

Analisis Perbandingan Mod Security dan Deep Packet Inspection pada Web Server Terhadap Serangan DDoS Slow Headers

Nanda Iryani^{1#}, Dhanar Yusuf Febriansyah², Eko Fajar Cahyadi³

¹Program Studi Teknik Komputer, Institut Teknologi Telkom Surabaya
Jl. Ketintang No. 156, Ketintang, Kec. Gayungan, Kota Surabaya, Jawa Timur 60231, Indonesia

^{2,3}Program Studi Teknik Telekomunikasi, Institut Teknologi Telkom Purwokerto
Jl. DI Panjaitan No. 128 Purwokerto Jawa Tengah, Indonesia

#nandandi@ittelkom-sby.ac.id

Abstrak

Pada proses pertukaran data dibutuhkan sebuah *web server* yang berfungsi sebagai penyedia layanan HTTP yang dapat diakses oleh *client* melalui *web browser*. Jumlah permintaan yang melebihi kapasitas menyebabkan *web server* mengalami *down* sehingga tidak dapat memproses setiap permintaan yang dikirimkan oleh *client*. Serangan *Distributed Denial of Service* (DDoS) membuat *client* sah dari sebuah jaringan tidak dapat mengakses layanan *web server*. Keamanan menjadi aspek yang harus dijaga dalam sistem jaringan komputer. Penerapan *mod security* dan *Deep Packet Inspection* (DPI) sebagai pengamanan jaringan menawarkan solusi pada isu keamanan jaringan. Penelitian ini membandingkan implementasi antara *mod security* dengan DPI yang bertujuan untuk mengetahui perbedaan performansi berdasarkan 3 parameter pengujian yaitu *CPU usage*, *delay*, dan *response time*. Pengujian dilakukan dengan 4 skenario dengan masing-masing skenario dilakukan sebanyak 10 kali. Hasil pengujian dengan parameter *CPU usage*, DPI lebih unggul dari *mod security* dengan perolehan rata-rata 12,1% untuk DPI dan 12,3% untuk *mod security*. Pada hasil pengujian dengan parameter *response time*, *mod security* lebih unggul dari DPI dengan perolehan rata-rata 146 detik untuk *mod security* dan 158,5 detik untuk DPI. Pada hasil pengujian dengan parameter *delay*, *mod security* lebih unggul dari DPI dengan perolehan nilai 2,57 ms untuk *mod security* dan 2,77 ms untuk DPI.

Kata kunci: *web server*, *mod security*, *Deep Packet Inspection*

Abstract

In the process of exchanging data, a *web server* is needed that functions as an HTTP service provider that can be accessed by clients via a *web browser*. The number of requests that exceed capacity causes the *web server* to experience a *downtime* so that it cannot process every request sent by the *client*. *Distributed Denial of Service* (DDoS) attacks prevent legitimate clients from a network from accessing *web server* services. Security is an aspect that must be maintained in a computer network system. The application of *mod security* and *Deep Packet Inspection* (DPI) as network security offers solutions to network security issues. This study compares the implementation of *mod security* with DPI which aims to determine differences in performance based on 3 test parameters, namely *CPU usage*, *delay*, and *response time*. Testing was carried out with 4 scenarios with each scenario being carried out 10 times. Test results with the *CPU usage* parameter, DPI are superior to *mod security* with an average gain of 12.1% for DPI and 12.3% for *mod security*. In the test results with the *response time* parameter, *mod security* is superior to DPI with an average acquisition of 146 seconds for *mod security* and 158.5 seconds for DPI. In the test results with the *delay* parameter, *mod security* is superior to DPI with a value of 2.57 ms for *mod security* and 2.77 ms for DPI.

Keywords: *web server*, *mod security*, *Deep Packet Inspection*

I. PENDAHULUAN

Lumpuhnya *web server* dapat mengganggu proses *client* dalam mengakses informasi dari *web server*. Jumlah permintaan yang melebihi kapasitas

menyebabkan lumpuhnya *web server*. Serangan DDoS membuat *client* sah dari sebuah jaringan tidak dapat mengakses layanan *web server* [1]. Serangan DDoS memiliki banyak varian, salah satunya yaitu serangan DDoS *slow headers* atau

sering disebut *slowris* yang bekerja dengan mengirim *header* HTTP, menambah jumlah pengiriman paket tetapi tidak pernah menyelesaikan permintaan sehingga memaksa *web server* untuk tetap menjaga koneksi tetap terbuka. Koneksi yang terbuka dapat dengan mudah untuk diambil sumber daya sehingga membuat *client* sah tidak dapat mengakses *web server* [2].

