IMPLEMENTASI KEAMANAN *WEBSITE* DENGAN METODE *FIREWALL* APLIKASI *WEB* (WAF) Studi kasus: WEB DESA WANTILAN

Yusuf Ardiansyah, M. Agus Sunandar, Yusuf Muhyidin
Program Studi Teknik Informatika S1, Fakultas Teknik, STT Wastukancana Purwakarta
Jalan Cikopak, Kabupaten Purwakarta, Jawa Barat, Indonesia
yusufardiansyah21@wastukancana.ac.id

ABSTRAK

Desa wantilan merupakan desa yang memanfaatkan teknologi sebagai media untuk menyampaikan program kerja, kegiatan, ataupun tramsparansi anggaran kepada masyarakat. Perkembangan telnologi digital terutama website sering kali digunakan sebagai target serangan seorang hacker untuk memperoleh informasi lebih secara illegal untuk tujuan tertentu. Website desa Wantilan sebelumnya mengalami peretasan selama dua kali oleh hacker yang mengakibatkan perubahan struktur judul dan gambar pada berita yang di muat di halaman website Wantilan. Untuk itu diperlukan suatu sistem yang mampu memberikan solusi dalam pengamanan website. Pada penelitian ini menggunakan metode web application firewall karena metode ini mampu untuk menjadi security sistem dalam mengamankan suatu website dari serangan. Hasil penelitian yang di lakukan, serangan SQL Injection dan DDoS yang telah diujicobakan pada website berhasil diblokir sehingga membuat website menjadi aman dari serangan tersebut

Kata kunci*:* *Website*, Web *Application Firewall (WAF), SQL Injection, Distributed Denial of Service (DDoS)*

1. PENDAHULUAN

Aplikasi web saat ini sudah menjadi bagian yang tidak mampu dipisahkan pada kehidupan sehari – hari, karena untuk memperoleh berita pada keadaan mirip saat ini semuanya dihasilkan melalui software berbasis web dan aplikasi *mobile*. Hal ini tentu saja mempermudah kegiatan insan untuk mendapatkan informasi dari pemanfaatan *software* web ini. tetapi, banyak sekali kelebihan aplikasi web dalam dunia internet juga memiliki kelemahan yang selaras dengan aspek keamanan yaitu sangat rentan terhadap ancaman [1].

Keamanan pada sebuah *website* adalah aspek krusial yang harus dimiliki. Mengamankan web dapat dilakukan dengan memasang *firewall*, anti *virus*, atau *software* serupa pada komputer ataupun *router* yang terhubung pribadi atau berada pada satu jaringan *server* perangkat *lunak* web tersebut. Keamanan pada web bisa dilakukan menggunakan memakai *web application firewall* (WAF) yang dipasangkan pada layanan web server. *Web application firewall* merupakan suatu metode pengamanan *website*, yang bisa mencegah adanya ancaman dari *attacker* ataupun *hacker*. *Web application firewall* sudah bisa bekerja terlebih dahulu tanpa melakukan konfigurasi tambahan di *server web* sebagai akibatnya tidak perlu lagi dilakukan perubahan atas *script default* web, sehigga dapat diterapkan pada software yang sudah berjalan [2].

Sebelum *Web Aplication Firewall* (WAF) di implementasikan, *website* wantilandesa.id mengalami peretasan sebanyak 2 kali oleh pihak yang tidak bertanggung jawab atau *attacker* yang mengakibatkan tampilan di judul serta gambar berita berubah, penyerang memanfaatkan celah di admin wantilandesa.id, salah satu faktor yang menyebabkan peretasan karena *username* dan *password* terlalu simpel dan mudah di tebak, tidak adanya *Web Aplication Firewall* (WAF) menjadi celah untuk penyerang untuk mengekploitasi *website* wantilandesa.id. penyerang melakukan hal yg sama hanya merubah judul berita dan gambar. Selain menu tersebut penyerang tidak merubah yang lain selain di menu berita.

Beberapa fungsi *Web application firewall* (WAF) seperti, *monitoring, blokir, filtering* pada permintaan yang terdeteksi mencurigakan dan membuat suatu lapisan keamanan yang bisa mendeteksi serta mencegah ancaman serangan *attacker*. Adapun tindakan yang bisa dilakukan mirip menghentikan *request* melalui status 403 *forbidden*. [3].

