule-rule* pada *firewall*.
3. Studi antar muka dan interaksi antara *client* dan *server* aplikasi.
   Konversi cara *setting* sistem ini dilakukan berbasis teks oleh administrator.
4. Studi nilai tambah
   Dengan adanya serangan yang telah terjadi, seharusnya dapat menjadi sebuah evaluasi bagi para sistem administrator dan *developer* dalam merancang server untuk masa yang akan datang. Segala hal berkaitan dengan tindakan penyerangan harus terdokumentasi dengan baik.

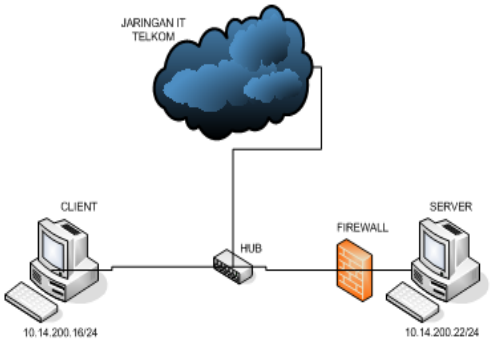
Selain itu kelebihan *port knocking* pada pengimplementasian *server* di antaranya:

1. Dimana komputer dapat *meremote* atauberkomunikasi dengan sebuah *server* melalui *port* yang tertutup.
2. Meskipun *port* ditutup, layanan yang disediakan tetap berjalan.

4.2 Perancangan Sistem

Pada peracangan sistem akan dijelaskan mengenai, perancangan simulasi dalam pengujian sistem. Perancangan dilakukan dengan menambahkan pengamanan *server* dengan penutupan semua *port* yang terbuka dengan penambahan metode *port knocking* untuk mengakses *server*.

.



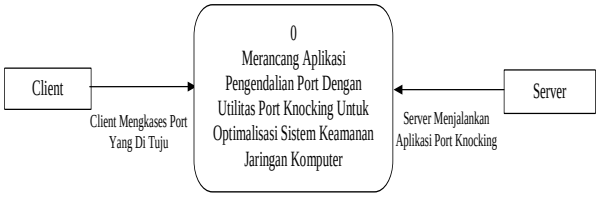
Gambar 5. Perancangan sistem

4.3 Desain Proses

Setelah tahapan analisa kebutuhan dan perancangan, maka akan diuraikan desain proses dalam membangun *PC Server*. Berikut ini alur proses yang digunakan dalam penelitian ini.

4.3.1 Context Diagram

CD memperlihatkan sistem yang di rancang secara keseluruhan, semua *external entity* harus digambarkan sedemikian rupa, sehingga terlihat data yang mengalir pada *input-proses-output*.

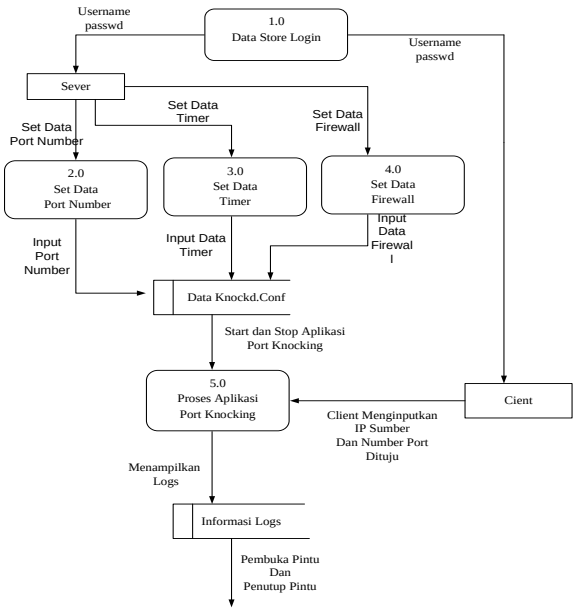


Gambar 6. Context diagram

Pada gambar 6 di atas menjabarkan proses *client* dan *server* dalam menjalakan atau menggunakan aplikasi *port knocking*. Di mana *server* menjalankan perintah untuk mengaktifkan apliksi *port* knocking, sedangkan *client* menjalankan perintah untuk mengakses port yang digunakan.