Penerapan *mod security* dan DPI sebagai pengamanan jaringan menawarkan solusi pada isu keamanan jaringan. Penggunaan *mod security* sebagai keamanan jaringan berfungsi untuk menyaring, memantau, dan memblokir lalu lintas HTTP pada *Open System Interconnection* (OSI) *layer* ke 7 [3]. *Mod security* akan menganalisis permintaan *get* dan *post* yang terdapat dalam HTTP yang kemudian dicocokkan dengan aturan *firewall* yang telah dikonfigurasi untuk kemudian dilakukan pemblokiran dan penolakan akses ke aplikasi *web* apabila terdapat sebuah lalu lintas yang mencurigakan [4]. *Mod security* mampu melakukan 80% pertahanan pada aplikasi *web*, *mod security* dapat digunakan sebagai *proxy* yang di *embed* [5]. DPI merupakan salah satu dari perkembangan *firewall* yang secara teknis merupakan kombinasi fungsi *firewall*, *Intrusion Detection System* (IDS) dan *Intrusion Prevention System* (IPS) [6]. DPI bekerja pada OSI *layer* ke 7 [7]. DPI memeriksa informasi *header* pada paket, memeriksa isi muatan pada paket dan melakukan penyaringan pada paket. DPI mencocokkan paket dengan set aturan yang telah dikonfigurasi untuk menentukan apakah paket tersebut dapat diizinkan untuk diteruskan ke *web server* atau paket tersebut diblokir [8]. Teknologi pada DPI dapat digunakan dalam mengidentifikasi secara spesifik dari mana asal muatan pada setiap paket yang dikirimkan melalui jaringan [9].

Penelitian [10] membahas mengenai perbandingan penerapan metode pengamanan *mod security* dan *mod evasive* pada *web server* terhadap serangan *slow headers* yang bertujuan untuk mengetahui metode pengamanan terbaik terhadap serangan DoS *slow headers*. Penelitian tersebut tidak terdapat parameter *CPU usage*, *delay* dan *request time*. Penelitian ini mengusulkan perbandingan dua metode untuk mengetahui penerapan pengamanan terbaik pada *web server*.

II. METODE PENELITIAN

A. Metode Penelitian

Metode penelitian yang digunakan adalah metode eksperimen. Penelitian ini mengimplementasikan *mod security* dan DPI sebagai pengamanan pada *web server*. Data pada penelitian diperoleh dengan melakukan percobaan

di laboratorium pengolahan sinyal digital Institut Teknologi Telkom Purwokerto.

B. Alur Penelitian

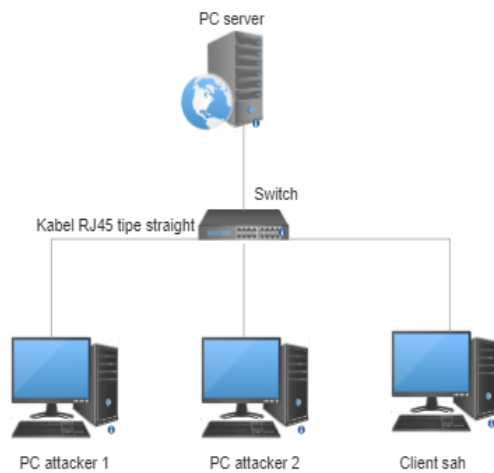
Alur penelitian terdiri dari beberapa langkah-langkah yang digunakan dalam perancangan sistem perbandingan *mod security* dengan DPI agar terstruktur dengan baik. Pada Gambar 1 disajikan alur penelitian dalam bentuk *flowchart*.

C. Topologi Jaringan

Perancangan topologi jaringan yang digunakan ditunjukkan pada Gambar 2. Perangkat yang digunakan yaitu menggunakan 1 personal komputer yang dioperasikan sebagai *web server*, 2 personal komputer masing masing berperan sebagai penyerang dan 1 personal komputer yang berperan sebagai *client* sah. Masing-masing perangkat dihubungkan menggunakan perangkat *switch* dengan media kabel RJ 45 dengan tipe *straight*.



Gambar 1. Alur tahapan penelitian



Gambar 2. Topologi jaringan

Pada *web server* berjalan dengan sistem operasi ubuntu versi 20.04 LTS. *Web server* yang digunakan adalah apache *web server* yang dilengkapi dengan sistem keamanan mod *security* dan DPI. Adapun rule pada ModSec disebut ‘SecRules’ untuk memantau traffic HTTP(S) secara real-time, pencatatan (logging), dan melakukan filter pada komunikasi HTTP(S) berdasarkan aturan yang sudah ditentukan. Sedangkan DPI atau dikenal dengan Inspeksi paket mendalam adalah jenis pemrosesan data yang memeriksa secara detail data yang dikirim melalui jaringan komputer, dan dapat mengambil tindakan seperti memperingatkan, memblokir, merutekan ulang, atau mencatatnya [7].

