Journal of Technology and Informatics (JoTI)
Vol. 3, No. 1, Bulan Oktober Tahun 2021
E-ISSN: 2686-6102
P-ISSN: 2712-4842 DOI: https://doi.org/10.37802/joti.v3i1.188

Intelligent Rule Firewall berbasis Linux menggunakan *Association Rule Mining* untuk Peningkatan *Adaptive Response Attack*

Slamet[1], **Norma Ningsih**[2]
[1, 2] Program Studi/Jurusan Sistem Informasi, Universitas Dinamika
Email: slamet@dinamika.ac.id*, norma@dinamika.ac.id

Abstrak: Kompleksitas jaringan dan kebutuhan transaksi bisnis yang berkembang membuat organisasi harus semakin terbuka terhadap dunia, sehingga potensi serangan dari dalam dan dari luar jaringannya semakin meningkat. Salah satu mekanisme perlindungan yang serius dan murah adalah dengan menerapkan firewall berbasis linux untuk menjaga pintu masuknya serangan. Umumnya, fitur dasar firewall tidak efektif untuk menjaga serangan yang dinamis dan terjadi terus-menerus, sehingga diperlukan sistem firewall cerdas agar bisa adaptif terhadap jenis serangan dan kondusif terhadap kondisi paket terkini. Pada dasarnya, konfigurasi firewall menerima atau menolak tindakan untuk paket secara *default*. Dalam paper ini, peneliti mengusulkan untuk menambahkan kecerdasan buatan pada *rule* konfigurasi firewall *default*. *Log* firewall sebagai representasi keluar masuknya trafik diolah menggunakan *Association Rule Mining*. Hasilnya, (a) *rule* firewall mampu adaptif terhadap perilaku serangan di jaringan, (b) firewall mampu membuat sekaligus memperbaiki *rule* kebijakan dirinya dari paket-paket anomali sehingga dapat diterapkan sebagai konfigurasi firewall yang efektif.

Kata Kunci: *rule* firewall, linux, cerdas, *association rule*

Abstract: *The complexity of network and the growing need for business transactions make organizations have to be more open to the world, so that the potential for attacks from inside and outside their network is increasing. One serious and inexpensive protection mechanism is to implement a linux-based firewall to guard against the entrance of attacks. The basic features of the firewall are not effective in preventing attacks that occur continuously and dynamically, so an intelligent firewall system is needed to be adaptive to the types of attacks and the real time packet conditions. Basically, Firewall sets accept or reject actions for packets by default. In this paper, the researcher proposes to add artificial intelligence in the firewall. Firewall logs as a representation of the entry and exit of traffic are processed using data association rule mining. As a result, (a) the firewall can be adaptive to the behavior of attacking on the network, (b) the firewall is able to create and fix its own policy rules and anomalous packets so that it can be applied as an effective firewall configuration.*

Keywords: *rule firewall, linux, intelligent, association rule*

PENDAHULUAN

Saat ini, infrastruktur siber dianggap sebagai aset terbesar bagi banyak organisasi sehingga pertahanan menjadi fokus perhatian utama. Untuk tujuan ini, organisasi menggunakan mekanisme pertahanan, termasuk firewall yang menjadi persyaratan utama untuk setiap sistem keamanan jaringan [1]. Firewall berbasis linux sering digunakan sebagai firewall oleh banyak perusahaan karena dapat digunakan secara gratis dan menawarkan fungsionalitas *customized* untuk mendeteksi ancaman atau peringatan umum [2]. Deteksi ancaman didapatkan dengan membuat *rule* dasar pada lalu lintas jaringan tertentu dan menganalisis statistik paket data yang melewatinya. Cara ini menghasilkan statistik informasi dengan volume yang besar tentang peristiwa keamanan yang dapat digunakan untuk analisis lebih lanjut. Namun sesungguhnya deteksi ancaman hanyalah fungsi dasar, sehingga membutuhkan kecerdasan tambahan dalam mengotomatisasi ekstraksi informasi firewall ini.

Selama ini cara otomatis ditawarkan oleh pihak ketiga dengan menggunakan alat yang mahal dan tidak bisa dikustomisasi oleh pengguna [3]. Oleh karena itu, perlindungan terhadap ancaman-ancaman keamanan hanya bersifat statis dan tidak bisa diekplorasi lagi. Teknologi firewall berada di garis depan dalam mengamankan jaringan. Apabila aturan *(rule)* firewall yang dikonfigurasi buruk maka menyebabkan efektivitas keamanan firewall menjadi terbatas. *Issue* utama dalam cara pengaturan ini adalah tentang seberapa banyak *rule* yang penting, berguna, *up-to-date*, dan terorganisir dengan baik sebagai cermin karakteristik dan volume paket di dalam jaringan saat ini. Namun yang sering terjadi adalah *trend* lalu lintas jaringan saat ini yang dinamis tidak diikuti dengan konfigurasi firewall yang dinamis. *Log server* seringkali menunjukkan bukti bahwa beberapa *rule* firewall sudah *out-of-date* atau tidak berguna lagi (usang) karena trafik yang dinamis, sehingga *Network* Administrator harus menghapus,

menggabungkan, atau menyusun ulang *rule* untuk mengoptimalkan kebijakan dan efektifitas firewall.

Fungsi *Association Rule Mining* kepada *log* trafik jaringan dapat mengungkapkan ketidak-sesuaian *rule* firewall saat ini seperti contoh: *update* konten *website* melalui *link* SSH atau mengizinkan lalu lintas terlarang melalui perangkat di belakang firewall. Selain itu, dikarenakan jumlah *rule filtering* yang meningkat dan diatur dengan proses manual, pengelolaan *rule* kebijakan firewall menjadi sangat sulit dan memakan waktu yang lama.

