Implementasi *Intrusion Prevention System* (IPS) Menggunakan Suricata Untuk Serangan SQL *Injection*

Faula Tanang Anugrah[1], Syariful Ikhwan[2], Jafaruddin Gusti A.G[3]

[1,2,3]Program Studi Teknik Telekomunikasi,
Fakultas Teknik Telekomunikasi dan Elektro,
Institut Teknologi Telkom Purwokerto
[1]faulatanang@gmail.com, [2]syariful@ittelkom-pwt.ac.id, [3]jafaruddin@ittelkom-pwt.ac.id

Abstrak

Keamanan jaringan merupakan hal yang penting untuk dilakukan sebagai pencegah penyalahgunaan sumber daya yang tidak sah. *Intrusion Prevention System* (IPS) merupakan salah satu *tool* pengamanan pada jaringan. Pada penelitian ini Suricata sebagai IPS untuk melindungi webserver dari serangan SQL *Injection* menggunakan SQLMap dengan melihat efektivitas *rules* dan parameter *response time*. Penelitian ini dilakukan di dalam Laboratorium PSD menggunakan topologi jaringan LAN yang di *setting static*. Suricata sebagai *tool* IPS terinstall pada PC yang berfungsi sebagai router sekaligus server IPS. *Normal user* dan *attacker* menggunakan sistem operasi Windows 10 sedangkan webserver dan server IPS menggunakan sistem operasi Ubuntu 20.04. Pengujian *response time* parameter dilakukan sebanyak 30 kali selama serangan SQL *Injection* berlangsung. Hasil dari pengujian tersebut menunjukkan nilai rata-rata ketika Suricata diterapkan 4,260633 *milliseconds*. Hal ini berarti Suricata membutuhkan waktu 4,2 ms untuk menanggapi suatu paket. Suricata pada penelitian ini berperan sebagai IPS yang bekerja setiap terjadinya serangan SQL *Injection* akan dideteksi oleh Suricata dengan melakukan pengecekan kecocokan paket terhadap *signature rules*. *Rules* yang dinilai efektif untuk menghadapi serangan SQL *Injection* adalah *rules* yang menggunakan beberapa kode ascii sebagai kata kuncinya.

Kata kunci: keamanan jaringan, *Intrusion Prevention System*, *Suricata rules*, SQL *Injection*, *response time*

Abstract

Network security is an important thing to prevent unauthorized use of resources. An intrusion Prevention System (IPS) is one of the security tools on the network. In this study, Suricata as an IPS to protect the webserver from SQL Injection attacks using SQLMap by looking at the effectiveness of the rules and response time parameters. This research was conducted in the PSD Laboratory using a LAN network topology that is set to static. Suricata is an IPS tool installed on a PC that functions as a router as well as an IPS server. Normal users and attackers use the Windows 10 operating system, while the web server and IPS server use the Ubuntu 20.04 operating system. The response time parameter testing was performed 30 times during the SQL Injection attack. The results of these tests show the average value when Suricata is applied at 4.260633 milliseconds. This means that Suricata takes 4.2 ms to respond a packet. Suricata in this study acts as an IPS that works every time a SQL Injection attack occurs, it will be detected by Suricata by checking the packet's compatibility against the signature rules. Rules that are considered effective to deal with SQL Injection attacks are rules that use some ASCII code as keywords.

Received: 22 Mei 2022 Revised: 16 Juli 2022 Accepted: 30 Juli 2022

Keywords: network security, Intrusion Prevention System, Suricata rules, SQL Injection, response time

1. Pendahuluan

Perkembangan teknologi informasi yang diiringi dengan kebutuhan akan teknologi dan jaringan komputer semakin meningkat. Salah satu dampak dari perkembangan teknologi adalah informasi dan data dapat dengan mudahnya diperoleh dari pengguna ke pengguna [1]. Kemudahan dalam pertukaran informasi tersebut juga dapat memunculkan *cybercrime*. Perlunya antisipasi risiko ancaman terhadap penyalahgunaan sumber daya yang tidak sah merupakan hal yang penting dilakukan. Terdapat beberapa sistem keamanan jaringan yang ada seperti *firewall* untuk menghentikan paket data yang tidak diizinkan, dan *cryptography* dengan enkripsi data. Selain itu juga terdapat sistem pendeteksian penyusup atau *Intrusion Detection System* (IDS) dan sistem pencegahan penyusup atau *Intrusion Prevention System* (IPS) yang dapat diterapkan untuk melakukan pengamanan pada jaringan. Salah satu *tools* yang digunakan untuk sistem monitoring jaringan dari ancaman penyusupan yaitu Suricata [2].

Suricata merupakan IDS, IPS, dan alat monitoring keamanan jaringan yang berbasis *open-source*. Suricata adalah sebuah *tool* keamanan jaringan dengan performa tinggi yang memiliki kemampuan *multi-threaded.* Suricata mampu mendeteksi gangguan secara *real-time*, pencegahan intrusi *inline* (IPS), pemantauan keamanan jaringan (NSM), dan pemrosesan PCAP *offline*. Suricata memerikas *traffic* jaringan menggunakan *rules* dan *signature* yang kuat dan Lua *scripting* untuk mendukung pendeteksianan serangan yang kompleks [3].