1.3.2 Data Flow Diagram Level 0

Data Flow Diagram (DFD) memberikan gambaran komponen-komponen dari sebuah sistem beserta aliran data. *Data Flow Diagram* (DFD) *level* 0 dapat dilihat pada gambar 7.

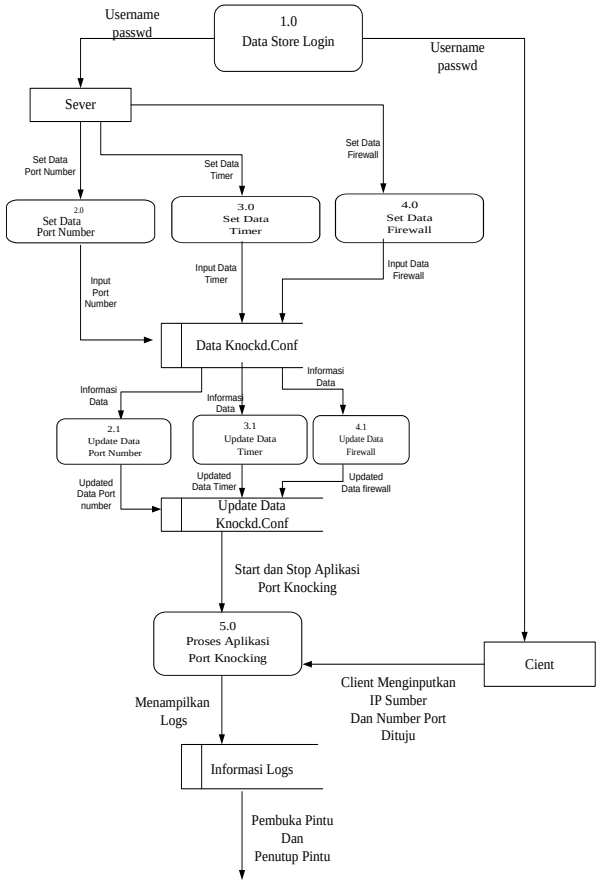


Gambar 7. DFD level 0

Pada gambar 7 di atas menjabarkan proses aplikasi *port knocking*.*Server* terlebih dahulu melakukan login kedalam sistem., kemudian *server* melakukan set data *port number*, *timer*, *firewall* didalam file *knockd.conf.* setelah data diset *server* menyimpan aplikasi *port knocking* dan menjalankan atau memberhentikan aplikasi *port knocking*.Client melakukan login kedalam sistem agar dapat menggunakan *port* yang ingin dituju. *Client* menjalankan perintah berdasarkan IP sumber dan *Port Number* untuk mengakses *port* yang digunakan dan *Logs* akan menampilkan informasi siapa yang masuk ke *port* yang dituju dan siapa yang menutup *port* yang dituju kedalam *server*.

1.3.3 Data Flow Diagram Level 1

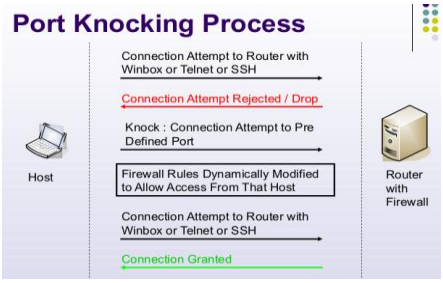
Data Flow Diagram (DFD) memberikan gambaran komponen-komponen dari sebuah sistem beserta aliran data. *Data Flow Diagram* (DFD) *level* 1 dapat dilihat pada gambar 8.



Gambar 8. DFD level 1

4.4 Model Proses Port Knocking

Tujuan utama *port knocking* ini adalah untuk mencegah dari *attacker* yang melakukan *scanning port* untk mencari informasi mengenai *port* yang terbuka pada *router/server*. Model Proses *Port Knocking* dapat dilihat pada gambar 9.