Berdasarkan permasalahan di atas, penulis menerapkan *firewall* sebagai sistem keamanan *website*, dimana salah satu metode pengamanan yang bisa diterapkan yaitu *web aplication firewall*. metode tadi akan diterapkan di *website* Desa Wantilan, bentuk simulasi pengujian pertahanan, akan dicoba disimulasikan menggunakan teknik penetrasi dan penyerangan kepada *website* desa Wantiilan. dengan sistem pencegahan ini, dapat menjadi rujukan rekomendasi yang efektif untuk keamanan *website* wantiandesa.id.

2. TINJAUAN PUSTAKA

2.1. Keamanan Informasi

Merupakan upaya untuk melindungi aset informasi dari potensi ancaman. Keamanan berita secara tidak eksklusif memastikan kelangsungan bisnis, mengurangi risiko yang timbul, serta

memungkinkan mengoptimalkan laba atas investasi [4].

2.2. Web *Server*

Merupakan sebuah *software server* yang melayani permintaan *HTTP* atau *HTTPS* dari browser dan mengirimkannya kembali pada bentuk halaman web. *Page web* yg dikirim oleh *web server* umumnya berupa file *HTML* serta *CSS* yg nantinya akan ditata browser sehingga menjadi laman web yang menarik serta simpel dibaca *Web server* memakai port 80. Fungsi primer sebuah Web Server ialah untuk transaksi berkas atas permintaan pengguna melalui protokol komunikasi yg telah ditentukan. disebabkan sebuah halaman web bisa terdiri atas berkas teks, gambar, video, serta lainnya pemanfaatan *Web Server* berfungsi juga untuk mentransfer seluruh aspek pemberkasan pada sebuah *page web* yg terkait, termasuk pada dalamnya teks, gambar, video, atau lainnya [5]

2.3. *Web Application Firewall* (WAF)

WAF digunakan untuk menyaring, memantau, serta memblokir ancaman-ancaman pada *website*. Salah satu perangkat lunak WAF yang ada adalah Cloudflare. Beberapa serangan yang bisa diatasi cloudflare adalah SQL Injection dan DDoS Attack [6].

Web Application Firewall seperti dengan firewall tradisional bekerja. *Firewall* bekerja berdasarkan suatu set aturan yang dikonfigurasi pada *firewall* atau yang diklaim menggunakan rule. hukum ini yg selektif mengizinkan atau menolak lalu lintas jaringan. aturan di WAF secara khusus dibuat untuk menyaring kemudian lintas jaringan dengan memakai protokol *HTTP*. WAF juga mampu mendeteksi serangan awam, seperti *probe* (upaya mendapatkan informasi awal sebelum melakukan serangan) dari serangan *SQL Injection* serta upaya XSS. *Firewall* mampu berupa *software* yg di-*instal* di *host*, atau perangkat keras khusus. WAF disebut juga satu prosedur pertahanan awal pada sistem *website* [7].

2.4. *Wireshark*

Wireshark digunakan untuk melakukan analisa paket data jaringan. *Wireshark* diklaim pula *Network packet analyzer* yg berfungsi menangkap paket-paket jaringan serta berusaha buat menampilkan seluruh informasi paket sedetail mungkin. Sebenarnya *network packet analyzer* sebagai indera untuk menyelidiki apa yang sebenarnya terjadi pada pada jaringan baik kabel juga wireless [8].

2.5. *OWASP*

OWASP adalah alat yang sangat baik untuk menyelesaikan masalah keamanan situs [1]. OWASP menjadi *framework* yang digunakan oleh pengembang dan pakar teknologi untuk mengamankan *website*. OWASP memberikan platform bagi pengembang menaikkan keamanan sistem melalui proyek open-*source* bersama dengan *tools.* OWASP menjadi pendukung dalam pengujian sistem [9].

2.6. *SQL Injection*

SQL Injection menyebabkan penyerang memiliki kemampuan untuk mempengaruhi *query* dari sebuah *SQL* yang akan dikirimkan melalui aplikasi ke *database* yang tidak hanya mempengaruhi aplikasi web nya, tapi juga mempengaruhi semua program lain yang menggunakan kalimat *SQL* [10].

2.7. *DDoS*

Agresi *distributed denial of service* berupa ancaman keamanan di mana penyerang mengirimkan sejumlah besar permintaan palsu ke *host* atau *server*, sebagai akibatnya *host* sasaran menolak akses asal pengguna yang berwenang. Layanan berasal host menjadi tak tersedia. oleh karena itu, serangan itu menghambat ketersediaan sistem. serangan DDoS yang bertujuan untuk pengguna yang legal, klien, pelanggan asal berhasil mengakses internet sudah menyebabkan tantangan serius bagi keamanan jaringan. Bila serangan *distributed denial of service* diluncurkan dari beberapa personal komputer , sering diklaim serangan *Distributed Denial of Service (DDoS)*[11]

2.8. *Cross Site Scripting (XSS)*

XSS merupakan kerentanan yang terjadi pada *website* tanpa dilapisi sistem keamanan yang kuat , sehingga penyerang dengan mudah memasukan kode yang dapat mengekploitasi *website*, kode tersebut berupa *javascript* pada halaman web dengan maksud untuk mendapatkan cookie dan session untuk digunakan sebagai hak akses yang sah kedalam *webisite* [12].

