IMPLEMENTASI *FIREWALL* DAN *PORT KNOCKING* SEBAGAI KEAMANAN DATA TRANSFER PADA FTP *SERVER* BERBASISKAN LINUX UBUNTU *SERVER*

Shah Khadafi[1)], S. Nurmuslimah[2)], Florian Kelvianto Anggakusuma[3)]
[1),2), 3)]*Institut Teknologi Adhi Tama Surabaya (ITATS)*
Jl. Arief rachman Hakim 100Surabaya, 60117
E-mail: *khadafi@itats.ac.id*[1)], *s.nurmuslimah@itats.ac.id*[2)], *floriankelvianto@gmail.com*[3)]

Abstrak

FTP (File Transfer Protocol) server menyediakan service file transfer antar mesin komputer dalam sebuah network. FTP protokol level aplikasi dalam OSI yang digunakan sebagai standar proses file transfer. Inisialisasi transfer FTP pada port nomor 21 menggunakan port TCP (Transmission Control Protocol) sebagai komunikasi data komputer client dan server. Port 21 yang aktif membuka layanan file transfer antara komputer client dan server. Ketika client melakukan pertukaran data harus melakukan koneksi ke port TCP nomor 21, setelah server mengijinkan maka terbentuk koneksi baru melalui port TCP sebagai jalur pertukaran data baik upload maupun download. Server FTP merupakan target peretas dikarenakan portnya yang selalu aktif dan terbuka. Terbukanya port 21, peretas dapat melakukan scanning port FTP yang digunakan untuk mengetahui nomor port FTP. Selanjutnya peretas melakukan sniffing untuk mencuri informasi username dan password, sehingga peretas dapat masuk ke FTP server yang mengakibatkan hilangnya data dalam FTP server. Solusinya yaitu menggunakan firewall untuk menutup seluruh port dengan memberikan hak akses client yang dapat mengakses server, penggunaan port knocking mengharuskan client melakukan otentifikasi sebelum menggunakan layanan FTP. Hasil dari pengujian, dengan mengaktifkan firewall membuat peretas tidak dapat mengetahui port berapa yang aktif. Menggunakan sistem otenfitikasi port knocking dapat melindungi hak akses penggunaan layanan FTP.

Kata kunci : *ftp, file transfer, firewall, port knocking.*

Abstract

FTP (File Transfer Protocol) server provides a file transfer services between computer machines in a network. FTP application level protocols in OSI are used as standard file transfer processes. Initialization of FTP transfers on port number 21 uses the TCP (Transmission Control Protocol) port as data communication for client and server computers. Port 21 which actively opens file transfer services between client and server computers. When the client wants to exchange data, it must connect to port TCP number 21, When the client exchanges data, it must connect to port TCP number 21, after the server allows to establish a new connection through the TCP port as a data exchange path both upload and download. The FTP server is the target of hackers because the ports are always active and open. The opening of port 21, hackers can scan the FTP port that is used to determine the FTP port number. Furtermore hackers do sniffing to steal username and password information, so hackers can enter the FTP server which results in loss of data in the FTP server. The solution is to use a firewall to close all ports by giving client access rights that can access the server, the use of port knocking requires the client to authenticate before using the FTP service. The results of the testing, by enabling the firewall makes a hacker can not figure out what ports are active. Using the port knocking authentication system can protect access rights to use FTP services.

Keywords : *ftp, file transfer, firewall, port knocking*

1. PENDAHULUAN

Salah satu akses pertukaran data melalui jaringan internet yang paling awal dikembangkan dan masih digunakan hingga saat ini adalah FTP (*file transfer protocol*), dimana dengan *protocol* ini *user* dapat melakukan pertukaran data baik pengunduhan (*download*) dan pengunggahan (*upload*). FTP merupakan protokol level aplikasi dalam standarisasi protokol OSI yang digunakan untuk standar proses *file transfer*. Dalam menjalankan fungsinya, inisialisasi transfer FTP yaitu pada port nomor 21 yang menggunakan *port* TCP (*transmission control protocol*) sebagai komunikasi data komputer *client* dan *server*. *Port* TCP nomor 21 pada