SQL *Injection* merupakan teknik penyerangan yang ditujukan ke webserver dengan menggunakan kode SQL untuk memanipulasi database. Serangan yang akan merugikan banyak orang karena jenis serangan ini akan menyerang webserver dan penyerang dapat mencuri maupun mengubah data yang ada. Injeksi SQL merupakan teknik yang memungkinkan penyerang untuk memasukan perintah (*query*) SQL yang berbahaya sehingga dapat memanipulasi logika perintah SQL untuk mendapatkan akses ke database dan informasi penting lainnya. Penyerang dapat mempengaruhi *syntax* SQL, kekuatan, fleksibilitas dari database, dan mempengaruhi sistem operasi untuk database [4].

Response time adalah waktu tanggap yang diberikan oleh *interface* ketika pengguna mengirimkan *request* ke server. *Response time testing* merupakan pengujian *response time* yang mengacu pada waktu yang dibutuhkan sebuah node dalam sistem untuk merespons permintaan dari node lain. Waktu yang dibutuhkan ketika sistem atau CPU mengambil langkah terhadap suatu *input* tertentu sampai proses selesai. *Response testing* memiliki dua karakteristik yaitu waktu tanggapan rata-rata dan waktu tanggapan maksimum [5].

Penelitian yang dilakukan oleh Bagas Suryo Anggoro dan Wiwin Sulistyo pada tahun 2019 yang berjudul "Implementasi *Intrusion Prevention System* Suricata dengan *Anomaly-Based* untuk Keamanan Jaringan PT. Grahamedia Informasi" bertujuan untuk mengimplementasikan IPS sebagai sistem keamanan jaringan karena IPS memanfaatkan firewall yang akan mendeteksi serangan berbasis *port* dan protokol dan menolak akses, serta mencatat log yang teridentifikasi negatif. Pada penelitian ini Suricata bekerja berdasarkan *anomaly-based*, setiap paket masuk diseleksi menggunakan *rules* Suricata dengan membandingkan aktivitas yang sedang di-monitoring dengan aktivitas sebelum dipantau [4] . Elsa Stephani, Fitri Nova, dan Ervan Asri melakukan penelitian pada tahun 2020 berjudul "Implementasi dan Analisa Keamanan jaringan IDS (*Intrusion Detection*

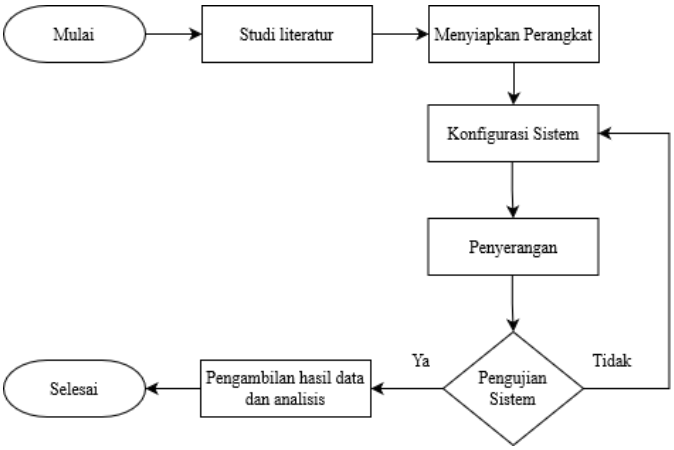
System) Menggunakan Suricata Pada Webserver" secara garis besar memaparkan tentang penerapan Suricata sebagai IDS yang dibangun menggunakan firewall OPNsense dapat digunakan untuk mendeteksi maupun mencegah anomali pada webserver dari serangan DDoS dan web *scanning*, serta Suricata tidak memiliki *shared object rules* [6].

Penelitian pada artikel ini dirancang berdasarkan beberapa penelitian yang telah dilakukan. Penelitian ini dilakukan dengan tujuan untuk mengetahui proses perancangan sebuah sistem pencegahan serangan SQL *Injection* dengan mengimplementasikan *rules* pada Suricata menggunakan metode *signature -based* serta untuk mengetahui *kinerja* sistem berdasarkan parameter *response time*.

2. Metode Penelitian

2.1 Alur Penelitian

Alur penelitian digunakan sebagai pedoman selama melakukan penelitian agar hasil yang diperoleh tidak menyimpang dari tujuan. Tahapan-tahapan kerja disusun dalam bentuk *flowchart* seperti pada Gambar 1.



Gambar 1. Diagram alur penelitian

2.2 Perangkat yang digunakan

Pada tahap menyiapkan perangkat ini perlu adanya sebuah sistem penunjang alat dan bahan yang sesuai dengan kebutuhan dalam proses melakukan implementasi meliputi *software* dan *hardware*. Berikut adalah spesifikasi perangkat yang akan digunakan:

a. Perangkat keras (*Hardware*)

Tabel 1 menampilkan spesifikasi dari *hardware* yang digunakan untuk menunjang penelitian Implementasi *Intrusion Prevention System* (IPS) menggunakan Suricata untuk serangan SQL *Injection*.