Untuk menjawab kebutuhan akan manajemen firewall yang efektif, optimal dan dapat memvalidasi *rule* firewall secara otomatis, dibutuhkan sebuah teknik keamanan dan manajemen *rule* yang efektif juga. Untuk menjembatani kesenjangan ini, hal pertama perlu dilakukan pengamatan di jaringan dengan menganalisis *trafik* paket data menggunakan A*ssociation Rule Mining*.

Penelitian ini menganalisis dan mengelola *rule* kebijakan firewall, tidak hanya dengan meminimalkan *rule*, tetapi juga untuk menghasilkan seperangkat *rule* efisien yang mencerminkan tren lalu lintas terkini. Selanjutnya dengan memberikan kemampuan pembaruan kebijakan secara *real time* dari firewall (misalnya, dapat mendeteksi banyaknya serangan DDOS pada *web server*, *Sniffing* pada *DNS Server*).

METODE PENELITIAN

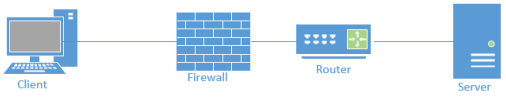
Metode penelitian ini dimulai dengan penjelasan dasar firewall, *intelligent rule firewall, algoritma association rule maining* dan algoritma apriori.

Dasar Firewall

Firewall adalah perangkat lunak atau sistem perangkat keras yang menyaring lalu lintas jaringan sesuai dengan kebijakan yang telah ditentukan sebelumnya. Untuk ini, firewall ditanamkan di tempat penting dari sistem operasi atau di posisi kunci dari arsitektur jaringan. Di posisi ini, firewall menganalisis apa yang terjadi melalui dan menerapkan tindakan dengan seperangkat *rule*.

Secara umum, fungsional firewall dapat memproses paket yang masuk atau keluar berdasarkan kebijakan yang telah dikonfigurasi sebelumnya. Kebijakan diwakili oleh seperangkat *rule* yang menentukan apa yang harus dilakukan dengan paket tertentu: apakah paket tersebut diizinkan/dilewati, atau dibuang (dengan atau tanpa pemberitahuan).

Biasanya, ada 3 tindakan *(action)* dasar yang dilakukan firewall terhadap paket data tersebut. Tindakan ini adalah bagian dari definisi *rule* [4]. Setiap *rule* menentukan satu tindakan. Tindakan memiliki nama yang cukup jelas. Tindakan *ALLOW* (atau *ACCEPT*) mengizinkan paket untuk melewati firewall masuk atau keluar (tergantung pada arah lalu lintas). Tindakan ini adalah tindakan *default*, seperti apabila firewall tidak dipasang. Tindakan *DENY* (atau *DROP*) akan menjatuhkan paket tanpa pemberitahuan kepada pengirim. Tindakan ini adalah tindakan yang paling berguna. Tindakan *REJECT*, mirip dengan *DENY*, melarang paket lewat, tetapi dengan pemberitahuan yang dikirim kembali kepada pengirim. Fungsionalitas firewall secara umum dapat dilihat pada gambar 1, dimana paket dari *server* akan dilakukan *filtering* sebelum menuju ke *client*. Demikian juga sebaliknya untuk paket yang berasal dari *client* yang menuju ke *server*.



Gambar 1. Fungsi dasar Firewall

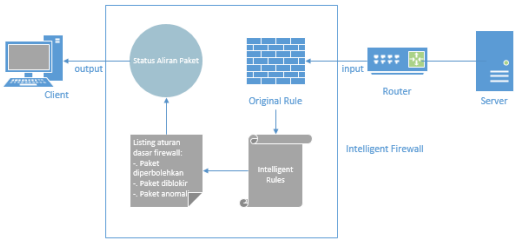
Sedangkan aturan-aturan *(rules)* umum yang sering digunakan oleh firewall merujuk pada área-area seperti IP sumber, IP tujuan, *port* sumber, *port* tujuan, *mask* sumber, *mask* tujuan, *action* dan protokol yang digunakan untuk melakukan *filtering*, sebagaimana dijelaskan pada tabel 1.

Tabel 1. Area umum dalam firewall:

Area	Keterangan
IP Sumber	Alamat IP sumber yang cocok dengan *rule*
IP Tujuan	Alamat IP tujuan yang cocok dengan *rule*
Port Sumber	Alamat *Port* sumber yang cocok dengan *rule*
Port Tujuan	Alamat *Port* tujuan yang cocok dengan *rule*
Mask Sumber	Alamat *Mask* sumber yang cocok dengan *rule*
Mask Tujuan	Alamat *Mask* tujuan yang cocok dengan *rule*
Action	*Accept, Deny* atau *Reject*
Protokol	Protokol yang cocok : TCP, UDP

Intelligent Rule Firewall

Intelligent Rule Firewall adalah firewall umum yang dilengkapi dengan kemampuan untuk melakukan *rule filtering*, memindai, dan mencari aktivitas yang berpotensi berbahaya, kemudian mengirimkan peringatan untuk memberi tahu administrator tentang ancaman yang masuk ke jaringan sehingga dapat segera didiagnosis masalahnya. Alat ini juga memiliki kemampuan untuk menegakkan kebijakan internal seperti mencegah pemakai menjelajahi dan mengakses situs *web* yang berpotensi berbahaya atau terlarang di tempat kerja. *Artificial Intelligent* digunakan sebagai pembelajaran di dalam *rule* firewall untuk melacak catatan serangan secara otomatis dan mencegah tindakan serupa di masa depan [5]. Algoritma yang digunakan dalam *intelligent rule firewall* pada penelitian ini adalah Algoritma *Association Rule Mining (ARM)* dan *Algoritma Apriori*. Flow Diagram dari *Intelligent Rule* Firewall dapat dilihat pada gambar 2.