D. Skenario Pengujian

Pada tahapan pengujian dilakukan dengan 3 skenario pengujian dengan masing masing skenario sebagai berikut.

1. Skenario 1

Skenario 1 dilakukan dengan pengukuran sebanyak 10 kali menggunakan 3 parameter yaitu CPU *usage*, *response time* dan *delay* dalam kondisi *web server* tanpa *firewall* dan tanpa meluncurkan serangan DDoS *slow headers*.

2. Skenario 2

Skenario 2, pengujian dilakukan dengan meluncurkan serangan DDoS *slow headers* sebanyak 10 kali dalam kondisi *web server* terpasang mod *security* sebagai *firewall* dan menonaktifkan DPI.

3. Skenario 3

Pada skenario 3 yaitu meluncurkan serangan DDoS *slow headers* sebanyak 10 kali dalam kondisi *web server* terpasang DPI sebagai *firewall* dan menonaktifkan mod *security*.

```

root@attacker: /home/attacker
[attacker@attacker]~$ sudo su
[sudo] password for attacker:
[root@attacker]~/home/attacker# nmap -sS 192.168.212.2
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-04 13:38 EDT
Nmap scan report for 192.168.212.2
Host is up (0.00023s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE
80/tcp    open  http
MAC Address: B4:2E:99:90:3F:70 (Giga-byte Technology)
Nmap done: 1 IP address (1 host up) scanned in 13.23 seconds
  
```

Gambar 3. Hasil pemindaian tools Nmap

```

192.168.212.5 - - [14 / Jul / 2022
:19:07:44 +0700] "GET /login.php
HTTP/1.1" 403 439 "https : //
github.com/shekya/ slowhttptest/"
"User-Agent: Mozilla/5.0 (Windows
NT 6.1; WOW64; Trident/7.0; rv :
11.0) like GeckoAppleWebKit/534.30
(KHTML, like Gecko)
Chrome/12.0.742.122 Safari/534.30"
  
```

Gambar 4. Access log Mod security

III. HASIL DAN PEMBAHASAN

Pada tahap ini membahas tentang hasil implementasi dari mod *security* dan DPI pada *web server*. Hasil uji coba mencakup 3 skenario pengujian berdasarkan parameter CPU *usage*, *response time*, dan *delay*. Hasil data kemudian dibandingkan antara mod *security* dengan DPI untuk menentukan jenis *firewall* yang memiliki performansi lebih baik pada implementasi *web server*.

A. Hasil Pemindaian

Pada Gambar 3 menunjukkan hasil dari pemindaian *port* jaringan yang terbuka menggunakan tools Nmap yang berjalan pada sistem operasi Linux.

Hasil pemindaian menunjukkan bahwa *port* pada IP *web server* yaitu 192.168.212.2 tertutup sebanyak 999 *port* dan pada *port* 80 berstatus terbuka.

B. Implementasi Mod Security

Pada *alert* serangan menggunakan mod *security* dimulai dengan pengenalan IP penyerang yaitu “192.168.212.5” dan keterangan waktu akses dari penyerang yang ditunjukkan dengan “[14/Jul/2022:19:07:04 +0700]”. Pada kode “GET/login.php HTTP/1.1” memiliki arti IP penyerang mencoba masuk ke laman *website* yang bernama login.php dengan permintaan GET (Gambar 4).

```
[Thu Jul 14 19:07:44.603548 2022]
[:error] [pid 11879] [client
192.168.212.5:50004] [client
192.168.212.5] ModSecurity: Access
denied. Operator GE matched 5 at
TX: anomaly_score . [file " / etc
/ modsecurity / rules/REQUEST-949-
BLOCKING-EVALUATION.conf"] [line
"93"] [id "949110"] [msg "Inbound
Anomaly Score Exceeded (Total
Score: 8)"] [ver "OWASP CRS/3.3.0"]
[tag "application-multi"] [tag
"language-multi"] [tag "platform-
multi"] [tag "attack-generic"]
[hostname "192.168.212.2"] [uri
"/login.php"] referer : https :
// github . com /
shekya/slowhttpstest/
```