Tabel 1. Spesifikasi perangkat keras

Jenis *hardware*	Kegunaan	*Operating System*	RAM
Personal Computer (PC)	a) *router & server* IPS	Linux Ubuntu 20.04	8 GB
	b) attacker	Windows 10	
	c) normal user		
Laptop	webserver	Linux Ubuntu 20.04	4 GB
Switch	sebagai penghubung antara webserver dengan server nips	-	-

b. Perangkat lunak (*Software*)

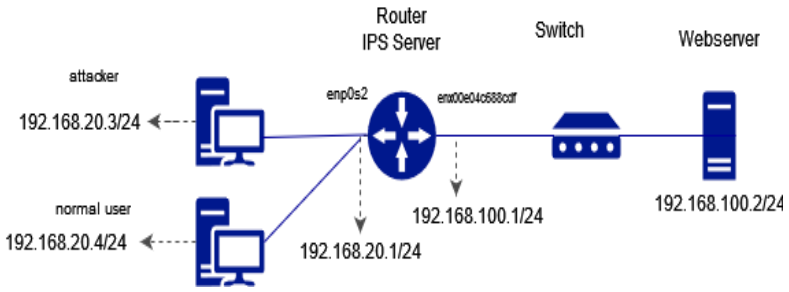
Tabel 2 menampilkan daftar dan spesifikasi dari *software* yang digunakan untuk menjalankan penelitian baik dari sisi server IPS, *client* (*attacker* dan *normal user*), maupun dari sisi webserver.

Tabel 2. Spesifikasi perangkat lunak

Jenis *Software*	Versi	Fungsi
Ubuntu server	20.04	Sistem operasi server IPS dan webserver
Windows	10	Sistem operasi *attacker* dan *normal user*
Suricata	6.0.4	Sistem keamanan
SQLMap	1.5.3.18	Penetrasi
Apache	2.4.41	Layanan webserver
MySQL	8.0.28	Database
DVWA	1.10	*Website*

2.3 Konfigurasi Jaringan

Gambar 2 menjelaskan bahwa penelitian ini menggunakan jenis topologi LAN dengan alamat IP di–*setting static* menggunakan *network address* 192.168.100.0/24. Konfigurasi jaringan dilakukan pada sebuah PC yang difungsikan sebagai router dengan ketentuan server IPS menggunakan alamat IP 192.168.100.1/24, dan 192.168.100.2/24 sebagai IP *address* dari webserver. *Network address* 192.168.20.0/24 digunakan sebagai IP *client*.



Gambar 2. Topologi jaringan

2.4 Konfigurasi Server IPS

a. Instalasi Suricata

Instalasi Suricata diawali dengan mengunduh file Suricata pada web resmi dan kemudian di-*install* dengan menggunakan perintah `apt-get install suricata -y`. Suricata pada penelitian ini berfungsi sebagai IPS, maka diperlukan beberapa paket tambahan agar IPS dapat berjalan dengan baik. Paket tambahan tersebut dapat di-*install* dengan menggunakan perintah seperti berikut

```
apt-get install libnetfilter-queue-dev libnetfilter-queue1   libnetfilter-
log-dev libnetfilter-log1 libnfnetlink-dev libnfnetlink0 -y
```

b. Konfigurasi file `suricata.yaml`

Suricata menggunakan format .yaml sebagai konfigurasi yang terletak pada direktori `/etc/suricata/`. File `suricata.yaml` merupakan file konfigurasi utama di mana di dalam file tersebut terdapat pengaturan seperti menentukan *network address* yang akan digunakan, menentukan interfaces yang akan dipindai oleh Suricata, letak file *rules* yang akan diterapkan, dan menentukan *output* file log.

c. Konfigurasi file *rules*

File konfigurasi *rules* berisikan *rules* yang akan diterapkan pada Suricata. Penyimpanan file ini memiliki direktori *default* yang berada di `/etc/suricata/rules/`. Sebuah *rules* pada Suricata terdiri dari tiga bagian utama yaitu *action*, *header* dan *rule option*. Bagian *action* merupakan bagian yang berfungsi untuk menentukan tindakan yang diperlukan ketika *signature* cocok dengan paket. Beberapa *action* yang dapat digunakan yaitu *alert*, *pass*, *drop*, dan *reject*. Selanjutnya bagian *header*, pada bagian ini dibagi menjadi beberapa bagian yaitu *protocol, source and destination, ports* (*source and destination*), dan *direction*. Protokol yang mendeklarasikan protokol apa yang menjadi fokus Suricata. Protokol dasar terdiri dari tcp, udp, icmp, dan ip, sedangkan untuk protokol pada layer 7 dapat menggunakan http, ftp, smb, dns, ssh, dan lain sebagainya. Setelah protokol terdapat *source and destination* yang mendeklarasikan tentang sumber dan tujuan paket. Bagian *ports* (*source and destination*) berarti *port* yang akan menjadi perhatian Suricata. Kemudian pada bagian direction merupakan bagian yang menentukan ke arah mana *signature* dicocokkan. Pada umumnya setiap *signature* memiliki arah ke kanan (->), namun terdapat kemungkinan untuk memiliki aturan yang cocok dua arah (< >). Bagian *rule* yang terakhir yaitu *rules option* yang diapit oleh tanda kurung () dan dipisahkan oleh *semicolons* (;). Pada bagian ini umumnya berisi seperti pesan (msg), *keyword*, sig, dan rev.

d. Konfigurasi file *output*

File *output* (*alert* dan *events*) Suricata secara *default* tersimpan pada direktori `/var/log/suricata/`. Direktori penyimpanan file *output* dapat diubah dengan menggunakan *command* `-l` atau juga dapat mengubah konfigurasi di dalam file suricata.yaml. Namun pada penelitian ini menggunakan direktori *default* sebagai direktori penyimpanan file *output*.