Gambar 2. Flow Diagram *Intelligent Rule* Firewall

Pada gambar 2, trafik data yang berasal dari *server* menuju *client*, melewati sebuah router terlebih dahulu, sebelum dilakukan *filtering* di dalam firewall yang sudah dilengkapi dengan sistem cerdas. Paket yang masuk ke dalam firewall diolah menggunakan teknik *mining* untuk menghasilkan *rule* terbaik dan adaptif.

Selanjutnya, paket dari firewall akan diteruskan kepada *client* untuk dieksekusi. Apabila aturannya memperbolehkan, maka paket akan diberikan kepada *client*. Paket akan diblokir apabila tidak diperkenankan oleh *rule* firewall. Apabila terdapat paket anomali, data akan diolah kembali untuk menghasilkan pola dan *rule* terbaik yang bisa dimasukkan sebagai usulan tambahan untuk *rule* terbaru.

Algoritma Association Rule Mining

Association Rule adalah pernyataan *if/then* yang membantu untuk mengungkap hubungan antara data yang tidak terkait dalam basis data, basis data relasional atau repository informasi lainnya [6]. Aturan asosiasi digunakan untuk menemukan hubungan antara benda-benda yang sering digunakan bersama-sama. Penggunaan *Association Rule* seperti analisis data dalam basis data, klasifikasi, *cross marketing*, *clustering*, desain katalog, analisis *lost-leader* dan sebagainya. Sebagai contoh misalnya, jika pelanggan membeli roti maka dia mungkin juga membeli mentega. Jika pelanggan membeli laptop maka dia mungkin juga membeli kartu memori.

Ada dua kriteria dasar yang digunakan oleh *association rule* yaitu *support* dan *confidence*. Dua kriteria ini menghasilkan identifikasi hubungan dan aturan dengan menganalisis *if/then* yang sering digunakan oleh pola. *Association rule* biasanya diperlukan untuk memenuhi dukungan minimum yang ditentukan pengguna dengan *confidence* minimum pada saat yang sama.

$$Rule: X => Y \begin{cases} Support = \frac{frq(X,Y)}{N} \\ Confidence = \frac{frq(X,Y)}{N} \end{cases}$$

1. *Support*

Support pada *Association Rule Mining* didefinisikan sebagai persentase *record* yang berisi *X Ũ Y* terhadap jumlah total record dalam basis data [6]. Hitungan untuk setiap item bertambah satu setiap kali item ditemukan dalam transaksi berbeda (*T*) di basis data (*D*) selama proses pemindaian. Hal ini berarti jumlah *support* tidak memperhitungkan jumlah *item*. Contoh sederhana, misalnya dalam sebuah transaksi seorang pelanggan membeli tiga botol air mineral akan ditambahkan *support* berjumlah satu air mineral saja. Dengan kata lain, jika transaksi berisi item maka jumlah dukungan item ini bertambah satu. *Support* dihitung dengan rumus berikut:

$$Support\ (X,Y) = \frac{Jumlah\ Support\ dari\ XY}{Total\ Transaksi\ pada\ D}$$

Dapat dilihat bahwa *support* dari suatu *item* adalah signifikansi statistik dari *association rule*. Misalkan *support* suatu barang adalah 0,1%, artinya hanya 0,1 persen transaksi yang mengandung pembelian barang ini. Pembeli tidak akan terlalu memperhatikan jenis barang yang tidak begitu sering dibeli, karena diperlukan dukungan yang tinggi untuk *association rule* yang lebih menarik. Sebelum proses mining, pengguna dapat menentukan *support* minimum sebagai *threshold*, yang berarti mereka hanya tertarik pada *aturan asosiasi* tertentu yang dihasilkan dari kumpulan *item* yang *support*-nya melebihi *threshold* itu. Namun, terkadang *itemset* tidak sesering yang didefinisikan oleh *threshold*, sehingga *association rule* yang dihasilkan darinya masih penting.

2. *Confidence*

Confidence didefinisikan sebagai persentase jumlah transaksi yang mengandung *XŨY* terhadap total jumlah record yang mengandung *X*, dimana jika persentase tersebut melebihi *threshold* dari *confidence* maka dapat dihasilkan aturan asosiasi X→Y [6].

$$Confidence\ (X|Y) = \frac{Support\ dari\ (XY)}{Support\ dari\ X}$$

Confidence adalah ukuran kekuatan aturan asosiasi, misalkan *confidence* aturan asosiasi X→Y adalah 60%, artinya 60% dari transaksi yang mengandung X juga bersama-sama mengandung Y, demikian pula untuk memastikan aturan dengan *confidence* minimum juga bisa ditentukan sebelumnya oleh pengguna.

Algoritma Apriori

Algoritma Apriori digunakan untuk melakukan *mining* kepada kumpulan item yang sering (*frequently*) dan *association rule learning*. Algoritma ini menggunakan pencarian *level-wise*, dimana *k-itemsets* (Sebuah *itemset* yang berisi *item* yang dikenal dengan *k*) digunakan untuk mengeksplorasi *(k+1)-itemsets*, melakukan *mining itemset* yang sering (*frequently*) dari transaksional basis data pada *boolean association rule*.