komputer *server* digunakan sebagai *port* pengatur. Sedangkan port TCP nomor 20 akan terbuka sebagai koneksi transfer data. Dalam keadaan normal *port* untuk layanan FTP nomor 21 selalu terbuka walaupun sistemnya tidak digunakan untuk proses transfer data. Maka dalam keadaan inilah adanya kelemahan dalam sistem komputer yang mengakibatkan rentan terjadinya peretasan [1]. *Port* pengatur pada *Port* TCP nomor 21 ini akan tetap terbuka dan berjalan meskipun tidak digunakan. Hal inilah yang menjadi kelemahan dari protokol ini. Hal inilah merupakan target peretas dikarenakan *port* nomor 21 yang selalu aktif dan terbuka, sehingga peretas dapat melakukan scanning *port* FTP yang digunakan untuk mengetahui nomor *port* FTP yang digunakan dan selanjutnya peretas dapat melakukan *sniffing* untuk mencuri informasi *username* dan *password*. Besarnya manfaat layanan FTP inilah yang mengharuskan seorang admin sistem jaringan perlu merancang sistem keamanan terhadap FTP *server* untuk mencegah *authorized access* dari *user* yang tidak bertanggung jawab.

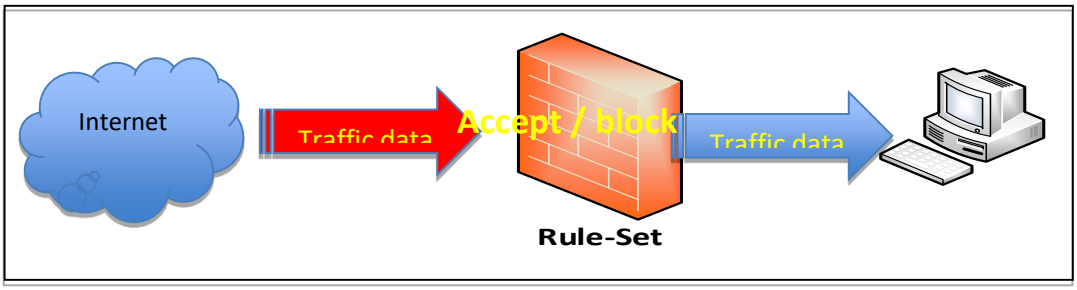
Rumusan masalah dalam penelitian yang dikembangkan ini, yaitu bagaimana mengimplementasikan sistem keamanan komputer FTP *server* yang dapat digunakan untuk *unauthorized access* menggunakan rules *firewall* yang dikombinasikan dengan metode *port knocking*. Tujuan penelitian yang dikembangkan ini, yaitu mengimplementasikan aturan-aturan atau *rules firewall* sebagai blokir terhadap komputer yang tidak diizinkan mengakses komputer FTP *server*. Dan juga menerapkan metode *port knocking* untuk *user* yang diizinkan mengakses komputer FTP *server*.

2. DASAR TEORI

a. Firewall

Firewall adalah sebuah sistem aplikasi di dalam sistem komputer yang berfungsi untuk melindungi komputer yang terkoneksi dalam jaringan komputer dari berbagai macam ancaman atau gangguan dari user yang tidak bertanggung jawab. Penggunan *firewall* merupakan suatu cara untuk memastikan informasi yang bersifat pribadi atau data yang terhubung dengan internet tidak dapat diakses oleh pihak yang tidak bersangkutan. Jika terdapat adanya percobaan akses oleh pihak yang tidak bersangkutan maka akan dilakukan pemblokiran oleh *firewall*. Sistem *firewall* dapat digunakan dalam berbagai entitas paket data jaringan komputer diantaranya yaitu, akses HTTP *page response*, email *download* / *upload responses time*, *packet delay*, *traffic drop*, *etc* [2].