Gambar 5. Error log Mod security

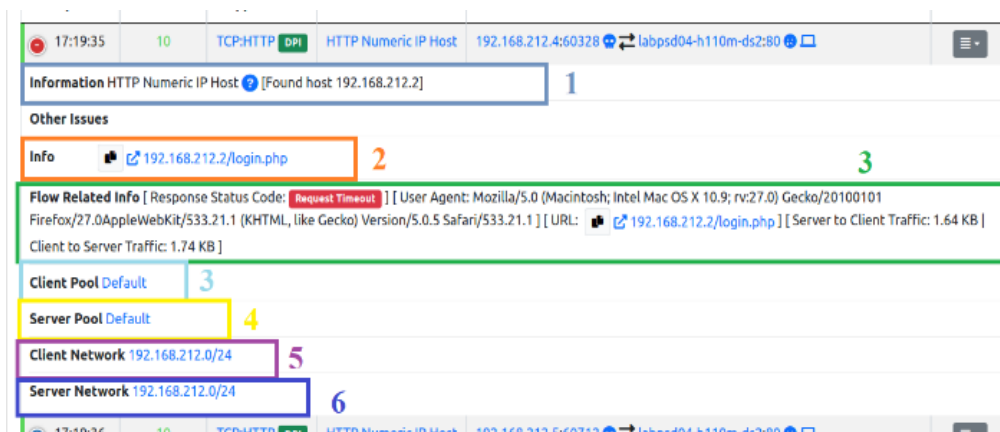
Kode “403” yang berarti alamat IP tersebut tidak memiliki izin untuk mengakses *website*. Kode “https://github.com/shekya/slowhttpstest/” menandakan bahwa *mod security* dapat mendeteksi *tools* yang digunakan oleh penyerang. Kode “User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like GeckoAppleWebKit/534.30 (KHTML, like Gecko) Chrome/12.0.742.122 Safari/534.30” adalah *user agent* yang merupakan karakteristik *string* yang memungkinkan *server* dapat mengidentifikasi sistem operasi, *browser* yang digunakan oleh penyerang.

Pada *file* konfigurasi *mod security*, dilakukan penambahan *script* “Include /etc/modsecurity/rules/*.conf” tepat dibawah *script* “IncludeOptional /etc/modsecurity/*.conf”. Penambahan *rules* tersebut bertujuan agar pada saat terjadi serangan maka seluruh *rules* yang terdapat pada *folder* *etc/modsecurity/rules* dapat bekerja menangani serangan tersebut.

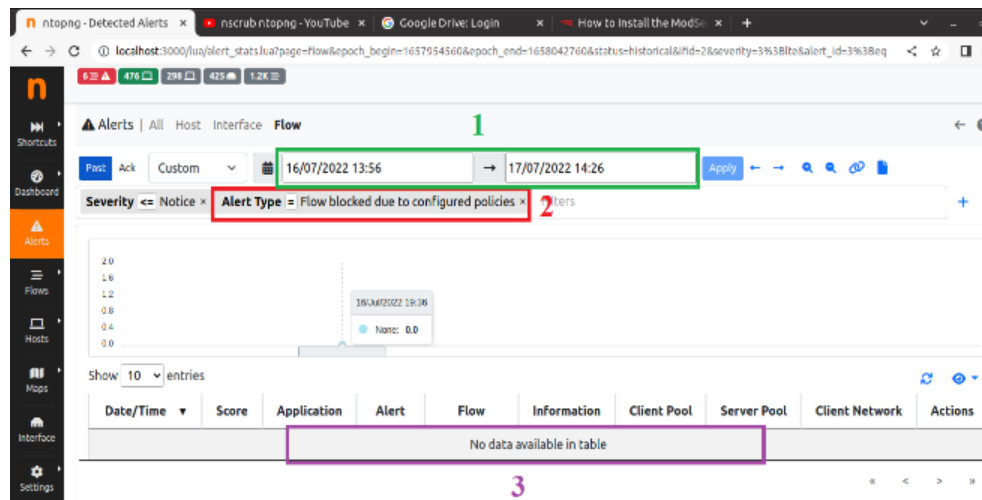
Pada Gambar 5 menunjukkan implementasi *mod security* pada *web server* berupa *error log*. *Error log* pada *mod security* berfungsi untuk menampilkan upaya penyerangan yang digagalkan oleh *mod security*. Pada Gambar 5 *error log* dimulai dengan menampilkan keterangan waktu pengiriman paket oleh penyerang dengan kode “[Thu Jul 14 19:07:44.602390 2022]”. Kode [client 192.168.212.5] menunjukkan IP pengirim. Kode “ModSecurity: Warning. Matched phrase “user-agent:” at REQUEST_HEADERS:User - Agent. [file “ / etc / modsecurity / rules / REQUEST - 913 -SCANNER-DETECTION.conf”] [line “54”] [id “913100”]” yang berarti *rules* *mod security* pada *file* *REQUEST-913-SCANNER-DETECTION.conf* dengan id 913100 sedang bekerja mencocokkan *user agent* dari paket yang dikirimkan oleh penyerang. Kode “referer : https://github.com/shekya/slowhttpstest/” menunjukkan bahwa *tools* yang digunakan oleh penyerang adalah *slowhttpstest*.