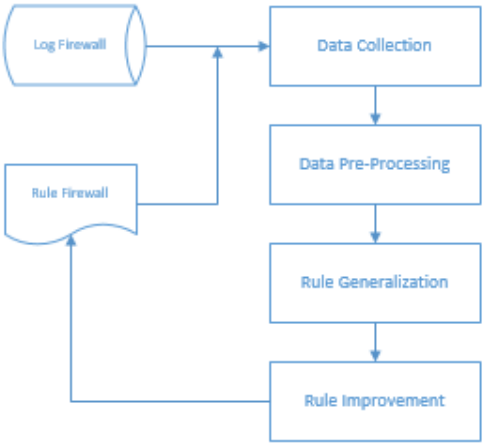
Dalam algoritma ini, himpunan bagian yang sering diperpanjang satu *item* pada saat yang sama. Langkah ini dikenal sebagai proses pembangkitan (*generate*) kandidat, selanjutnya kelompok kandidat itu diujikan kepada data tertentu. Untuk menghitung calon *set item* secara efisien, Apriori menggunakan metode *breadth-first search* dan *hash tree structure* [7]. Cara ini

digunakan untuk mengidentifikasi *item* individu yang sering berada di basis data dan memperluasnya ke *set item* yang lebih besar sebagai *set item* yang sering muncul di basis data. Algoritma Apriori menentukan set item yang sering muncul dan dapat digunakan untuk menentukan aturan asosiasi yang menjadi tren umum di dalam basis data.

Berikut ini adalah prosedur untuk algoritma Apriori:

CI_k : Candidate itemset having size k
FI_k : Frequent itemset having size k
FI_1 = {frequent items};
For (k=1; FI_k != null; k++) do begin
CI_k+1 = candidates generated from FIk;
For each transaction t in database D do
Increment the count value of all candidates in CI_k+1 that are contained in t
FI_k+1 = candidates in CIk+1 with min_support
End
Return FI_k ;

Proses penelitian yang dilakukan pada paper ini terdiri dari empat iterasi, yaitu: (1). Mengumpulkan data mentah dari *log firewall* menggunakan tools TCPDump yang ada di linux, (2). Melakukan *Pre-Processing* dengan mengekstraksi atribut dari *log firewall* menggunakan Algoritma *Association Rule Mining* dan Algoritma *Apriori* untuk menghasilkan *rule* awal, (3). Untuk menemukan kumpulan *rule* awal pada firewall digunakan *Filtering Rule Generalization* (4) Melakukan perbaikan *rule* dengan mengidentifikasi *rule* yang *out-of-date* dan *rule* yang terbaik. Hasil identifikasi terbaik digunakan sebagai perbaikan *rule* firewall yang sudah ada. Framework penelitian dapat dilihat pada gambar 3.



Gambar 3. *Framework* Penelitian

HASIL DAN PEMBAHASAN

Pada bagian ini dijelaskan dua hal yaitu: a. proses dan analisis dataset; b. proses pembuangan *rule* yang membusuk (usang) dan pemakaian *rule* yang dominan pada konfigurasi firewall.

A. Proses dan Analisis *Dataset*

A.1 *Data Collection*

Pada tahap pertama memproses kebijakan aturan *(rule)* firewall yang diambil dari *log file* trafik data di dalam firewall linux. Tercatat sebanyak 40.150 catatan dari *log* firewall Linux. Untuk mengurangi jumlah status penelitian dan prototipe, kami hanya menggunakan area penting dalam *rule* firewall dan header paket IP seperti: protokol (TCP atau UDP), arah paket (masuk atau keluar), alamat IP sumber, *port* sumber, alamat IP tujuan, *port* tujuan, dan tindakan *(Deny/Drop/Accept)*. Setiap *rule* terdiri dari atribut dalam format berikut *"<direction> <protocol> <source-IP> <source-port> <dest-IP> <dest-port> <action>"*. Tidak adanya atribut di atas dalam *rule* menunjukkan bahwa *rule* tersebut tidak dipengaruhi oleh atribut tersebut.