Sistem kerja *firewall* dilakukan dengan *studying the deployment rule-set*[3]. *The rule-set* dirancang sedemikian rupa sehingga *firewall* harus memeriksa dan menganalisa setiap *traffic data* yang masuk melalui *rule-set*. Filter *firewall* mempunyai kebijakan yang dibuat (*rule-set*) untuk mengontrol *traffic data* yang masuk sebelum mengizinkan *traffic data* tersebut masuk. *Firewall* juga dapat memblokir *traffic data* serta melakukan pencatatan bilamana *traffic* data yang masuk berisikan paket data yang mencurigakan. *The Rule-set* untuk *firewall* tersebut kemudian dikonversi menjadi sintaks khusus untuk mesin *firewall* yang digunakan oleh perangkat komputer.



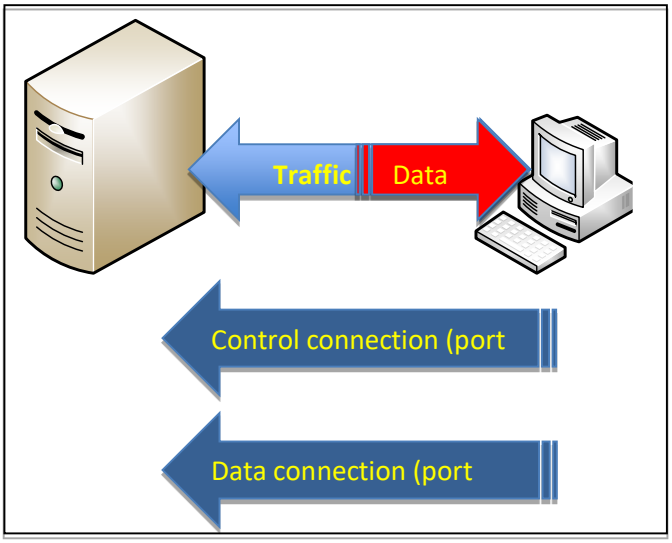
Gambar 1. Cara Kerja *Firewall*

b. FTP (*File Transfer Protocol*)

FTP adalah suatu *protocol* yang digunakan komputer untuk melakukan komunikasi ataupun pertukaran data atau *file* diantara komputer yang terkoneksi dengan jaringan. FTP juga merupakan lapisan *application layer protocol* yang umum digunakan untuk *transfer* data. FTP dikembangkan untuk memungkinkan transfer data diantara FTP *client* dan FTP *server*. FTP *server* adalah sebuah komputer *server* yang menyediakan layanan penyimpanan untuk *transfer* data atau file. Sedangkan FTP *client* adalah aplikasi yang berjalan di dalam komputer yang digunakan untuk *download* dan *upload* file dari FTP *server* yang menjalankan *daemon* FTP (FTPd). Untuk berhasil *transfer file*, FTP memerlukan dua koneksi antara *client* dan *server*, berikut ini teknik koneksi diantara *client* dan *server* menggunakan FTP:

- *Client* membuat koneksi pertama ke *server* pada port TCP nomer 21. Dimana, koneksi ini nantinya digunakan untuk mengontrol *traffic data*, yang berisikan perintah dari *client* dan balasan dari *server*.
- Selanjutnya, *client* membuat koneksi kedua ke *server* melalui port TCP nomer 20. Dimana, koneksi ini digunakan untuk *transfer* file aktual, dan koneksi ini dibuat setiap kali terdapat file yang ditransfer diantara *client* dan *server*.

Transfer data pada FTP dapat terjadi pada dua sisi, baik dari sisi *client* maupun *server*. *Client* dapat *download* (mengunduh) data dari komputer FTP *server* ataupun sebaliknya komputer *client* dapat *upload* (mengunggah) data ke FTP *server*. Dalam FTP dua kejadian tersebut dapat terjadi secara bersama-sama.



Gambar 2. Cara Kerja *FTP*

c. Serangan pada FTP *server*

Dikarenakan kehandalan FTP yang digunakan untuk *transfer* data, maka terdapat celah yang dapat digunakan oleh seorang *hacker* untuk dapat mengganggu sistem FTP. Diantara beberapa serangan yang terjadi pada FTP server yaitu *ping of death*, *packet sniffing*, dan *port scanning* [4].