C. Implementasi DPI

Gambar 6 menunjukkan implementasi DPI pada *web server* berupa *error log*. Pada detail serangan *tools* *ntopng* bagian 1 menunjukkan informasi IP host yaitu 192.168.212.2. Bagian 2 menunjukkan info *website* yang dikunjungi yaitu 192.168.212.2/login.php yang merupakan alamat *website*. Pada bagian 3 menunjukkan status : *request timeout* dengan *user agent* yang merupakan karakteristik *string* yang memungkinkan *server* dapat mengidentifikasi sistem operasi, *browser* yang digunakan oleh penyerang. Bagian 4 menunjukkan jangkauan IP pada *client*. Bagian 5 menunjukkan jangkauan IP *server*. Bagian 6 menunjukkan IP *network* dari *client* yaitu 192.168.212.0 dengan prefix /24. Bagian 7 menunjukkan IP *network* dari *server* yaitu 192.168.212.0 dengan prefix /24. Proses instalasi dari *ntopng* yang merupakan *tools open source* dari DPI pada *Apache web server*.

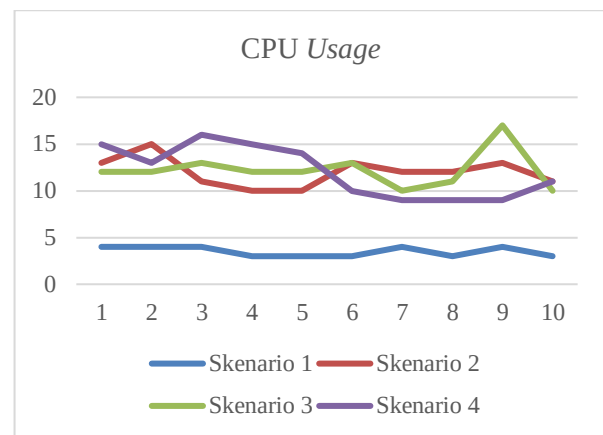


Gambar 6. Access log pada tools Ntopng



Gambar 7. Error log tools Ntopng

Pada Gambar 7 ditunjukkan *error log* pada DPI menggunakan tools ntopng untuk mengetahui upaya penyerangan yang digagalkan oleh DPI pada tools ntopng. Pada bagian 1 menunjukkan waktu serangan yang terjadi pada tanggal 16/07/2022 sampai tanggal 17/07/2022. Pada bagian 2 menunjukkan jenis peringatan yang dipilih adalah pemblokiran aliran data oleh kebijakan yang berlaku. Pada bagian 3 menunjukkan hasil dari peringatan yang dihasilkan pada bagian 2, pada bagian 3 tidak terdapat aliran data yang terblokir, hal tersebut menandakan bahwa DPI tidak dapat memblokir serangan DDoS *slow headers* yang dikirimkan oleh penyerang.



Gambar 8. Grafik CPU usage

D. CPU Usage

Pada pengujian dengan parameter *CPU Usage* dilakukan sebanyak 10 kali dalam 4 skenario pengujian. Hasil pengujian disajikan pada Gambar 8. Pada skenario 1 pengujian *CPU usage* dengan nilai rata rata *CPU usage* sebesar 3,5% yang dihasilkan dari 10 kali percobaan. Perolehan nilai tersebut dikarenakan PC *web server* sedang tidak menerima banyak permintaan yang merupakan serangan DDoS *slow headers* sehingga CPU pada PC *web server* tidak bekerja secara intens. Pada skenario 2 diperoleh nilai rata-rata sebesar 12,3%, perolehan nilai tersebut dikarenakan saat terjadi serangan DDoS *slow headers*, mod *security* yang telah terpasang pada *web server* bekerja dengan memindai dan memblokir serangan seperti yang tertera pada Gambar 5. Hal tersebut dapat menaikkan *CPU usage*. Pada skenario 3 diperoleh nilai rata-rata sebesar 12,1%, perolehan nilai tersebut dikarenakan DPI bekerja pada *web server* dengan memindai paket pada DPI tidak terdapat mekanisme pemblokiran serangan, sehingga *CPU usage* yang didapatkan lebih rendah dari skenario 3.

Pada Gambar 9 disajikan grafik *CPU usage* dari 3 skenario pengujian untuk mengetahui perbandingan implementasi mod *security* dengan DPI pada *web server*.

