Komparasi *Mod Evasive* dan *DDoS Deflate* Untuk Mitigasi Serangan *Slow Post*

Comparison of Mod_Evasive and DDoS Deflate for Slow Post Attack Mitigation

Jupriyadi[1], Budi Hijriyanto[2], Faruk Ulum[3]
[1]Program Studi S1 Teknologi Informasi Universitas Teknokrat Indonesia
[2]Program Studi S1 Informatika Universitas Teknokrat Indonesia
[3]Program Studi S1 Sistem Informasi Universitas Teknokrat Indonesia
E-mail : [1]jupriyadi@teknokrat.ac.id, [2]budihijriyanto@gmail.com, [3]faruk.ulum@teknokrat.ac.id

Abstrak

Web server merupakan server yang memberikan layanan berbasis web dan harus mampu melayani pengguna saat dibutuhkam. Namun tidak menutup kemungkinan web server dapat mengalami gangguan akibat ancaman dan serangan yang dilakukan oleh pihak yang tidak bertanggungjawab. Salah satu ancaman yang dapat menggangu web server adalah serangan Denial of Service (DOS) menggunakan teknik slow post yang dapat menyebabkan layanan pada web server tidak dapat diakses. Penelitian ini akan menguji dua buah metode pengamanan yang dapat digunakan untuk mengurangi dampak serangan DoS yaitu mod-evasive dan ddos deflate. Berdasarkan eksperimen yang telah dilakukan dapat diambil kesimpulan bahwa ddos deflate merupakan metode yang lebih baik dibandingkan dengan mod-evasive dalam mengatasi serangan DOS karena ddos deflate mampu mendeteksi dan dapat memutus koneksi yang berlebihan sesuai dengan konfigurasi yang dilakukan.
Kata kunci: *denial of service, slow post, web server, ddos deflate, mod-evasive*

Abstract

Web server is a server that provides web-based services and must be able to serve users when needed. However, it is possible that the web server may experience interference due to threats and attacks carried out by irresponsible parties. One of the threats that can interfere with a web server is a Denial of Service (DOS) attack using the slow post technique which can make services on the web server inaccessible. This study will examine two security methods that can be used to reduce the impact of a DoS attack, namely mod-evasive and ddos deflate. Based on the experiments that have been carried out, it can be concluded that ddos deflate is a better method than mod-evasive in overcoming DOS attacks because ddos deflate is able to detect and disconnect excessive connections according to the configuration performed.

Keywords: denial of service, slow post, web server, ddos deflate, mod-evasive

1. PENDAHULUAN

Perkembangan teknologi saat ini membuat orang dapat berinteraksi melalui jaringan internet dengan berbagai macam layanan yang disediakan. Salah satu layanan yang ada dan dapat diakses melalui jaringan internet adalah layanan berbasis web contohnya seperti *elearning, e-commerce,* layanan email dan masih banyak lagi. Mengingat layanan berbasis web dapat diakses melalui jaringan internet sehingga dapat diakses oleh siapapun yang terhubung ke internet, maka penting untuk menjaga keamanan layanan berbasis web dari ganguan yang tidak diinginkan. Salah satu ancaman yang dapat mengganggu komunikasi data yaitu serangan *denial of service* (DoS)[1].

DoS adalah jenis serangan yang dapat mengganggu layanan yang disediakan sehingga layanan yang ada tidak dapat diakses atau digunakan. *Slow Post* adalah salah satu jenis serangan *DoS*. *Slow Post* tidak melakukan eksploitasi pada lapisan jaringan seperti serangan *DoS / DDoS*, tetapi mengeksploitasi lapisan aplikasi dengan mengirimkan permintaan HTTP yang tidak lengkap, atau *transfer rate* yang sangat rendah. Hal ini menyebabkan *web server* akan menggunakan banyak sumber daya dalam menunggu sisa data, jadi ketika *web server* sudah terlalu banyak atau mengalami kekurangan sumber daya, maka akan terjadi penolakan layanan atau *denial of service* pada server [2].