Gambar 9. Model proses port knocking

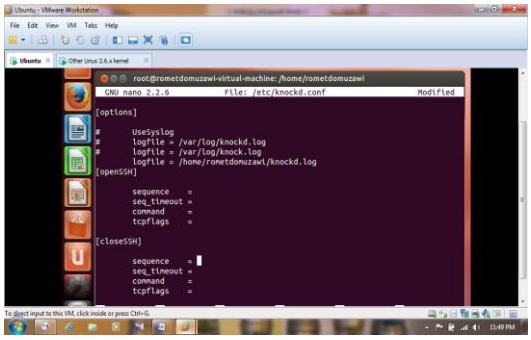
Gambar 9 di atas adalah sangat high-end pelabuhan SSH perspektif narasi proses mengetuk, jika tidak ada perubahan pelabuhan mengetuk ide, server SSH akan memungkinkan *port* SSH (22) terus mempertahankan keadaan terbuka, tetapi jika kita menggunakan pelabuhan teknologi mengetuk, klien harus mengirim satu set kombinasi *port* tetap (sinyal), dan kemudian menunggu *server* untuk mengkonfirmasi kombinasi *port* koneksi benar hanya setelah *port* SSH terbuka untuk membiarkan koneksi *client.*

5. Implementasi Dan Pengujian

5.1 Hasil Pengujian

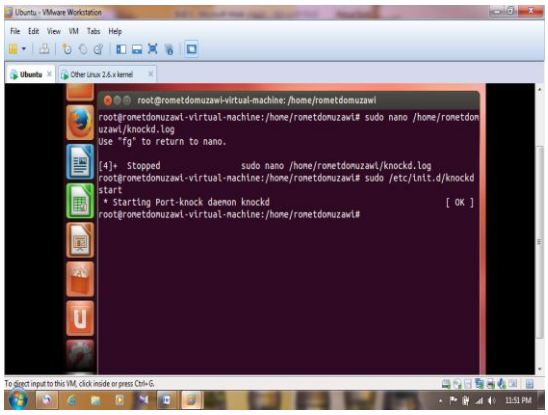
Pada bagian ini akan dijelaskan hasil yang diperoleh dari implementasi sistem terhadap *server*. Langkah langkah pengujian dengan langkah-langkah sebagai berikut :

1. Berikut ini adalah hasil sebelum mengisi file ***knockd.conf*** tersebut dengan beberapa baris teks yang ditampilkan dalam bentuk file.