Pada skenario 1, *CPU usage* stabil di angka 3% hingga 4%. Pada skenario 2 angka *CPU usage* dimulai pada 13% dan mengalami kenaikan pada percobaan kedua menjadi 15%, *CPU usage* mengalami penurunan pada percobaan ketiga hingga percobaan kelima. Pada percobaan keenam hingga percobaan kesepuluh *CPU usage* mengalami peningkatan dan penurunan sebesar 1%. Pada skenario 2 implementasi mod *security* pada *web server*, *CPU usage* dimulai pada 12% pada percobaan pertama hingga mengalami kenaikan menjadi 13% pada percobaan ketiga dan mengalami penurunan menjadi 12% pada percobaan keempat dan kelima, pada percobaan ketujuh *CPU usage* mengalami peningkatan hingga terjadi peningkatan pada percobaan kesembilan dan mengalami penurunan pada percobaan terakhir. Pada skenario 3 dimana *web server* diamankan dengan DPI, pada percobaan pertama nilai *CPU usage* sebesar 15% dan

mengalami penurunan pada percobaan kedua menjadi sebesar 13% dan mengalami kenaikan pada percobaan ketiga menjadi 16%. CPU *usage* mengalami penurunan dari percobaan ketiga hingga percobaan kesembilan dan mengalami kenaikan pada percobaan kesepuluh menjadi 11%. Berdasarkan grafik yang diperoleh maka DPI lebih unggul dari mod *security* pada parameter CPU *usage*.

E. Response Time

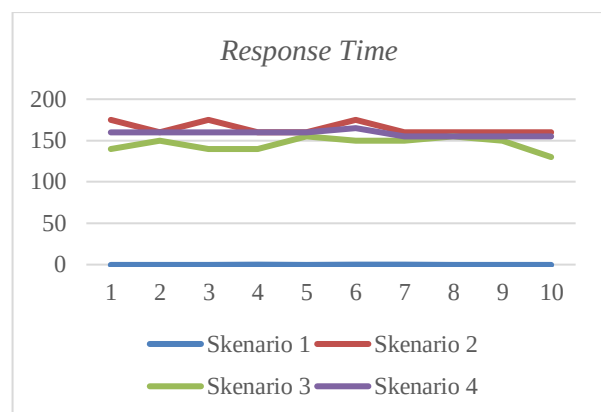
Pada pengujian dengan parameter *response time* dilakukan sebanyak 10 kali dalam 4 skenario pengujian. Hasil pengujian disajikan pada Gambar 9. Pada skenario 1 pengujian *response time* dengan nilai terendah diperoleh rata-rata 0,0024 detik. Perolehan nilai tersebut dikarenakan PC *web server* sedang tidak menerima serangan DDoS *slow headers*, sehingga *web server* dapat dengan cepat melayani permintaan yang dikirimkan oleh *client* sah. Pada skenario 2 diperoleh nilai rata-rata sebesar 146 detik, perolehan nilai tersebut dikarenakan saat terjadi serangan DDoS *slow headers*, PC *web server* dilengkapi dengan mod *security* sebagai pengaman. Mod *security* dapat memblokir serangan berdasarkan *rules 949 blocking evaluation* yang tertera pada Gambar 5 sehingga membuat *web server* dapat melayani permintaan dari *client* sah dengan lebih cepat dari skenario 2. Pada skenario 3 diperoleh nilai rata-rata sebesar 158,5 detik, perolehan nilai tersebut dikarenakan DPI tidak memiliki mekanisme pemblokiran serangan yang ditunjukkan pada Gambar 7 sehingga waktu yang dibutuhkan *web server* dalam melayani permintaan *client* sah tidak lebih baik dari skenario ketiga.

Pada skenario 1, *response time* mengalami kenaikan dari 0,0021 detik hingga 0,0024 detik dan mengalami pelonjakan hingga 0,0033 detik dan mengalami penurunan hingga 0,0025 detik. Pada skenario 2 percobaan pertama hingga percobaan

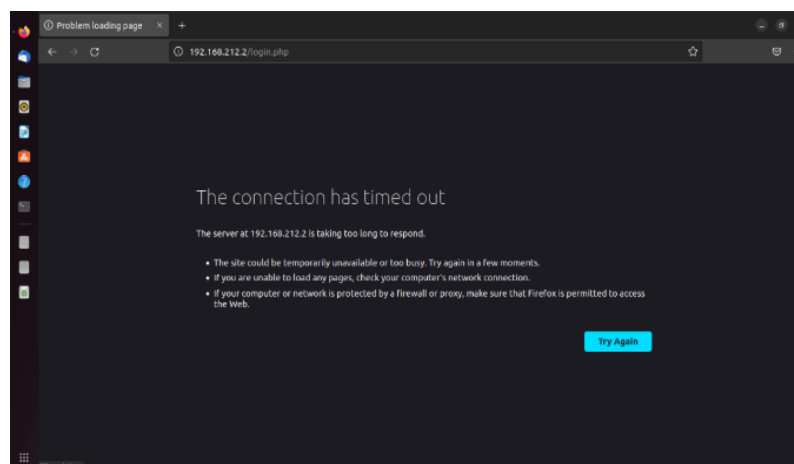
keenam, angka *response time* mengalami kenaikan dan penurunan dari 175 detik hingga 160 detik.