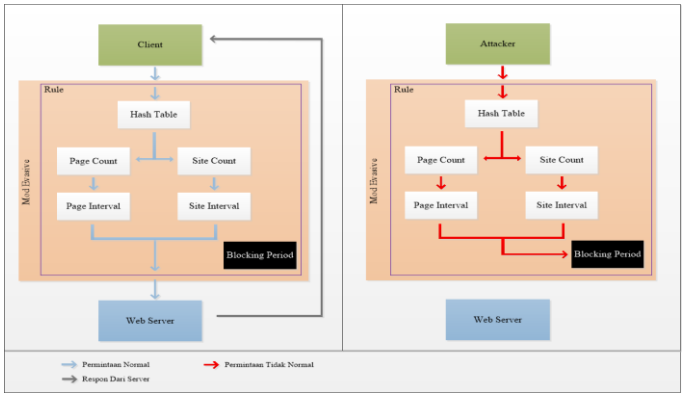
Berdasarkan permasalahan yang telah diuraikan sebelumnya, penting untuk sebuah *web server* menerapkan suatu metode yang dapat digunakan untuk mengamankan *web server* dari ancaman serangan *Slow Post*. Ada beberapa metode pengamanan yang dapat digunakan untuk menangkal atau mengurangi serangan *DoS* yaitu *DDoS Deflate* dan *Mod Evasive*. Pada penelitian ini akan menerapkan *DDOS Deflate* dan *Mod-Evasive* pada sebuah web server untuk mengetahui kinerjanya dalam mitigasi serangan DoS khusunya serangan *slow post*. Hal ini ditujukan untuk mengetahui teknik yang terbaik dalam mitigasi serangan *slow post* menggunakan 2 tool tersebut sehingga dapat dijadikan rujukan bagi admin jaringan dalam menggunakan tool untuk mitigasi serangan DoS. *DDoS deflate* dan *mod_evasive* telah diuji untuk mitigasi serangan terhadap koneksi http, namun masih perlu dilakukan pengujian terhadap beberapa jenis serangan http menggunakan DDoS deflate dan mod_evasive [3]. Mod_evasive dapat bekerja pada sistem operasi windows dan linux dan mampu melakukan pencegahan serangan DoS berdasarkan parameter tertentu [4].

1.1 Serangan *Denial of Service* (DoS)

Denial of Service (DoS) atau *Distributed Denial of Service* (DDoS) merupakan serangan yang membanjiri server dengan mengirimkan permintaan yang sangat banyak sehingga menghabiskan sumber daya pada server tersebut sampai server tersebut tidak dapat menjalankan fungsi dan tugasnya dengan benar. Kemudian server yang tidak bisa menangani permintaan, maka akan mengalami penolakan layanan (*denial of service*) [5]. Serangan DDoS *flooding* juga dapat dilakukan pada environement *singlehoming* dan *multihoming* honeypot dimana pada environment *multihoming* mitigasi serangan DoS dapat dilakukan karena trafik di bagi kedalam beberapa *path* jaringan[6].

1.2 Mod Evasive

Berikut ini adalah cara kerja dari *mod evasive* dalam melakukan tugasnya menjaga web server, pada gambar 1.