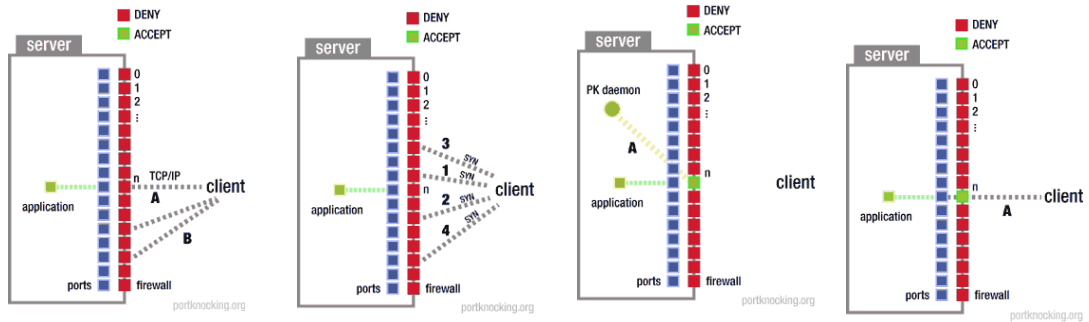
- *Ping of Death*: teknik penyerangan ini mengirim paket secara terus-menerus agar kinerja sistem pada target mengalami *hang* atau berhenti bekerja seperti penyerangan jenis *Denial of Service*. Serangan ini dilakukan dengan melakukan pengiriman paket ICMP sebesar mungkin dan secara terus-menerus melalui *command prompt*.

- *Packet Sniffing*: teknik pencurian data yang dilakukan dengan memonitoring atau melakukan analisis terhadap *packet* data yang ditransmisikan dari komputer client ke *server*. Serangan ini dilakukan *hacker* untuk melakukan tindakan yang dilarang seperti mencuri *password* dan pengambilan data-data penting lainnya

- *Port Scanning*: teknik penyerangan dengan melakukan *scanning* pada port untuk mengindentifikasi *port* apa saja yang terbuka dan mengenali sistem operasi target. *Port scanning* ini dapat dilakukan pada *port* TCP maupun UDP.

d. Metode *port knocking*

Port knocking merupakan sebuah metode otorisasi *user* berdasarkan *firewall* untuk melakukan komunikasi melalui *port* yang tertutup [5]. Metode *port knocking* menggunakan sistem *authentication* yang secara khusus dibuat untuk koneksi *client* dan *server*. Dalam hal ini metode *port knocking* ini digunakan untuk autenifikasi pengiriman informasi atau data, dimana informasi yang dikirimkan dikodekan (mungkin juga dienkripsi) ke dalam urutan nomor-nomor *port*. Urutan ini disebut dengan *knock* (ketukan). Sistem kerja dari *port knocking* dapat dijelaskan sesuai dari urutan-urutan gambar 3 [6]. Secara normal komputer *server* tidak menyajikan *port* terbuka terhadap *public network* dan server juga memantau semua upaya koneksi yang masuk ke dalamnya (gambar 3.a). Untuk memulai koneksi *client*, dimulai dengan upaya koneksi ke *server* dengan serangkaian urutan ke *well-defined set of ports*, yang juga mengirimkan paket SYN atau *synchronus* ke *port* yang ditentukan sebelumnya dalam ketukan atau disebut dengan *knock*. Proses *knock* inilah yang menyebabkan *port* mengetuk nomornya (gambar 3.b). Sedangkan yang terjadi di sisi *server*, yaitu *server* tidak memberikan respons terlebih dahulu kepada *client* selama fase *knock*, namun secara tidak diketahui dari sisi *server* menafsirkan yaitu *decodes* dan *decryptes* pada serangkaian urutan *port-port* untuk diperiksa otentifikasinya. Sehingga ketika *server* menerima dan menterjemahkan ketukan *well-defined set of ports* yang valid, selanjutnya *server* memicu terbentuknya koneksi antara *client* dan *server* melalui proses transfer data (gambar 3.d).