Pada percobaan ketujuh hingga percobaan 100 *response time* stabil di angka 160 detik. Lamanya waktu *response time* tersebut membuat *web server* *down* selama 160 hingga 175 detik sehingga mengakibatkan *client* sah tidak dapat mengakses ke *web server* seperti pada Gambar 10.

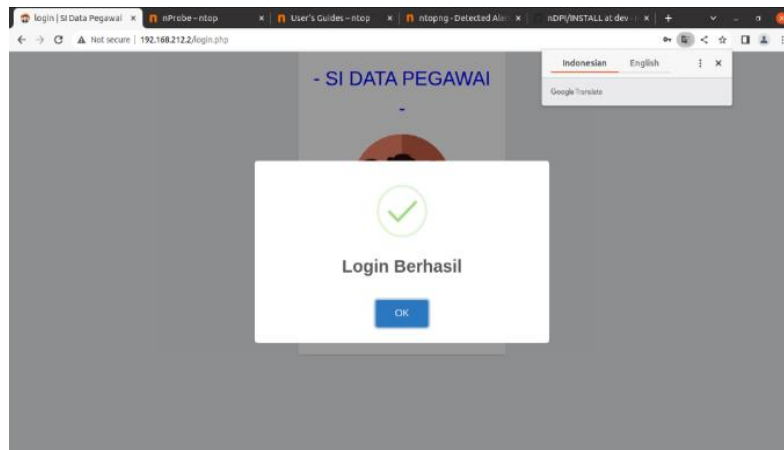
Pada skenario 2 implementasi mod *security* pada *web server*, *response time* mengalami kenaikan dan penurunan nilai *response time* namun waktu yang dibutuhkan *server* untuk kembali lagi beroperasi lebih cepat. Pada skenario 3 dimana *web server* diamankan dengan DPI, pada percobaan pertama hingga percobaan kelima *response time* yang dihasilkan stabil pada 160 detik dan mengalami kenaikan pada percobaan keenam, pada percobaan ketujuh hingga kesepuluh *response time* yang dihasilkan stabil pada 155 detik. Berdasarkan grafik yang diperoleh maka mod *security* lebih unggul dari DPI pada parameter CPU *response time* namun sama-sama dapat mempersingkat durasi *web server down* seperti pada Gambar 11.



Gambar 9. Grafik *response time*



Gambar 10. *Web server down*

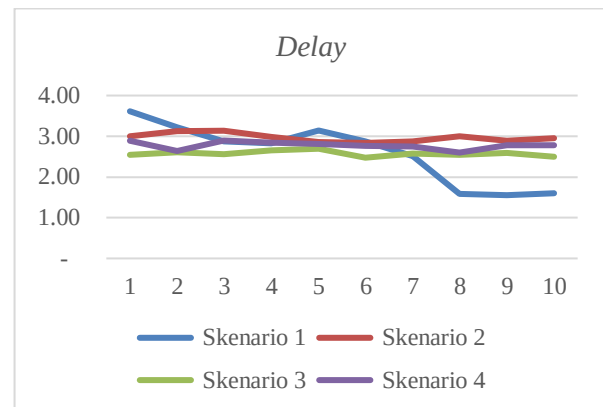


Gambar 11. Akses web server

F. Delay

Pada pengujian dengan parameter *delay* dilakukan sebanyak 10 kali dalam 3 skenario pengujian. Hasil pengujian disajikan pada Gambar 12. Pada skenario 1 diperoleh pengujian parameter *delay* dengan nilai rata-rata sebesar 2,58 ms, nilai tersebut diperoleh karena tidak terdapat banyak paket yang masuk ke *web server*, sehingga waktu yang dibutuhkan dalam pengiriman permintaan client sah menuju *web server* menjadi cepat. Pada skenario 2 diperoleh nilai rata-rata sebesar 2,57 ms, perolehan nilai tersebut dapat dipengaruhi oleh waktu yang dibutuhkan *mod security* untuk mencocokkan serangan dengan *rules* yang sudah ada. *Mod security* dapat memblokir paket yang dikirimkan oleh penyerang berdasarkan *rules* 949 *blocking evaluation* yang tertera pada Gambar 4.5 sehingga dapat memperbaiki waktu pengiriman dari client sah menuju *web server*. Pada skenario 3 diperoleh nilai rata-rata sebesar 2,77 ms, perolehan nilai tersebut dikarenakan DPI hanya mendeteksi serangan tanpa terdapat mekanisme pemblokiran paket yang terdapat pada Gambar 5 sehingga DPI tidak dapat menghalangi permintaan yang dikirimkan oleh penyerang yang membuat waktu pengiriman dari client sah menuju *web server* tidak lebih baik dari skenario 3.