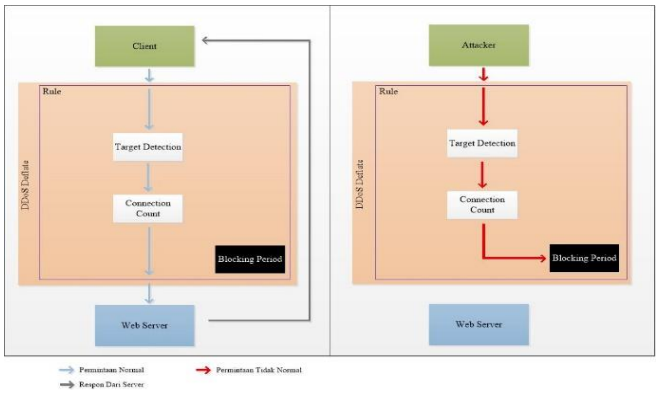


Gambar 1 Cara Kerja *mod_evasive*

Dari gambar tersebut menggambarkan mengenai cara kerja *mod_evasive*. *Request* yang akan masuk ke *web server*, akan melewati pemeriksaan pada setiap *rule* yang ada pada *mod evasive*. Pada tahap awal akan dibuat *Tabel Hash* dinamis internal dari Alamat IP dan URL sebagai deteksi awal. Kemudian dilakukan perhitungan jumlah permintaan halaman pada bagian *page count* dan permintaan pada situs pada bagian *site count*. Selanjutnya akan di hitung rentang waktu antar permintaan halaman dan situs pada *page interval* dan *site interval*. Apabila jumlah dan waktu permintaan tidak melebihi batas yang sudah ditentukan, maka akan diteruskan ke *web server* dan *web server* akan mengirim respon. Tetapi apabila jumlah permintaan melebihi batas maka permintaan selanjuntnya akan ditolak dan dimasukkan dalam daftar hitam sementara atau *blocking period* [7]. Mod_evasive dapat digunakan untuk mendeteksi dan melakukan bloking terhadap serangan DoS berupa koneksi yang banyak dan berulang dari alamat IP tertentu [8]. Mod_evasive telah diuji untuk pencegahan serangan *flooding* terhadap keamanan layanan video streaming pada web server[9].

1.3 DDoS Deflate

Berikut ini adalah cara kerja dari *DDoS Deflate* dalam melakukan tugasnya menjaga *web server*, pada gambar 2.

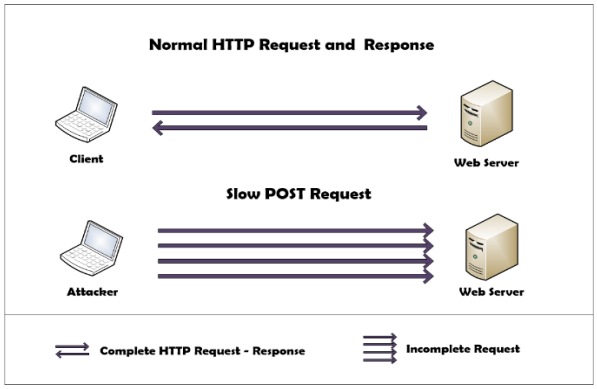


Gambar 2 Cara Kerja DDoS Deflate.

Setelah menetapkan frekuensi target, DDoS Deflate menetapkan kriteria berdasarkan batas untuk jumlah total koneksi, di mana *DDoS Deflate* harus menentukan jumlah maksimum koneksi untuk alamat IP. Jumlah default node diatur ke 200. Jika alamat IP mencapai jumlah maksimum batas *node*, maka *DDoS Deflate* memperlakukan alamat IP sebagai alamat IP yang buruk dan memblokirnya [10].

1.4 Slow Post

Serangan *slow Post* dalam melakukan aksinya akan membanjiri target dengan permintaan yang tidak lengkap atau parsial, seperti yang terlihat pada gambar 3. Pada dasarnya sebelum panjang konten (*content-length)* belum terkirim seluruhnya, maka *web server* akan menunngu isi pesan yang tersisa untuk dikirm. Selama proses menunggu tersebut, *web server* akan membuka koneksi yang lambat. Tidak seperti *slow headers*, karena tidak adanya penundaan dalam pengiriman *HTTP header*. Serangan *slow post* merupakan salah serangan yang sangat mudah dilakukan oleh penyerang dengan memanfaatkan kelemahan protokol HTTP [11]. Serangan ini sulit untuk di deteksi karena sulit untuk dibandingkan trafik normal HTTP dengan trafik serangan[12].



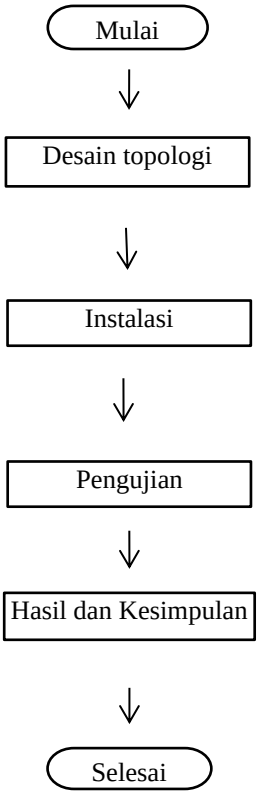
Gambar 3 Cara Kerja Serangan *slow Post.*

Permintaan yang tidak lengkap tersebut akan membuat *web server* menunggu paket tersebut. Sehingga akan terjadinya penumpukan paket yang tidak lengkap. Hal tersebut akan mengakibatkan *web server* menjalankan banyak layanan httpd yang akan menghabiskan banyak sumber daya. Ketika *web server* sudah tidak mampu melayani permintaan maka akan terjadi yang namanya *denial of service.*

Berdasarkan laporan yang dirilis oleh NETSCOUT untuk tahun 2018 menyatakan bahwa jumlah serangan *Denial of Servie* (DoS) mengalami kenaikan sebesar 26 persen [13]. Hal ini menunjukkan bahwa serangan DoS masih banyak di gunakan oleh penyerang.