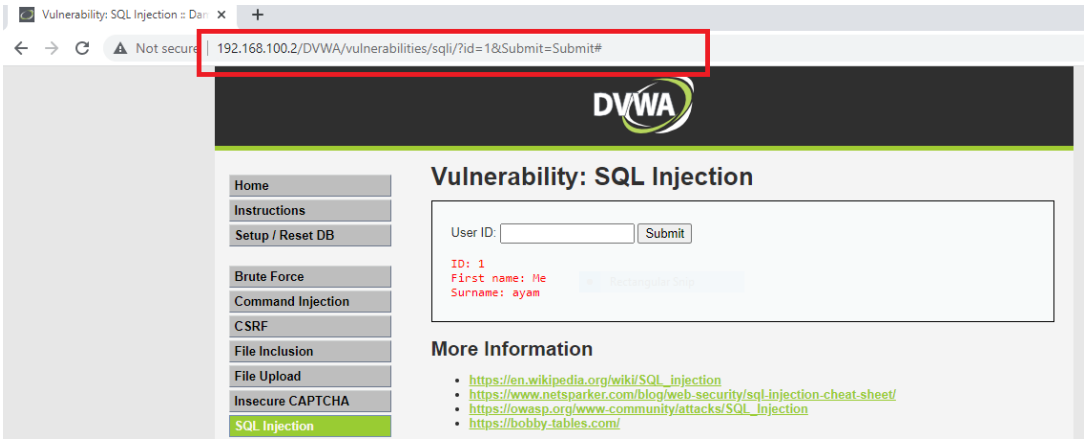
e. Konfigurasi IPTables

IPTables pada penelitian ini menggunakan sebuah *tool* yaitu NFQUEUE, di mana *tool* tersebut nantinya yang akan mengirim paket ke Suricata. IPTables bertugas sebagai firewall akan melakukan penyaringan paket (*packet filtering*) yang masuk ke *interface* `enxd03745243baf` (LAN) kemudian paket tersebut diarahkan ke NFQUEUE dan dikirimkan ke Suricata untuk dicocokan dengan *signature rules* yang telah dibuat. Ketika paket tersebut sesuai dengan *signature* maka akan di-*drop*.

3. Hasil dan Pembahasan

Pengujian dilakukan untuk mengetahui kinerja sistem Suricata yang telah dirancang sesuai dengan skenario pengujian yang telah dibuat. Skenario pertama pengujian dilakukan tanpa adanya keamanan dari Suricata (Suricata nonaktif) dan skenario kedua dilakukan ketika sesudah adanya keamanan dari Suricata (Suricata aktif). Saat kondisi

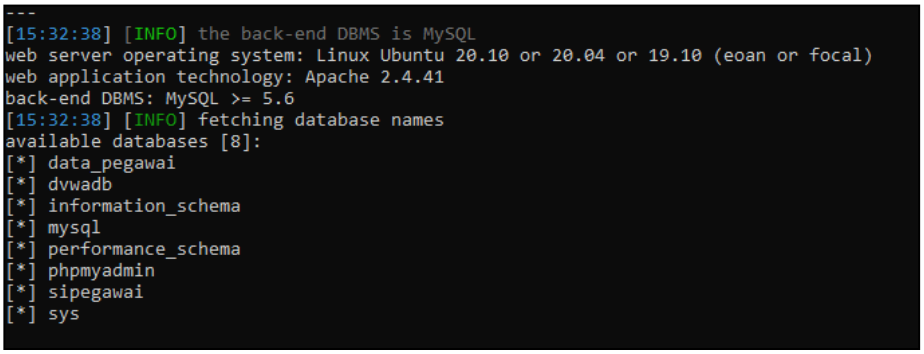
skenario pengujian tersebut dilakukan pengambilan nilai parameter *response time* ketika terjadi serangan SQL *Injection* yang bertujuan untuk mengetahui performa sistem keamanan dari sisi normal user. Pengujian serangan SQL *Injection* dilakukan dari sisi penyerang (*attacker*) menggunakan sebuah *tool* SQLMap. Selain melalui parameter *response time* kinerja sistem Suricata juga diukur dari efektivitas *rules* yang diterapkan untuk menghadapi berbagai jenis serangan SQL *Injection*.



Gambar 3. Tampilan DVWA web

DVWA web digunakan sebagai objek penyerangan SQL *Injection* pada penelitian ini. DVWA web merupakan sebuah *website* yang memiliki kelemahan atau *vulnerability* yang rentan untuk diserang. Gambar 3 merupakan tampilan dari DVWA web ketika user sudah berhasil *login*. Pada menu SQL *Injection* ketika memasukan user id 1 maka akan muncul informasi dari pengguna tersebut. Ketika melakukan submit maka akan menampilkan url seperti pada gambar tersebut yang merupakan salah satu bentuk celah keamanan pada *website*. Celah keamanan tersebut yaitu munculnya parameter `id=1&Submit=Submit` pada url ketika memasukan user id yang termasuk kedalam query sql. Kelemahan tersebut dapat dimanfaatkan oleh penyerang untuk melakukan SQL *Injection* dengan menggunakan SQLMap yang dijalankan pada *command prompt* di Windows. Perintah pada SQLMap memiliki format `sqlmap.py -u "http://192.168.100.2/DVWA/vulnerability/sqli/" --data="id=1& Submit=Submit" --(parameter injeksi).`

Pada *command* tersebut `-u` merupakan definisi untuk url, selanjutnya `--data="id=1&Submit=Submit"` merupakan celah keamanan yang terdapat pada url *website*. Pada penelitian ini menggunakan beberapa parameter injeksi seperti `--p id` untuk mengetahui ada tidaknya celah keamanan pada parameter id, `--dbs` untuk melihat database apa saja yang terdapat didalam *website*, `--tables` untuk mengetahui isi tabel yang berada didalam suatu database, `--column` untuk mengetahui isi kolom dalam suatu tabel dari suatu database, dan `--dump` berfungsi untuk melakukan pencatatan log pada database.