(a) (b) (c) (d)

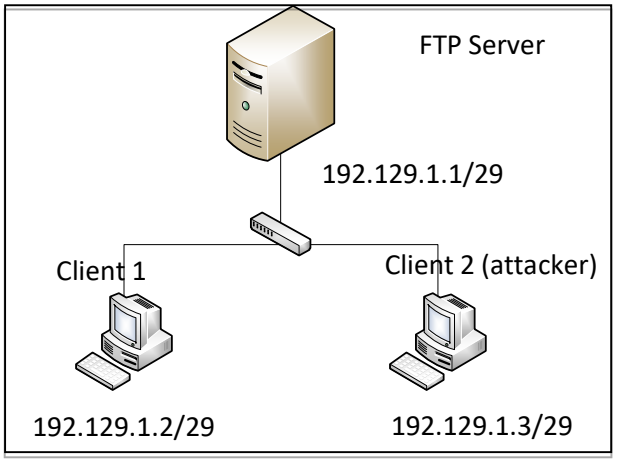
Gambar 3. Cara Kerja *Port Knocking*

3. METODOLOGI PENELITIAN

Pada penelitian ini akan dibuat sebuah sistem keamanan *transfer* data pada FTP *server* dengan menerapkan *firewall* dan metode *port knocking*. Terdapat 3 bagian untuk perencanaan untuk menyelesaikan permasalahan penelitian ini yaitu, yang pertama perencanaan topologi jaringan, rencana yang kedua perencanaan sistem keamanan *firewall* dan metode *port knocking*, dan yang terakhir perencanaan penyerangan terhadap sistem yang telah dibangun.

a. Topologi jaringan

Dari gambar 4, bahwa topologi yang digunakan terdiri dari 1 buah FTP server dan 2 client. Komputer FTP server ini digunakan sebagai penyedia file atau data yang nantinya dapat diakses oleh komputer client. Untuk komputer client 1 menggunakan sistem operasi Windows, dan komputer client 2 menggunakan sistem operasi Linux Desktop Ubuntu 16.04 LTS.



Gambar 4. Topologi Jaringan yang Dikembangkan

b. Implementasi sistem keamanan linux ubuntu server

Sistem keamanan yang dikembangkan dalam penelitian ini mencoba mengkombinasikan *rules-set* dari *firewall* dengan serangkaian urutan *well-defined set of ports* dari metode *port knocking*, dari dua teknik tersebut nantinya diimplementasikan di dalam komputer FTP *server* yang diinstall sistem operasi Linux Ubuntu 16.04 *Server*.

The rule-set dari *firewall* dalam sistem ini difungsikan untuk menutup semua *port* input pada komputer FTP *server* dan mengijinkan *port* tujuan terbuka kepada client dengan IP Address yang telah terautentikasi setelah terjadi perubahan file konfigurasi. File konfigurasi *firewall* pada Ubuntu 16.04 terletak pada file konfigurasi */etc/iptables*. *The rule-set firewall* untuk rancangan keamanan sistem yaitu:

- *Iptables –L INPUT DROP*
- *Iptables –A INPUT –m state –state ESTABLISHED, RELATED, -j ACCEPT*

Rangkaian urutan *well-defined set of ports* merupakan urutan *port* yang nantinya digunakan untuk autentikasi agar komputer *client* dapat mengakses layanan FTP port 21 pada komputer FTP *server*. *Port knocking daemon* mencatat percobaan koneksi yang dilakukan oleh *client*, dari sisi *server* melakukan autentikasi terhadap percobaan dari *client*, bila autentikasi berhasil. Dalam hal ini urutan *port* yang dicoba untuk dikoneksikan sesuai dengan aturan tertentu pada *port knocking daemon*, maka *daemon* akan melakukan *overwrite* terhadap file konfigurasi *firewall* agar mengijinkan *port* tujuan untuk dibuka kepada *client* dengan IP *Address* ter-autentikasi. *File* konfigurasi *port knocking* pada Ubuntu 16.04 bernama *knock* yang harus diinstall sebelumnya, *file* tersebut terletak pada direktori */etc/knockd.conf*. perintah untuk serangkaian urutan *well-defined set of ports* untuk keamanan sistem yaitu:

- *[options]*
  useSyslog
  logfile = /var/log/knockd.conf
- *[openFTP]*
  sequence = 4, 8, 12
  seq_timeout = 5
  command = /sbin/iptables –A INPUT –s %IP% –p tcp –dport 21 –j ACCEPT
  tcpflags = syn
- *[closeFTP]*
  sequence = 12, 8, 4
  seq_timeout = 5
  command = /sbin/iptables –D INPUT –s %IP% –p tcp –dport 21 –j ACCEPT
  tcpflags = syn

c. Perencanaan penyerangan terhadap sistem

Perencanaan pengujian terhadap sistem ini nantinya melibatkan komputer *client* sebagai komputer yang akan menyerang komputer FTP *server*. Skenario pengujian disajikan dalam bentuk tabel 1 dibawah ini. Dari tabel skenario perencanaan pengujian tabel 1, bahwa nantinya pengujian dilakukan dengan beberapa keadaan sistem yaitu *firewall & port knocking stop* dan *firewall & port knocking start*, dalam hal ini sistem keamanan untuk FTP *server* belum aktif dan sudah aktif. Jenis pengujian terkait dengan beberapa objek yang dijadikan pengujian pada komputer FTP *server*. Sedangkan aplikasi pengujian merupakan aplikasi atau *tools* yang dapat digunakan oleh komputer *client* maupun komputer *server* untuk melakukan pengujian sistem keamanan yang dirancang ini.

Tabel 1. Skenario Pengujian Sistem Keamanan *FTP Server*

Status sistem	**Jenis pengujian**	**Aplikasi pengujian**	**Komputer penguji**
Firewall dan *port knocking start*	periksa *port* yang terbuka	Nmap, wireshark	Client 1, Client 2
	Port knocking buka ke *server*	Wireshark	Client 1, Client 2
	Login FTP & *transfer file*	Filezilla, cmd, terminal linux, windows explorer, nautilus	Client 1, Client 2

4. HASIL DAN PEMBAHASAN

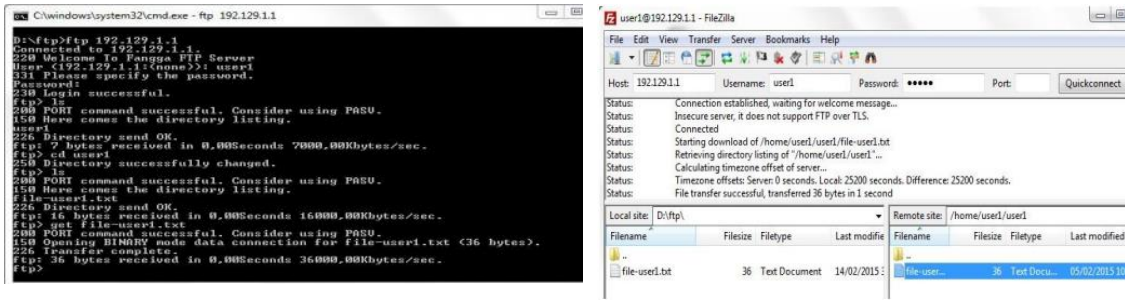
Pada bahasan pengujian mengacu pada tabel 1, dimana pengujian dilakukan terhadap sistem keamanan *firewall* dan *port knocking* sebelum diaktifkan dan pengujian sesudah diaktifkan. Pengujian sistem keaman ini dilakukan oleh komputer *client* yang juga bertindak sebagai *attacker*.

a. Sistem keamanan *firewall* dan *port knocking stop*

Pengujian pertama menguji status *port* untuk layanan FTP server yaitu port 21.

Gambar 4. Pengujian *Port FTP* Sebelum Sistem Keamanan Diaktifkan

Tampak dari hasil pengujian port FTP yang nampak pada gambar 4, setelah dilakukan *port scanning* menggunakan *tools Zenmap* (*nmap*) bahwa status layanan FTP *port* 21 sedang terbuka. Pengujian selanjutnya yaitu ketika komputer *client* melakukan *login* dan *transfer* data pada komputer FTP *server* menggunakan *port* 21.