2. METODE PENELITIAN

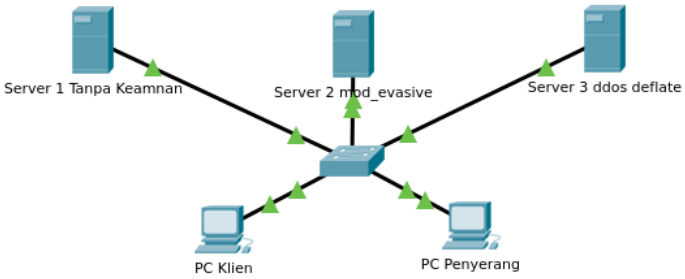
Gambar 4 berikut ini adalah tahapan yang dilakukan dalam melaksanakan penelitian.



Gambar 4 Tahapan penelitian

2.1 Desain Topologi

Dalam melaksanakan penelitian ini menggunakan 3 server, dan 2 klien (1 sebagai klien dan 1 lagi sebagai penyerang). Desain topologi yang digunakan dalam melaksanakan penelitian dapat dilihat pada gambar 5.



Gambar 5 Topologi pengujian

2.2 Instalasi

Tahap selanjutnya adalah instalasi pada server 1 (ubuntu OS, apache web server, LMS moodle), server 2 (ubuntu OS, apache web server, LMS moodle dan mod_evasive), server 3 (ubuntu OS, apache web server, LMS moodle dan ddos deflate). Berikut ini adalah perintah yang digunakan untuk instalasi mod_evasive dan ddos deflate.

Instalasi ddos deflate[14] :

```
# wget http://www.hostnic.id/download/ddos/install.sh

# chmod 0700 install.sh

# ./install.sh
```

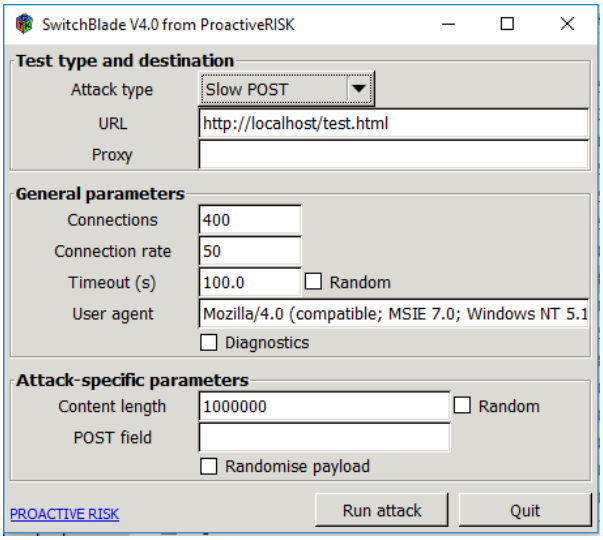
Mod evasive merupakan modul apache yang dapat diinstall pada server dengan sistem operasi linux, berikut ini adalah perintah untuk instalasi mod_evasive[15]:

```
# apt-get install libapache2-mod-evasive
```

Dalam eksperimen konfigurasi mod_evasive dan ddos deflate menggunakan konfigurasi default instalasi.

2.3 Pengujian

Pengujian dilakukan menggunakan aplikasi *Switchblade* untuk melakukan serangan *Slow Post*. Tampilan antarmuka dari aplikasi *Switchblade* dapat dilihat pada gambar 6 berikut ini.



Gambar 6 Tampilan Antarmuka Switchblade

Kemudian pengujian akan dilakukan dengan beberapa kombinasi pengaturan serangan yang akan terapkan pada *Switchblade*. Kombinasi pengaturan serangan pada *Switchblade* dapat dilihat pada tabel 1 berikut.