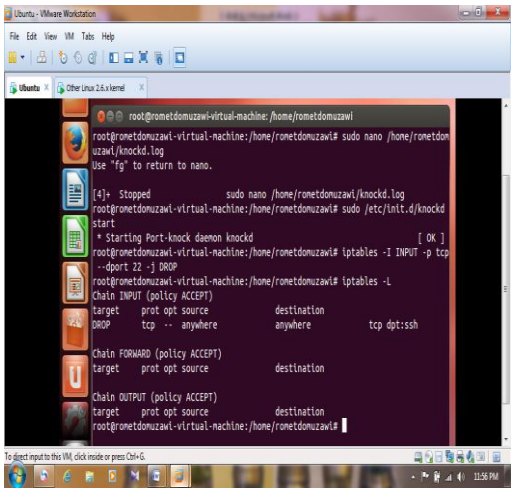
Gambar 10. File knockd yang kosong

2. *Server* Menjalankan file *knockd* pada komputer *server.*



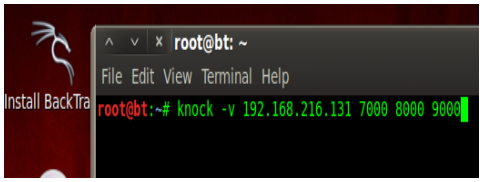
Gambar 11. File knockd pada server

3. Server Melakukan Penutupan Port 22 SSH (DROP).



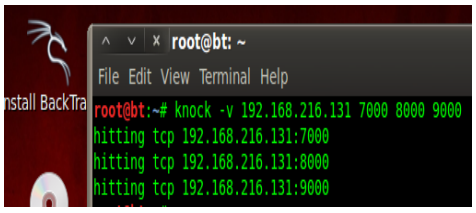
Gambar 12. File knockd pada server

4. *Client* menjalankan file *knockd* pada komputer *client* : **Knock –v (Ip Address Server) (Urutan Port Sesuai Role Untuk Membuka Akses SSH)**.



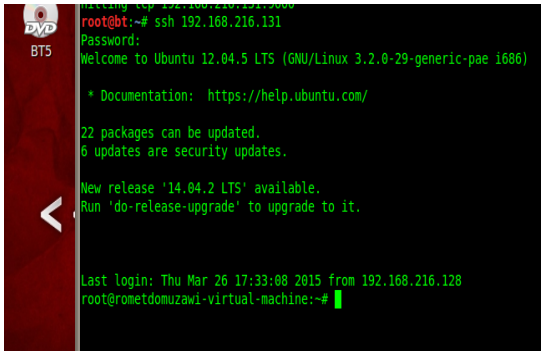
Gambar 13. File knockd pada client

5. *client* melakukan pengiriman paket urutan port 7000, 8000, 9000 pada komputer *server*, maka komputer *server* akan melakukan *monitoring port* apakah urutan *port* yang dikirim sesuai dengan format urutan *port openSSH.*



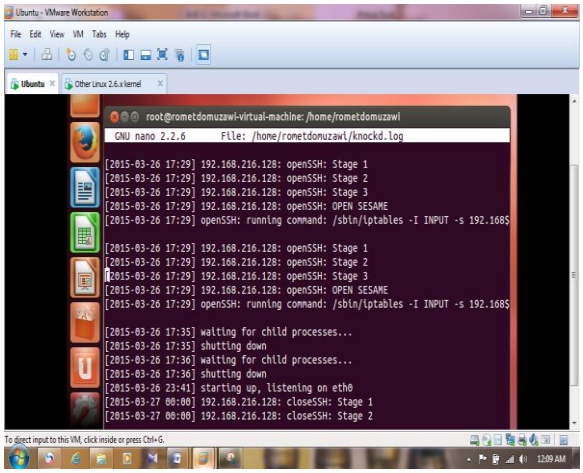
Gambar 14.Pengiriman urutan port

6. *Client* berhasil mencoba mengakses *port* 22 SSH.

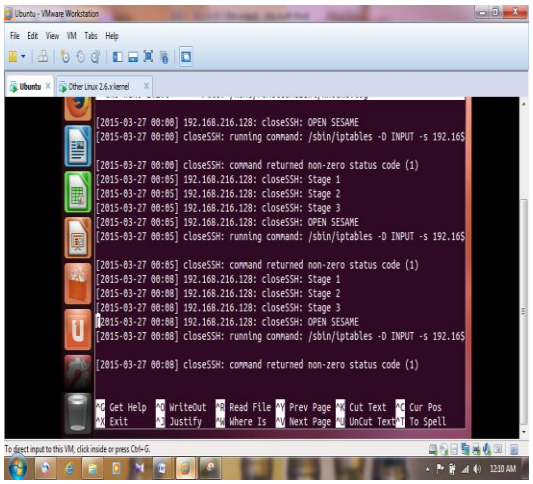


Gambar 15. Client mengakses port 22

7. Hasil tampilan *Logs* pada *Server* pada file ***/home/rometdomuzawi/knockd.log*** .



Gambar 16. Logs client membuka port ssh



Gambar 17. Logs client menutup port ssh

6. Simpulan

Berdasarkan analisa dan hasil dari aplikasi pengendalian *port* dengan utilitas *port knocking* telah berhasil diuji dan menghasilkan kondisi *port* yang terbuka untuk digunakan dengan fakta ini bisa disimpulkan bahwa optimalisasi sistem keamanan jaringan komputer bisa memanfaatkan aplikasi pengendalian *port* ini.