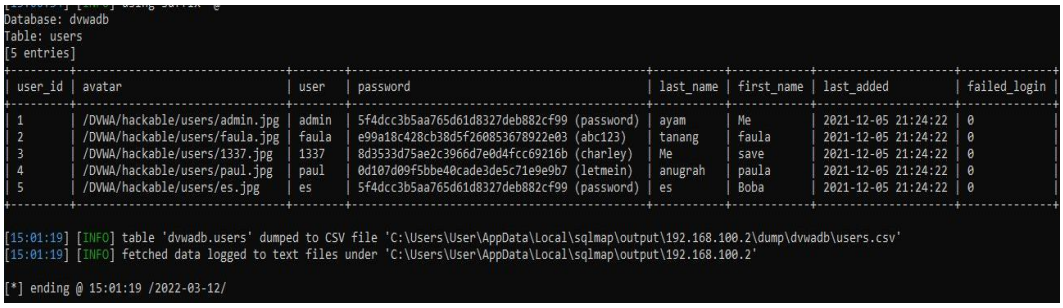
Gambar 4. Log informasi webserver

Tabel 3. Log Informasi tables serangan SQL *Injection* dengan SQLMap

Database Name	***Tables***
dvwadb	*guestbook*
	users

Tabel 4. Log informasi column serangan SQL *Injection* dengan sqlmap

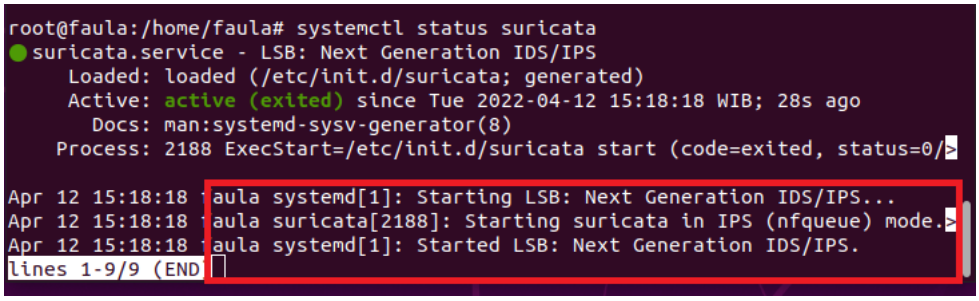
Database Name	***Column***	***Type***
dvwadb	*User*	*varchar(15)*
	Avatar	*varchar(15)*
	failed_login	*int*
	first_name	*varchar(15)*
	last_login	*timestamp*
	last_name	*varchar(15)*
	Password	*varchar(15)*
	user_id	*int*



Gambar 5. Log informasi *username & password* pada database dvwadb

Gambar 4, Tabel 3, Tabel 4, dan Gambar 5 merupakan log hasil dari pengujian serangan SQL *Injection* dengan menggunakan SQLMap. Gambar 4 merupakan informasi dari webserver yang menggunakan OS berbasis Linux Ubuntu versi 20.10/20.04/19.10 (eoan atau focal), Apache versi 2.4.41, dan MySQL versi lebih dari 5.6 untuk databasenya. Selain mendapatkan informasi sistem webserver juga mendapatkan informasi mengenai database apa saja yang ada di dalam server salah satunya yaitu dvwadb yang merupakan database dari DVWA web. Pada Tabel 3 dan 4 adalah log informasi tabel dan kolom yang berada di dalam database dvwadb. Gambar 5 merupakan tampilan informasi paling penting dari sebuah database dvwadb dapat terpecahkan. Berdasarkan log informasi yang telah didapatkan terbukti bahwa melalui serangan SQL *Injection* penyerang dapat

mengetahui mulai dari informasi umum seperti jenis dan versi *Operating System* yang digunakan oleh webserver hingga informasi penting seperti *username* dan *password* dari masing-masing pengguna.



Gambar 6. Status Suricata

Gambar 6 merupakan hasil konfigurasi server IPS. Suricata berfungsi sebagai nips yang merupakan sistem pencegahan untuk memantau dan melindungi *host* di jaringan global daripada secara khusus memantau satu *host*. *Tool* yang digunakan untuk menjalankan fungsi tersebut adalah nfqueue. Setelah mendapatkan hasil pengujian serangan SQL *Injection* dengan SQLMap tersebut selanjutnya melakukan pengujian terhadap server IPS. Pengujian ini untuk melihat seberapa efektif *rules* yang telah diterapkan ke dalam Suricata untuk menghadapi berbagai jenis serangan SQL *Injection* (blind sqli, error sqli, dan *UNION* sqli). Pengujian menggunakan tiga macam *rules* yang diimplementasikan kedalam sistem. Ketiga *rules* tersebut didapat dari sumber yang berbeda yang bertujuan untuk membandingkan kinerja dari masing-masing *rule*. Gambar 7 merupakan kode program *rule* yang pertama yang diterapkan berasal dari *rules default* bawaan Suricata.

```
drop http any any -> 192.168.100.2 $HTTP_PORTS (msg:"ET WEB_SERVER Possible SQL Injection
Attempt UNION SELECT"; flow:established,to_server; content:"UNION"; http_uri; nocase;
content:"SELECT"; http_uri; nocase; pcre:"/UNION.+SELECT/Ui";
reference:url,doc.emergingthreats.net/2006446; classtype:web-application-attack; sid:2006446;
rev:13; metadata:affected_product Web_Server_Applications, attack_target Web_Server, created_at
2010_07_30, deployment Datacenter, signature_severity Major, tag SQL_Injection, updated_at 2020_
09_01;)
```