3. METODE PENELITIAN

3.1. Metode Penelitian

Metode yang digunakan yaitu metode eksprimen atau pengujian langsung pada *website* wantilandesa.id. penerapan metode *web application firewall* sebagai sistem keamanan *website*. Hasil dari simulasi pengujian dokumentasikan untuk di analisa sebagai bahan rekomendasi firewall yang efektif untuk *website*. Simulasi pengujian di lakukan untuk peneliti menarik kesimpulan berkenaan dengan implementasi metode WAF sebagai *web security*.

3.2. Kerangka Penelitian

Berikut kerangka penelitian yang menjadi gambaran besar mengenai penelitian yang dilakukan dapat dilihat pada gambar 1.

Gambar 1. Kerangka Penelitian

3.3. Observasi

Pada penelitian ini dilakukan observasi lapangan untuk mengetahui lebih detail terkait penelitian yang di lakukan serta melalui media internet. Dengan tujuan mencari informasi agar informasi tersebut dapat memenuhi kelengkapan untuk bahan penelitian.

3.4. Studi Pustaka

Tahap melakukan pendalaman teori sebagai dasar studi literatur keilmuan untuk dapat menunjang penelitian ini.

3.5. Pengujian Keamanan Sistem

Dalam tahap ini penulis akan mencoba melakukan pengujian keamanan sistem *website* dengan melakukan penetrasi pada *website* wantilandesa.id, penetrasi ini didefinisikan kegiatan pengujian keamanan dari sebuah *software*. Selain penetrasi akan dilakukan juga penyerangan langsung dengan menggunakan *SQL Injection* dan *Distributed Denial of Service (DDoS)*.

3.6. Penarikan Kesimpulan

Dalam tahap ini penulis akan melakukan analisis hasil dan mengambil kesimpulan dari hasil penyerangan tersebut.

4. HASIL DAN PEMBAHASAN

Analisis ini perlu dilakukan untuk mengetahui bagaimana implementasi metode WAF sebagai keamanan *website* wantilandesa.id. Proses implementasi sistem di mulai dengan instalasi sistem operasi *server* yang menggunakan kali linux, yang kemudian dilakukan pembangunan web *server* nginx dan implementasi mod security pada *server*. Setelah itu akan dilakukan konfigurasi basic rule pada mod security agar mengenali serangan dari *attacker* dan konfigurasi pada htacces di *server website*, sehingga modsecurity dapat melakukan penyaringan dan blocking terhadap ancaman serangan.

4.1. Persiapan Alat dan Bahan

Tahapan selanjutnya adalah menyiapkan perangkat maupun tools yang dibutuhkan untuk menunjang penelitian ini diantaranya yaitu:

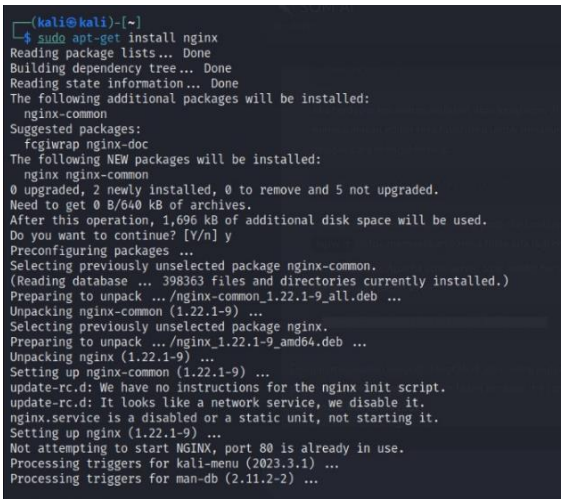
Tabel 1. Kebutuhan *Hardware*

No	Nama perangkat	Spesifikasi
1	Laptop	Acer Swiff 3
2	RAM	8 GB
3	SSD	500 GB
4	*Prosessor*	Intel Core i5