Tabel 1 Kombinasi Serangan

Connections	***Connection Rate***	***Timeout(s)***	***Content Length (***bytes***)***
400	50	100	1000000
1000	200	110	8192
20000	50	110	1000

Lama waktu pengujian yang dilakukan dapat dilihat pada tabel 2 berikut ini:

Tabel 2 Lama Waktu Penngujian

Pengujian	**Lama Waktu**
Pengujian 1	6 Menit
Pengujian 2	8 Menit
Pengujian 3	13 Menit

2.4 Hasil dan Kesimpulan

Pada tahap ini dilakukan analisa terhadap hasil eksperimen yang telah diperoleh untuk diambil kesimpulan. Hasil dan pembahasan secara lengkap dijelaskan pada bagian berikutnya.

3. HASIL DAN PEMBAHASAN

3.1. Pengujian Web Server Tanpa Pengamanan (web server 1)

Hasil pengujian terhadap *web server 1* yang dilakukan, kemudian dijadikan sebagai acuan dan pembanding terhadap hasil pengujian yang dilakukan terhadap *web server 2* dan *web server 3* yang diimplementasikan metode pengamanan. Hal ini dilakukan untuk membantu dalam proses analisis. Hasil pengujian yang sudah dilakukan terhadap *web server* 1 dapat dilihat

pada tabel 3 sampai table 5 berikut:

Tabel 3 Hasil pengujian pertama *web server* 1

Penyerang	Tujuan	Koneksi				
		Target	Aktif	Terhubung	Terputus	Error
Penyerang 1	Server 1	400	216	216	184	0
Penyerang 2	Server 1	400	82	82	318	0

Tabel 4 Hasil pengujian kedua *web server* 1

Penyerang	Tujuan	Koneksi				
		Target	Aktif	Terhubung	Terputus	Error
Penyerang 1	Server 1	1000	246	246	754	0
Penyerang 2	Server 1	1000	41	41	959	0

Tabel 5 Hasil pengujian ketiga *web server* 1

Penyerang	Tujuan	Koneksi				
		Target	Aktif	Terhubung	Terputus	Error
Penyerang 1	Server 1	20000	217	217	19783	0
Penyerang 2	Server 1	20000	62	62	19938	0

Dampak yang terjadi pada *web server* 1 saat sedang dilakukan penyerangan yaitu, halaman situs dari *web server* 1 tidak dapat diakses melalui *web browser* pada *virtual client*. Hal ini dikarenakan *web server* 1 tidak diimplementasikan pengamanan apapun.

3.2. Pengujian Web Server dengan Mod_Evasive (web server 2)

Pada tahap ini dilakukan pengujian yang dilakukan terhadap *web server* 2. *Web server* 2 merupakan *web server* yang diimplementasikan metode pengamanan *Mod Evasive*. *Mod Evasive* merupakan modul tambahan dari *Apache*, sehingga saat *Mod Evasive* dijalankan, masih menggunakan *service Apache.* dalam melakukan tugasnya, *Mod Evasive* memiliki sebuah aturan. Apabila ada yang melanggar aturan tersebut, maka *Mod Evasive* akan melakukan tindakan pemblokiran. Hasil pengujian serangan terhadap *web server 2* dapat dilihat pada tabel 6 sampai tabel 8 di bawah ini:

Tabel 6 Hasil pengujian pertama *web server 2*

Penyerang	Tujuan	Koneksi				
		Target	Aktif	Terhubung	Terputus	Error
Penyerang 1	Server 2	400	159	159	241	0
Penyerang 2	Server 2	400	148	148	252	0

Tabel 7 Hasil pengujian kedua *web server 2*

Penyerang	Tujuan	Koneksi				
		Target	Aktif	Terhubung	Terputus	Error
Penyerang 1	Server 2	1000	258	258	742	0
Penyerang 2	Server 2	1000	71	71	929	0

Tabel 8 Hasil pengujian ketiga *web server 2*

Penyerang	Tujuan	Koneksi				
		Target	Aktif	Terhubung	Terputus	Error
Penyerang 1	Server 2	20000	128	128	19872	0
Penyerang 2	Server 2	20000	151	151	19849	0