Gambar 7. Kode program *rule* SQL *Injection* pertama

Rule tersebut akan memberikan *alert* `ET WEB_SERVER Possible SQL Injection Attempt UNION SELECT` saat terjadinya serangan SQL *Injection* yang membawa paket berisikan keyword *UNION* dan *SELECT*. Ketika melakukan pengujian dengan menggunakan *rule* ini, sistem Suricata hanya dapat mengenali satu jenis serangan SQL *Injection* yaitu *UNION* SQL *Injection* sehingga serangan SQL *Injection* yang lain masih dapat berhasil lolos. Sehingga dilanjutkan dengan penerapan *rule* yang kedua yang ditunjukkan Gambar 8.

```
drop tcp any any -> 192.168.100.2 $HTTP_PORTS (msg:"SQL Injection
Detected";flow:established,to_server; content:"id"; nocase; http_uri;
pcre:"/(and\W+select)|(union.*select)|(or|and\d+=\d+)|(\'.\-\-)/Ui";
classtype:web-application-attack; sid:1000005; rev:1;)
```

Gambar 8. Kode program *rule* SQL *Injection* kedua

Rule kedua ini merupakan *rule* pengembangan dari *rule* pertama dengan keyword yang lebih spesifik. Ketika dilakukan pengujian serangan SQL *Injection* dengan menggunakan SQLMap, Suricata masih dapat meloloskan semua serangan yang SQLMap kirimkan. Namun ketika melakukan serangan SQL *Injection* secara manual (tanpa *tool*) yaitu dengan memasukkan *query* SQL pada website, sistem Suricata dapat mendeteksi dan juga memblokir semua jenis serangan SQL *Injection*. Kemudian penerapan *rule* yang terakhir berikut ini dibuat dengan meng-encode kode ASCII kedalam pengkodean UTF-8 pada subbagian *rule option* pcre, seperti ditunjukkan Gambar 9.

```
drop tcp any any -> 192.168.100.2 $HTTP_PORTS (msg:"SQLInjection - Start Attacks 1..... - SQL";
pcre:"/(((((((\%6f)|(o)|(\%4f))((\%52)|(r)|(\%72))(%20))|(((\%6f)|(o)|(\%4f))((\%52)|(r)|(\%72))((%
2b)|(\+)))))|((\%0a))|(((((\%6f)|(o)|(\%4f))((\%72)|(r)|(\%52))((\/)|(\%2f))((\*)|(\%2a)))))|(((((\%
4c)|(l)|(\%6c))((\%69)|(i)|(\%49))((\%6b)|(k)|(%4b))((\%65)|(e)|(\%45)))))|(((((\%63)|(c)|(\%43))((\%
6f)|(o)|(\%4f))((\%6e)|(n)|(\%4e))((\%63)|(c)|(\%43))((\%61)|(a)|(\%41))((\%74)|(t)|(\%54))((\%
76)|(v)|(\%56))((\%65)|(e)|(\%45))((\%72)|(r)|(\%52))((\%73)|(s)|(\%53))((\%69)|(i)|(\%49))((\%
6f)|(o)|(\%4f))((\%6e)|(n)|(\%4e)))|(((\%68)|(h)|(\%48))((\%6f)|(o)|(\%4f))((\%73)|(s)|(\%53))((\%
74)|(t)|(\%54))((\%6e)|(n)|(\%4e))((\%61)|(a)|(\%41))((\%6d)|(m)|(\%4d))((\%65)|(e)|(\%45)))|(((\%
55)|(u)|(\%75))((\%55)|(u)|(\%75))((\%49)|(i)|(\%69))((\%44)|(d)|(\%64)))|(((\%64)|(d)|(\%44))((\%
61)|(a)|(\%41))((\%74)|(t)|(\%54))((\%61)|(a)|(\%41))((\%64)|(d)|(\%44))((\%69)|(i)|(\%49))((\%
72)|(r)|(\%52)))))))/i";classtype:Web-application-attack; sid:1; rev:1;)
```

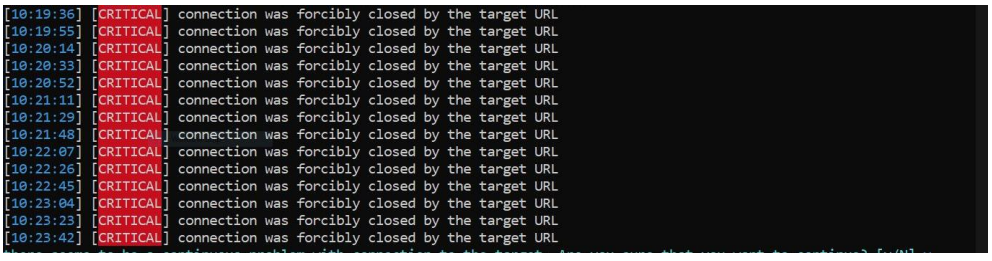
Gambar 9. Kode program *rule* SQL *Injection* ketiga

Rule tersebut dapat mendeteksi semua paket yang mengandung (OR%20), (OR+), (%0A), (OR/*), (LIKE), (CONCAT), maupun *hexadecimal* untuk huruf besar dan huruf kecil. Ketika dilakukan pengujian terhadap serangan SQL *Injection*, Suricata dapat mendeteksi dan juga melakukan *drop* paket semua jenis serangan SQL *Injection* yang masuk baik secara manual maupun dengan SQLMap. Hasil dari perbandingan *rules* dapat dilihat pada Tabel 5.