Tabel 2. Kebutuhan *Sofware*

No	Spesifikasi *Sofware*	Komponen
1	Kali Linux	Sistem Operasi
2	*OWASP*	Pengujian WEB untuk melihat kerentanan dan simulasi penyerangan
3	*Wireshark*	memantau lalu lintas yang melibatkan. meliputi pengaturan filter untuk menganalisis lalu lintas yang berkaitan dengan WAF, perekaman lalu lintas, dan analisis hasil perekaman untuk mendeteksi serangan
4	*MySQL*	*Database*
5	*Nginx*	*Web Server*
6	Mozila Firefox	*Browser*
7	*Web Aplication Firewall* *Mod security* dan *htacces*	Aplikasi yag digunakan untuk mencegah serangan

4.2. Instalasi Web *Server*



Gambar 2. Instalasi Web *Server* Nginx

4.3. Instalasi *Web Aplication Firewall* (WAF)

Skenario pengujian dilakukan setelah *website* wantilandesa.id menggunakan *Web Aplication Firewall* (WAF) Dalam skenario ini, tools yang di gunakan adalah mengkonfigurasi source code *Web Aplication Firewall* (WAF) pada htaccess *website* wantilandesa.id dan mengaftifkannya di cloudflare.

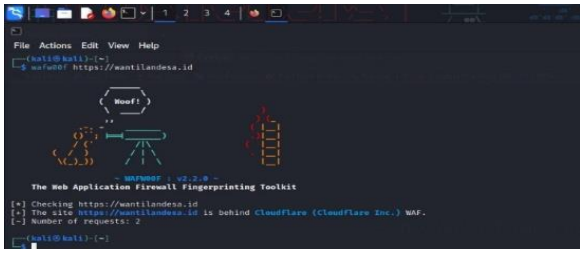
1. Konfigurasi WAF pada *htaccess*

Gambar 3. Konfigurasi WAF pada htaccess

Kemudian aktifkan *Web Aplication Firewall* (WAF) pada menu *security* di cloudflare dan pastikan WAF *enabled* atau aktif

Gambar 4. Mengaktifkan WAF pada cloudflare

2. Cek Aktivasi WAF



Gambar 5. Cek Aktivasi WAF

Dengan meng*input*kan perintah wafw000f wantilandesa.id pada terminal kali linux, akan muncul "*The site is behind WAF*", hal itu menandakan bahwa WAF nya sudah aktif.

4.4. Pengujian Sistem Keamanan *Website*

Pengujian dilakukan untuk mengetahui perbandingan sistem keamanan *website* dan pengaruh penerapan metode *Web Aplication Firewall* terhadap pertahanan sistem keamanan *website* wantiandesa.id pada sistem operasi *kali linux*. Pada tahap pengujian ini dilakukan dengan 2 tahapan, pertama tanpa menggunakan *Web Aplication Firewall* (WAF) dan kedua menggunakan *Web Aplication Firewall* (WAF), untuk melihat apakah *website* wantilandesa.id rentan terhadap serangan, simulasi penetrasi dilakukan dengan tools OWASP yang dapat digunakan untuk menemukan berbagai celah lubang keamanan *website* kemudian simulasi penyerangan menggunakan *SQL Injection* dan DDoS untuk pengujian pada sistem keamanan *website* wantilandesa.id.

4.5. Pengujian Sistem Keamanan *Website* Menggunakan OWASP *Penetration Testing*

Dalam skenario ini, *tools* yang di gunakan adalah Open *Web Application Security Project* sebagai *tools* untuk melakukan pengujian keamanan *website*. OWASP menyediakan panduan, alat, dan metode terstandar untuk mengidentifikasi, mengevaluasi, dan melindungi aplikasi web dari serangan yang berpotensi di exploitasi oleh *attacker.*

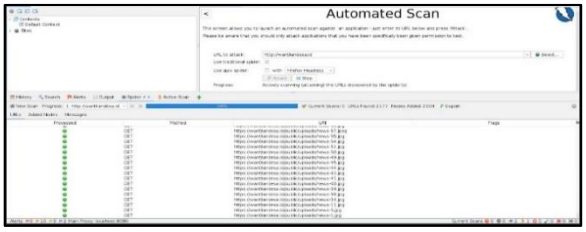
1. Masukan *URL Website*

Masukan alamat *url website* wantilandesa.id untuk dipenetrasi oleh OWASP, kemudian pilih perintah *attack* untuk memulai *scanning*, seperti terlihat pada Gambar 6.