Dampak yang terjadi pada *web server 2* saat sedang dilakukan penyerangan yaitu halaman situs dari *web server* 2 tidak dapat diakses melalui *web browser* pada *virtual client*. Hal ini dikarenakan pada *Mod Evasive* menggunakan *rule* yang mengatur tentang banyaknya jumlah *page request* maupun *site request* dan juga jarak waktu dari setiap banyaknya jumlah *page request* dan *site request* yang diterima oleh *web server* 2. Dimana parameter ini tidak cocok untuk menangkal atau mengurangi serangan *Slow Post*. Karena serangan *Slow Post* hanya mengirim sebuah permintaan dari setiap koneksi sampai serangan berakhir. Jadi setelah *web server 2* menerima paket tidak lengkap untuk pertama kali, maka paket yang dikirim selanjutnya yaitu untuk melengkapi permintaan sebelumnya, dan akan berlangsung selama paket tersebut belum terlengkapi atau utuh. Dan setelah paket lengkap maka serangan *Slow Post* akan berhenti. Hal tersebut hanya terhitung sebagai satu atau sebuah permintaan. Nilai terkecil dari parameter yang dapat diatur pada *Mod Evasive* bernilai 1. Sedangkan pada saat sebuah permintaan serangan *Slow Post* telah lengkap, maka serangan akan berhenti. Sehingga *Mod Evasive* tidak sempat untuk melakukan pemblokiran terhadap penyerang yang menggunakan serangan *Slow Post*.

3.3. Pengujian Web Server DDoS Deflate (web server 3)

Pengujian ini merupakan pengujian terakhir yang dilakukan pada *web server 3*. *web Server* ini menerapkan metode pengamanan *DDoS Deflate*. *DDoS Deflate* dijalankan dalam *service* yang bernama *Cron*. Dalam melakukan tugasnya, *DDoS Deflate* memiliki sebuah aturan. Apabila ada yang melanggar aturan tersebut, maka *DDoS Deflate* akan melakukan tindakan pemblokiran. Tabel 9 sampai 11 berikut adalah hasil pengujian *DDoS Deflate* terhadap serangan *Slow Post*:

Tabel 9 Hasil pengujian pertama *web server 3*

Penyerang	Tujuan	Koneksi				
		Target	Aktif	Terhubung	Terputus	Error
Penyerang 1	Server 3	400	0	0	400	0
Penyerang 2	Server 3	400	0	0	400	0

Tabel 10 Hasil pengujian kedua *web server 3*

Penyerang	Tujuan	Koneksi				
		Target	Aktif	Terhubung	Terputus	Error
Penyerang 1	Server 3	1000	0	0	1000	0
Penyerang 2	Server 3	1000	0	0	1000	0

Tabel 11 Hasil pengujian ketiga *web server 3*

Penyerang	Tujuan	Koneksi				
		Target	Aktif	Terhubung	Terputus	Error
Penyerang 1	Server 3	20000	0	0	20000	0
Penyerang 2	Server 3	20000	0	0	20000	0

Pada *web server* 3 saat sedang dilakukan penyerangan yaitu halaman situs dari *web server* 3 berhasil diakses melalui *web browser* pada *virtual client*. W*eb server* 3 berhasil diakses setelah *DDoS* melakukan tindakan pemblokiran terhadap alamat IP penyerang, karena penyerang mencoba membuat koneksi lebih dari yang sudah ditetapkan pada *rule DDoS Deflate*, yang dimana hal tersebut melanggar *rule* yang ada pada *DDoS Deflate*. Kemudian *DDoS Deflate* mengambil tindakan untuk memutus koneksi dari penyerang yang sudah berhasil terhubung dengan *web server 3*. Sehingga penyerang tidak dapat mengirim paket tidak lengkap kepada *web server 3*.

4. KESIMPULAN DAN SARAN

Berdasarkan dari seluruh kegiatan penelitian yang dilakukan maka dapat diambil beberapa kesimpulan antara lain sebagai berikut:

1. Metode pengamanan terbaik dari penelitian ini yang digunakan mitigasi serangan *Slow Post* adalah *DDoS Deflate*. Hal ini dikarenakan *DDoS Deflate* dapat memutus koneksi yang dibuat oleh penyerang ketika jumlah koneksi melebihi *rule* yang telah ditentukan, sehingga penyerang tidak dapat melakukan pengiriman paket yang tidak lengkap kepada *web server*.
2. Metode pengamanan DDoS *deflate* mampu menanggulangi serangan HTTP *slow post*, hal ini dibuktikan meskipun serangan dilakukan web server tetap dapat melayani klien dengan baik.
3. *Mod_evasive* mampu melakukan mitigasi terhadap serangan slow HTTP *post* namun tidak 100% mengatasinya karena masih terdapat beberapa koneksi yang aktif dan saat serangan diaktifkan server tidak dapat diakses oleh klien lain.
4. Dalam penelitian ini mod_evasive dan ddos deflate menggunakan konfigurasi default saat instalasi dalam menjaga web server. Parameter yang digunakan *mod_evasive* dan *ddos deflate* belum diatur sedemikian rupa sehingga perlu dikaji kembali dengan merubah parameter yang dimiliki masing-masing metode dalam mitigasi serangan *slow post.*