Daftar *rule* firewall Linux yang digunakan pada penelitian ini ditunjukkan pada skrip berikut.

```
#!/bin/bash
# Clear any previous rules.
/sbin/iptables -F
# Default drop policy.
/sbin/iptables -P INPUT DROP
/sbin/iptables -P OUTPUT ACCEPT
# Allow anything over loopback and vpn.
/sbin/iptables -A INPUT -i lo -s 127.0.0.1 -d 127.0.0.1 -j ACCEPT
/sbin/iptables -A OUTPUT -o lo -s 127.0.0.1 -d 127.0.0.1 -j ACCEPT
/sbin/iptables -A INPUT -i tun0 -j ACCEPT
/sbin/iptables -A OUTPUT -o tun0 -j ACCEPT
/sbin/iptables -A INPUT -p esp -j ACCEPT
/sbin/iptables -A OUTPUT -p esp -j ACCEPT
# Drop any tcp packet that does not start a connection with a syn flag.
/sbin/iptables -A INPUT -p tcp ! --syn -m state --state NEW -j DROP
# Drop any invalid packet that could not be identified.
/sbin/iptables -A INPUT -m state --state INVALID -j DROP
# Drop invalid packets.
/sbin/iptables -A INPUT -p tcp -m tcp --tcp-flags FIN,SYN,RST,PSH,ACK,URG NONE -j DROP
/sbin/iptables -A INPUT -p tcp -m tcp --tcp-flags SYN,FIN SYN,FIN -j DROP
/sbin/iptables -A INPUT -p tcp -m tcp --tcp-flags SYN,RST SYN,RST -j DROP
/sbin/iptables -A INPUT -p tcp -m tcp --tcp-flags FIN,RST FIN,RST -j DROP
/sbin/iptables -A INPUT -p tcp -m tcp --tcp-flags ACK,FIN FIN -j DROP
/sbin/iptables -A INPUT -p tcp -m tcp --tcp-flags ACK,URG URG -j DROP
# Reject broadcasts to 224.0.0.1
/sbin/iptables -A INPUT -s 224.0.0.0/4 -j DROP
/sbin/iptables -A INPUT -d 224.0.0.0/4 -j DROP
/sbin/iptables -A INPUT -s 240.0.0.0/5 -j DROP
# Blocked ports
/sbin/iptables -A INPUT -p tcp -m state --state NEW,ESTABLISHED,RELATED --dport 8010 -j DROP
# Allow TCP/UDP connections out. Keep state so conns out are allowed back in.
/sbin/iptables -A INPUT -p tcp -m state --state ESTABLISHED -j ACCEPT
/sbin/iptables -A OUTPUT -p tcp -m state --state NEW,ESTABLISHED -j ACCEPT
/sbin/iptables -A INPUT -p udp -m state --state ESTABLISHED -j ACCEPT
/sbin/iptables -A OUTPUT -p udp -m state --state NEW,ESTABLISHED -j ACCEPT
# Allow only ICMP echo requests (ping) in. Limit rate in. Uncomment if needed.
/sbin/iptables -A INPUT -p icmp -m state --state NEW,ESTABLISHED --icmp-type echo-reply -j ACCEPT
/sbin/iptables -A OUTPUT -p icmp -m state --state NEW,ESTABLISHED --icmp-type echo-request -j ACCEPT
# or block ICMP allow only ping out
```

```
/sbin/iptables -A INPUT  -p icmp -m state --state NEW -j DROP
/sbin/iptables -A INPUT  -p icmp -m state --state ESTABLISHED -j ACCEPT
/sbin/iptables -A OUTPUT -p icmp -m state --state NEW,ESTABLISHED -j ACCEPT
# Allow ssh connections in.
#/sbin/iptables -A INPUT -p tcp -s 1.2.3.4 -m tcp --dport 22 -m state --state NEW,ESTABLISHED,RELATED -m limit --limit 2/m -j ACCEPT
# Drop everything that did not match above or drop and log it.
#/sbin/iptables -A INPUT    -j LOG --log-level 4 --log-prefix "IPTABLES_INPUT: "
/sbin/iptables -A INPUT  -j DROP
#/sbin/iptables -A FORWARD -j LOG --log-level 4 --log-prefix "IPTABLES_FORWARD: "
/sbin/iptables -A FORWARD -j DROP
#/sbin/iptables -A OUTPUT  -j LOG --log-level 4 --log-prefix "IPTABLES_OUTPUT: "
/sbin/iptables -A OUTPUT -j ACCEPT
iptables-save > /dev/null 2>&1
```

A.2 *Data Pre-Processing*

Tahap kedua memproses dataset dengan mengekstraksi fitur paket dari *log* firewall menggunakan algoritma *Association Rule Mining* dan algoritma Apriori. Setiap baris *dataset log* firewall menunjukkan informasi setiap paket dalam: jangka waktu, *action* (DROP_LOGIN), arah (OUT=), alamat IP sumber (SRC=), IP tujuan (DST=), ukuran paket (LEN=), waktu berangkat (TTL=), ID paket (ID=), protokol (PROTO=), port sumber (SPT=), dan port tujuan (DPT=). Hasil sebagian dari proses untuk Alamat IP 202.110.* ditunjukkan pada gambar 4.

# of count	Protocol	Direction	Source IP	Source Port	Destination IP	Destination Port	Action
7071	UDP	INPUT	202.110.90.100	1058	202.110.90.255	1212	DENY
3	UDP	INPUT	202.110.90.91	61502	202.110.90.255	177	DENY
3	UDP	INPUT	202.110.90.80	61505	202.110.90.255	177	DENY
179	UDP	OUTPUT	202.110.90.90	123	202.110.100.10	123	DENY
340	UDP	INPUT	202.110.90.80	61501	202.110.90.255	126	DENY
3	UDP	INPUT	202.110.90.80	61545	202.110.90.255	177	DENY
1324	UDP	INPUT	202.110.90.80	61550	202.110.90.255	177	DENY
3	UDP	INPUT	202.110.90.80	61455	202.110.90.255	177	DENY
3	UDP	INPUT	202.110.90.80	61534	202.110.90.255	177	DENY
3	UDP	INPUT	202.110.90.80	61544	202.110.90.255	177	DENY
3	UDP	INPUT	202.110.90.80	61234	202.110.90.255	177	DENY
3	UDP	INPUT	202.110.90.80	61453	202.110.90.255	177	DENY
3	UDP	INPUT	202.110.90.80	61765	202.110.90.255	177	DENY
3	UDP	INPUT	202.110.90.80	61501	202.110.90.255	177	DENY
3	UDP	INPUT	202.110.90.80	61985	202.110.90.255	177	DENY
3	UDP	INPUT	202.110.90.80	61544	202.110.90.255	177	DENY
3	UDP	INPUT	202.110.90.80	61862	202.110.90.255	177	DENY
3	UDP	INPUT	202.110.90.80	61534	202.110.90.255	177	DENY
3	UDP	INPUT	202.110.90.80	61231	202.110.90.255	177	DENY
3	UDP	INPUT	202.110.90.80	61500	202.110.90.255	177	DENY
3	UDP	INPUT	202.110.90.80	61571	202.110.90.255	177	DENY
3	UDP	INPUT	202.110.90.80	61522	202.110.90.255	177	DENY
3	UDP	INPUT	202.110.90.80	61532	202.110.90.255	177	DENY
3	UDP	INPUT	202.110.90.80	61231	202.110.90.255	177	DENY
3	UDP	INPUT	202.110.90.80	61834	202.110.90.255	177	DENY
3	UDP	INPUT	202.110.90.80	61544	202.110.90.255	177	DENY
3	UDP	INPUT	202.110.90.80	61571	202.110.90.255	177	DENY
3	UDP	INPUT	202.110.90.80	61582	202.110.90.255	177	DENY
3	UDP	INPUT	202.110.90.80	61245	202.110.90.255	177	DENY
3	UDP	INPUT	202.110.90.80	61508	202.110.90.255	177	DENY
3	UDP	INPUT	202.110.90.80	61509	202.110.90.255	177	DENY
153	UDP	OUTPUT	202.110.90.70	65	202.110.100.12	66	DENY
3	UDP	INPUT	202.110.90.80	61515	202.110.90.255	177	DENY
3	UDP	INPUT	202.110.90.80	61842	202.110.90.255	177	DENY