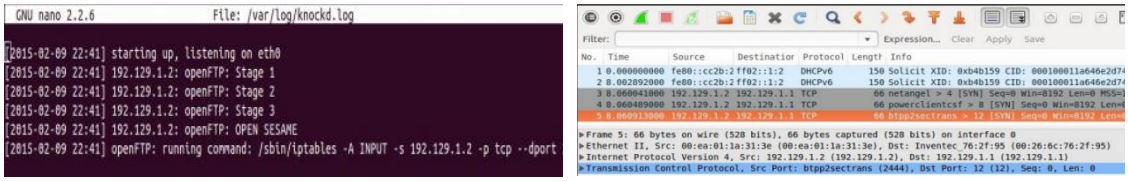


Gambar 5. Pengujian Login dan *Transfer FTP* Sebelum Sistem Keamanan Diaktifkan

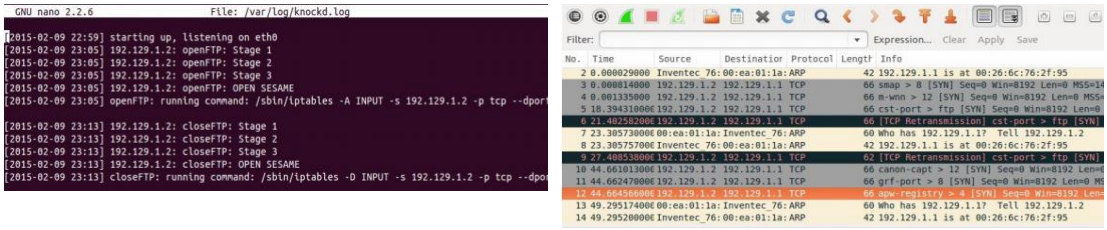
Tampak dari hasil pengujian *login* dan *transfer file* yang nampak pada gambar 5, setelah berhasil *login* dilakukan *transfer file* menggunakan *tools* Filezilla dan *command prompt* (CMD) sistem operasi Windows *client* berhasil melakukan *transfer* file ke FTP *server*.

b. Sistem keamanan *firewall* dan *port knocking start*

Pengujian dilakukan ketika layanan *rules-set firewall* dan layanan rangkaian urutan *well-defined set of ports* diaktifkan atau *start*. Komputer *client* 1 mencoba mengakses komputer FTP *server* yang telah mengaktifkan sistem keamanannya sehingga layanan FTP tidak bisa diakses dan juga port 21 statusya *close*. Komputer *client* 1 selaku komputer yang menguji sistem keamanan FTP *server* melakukan *knock*, yang sebelumnya telah terinstall *tools knock-win32* [7]. *Client* 1 mencoba menggunakan layanan FTP *server* dengan melakukan teknik *port knocking* yang memasukan urutan rangkaian *well-defined set of ports* (4, 8, 12). Selanjutnya dari sisi komputer FTP melakukan pencatatan dan melakukan autentikasi percobaan *knock client* 1, bila autentikasi berhasil maka *port* 21 untuk layanan FTP statusnya menjadi *open* atau terbuka. Hasil pengujian ini nampak pada gambar 6, dimana komputer FTP *server* melakukan pencatatan pada *file log* yang dapat dilihat pada file */var/log/knockd.log*. Selain itu, untuk mengetahui autentikasi pada komputer FTP *server* juga dapat dilihat menggunakan *tools* Wireshark [7].



Gambar 6. *Authentication Open* Komputer *Client 1* Yang Tercatat Pada *File Log FTP Server* dan *wireshark*



Gambar 7. *Authentication Close* Komputer *Client 1* Yang Tercatat Pada *File Log FTP Server* dan *wireshark*

Setelah *client* 1 menggunakan layanan FTP server dengan melakukan teknik *port knocking* yang memasukan urutan rangkaian *well-defined set of ports*. Untuk mengamankan kembali layanan FTP, maka *client* 1 harus melakukan *knock* kembali untuk menutup *port* 21, yaitu dengan memasukan urutan rangkaian *well-defined set of ports* (12, 8, 4). Kemudian dari sisi komputer FTP *server* melakukan pencatatan dan melakukan autentikasi percobaan *knock client* 1, bila autentikasi berhasil maka *port* 21 untuk layanan FTP statusnya menjadi *close* atau tertutup. Hasil pengujian nampak pada gambar 7.