UCAPAN TERIMA KASIH

Ucapan terimakasih penulis sampaikan kepada Universitas Teknokrat Indonesia yang telah berkenan memberikan bantuan dana sehingga penelitian ini dapat terlaksana.

DAFTAR PUSTAKA

[1] Arman, M., 2020. Metode Pertahanan Web Server Terhadap Distributed Slow HTTP DoS Attack. JATISI (Jurnal Teknik Informatika dan Sistem Informasi), 7(1), pp.56-70.
[2] J. Park, 2015, Analysis of Slow Read DoS Attack and Countermeasures on Web servers, *Int. J. Cyber-Security Digit. Forensics*, vol. 4, no. 2, hal. 339–353
[3] Unrein, E., Fish, D., Boeker, J., & Sun, W. (2012). Living in denial-A comparison of distributed denial of service mitigation methods. Issues in Information Systems, 13(1), 190-198.
[4] M. Yeasir, M. Morshed, dan M. Fakrul, 2015, A Practical Approach and Mitigation Techniques on Application Layer DDoS Attack in Web Server. *Int. J. Comput. Appl.*, vol. 131, no. 1, hal. 13–20
[5] Hari Siswantoro, A Sumarudi, Agus Mulyanto, Riri Fitri Sari, 2012, Comparison of Singlehoming and Multihoming Honeypot Defense System to Mitigate Flooding based DDoS Attacks, International Conference on Computer and Management (CAMAN)
[6] D. P. Mishra dan K. Sourav, 2012, DDoS Detection and Defense: Client Termination Approach, hal. 2–6.
[7] D. S. B. Sangeetha, 2015, DDoS Deflate and APF (Advanced Policy Firewall) : A Report *Int. J. Comput. Trends Technology*, vol. 27, no. 2, hal. 64–69
[8] Yevsieieva O, Helalat SM. 2017, Analysis of the impact of the slow http dos and ddos attacks on the cloud environment, 4th international scientific-practical conference problems of infocommunications. Science and technology (PICSI&T), IEEE, pp. 519–23
[9] S.N.M.P. Simamora, L. N. Lubis, A. Sularsa, 2012, Implementasi Pencegahan Serangan Interruption Jenis Flooding Terhadap Keamanan Layanan Video-Streaming, Jurnal Ilmiah Ilmu Komputer, Vol. 8 No. 2 Maret 2012: 153-165
[10] Marquette S. 2017, Types of DDoS attacks, https://www.esecurityplanet.com/network-security/types-of-ddos-attacks.html, Diakses tanggal 30 Juli 20 September 2020
[11] Jefferson González, 2018, Fork of DDoS Deflate with fixes, improvements and new

features, https://github.com/jgmdev/ddos-deflate, Diakses tanggal 30 Agustus 2020
[12] Francisndungu, 2018, How to Secure Apache Web Server with ModEvasive on Ubuntu 16.04, https://www.alibabacloud.com/blog/how-to-secure-apache-web-server-with-modevasive-on-ubuntu-16-04_59405, Diakses 10 September 2020
[13] Modi H. NETSCOUT threat intelligence report. Technical report. 2018, https://www.netscout.com/sites/default/files/2019-02/SECR_001_EN-1901%20-%20NETSCOUT%20Threat%20Intelligence%20Report%202H%202018.pdf. Diakses tanggal 25 September 2020.
[14] Muhammad Suyuti Ma'sum, M. Azhar Irwansyah, Heri Priyanto, 2017, Analisis Perbandingan Sistem Keamanan Jaringan Menggunakan Snort dan Netfilter, Jurnal Sistem dan Teknologi Informasi (JUSTIN) Vol.5,No.1, (2017)
[15] Micky Barzilay, 2019, Mitigating_HTTP_Flooding_Attacks on Apache HTTP Web Server, London Metropolitan University