Kelemahan dari sistem yang dibuat masih dilakukan secara manual dengan memanfaatkan utilitas *port knocking* untuk membuka *port* yang telah diblok oleh sistem *firewall*. Saran Perbaikan adalah dengan membangun suatu aplikasi yang terintegrasi dengan utilitas *port knocking* untuk melakukan koneksi ke *port* yang diblok oleh *firewall*.

Referensi

Al-bahadili, H., & Hadi, A. H. (2010). Network Security Using Hybrid Port Knocking. *IJCSNS International Journal of Computer Science and Network Security*, *10*(8), 8–12.

Anggoro, R., Informatika, J. T., & Informasi, F. T. (2010). IMPLEMENTASI REMOTE SERVER MENGGUNAKAN METODE PORT KNOCKING DENGAN ASYMMETRIC ENCRYPTION.

Aswin, M., & Ub, T. E. (2010). Firewall, Client, Server, Port knocking, Iptables . *Jurnal TIF*, *1*, 22–37.

Boroumand, L., Shiraz, M., Gani, A., & Khokhar, R. (2014). Virtualization Technique for Port Knocking in Mobile Cloud Computing. *Ist.J.Advance.Soft.Comput.Appl*, *6*(1), 1–25.

Khan, Z. A., Javaid, N., Arshad, M. H., Bibi, A., & Qasim, B. (2012). Performance Evaluation of Widely used Portknoking Algorithms, 1–5.

Krzywinski, M. (2003). krzywinski-portknocking-sysadmin2003.pdf. Sys Admin the journal for UNIX system administrators.

Muhammad Denny Adisety. (2011). Port Knocking, *1*(611060074), 12.

Otp, O. P. (2014). INTERNATIONAL JOURNAL OF CURRENT LIFE SCIENCES. *International Journal of Current Life Sciences*, *4*(3), 504–511.

Prihanto, A., & Knocking, P. (2013). Implementasi Port-Knocking di Mikrotik dengan Menggunakan Komponen Delphi TcpClient. *Prosiding Seminar Teknik Elektro Dan Pendidikan Teknik Elektro*, (Ste), 533–538.

Sahu, M. P., Singh, M. M., & Deepak, P. (2013). A Result Analysis of Modified Hybrid Port Knocking (MHPK) with Strong Authentication. *International Journal of Scientific & Engineering Research*, *4*(5), 2091–2097.

Sahu, P., Singh, M., & Kulhare, D. (2013). Implementation of Modified Hybrid Port Knocking (MHPK) with Strong Authentication. *International Journal of Computer Applications*, *64*(22), 31–36.

Saleh, M., Fajri, H., Suhatman, R., Putra, Y. E., Studi, P., Informatika, T., … Caltex, P. (2014). Analisa Port Knocking Pada Sistem Operasi Linux Ubuntu Server, *2*(1), 59–67.

Sembiring, I., Widiasari, I. R., Prasetyo, S. D., & Diponegoro, J. (2009). Analisa dan Implementasi Sistem Keamanan Jaringan Komputer dengan Iptables sebagai Firewall Menggunakan Metode Port Knocking. *Jurnal Informatika*, *5*, 15.

W, T. W., & Sanjaya, A. (2008). Studi sistem keamanan komputer. *Jurnal Artificial ,ICT Research Center UNAS*, *2*(2), 70–77.

Yewale, M. P. R. (2014). A Modified Hybrid Port Knocking Technique for Host Authentication : A Review. *IJRITCC International Journal on Recent and Innovation Trends in Computing and Communication*, *2*(3), 673–677.

Zorkta, H., & Almutlaq, B. (2012). Harden Single Packet Authentication (HSPA). *International Journal of Computer Theory and Engineering*, *4*(5), 5.