Gambar 4. Contoh *Rule Filtering* dari *Pre-processing*

A.3 *Rule Generalization*

Tahap ketiga adalah menggeneralisasi *rule* menggunakan *Filtering Rule Generalization*. Pada langkah ini, setiap *rule* dasar dengan frekuensi tertentu digeneralisasikan atau diagregasi lebih mendalam. Misalnya, alamat IP dalam *rule* dapat berupa host tertentu (alamat IP unik seperti 202.110.90.100) atau menghasilkan alamat jaringan gabungan (misalnya 202.110.90.*) dengan menggabungkan sekelompok *rule* serupa dengan *rule* yang sama (Alamat IP). *Port* dapat berupa nomor *port* tunggal tertentu atau digabungkan menjadi "*anything*" untuk menjadi nomor *port* apa pun.

Untuk membagi *rule* pemfilteran *multi-value* menjadi beberapa *rule* dimana setiap *rule* memiliki bidang bernilai tunggal. Atribut terdiri dari tujuh area untuk menganalisis dan menemukan pengetahuan dari *log* trafik data dimana *instance*-nya adalah informasi paket dari *log file* firewall. *Rule* dikelompokkan dari semua *rule* yang diidentifikasi dan digeneralisasi dalam *superset*. Misalnya, mengelompokkan semua *rule* IP alamat 202.110.90.80 (apakah itu IP sumber atau IP tujuan) dengan port tujuan (=21), dan *action* (=*accept*) adalah menghasilkan *rule* yang ditunjukkan pada gambar 5.

```
1 TCP,INPUT,202.110.90.80 ,ANY,202.110.90.45,21,DENY
2 TCP,INPUT,202.110.90.80 ,ANY,202.110.90.45,8080,DENY
3 UDP,INPUT,*.*.*.*,ANY,10.110.96.255,ANY,DENY
4 UDP,OUTPUT,202.110.90.45,ANY,202.110.10.*,ANY,DENY
5 TCP,INPUT,*.*.*.*,ANY,202.110.90.45,8080,ACCEPT
6 UDP,INPUT,202.110.10.*,53,202.110.90.45,ANY,ACCEPT
7 UDP,OUTPUT,202.110.90.45,ANY,129.110.96.*,110,ACCEPT
```

Gambar 5. Generalisasi *rule* (aturan)

A.4 *Rule Improvement*

Pada tahap ini, pertama-tama dilakukan penggabungan *rule* kebijakan yang ditemukan dari *log file* dari *rule* awal firewall. Penggabungan *rule* yang dihasilkan (15 *rule*) dapat dilihat pada gambar 6.

```
TCP, INPUT, 202.110.90.80, ANY, *.*.*.*, 53, DENY
UDP, INPUT, *.*.*.*, ANY, *.*.*.*, 53, ACCEPT
TCP, INPUT, *.*.*.*, ANY, *.*.*.*, 22, DENY
TCP, INPUT, 202.110.90.80, ANY, *.*.*.*, 443, DENY
TCP, INPUT, *.*.*.*, ANY, *.*.*.*, 21, ACCEPT
TCP, INPUT, 202.110.90.86, ANY, *.*.*.*, 53, DENY
TCP, INPUT, *.*.*.*, ANY, *.*.*.*, 80, DENY
UDP,OUTPUT, *.*.*.*, ANY, *.*.*.*, ANY, DENY
TCP, OUTPUT, *.*.*.*, 53, *.*.*.*, ANY, ACCEPT
TCP, INPUT, *.*.*.*, ANY, *.*.*.*, 22, ACCEPT
TCP, OUTPUT, 202.110.90.100, ANY, 202.110.90.86, 21, ACCEPT
TCP, INPUT, 202.110.90.100, ANY, 202.110.90.86, 80, ACCEPT
TCP, INPUT, *.*.*.*, ANY, 202.110.90.86, ANY, DENY
TCP, INPUT, *.*.*.*, ANY, 202.110.90.86, 8080, ACCEPT
UDP, OUTPUT, 202.110.90.100, ANY, 202.110.90.86, 8080, ACCEPT
```