Gambar 6. Tampilan OWASP

2. Proses Penetrasi

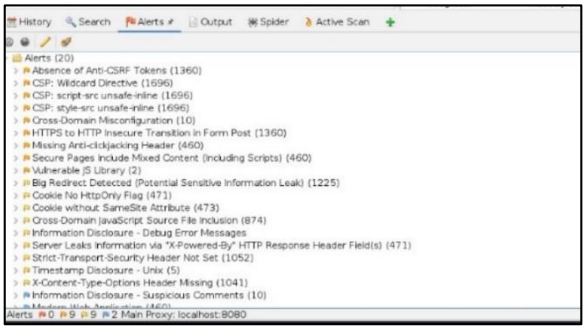
Setelah *memasukan URL website* wantilandesa.id dan melakukan perintah *Start*, maka dengan otomatis OWASP melakukan *scanning* dan penetrasi pada *website* wantilandesa.id untuk mencari celah keamanan yang ada.



Gambar 7. Proses Penetrasi OWASP

3. Hasil Penetrasi (*Alert*)

Setelah proses penetrasi selesai maka OWASP akan menampilkan hasil scaning dan menampilkan berbagai jenis celah keamanan yang terdapat di *website* wantilandesa.id tanpa menggunaka WAF.



Gambar 8. Hasil Penetrasi OWASP tanpa WAF

Hasil penetrasi OWASP pada kolom *alert* merupakan inti dari penetrasi yang di lakukan, OWASP berhasil menemukan banyak celah lubang kerentanan pada keamanan *website* wantilandesa.id . Kerentanan yang di temukan oleh OWASP pada colom *alert* terdapat 20 jenis celah yang beresiko di serang oleh *attacker*, Ancaman yang di temukan pada *website* dapat dilihat pada tabel 3.

Tabel 3. Hasil Penetrasi OWASP Tanpa WAF

No	Jenis Peringatan	Bukti	Celah terdeteksi
1	*Absence of Anti-CSRF Tokens*	<form action=”http://wantilandesa.id/search” method=”post” accept-charset=”utf-8”>	1360
2	*CSP: Wildcard Directive*	*Upgrade-insecure-requests*	1696
3	*CSP: script-src unsafe-inline*	*Upgrade-insecure-requests*	1696
4	*CSP: style-src unsafe-inline*	Upgrade-insecure-requests	1696
5	*Cross-Domain Misconfiguration*	*access-control-allow-origin*: *	10
6	*HTTPS to HTTP Insecure Transition in Form Post*	http://wantilandesa.id/search	1360
7	*Missing Anti-clickjacking Header*		460
8	*Secure Pages Include Mixed Content (Including Scripts)*	http://wantilandesa.id/public/uploads/logo.jpg	460
9	*Vulnerable JS Library*	Jquery-2.2.4.min.js	2
10	*Big Redirect Detected (Potential Sensitive Information Leak)*		1225
11	*Cookie No HttpOnly Flag*	*set-cookie: csrf_cookie_name*	471
12	*Cookie without SameSite Attribute*	*Set-cookie: csrf_cookie_name*	473
13	*Cross-Domain JavaScript Source File Inclusion*	<script type=”text/javascript” src=”//platform-api.sharethis.com/js/sharethis.js#property=5993ef01e2587a001253a261&product=inline-share-buttons”></script>	874
14	*Information Disclosure – Debug Error Messages*	*Under Construction*	
15	*Server Leaks Information via “X-Powered-By” HTTP Response Header Field(s)*	x-powered-by: PHP/7.4.33	471
16	*Strict-Transport-Security Header Not Set*		1052
17	*Timestamp Disclosure – Unix*	1628144437	5
18	*X-Content-Type-Options Header Missing*		1041
19	*Information Disclosure – Suspicious Comments*	Db	10
20	*Modern Web Application*	Statistik	460

Kemudian pengujian dilakukan kembali setelah *website* wantilandesa.id menggunakan *Web Aplication Firewall (WAF)*, Tahap ini merupakan tahap pengujian menggunakan *SQL injection* untuk mengetahui WAF berjalan atau tidak. Bertikut hasil penetrasi OWASP setelah menggunakan WAF.