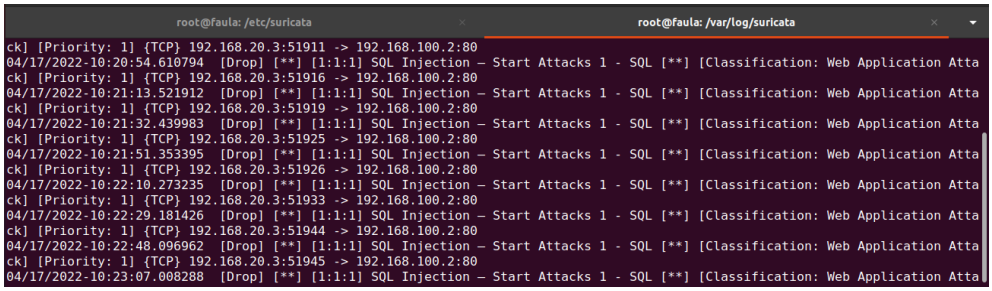
Tabel 5. Hasil penerapan *rules* terhadap SQL *INJECTION*

Rule	***Blind SQL Injection***	***Error SQL Injection***	***UNION SQL Injection***
1	*No*	*Yes*	*Yes*
2	*Yes*	*Yes*	*Yes*
3	*Yes*	*Yes*	*Yes*

Hasil pengujian server IPS ini juga dapat dilihat dari log Suricata dan tampilan SQLMap gagal melakukan serangan seperti pada Gambar 10.



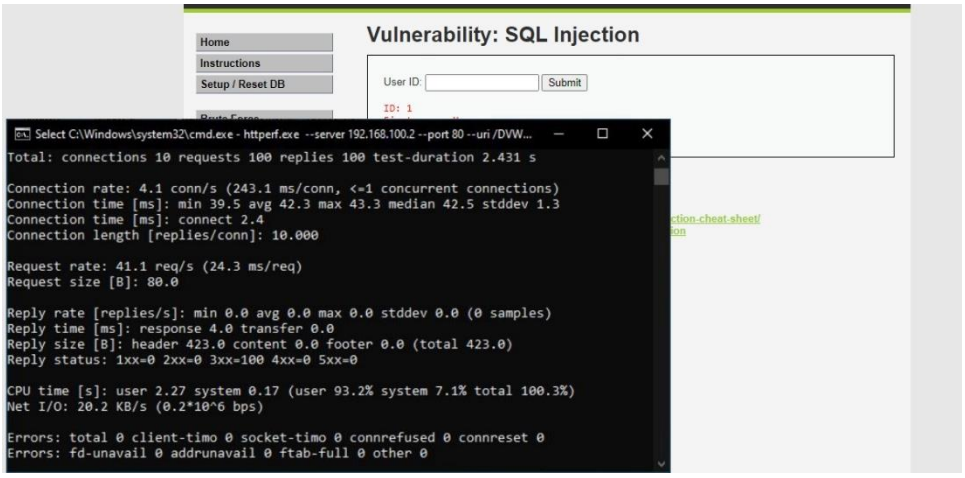
(a)



(b)

Gambar 10. (a) Tampilan ketika SQLMap terblokir, (b) Tampilan Log Suricata

Gambar 10 merupakan tampilan ketika paket yang dikirimkan oleh SQLMap sesuai dengan *signature rules*, maka Suricata akan langsung melakukan *dropping packet*. Gambar 10 (a) tampilan SQLMap dari sisi *attacker* yang gagal melakukan penyerangan sedangkan Gambar 10 (b) tampilan file `hasil.log` dari sisi server IPS Suricata ketika terjadi serangan.



Gambar 11. Tampilan *tool* httperf

Kemudian melakukan pengujian untuk mengetahui performa server IPS melalui parameter *response time*. Gambar 11 merupakan tampilan ketika mengambil nilai berdasarkan pengujian parameter *response time* dengan *tool* httperf. Pengujian parameter ini dilakukan selama 30 kali saat Suricata aktif. Parameter *response time* diambil dalam satuan *milliseconds*. Tabel 6 merupakan hasil pengujian parameter *response time*. Ketika serangan SQL *Injection* berlangsung terdapat kenaikan *response time* dari sisi *normal user*, hal ini dapat mengganggu *traffic* yang berjalan pada *user* saat akan mengakses *website*. Meskipun tidak terlalu berpengaruh terhadap kinerja server IPS dan webserver, namun tetap saja *request* yang dilakukan oleh SQLMap dalam melakukan serangan SQL *Injection* dapat membuat kinerja database DVWA web menjadi terganggu. Ketika Suricata aktif nilai *response time* terendah sebesar 3,563 *milliseconds* pada pengujian ke 2 dan untuk nilai tertingginya sebesar 4,863 *milliseconds* pada pengujian ke 30. Hasil pengujian saat terjadinya serangan pada tabel menunjukkan bahwa ketika Suricata aktif rata-rata *response time* 4,260633 *milliseconds*.