Gambar 6. *Rule* Kebijakan Gabungan Firewall

Langkah selanjutnya adalah mendeteksi kebijakan anomali dari *rule* firewall gabungan seperti yang ditunjukkan pada Gambar. 7.

```
Rule 1, Rule 2: => Generalisasi
Rule 1, Rule 14: => Shadowed
Rule 2, Rule 1: => Korelasi
Rule 2, Rule 12: => Generalisasi
Rule 3, Rule 5: => Shadow
Rule 4, Rule 7: => Shadow
Rule 4, Rule 5: => Generalisasi
Rule 5, Rule 7: => Generalisasi
Rule 7, Rule 5: => Korelasi
Rule 8, Rule 2: => Shadowed
Rule 10, Rule 12: => Korelasi
Rule 12, Rule 1: => Generalisasi
Rule 13, Rule 12: => Generalisasi
Rule 14, Rule 12: => Shadow
Rule 14, Rule 13: => Generalisasi
```

Gambar 7. Output Algoritma Deteksi Anomali

Dengan menggunakan pencarian trafik anomali, telah dideteksi 7 *rule* generalisasi, 3 *rule* korelasi, dan 5 *rule shadowed*. Beberapa *rule* yang terdeteksi ini, dimanfaatkan untuk dua hal. Pertama, digunakan sebagai dasar membuat *rule* umum untuk

memperbarui *rule* kebijakan firewall. Kedua, digunakan sebagai dasar membuat *rule* yang lebih spesifik untuk *rule* final dari konfigurasi firewall. Cara ini digunakan untuk mendeteksi lalu lintas yang tidak diinginkan. *Rule* kebijakan firewall yang ada (dari 10 *rule* awal) digabungkan dengan *rule* umum (7 rule yang dihasilkan dari *file log* firewall (40.150 catatan paket)). Kemudian untuk mendeteksi anomali, terdapat 7 dari 15 *rule* digunakan untuk *rule* akhir firewall. Hasilnya adalah *rule* baru yang diinginkan, telah digeneralisasi dan bebas dari anomali sehingga menjamin kebenaran dan efisiensi konfigurasi dan proses *filtering* firewall.

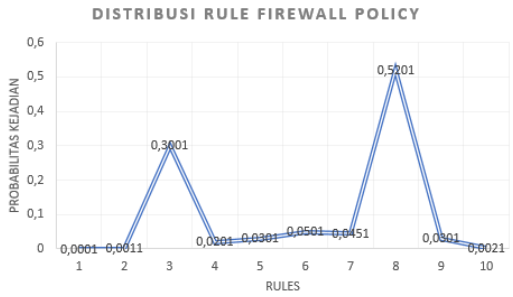
B. Rule Firewall yang dibuang dan *Rule* Firewall yang digunakan

Salah satu pertanyaan menarik yang diteliti pada makalah ini adalah bagaimana *rule* firewall pada konfigurasi awal masih berguna dan efektif untuk pola lalu lintas jaringan saat ini. Jawaban dari pertanyaan ini adalah: (1) terdapat *rule* di dalam konfigurasi yang jarang atau tidak pernah dipakai. *Rule* ini disebut sebagai rule yang membusuk, dan akhirnya *rule* ini yang dibuang dari konfigurasi firewall. (2) *rule* yang dipakai di dalam konfigurasi adalah *rule* yang mendominasi dan sering digunakan, dimana *rule* ini sebagai representasi dari sebagian besar trafik jaringan.

Hasil dari eksperimen bahwa distribusi probabilitas *log file* firewall dari 40.150 paket ditunjukkan pada tabel 2 dan gambar 8.

Tabel 3. Distribusi paket pada *filtering rule* awal

No	Src-IP	Src-Port	Dst-IP	Dst-Port	Prob
1	202.110.90.80	ANY	*.*.*.*	80	0,0001
2	*.*.*.*	ANY	*.*.*.*	80	0,0011
3	*.*.*.*	ANY	*.*.*.*	22	0,0451
4	202.110.90.80	ANY	*.*.*.*	23	0,0201
5	*.*.*.*	ANY	*.*.*.*	53	0,0301
6	202.110.90.80	ANY	*.*.*.*	22	0,0501
7	*.*.*.*	22	*.*.*.*	1024-65535	0,3001
8	*.*.*.*	1024-65535	*.*.*.*	ANY	0,0021
9	*.*.*.*	ANY	*.*.*.*	ANY	0,0301
10	*.*.*.*	ANY	*.*.*.*	53	0,5201
Total Probabilitas					1



Gambar 8. Distribusi Paket pada *Rules Filtering* Awal

Gambar 8 menampilkan distribusi probabilitas setiap rule dari 10 *rule* firewall awal, di mana sumbu x adalah nomor urutan *rule* dan sumbu y adalah skala probabilitas. Rangking setiap *rule* berdasarkan frekuensi kemunculan *rule* di dalam *log file* firewall. Probabilitas kemunculan setiap paket dihitung sebagai berikut: P = f / N, dimana probabilitas (P) sama dengan frekuensi kemunculan paket (f) dibagi dengan jumlah total paket (N) yang tercatat dalam *log file* firewall.

Dalam percobaan yang dihasilkan selama satu minggu, didapatkan 10 *rule* awal dari *log file* firewall. Gambar 8 menunjukkan bahwa *rule* 3 dan *rule* 7 adalah *rule* yang dominan, mencakup lebih dari 70% lalu lintas jaringan, dan *rule* 1 bisa menjadi *rule* yang usang atau dibuang.

Untuk setiap *rule* dalam konfigurasi umum, ditemukan frekuensi masing-masing *rule* tertentu (sumber dan tujuan tertentu) yang digunakan untuk analisis lebih lanjut. Sebagai contoh, *rule* spesifik cocok dengan *rule* 3 sebagai salah satu *rule* paling banyak digunakan. Informasi ini membantu memperbarui urutan *rule* guna mengoptimalkan pencocokan pemfilteran secara *real-time* di firewall. Grafik pada gambar 8 adalah distribusi probabilitas dari *rule* 10 yang menghasilkan 16 *rule*.