Gambar 9. Hasil Penetrasi OWASP dengan WAF

Tabel 4. Hasil Penetrasi OWASP dengan WAF

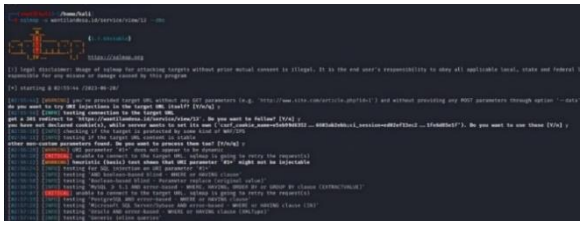
No	Jenis Peringatan	Bukti	Celah terdeteksi
1	*Absence of Anti-CSRF Tokens*	<form action=”http://wantilandesa.id/search” method=”post” accept-charset=”utf-8”>	173
2	*Content Security Policy (CSP)*	*Upgrade-insecure-requests*	92
3	*HTTP to HTTPS Insecure Transition in Form Post*	*Upgrade-insecure-requests*	
4	*Cross-Domain JavaScript Source File Inclusion*	<script async src="https://www.googletagmanager.com/gtag/js?id=G-0VSYWXVH4F" type="da92ebbc01fb50950798d251-text/javascript"></script>	257
5	*Strict-Transport-Security Header Not Set*	access-control-allow-origin: *	

No	Jenis Peringatan	Bukti	Celah terdeteksi
6	*Timestamp Disclosure - Unix*		181
7	*Information Disclosure - Suspicious Comments*	Admin	168
8	*Modern Web Application*		85
9	*Re-examine Cache-control Directives*	public, max-age=604800, s-maxage=604800	107
10	*Retrieved from Cache*		127

Hasil penetrasi OWASP menunjukan dengan diterapkanya *Web Aplication Firewall* (WAF) pada *website* wantilandesa.id dapat mengurangi banyak celah yang riskan terhadap serangan *attacker* yang berpotensi melakukan ekploitasi terhadap *website* wantilandesa.id, ditunjukan pada tabel diatas.

4.6. Pengujian Sistem Keamanan *Website* Tanpa WAF, dengan *SQL Injection*

Pengujian keamanan *website* dilakukan untuk memastikan bahwa sistem yang dibangun aman dari serangan *SQL injection* dan memiliki tingkat keamanan yang optimal. Berikut adalah skenario pengujian yang dilakukan.



Gambar 10. Simulasi penyerangan SQL Injection tanpa WAF

Simulasi pengujian penyerangan dengan *SQL Injection* pada *website* wantilandesa.id dilanjutkan pada saat *website* menerapkan *Web Aplication Firewall* (WAF)

Gambar 11. Simulasi penyerangan SQL Injection Setelah Menggunakan WAF

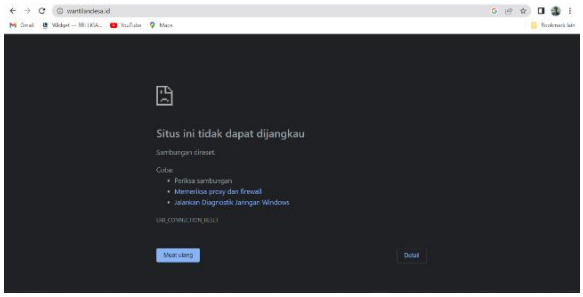
Hasil dari Kedua simulasi pengujian sistem keamanan *website* menggunakan *SQL Injection* pada *website* wantilandesa.id tidak berhasil, hal tersebut karena *website* wantilandesa.id menerapkan *Secure Socket Layer* (SSL) yang merupakan protokol keamanan internet berbasis *enkripsi*.

4.7. Pengujian Sistem Keamanan *Website*, dengan *Distributed Denial of Service* (DDoS)

Serangan DDoS dapat merusak tampilan dan performa suatu situs web sehingga tidak bisa diakses oleh pengguna internet secara normal. Simulasi serangan DDoS pada *server* lokal untuk mengetahui sejauh mana sistem keamanan *website* wantilandesa.id mampu melindungi diri dari serangan DDoS.

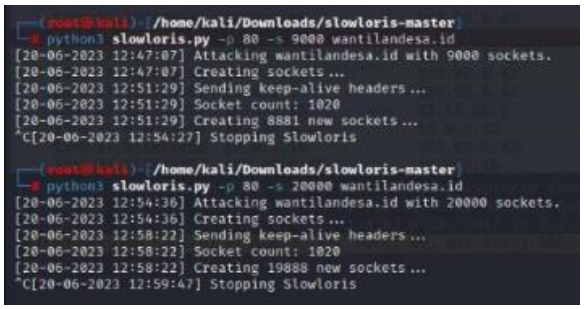
Gambar 12. Simulasi pengujian dengan DDos tanpa WAF

Hasil simulasi penyerangan pada *website* wantilandesa.id berhasil di serang menggunakan DDoS dan menampilkan pesan “Situs ini tidak dapat dijangkau” menunjukan *website* wantilandesa.id tidak dapat di akses atau “Down” karena *Server* diserang menggunakan DDoS Attack selama 3 menit dengan “Attack Limit” dan “Connection Limit” tidak terbatas, Dapat dilihat pada gambar 13.