Pada skenario 1, *delay* mengalami penurunan dari pengujian pertama hingga pengujian kesepuluh. Pada skenario 2 implementasi *mod security* pada *web server*, *delay* mengalami penurunan dibandingkan dengan *delay* pada skenario kedua yang menandakan *mod security* dapat menurunkan nilai *delay*. Pada skenario 3 dimana *web server* diamankan dengan DPI. Semakin kecil nilai *delay* yang diperoleh maka performansi pada suatu layanan semakin membaik, dikarenakan *delay* merupakan waktu yang dibutuhkan untuk mengirimkan data dari client ke server, dalam hal ini *mod security* lebih unggul dari DPI.



Gambar 12. Grafik delay

IV. KESIMPULAN

Berdasarkan analisis yang diperoleh dari hasil pengujian. *Mod security* memiliki kemampuan dalam memindai dan memblokir serangan, pada DPI hanya dapat melakukan pemindaian tanpa melakukan pemblokiran serangan sehingga *mod security* unggul dari DPI pada parameter *response time* dan *delay*. Pada parameter *CPU usage*, DPI unggul dari *mod security*. Penelitian selanjutnya dapat menggunakan aplikasi DPI berbayar seperti palo alto untuk mendapatkan fitur terbaik pada metode DPI dalam mengamankan *web server*.

UCAPAN TERIMA KASIH

Ucapan terima kasih diberikan kepada pihak pengurus laboratorium pengolahan sinyal digital Institut Teknologi Telkom Purwokerto selaku penyedia subjek pada penelitian ini.

REFERENSI

- [1] J. Jupriyadi, B. Hijriyanto, and F. Ulum, "Komparasi Mod Evasive dan DDoS Deflate Untuk Mitigasi Serangan Slow Post," *Techno.Com*, vol.

- 20, no. 1, pp. 59–68, 2021.
- [2] M. Arman, “Metode Pertahanan Web Server Terhadap Distributed Slow HTTP DoS Attack,” *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 7, no. 1, pp. 56–70, 2020.
- [3] R. Riska and H. Alamsyah, “Penerapan Sistem Keamanan Web Menggunakan Metode Web Application Firewall,” *J. Amplif. J. Ilm. Bid. Tek. Elektro Dan Komput.*, vol. 11, no. 1, pp. 37–42, 2021.
- [4] A. Aryapranata, “Web Application Firewall pada Situs Web Institut Bisnis Nusantara www. ibn. ac. id,” *Esensi Komputasi*, vol. 4, no. 1, pp. 55–59, 2020.
- [5] Triyadi, “Pengertian, Fungsi dan Cara Kerja ModSecurity,” 2019. <https://www.rumahweb.com/journal/pengertian-fungsi-dan-cara-kerja-modsecurity/> (accessed Apr. 20, 2022).
- [6] R. Sun, L. Shi, C. Yin, and J. Wang, “An improved method in deep packet inspection based on regular expression,” *J. Supercomput.*, vol. 75, no. 6, pp. 3317–3333, 2019.
- [7] G. De La Torre Parra, P. Rad, and K. K. R. Choo, “Implementation of deep packet inspection in smart grids and industrial Internet of Things: Challenges and opportunities,” *J. Netw. Comput. Appl.*, vol. 135, no. October 2018, pp. 32–46, 2019.
- [8] R. E. M. Taher, N. Mostafa, and A. Baha, “A Survey on Deep Packet inspection,” *Comput. Eng. Dep.*, no. Deep Packet Inspection, 2017.
- [9] I. P. A. E. Pratama and P. A. Dharmesta, “Implementasi Teknik Deep Packet Inspection Dengan Menggunakan Wireshark Pada Sistem Operasi Ubuntu,” *J. Resist. (Rekayasa Sist. Komputer)*, vol. 1, no. 2, pp. 79–85, 2018.
- [10] P. P. Pahlawan, “Perbandingan Penerapan Metode Pengamanan Mod Security Dan Mod Evasive Pada Web Server Terhadap Serangan Slow Headers,” *J. Eng. Comput. Sci. ...*, vol. 1, no. 1, pp. 93–100, 2021.