Gambar 9. Distribusi Paket pada *Rules Filtering* Awal

Sumbu x adalah sumbu setiap *rule* sesuai dengan urutannya dan sumbu y adalah skala probabilitas. Distribusi probabilitas dari setiap *rule* berbeda menggambarkan populasi pola lalu lintas jaringan yang tidak terdistribusi secara merata tetapi terkonsentrasi menjadi beberapa *rule*. Hal ini menjadi indikator yang positif dan kuat dengan berfokus pada beberapa *rule* unik. Dengan menyusun ulang atau memprioritaskan sebagian kecil dari *rule* firewall, seseorang dapat mengharapkan peningkatan kinerja dari fungsi firewall.

Dalam contoh ini, *rule* 16 menjadi kandidat pertama yang dianggap sebagai *rule* dominan. Sementara itu, *rule* 1 menjadi kandidat *rule* yang membusuk (*rule* kandidat yang akan dibuang) dari paket yang mengalir selama periode waktu tertentu, sebagaimana ditunjukkan pada gambar 10.

Gambar 10. Frekwensi Paket pada *Rule* 1 dalam 4 minggu

Gambar 10 menunjukkan distribusi probabilitas *rule* 1 yang ditambang (*mining)* dari *file log* yang dihasilkan setiap minggu selama sebulan, dimana sumbu x adalah dalam mingguan dan sumbu y adalah skala probabilitas.

Distribusi tersebut menegaskan kemungkinan *rule* yang mengikuti siklus hidup dan akhirnya membusuk setelah jangka waktu tertentu. Ini adalah salah satu indikator yang menjanjikan untuk mengkonfirmasi salah satu tujuan penelitian ini untuk menemukan rule yang membusuk dan memperbaharui *rule* firewall dalam waktu yang tepat. *Rule* yang tidak berguna dibuang dan *rule* yang mendominasi adalah *rule* yang mencakup bagian dari lalu lintas jaringan signifikan.

Dengan demikian telah didapatkan *rule* kebijakan yang lebih berguna, efektif dari *rule* firewall awal. Hal ini karena *rule* terakhir yang ada telah memberikan respon yang adaptif, dalam arti sudah sesuai dengan pola lalu lintas atau serangan (*attacking*) pada jaringan saat ini.

KESIMPULAN DAN SARAN

Dalam makalah ini, kami telah menyajikan proses baru dari pengelolaan kebijakan *rule* firewall, yang terdiri dari deteksi anomali, generalisasi dan perbaikan kebijakan menggunakan Algoritma *Association Rule Mining* dan dibantu Algoritma Apriori. Hasil yang dicapai: (a) *rule* mengambil data awal dari *log* firewall linux yang mencerminkan kapasitas tren lalu lintas jaringan saat ini, (b) membuat keputusan pola trafik data berupa analisis dasar dan deteksi anomali termasuk lalu lintas yang tersembunyi, (c) menggunakan teknik mining untuk menangani atribut diskrit dan *real time* agar beroperasi secara efisiensi dan fleksibel, dan (4) menganalisis *rule* firewall lama dan juga trafik anomali atau yang tersembunyi untuk mendapatkan *rule* yang lebih akurat dan efektif.

Kesimpulannya, *rule* mining ini terbukti bukan hanya salah satu dari pilihan yang layak, tetapi juga praktis, efektif dan optimal untuk diterapkan sebagai kebijakan *rule* firewall secara *real time*.

DAFTAR PUSTAKA

[1] K. Neupane, R. Haddad dan L. Chen, *Next Generation Firewall for Network Security: A Survey*, Southeast Conf. 2018, 2018, pp. 1-6, doi: 10.1109/SECON.2018.8478973.

[2] M. G. Mihalos, S. I. Nalmpantis dan K. Ovaliadis, *Design and Implementation of Firewall Security Policies using Linux Iptables*, Journal of Engineering Science and Technology Review 12 (1) (2019) 80 – 86.
(2021) The E-security Planet website. [Online], *Top Threat Intelligence Platform 2021*, https://www.esecurityplanet.com/products/threat-intelligence-platforms/, tanggal akses: 16 Juni 2021.

[4] Kristian Valentin, Michal Maly, *Network Firewall Using Artificial Neural Networks*, Computing and Informatics, Vol. 32, 2013, 1312-1327

[5] Partha Chakraborty, Md. Zahidur Rahman, dan Saifur Rahman, *Building New Generation Firewall Including Artificial Intelligence*, International Journal of Computer Applications (0975 - 8887) Volume 178 - No.49, September 2019.

[6] Sanjay Rathee, Arti Kashyap, *Adaptive-Miner: An Efficient Distributed Association Rule Mining Algorithm on Spark*, Journal of Big Data, Springer (2018) 5:6, https://doi.org/10.1186/s40537-018-0112-0.

[7] Ish Nath Jha, Samarjeet Borah, *An Analysis on Association Rule Mining Techniques*, Special Issue of International Journal of Computer Applications (0975 – 8887) International Conference on Computing, Communication and Sensor Network (CCSN) 2012.

[8] C. Wang, X. Zheng, *Application of Improved Time Series Apriori Algorithm by Frequent Itemsets in Association Rule Data Mining Based on Temporal Constraint*. Evolutionary Intelligent 13, Springer, 39–49 (2020). https://doi.org/10.1007/s12065-019-00234-5