Gambar 13. Respon website wantilandesa.id

Simulasi pengujian keamanan sistem selanjutnya dilakukan setelah *website* wantilandesa.id menerapkan *Web Aplication Firewall* (WAF) dapat dilihat pada gambar 14.

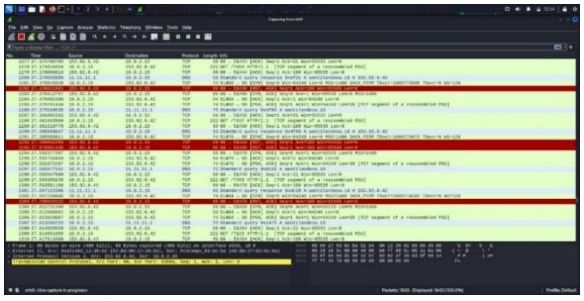


Gambar 14. Simulasi Penyerangan DDoS setelah menggunakan WAF

Pada simulasi penyerangan menggunakan DDoS dapat di bandingkan bahwa implementasi metode WAF dapat mempengaruhi pertahanan dan keamanan *website* dari serangan DDoS.

4.8. Monitoring Pemantauan WAF menggunakan Wireshark

Pada tahap *monitoring*, pemantauan terhadap lalu lintas jaringan yang masuk ke dalam sistem menggunakan perangkat lunak *Wireshark.* Untuk melakukan pemantauan pada WAF, dilakukan instalasi *Wireshark* pada *server* tempat WAF di implementasikan dan kemudian dikonfigurasi *filter* sesuai dengan *port* dan protokol yang digunakan oleh *website* wantilandesa.id. Selama proses *monitoring* berlangsung, data paket jaringan yang diterima oleh server akan ditampilkan secara *real-time* pada antarmuka *Wireshark*. Selanjutnya, menganalisis setiap paket tersebut untuk mengetahui apakah ada serangan atau celah keamanan yang mencoba dieksploitasi.



Gambar 15. Monitoring WAF dengan Wireshark

4.9. Implementasi Metode WAF

Performa penggunaan WAF dapat dilihat melalui hasil *benchmarking* yang telah dilakukan pada saat proses instalasi dan pengujian. Berdasarkan hasil *benchmarking* tersebut, bahwa *Web Aplication Firewall* (WAF) sangat membantu dalam menjaga keamanan *website* wantilandesa.id.

Berikut hasil pengujian pada *website* wantilandesa.id terkait penerapan system keamana *website* menggunakan metode WAF:

Tabel 5. Hasil Pengujian Performansi

Simulasi Pengujian	***Alert* hasil penetrasi OWASP**	***SQL Injection***	***DDos***
Tanpa WAF	20 *Alert*	Gagal	Berhasil
Menggunakan WAF	10 *Alert*	Gagal	gagal

Hasil pengujian performansi dilakukan untuk mengetahui pengaruh dari implementasi metode WAF pada *website* wantilandesa.id. Adapun celah yang berhasil di temukan dari penetrasi menggunakan OWASP, Terlihat pada tabel di atas simulasi tanpa penerapan WAF mendapatkan celah keamanan yang banyak sehingga *website* wantilandesa.id tanpa menggunakan WAF rentan terhadap serangan, dan ketika *website* menggunakan WAF lebih sedikit celah yang di temukan.

Pada simulasi penyerangan *SQL injection* penyerangan tahap 1 dan 2 terbukti gagal karena *website* wantilandesa.id menggunakan SSL sehingga pada tahap penyerangan *SQL Injection* gagal di lakukan. Untuk penyerangan DDoS pada *website* wantilandesa.id simulasi tahap pertama tanpa menggunakan *web aplication firewall* (WAF) berhasil di serang sehingga *website* wantilandesa.id *down* tidak dapat di akses, kemudian setelah *website* menggunakan *web application firewall* (WAF), pengujian serangan DDoS kepada *website* wantilandesa.id gagal diserangan dan aman dari serangan DDoS.

sehingga terbukti penerapan sistem keamanan *website* wantilandesa.id menggunakan *Web Aplication Firewall* (WAF) dapat meningkatkan keamanan *website* membantu dan mengatasi dari serangan *SQL Injection* dan DDoS.