Hasil dari skenario pengujian yang dilakukan oleh komputer *client* 1 yang melakukan teknik *port knocking* sebelum mengakses layanan FTP disajikan ke dalam tabel 2. Sedangkan hasil pengujian yang dilakukan oleh *client* 2 yang tanpa menggunakan *port knocking* mengakses layanan FTP disajikan ke dalam tabel 3.

Tabel 2. Hasil Pengujian *FTP server* Oleh *Client* 1

Status sistem	**Urutan *Well-defined set of ports***	**Jenis pengujian**	**Hasil pengujian**
Firewall dan *port knocking start*	4 8 12	Cek *port* yang terbuka	21
		Port knocking buka ke *server*	4 [syn]; 8 [syn]; 12 [syn]
		Login FTP & *transfer* file	Success
	12 8 4	Cek *port* yang terbuka	Close all

Firewall dan *port knocking stop*		*Login* FTP & *transfer* file	Close
		Login FTP & *transfer* file	Close

Tabel 3. Hasil Pengujian *FTP server* Oleh *Client* 2

Status sistem	**Urutan *Well-defined set of ports***	**Jenis pengujian**	**Hasil pengujian**
Firewall dan port knocking start	-	Cek *port* yang terbuka	Close all
		Port knocking buka ke *server*	-
		Login FTP & *transfer* file	Failed
Firewall dan port knocking stop	-	Cek *port* yang terbuka	Close all
		Login FTP & *transfer file*	Close
		Login FTP & *transfer file*	Close

5. KESIMPULAN

Kesimpulan dari penelitian yang telah dilakukan ini dapat dijelaskan sebagai berikut antara lain, yaitu:

- *The rule-set firewall* yang telah diimplementasikan pada FTP *server* membuat penyerang tidak dapat mengetahui *port* mana saja yang sedang *open*.
- Metode *port knocking* dapat melindugi akses FTP *server* walaupun *client* mengetahui *username* dan *password* namun tetap dapat mendapatkan akses untuk *transfer* file atau data.
- Dengan menggunakan urutan nomor *port knocking* baik untuk membukak akses maupun menutup akses layanan FTP menambah keamanan dalam *transfer file*.

Daftar Pustaka

[1] Hendy Djaja Siswaja, "Pembuatan FTP *Server* Pada *Server* Redhat 4 Dengan VSFTPD" Media Informatika Vol.13 No.1, Sekolah Tinggi Manajemen Informatika dan Komputer, Bandung, 2014.

[2] Francis Kwadzo Agbenyegah, Michel Asante, "Impact of firewall on network performance", International Journal Of Scientific & Technology Research Volume 6, Issue 03, 2017.

[3] L.Saliou, W.J.Buchanan, J.Graves and J.Munoz, "*Scenario Analysis using Out-of-line Firewall Evaluation Framework*", Centre for Distributed Computing and Security, Napier University, Edinburgh, United Kingdom.

[4] Yuni Dian Pratiwi, Enggar Alfianto, Shah Khadafi. "Implementasi Metode *Intrusion Detection System* (IDS) Menggunakan Perangkat Lunak Portsentry Dan Snort Berbasis Sistem Operasi Linux Ubuntu 16.04 LTS", Jurusan Sistem Komputer, Fakultas Teknologi Informasi, Institut Teknologi Adhi Tama Surabaya, Surabaya.

[5] Edy Haryanto, Widyawan, Dani Adhipta, "Meningkatkan Keamanan *Port Knocking* Dengan Kombinasi *Special Features* ICMP, *Source Port*, dan *Tunneling*", Seminar Riset Teknologi Informasi (SRITI), 2016.

[6] http://portknocking.org/.

[7] http://www.zeroflux.org/proj/knock/files/knock-win32.zip.