Tabel 6. Hasil pengujian *response time* saat terjadinya serangan

Pengujian	***Response time (ms) Suricata aktif***
Pengujian 1	3,823
Pengujian 2	3,563
Pengujian 3	3,586
Pengujian 4	3,956
Pengujian 5	4,013
Pengujian 6	3,926
Pengujian 7	3,943
Pengujian 8	3,96
Pengujian 9	4,566
Pengujian 10	4,51
Pengujian 11	4,616
Pengujian 12	4,536
Pengujian 13	4,586
Pengujian 14	4,456
Pengujian 15	4,783
Pengujian 16	4,163
Pengujian 17	4,25
Pengujian 18	4,266
Pengujian 19	4,3
Pengujian 20	4,183
Pengujian 21	4,22
Pengujian 22	4,183
Pengujian 23	4,183
Pengujian 24	4,17
Pengujian 25	4,293
Pengujian 26	4,296
Pengujian 27	4,22
Pengujian 28	4,65
Pengujian 29	4,756
Pengujian 30	4,863

4. Kesimpulan

Suricata pada penelitian ini berhasil berperan sebagai NIPS dalam melakukan deteksi dan juga pemblokiran terhadap beberapa jenis serangan SQL *Injection* yang ditujukan ke webserver. Setiap terjadinya serangan SQL *Injection* akan dideteksi oleh Suricata dengan melakukan pengecekan terhadap *signature rules* apakah terdapat kecocokan atau tidak. Berdasarkan ketiga *rules* yang telah diterapkan hanya *rule* dengan kode ASCII yang ter-encode yang dinilai efektif untuk menghadapi SQL *Injection* dikarenakan mampu mendeteksi sekaligus mem-blokir tiga jenis serangan SQL *Injection* yaitu *Blind* SQL *Injection*, *Error* SQL *Injection*, dan *UNION* SQL *Injection* baik serangan yang dilakukan dengan menggunakan *tool* SQLMap maupun serangan SQL *Injection* secara manual. Kinerja server IPS membutuhkan waktu yang lebih lama saat kondisi Suricata aktif dalam merespons suatu paket berdasarkan nilai *response time* yang telah diperoleh. Berdasarkan nilai *response time* yang telah diperoleh, Suricata membutuhkan waktu rata-rata 4,260633 *milliseconds* untuk menanggapi serangan yang masuk. Waktu respons tertingginya yaitu 4,863 *milliseconds* pada pengujian ke-30.

Daftar Pustaka

[1] M. A. Fazizi, S. J. I. Ismail and A. Sularsa, "Implementasi Sistem Pendeteksian Serangan Pada Jaringan dengan Briarids Berbasis Raspberry Pi," in *e-Proceeding of Applied Science*, Bandung, 2018.

[2] M. Syani, "Implementasi Intrusion Detection System (IDS) Menggunakan Suricata Pada Linux Debian 9 Berbasis Cloud Virtual Private Servers (VPS)," *Jurnal Infokar,* vol. 1, no. 1, pp. 13-20, 2020.

[3] D. Utomo, M. Sholeh and A. Avorizano, "Membangun Sistem Mobile Monitoring Keamanan Web Aplikasi Menggunakan Suricata dan Bot Telegram Channel," *Seminar Nasional TEKNOKA,* vol. 2, pp. 81-87, 2017.

[4] B. S. Anggoro and W. Sulistyo, "Implementasi Intrusion Prevention System Suricata dengan Anomaly-Based untuk Keamanan Jaringan PT. Grahamedia Informasi," *Seminar Nasional APTIKOM (SEMNASTIK),* pp. 280-288, 2019.

[5] R. Suwanto, I. Ruslianto and M. Diponegoro, "Implementasi Intrusion Prevention System (IPS) menggunakan SNORT dan Iptables pada Monitoring Jaringan Lokal Berbasis Website," *Coding : Jurnal Komputer dan Aplikasi,* vol. 07, no. 1, pp. 97-107, 2019.

[6] E. Stephani, F. Nova and E. Asri, "Implementasi dan Analisa Keamanan Jaringan IDS (Intrusion Detection System) Menggunakan Suricata Pada Web Server," *Jitsi,* vol. 1, no. 2, pp. 67-74, 2020.

[7] O. W. Purbo, "onno center," 30 Maret 2020. [Online]. Available: https://lms.onnocenter.or.id/wiki/index.php/Suricata_(software).

[8] anonim, "Suricata-Update," [Online]. Available: https://rules.emergingthreats.net/open/suricata/rules/. [Accessed Januari 2022].

[9] M. A. M. Yunus, M. Z. Brohan, N. M. Nawi, E. S. M. Surin, N. A. M. Najib and C. W. Liang, "215Review of SQL Injection : Problems and Prevention," *International Journal on Informatics Visualization,* vol. 2, no. 3-2, pp. 215-219, 2018.

[10] R. M. Thiyab, M. A. M. Ali, F. Basil and A. , "The impact of SQL injection attacks on the security of databases," in *Proceedings of the 6th International Conference of Computing & Informatics*, Malaysia, 2017.