5. KESIMPULAN DAN SARAN

Berdasarkan hasil implementasi dan pengujian *website* menggunakan *Web Aplication Firewall* (WAF), Dengan adanya WAF secara signifikan tingkat keamanan web wantilandesa.id meningkat, dibuktikan dari beberapa pengujian yang di lakukan oleh peneliti yaitu *SQL injection*, *cross-site scripting* (XSS), dan serangan DDoS. dimana WAF dapat mendeteksi dan mecegah serangan-serangan tersebut. WAF memberikan lapisan tambahan yang efektif terhadap serangan yang mungkin terlewatkan oleh *firewall*. WAF juga dapat mengurangi jumlah serangan terhadap *website* dimana hasil dari pengujian yang di lakukan serangan tersebut lebih sering gagal karena *website* sudah di lindungi oleh *Web Aplication Firewall* (WAF). Hasil ini membuktikan pentingnya untuk dapat membuat *website* menggunakan sistem keamanan yang lebih baik agar dapat mengantisipasi jenis serangan yang lain, dan dapat mengembangkan dari segi keamanan *database* menggunakan tools lain untuk mencari mencelah keamanan *website* selain OWASP sehingga aplikasi web yang dibangun tidak

hanya mempunyai desain yang baik namun juga terjaga integritas datanya.

DAFTAR PUSTAKA

[1] R. Riska and H. Alamsyah, "Penerapan Sistem Keamanan Web Menggunakan Metode Web Aplication Firewall," *J. Amplif. J. Ilm. Bid. Tek. Elektro Dan Komput.*, vol. 11, no. 1, pp. 37–42, 2021, doi: 10.33369/jamplifier.v11i1.16683.

[2] Widianto, Septian Rheno, and Izzudin Abdullah Azzam. 2018. "Analisis Upaya Peretasan Web Application Firewall Dan Notifikasi Serangan Menggunakan Bot Telegram." Elektra 3(2): 19–28.

[3] S. J. I. I. Risma Yanti Jamain, Periyadi, "Implementasi Keamanan Aplikasi Web Dengan Web Application Firewall," *e-Proceeding Appl. Sci.*, vol. 1, no. 3, pp. 2191–2195, 2015.

[4] Puriwigati, A. N., & Buana, U. M. (2020). Sistem Informasi Manajemen-Keamanan Informasi. May.

[5] R. Dimas Prakoso and Asmunin, "Implementasi dan Perbandingan Performa Proxmox dalam Virtualisasi dengan Tiga Virtual *Server* (Studi Kasus : Jurusan Teknik Informatika UNESA)," *J. Manaj. Inf.*, vol. 8, no. 1, pp. 79–86, 2017, [Online]. Available: https://jurnalmahasiswa.unesa.ac.id/index.php/11/article/view/22864

[6] A. Hamzah, S. Juli, I. Ismail, L. Meisaroh, S. Si, and M. Si, "Implementasi Sistem Monitoring Jaringan Menggunakan Zabbix dan Web *Web Application Firewall* di PT PLN (Persero) Transmisi Jawa Bagian Tengah," e-Proceeding Appl. Sci., vol. 5, no. 3, pp. 2378– 2384, 2019.

[7] M. M. M. H. F. C. S. Andy Rachman, "IMPLEMENTASI DAN ANALISIS JARINGAN MENGGUNAKAN WIRESHARK, CAIN AND ABELS, NETWORK MINNER," Jurnal Cendikia Vol.XVI, p. 2, 2014.

[8] A. Elanda and R. Lintang Buana, "ANALISIS KUALITAS KEAMANAN SISTEM INFORMASI E-OFFICE BERBASIS *WEBSITE* PADA STMIK ROSMA DENGAN MENGGUNAKAN OWASP TOP 10," 2021

[9] Ghozali, Bahrun, Kusrini Kusrini, and Sudarmawan Sudarmawan. 2019. "Mendeteksi Kerentanan Keamanan Aplikasi *Website* Menggunakan Metode Owasp (Open Web Application Security Project) Untuk Penilaian Risk Rating." Creative Information Technology Journal 4(4): 264..

[10] Yulianingsih, "Menangkal Serangan *SQL Injection* dengan Parameterized Query," vol. 2, no. 1, pp.

[11] Suroto, S. (2018) 'A Review of Defense Against Slow *HTTP* Attack', JOIV : International Journal on Informatics Visualization, 1(4), p. 127. doi: 10.30630/joiv.1.4.51.

[12] Wasef, Abdalla, Zarul Fitri, and Herman Khalid. 2017. "Web Security: Detection of Cross Site Scripting in PHP Web Application Using Genetic Algorithm." International Journal of Advanced Computer Science and Applications 8(5).