Jutisi: Jurnal Ilmiah Teknik Informatika dan Sistem Informasi
Jl. Ahmad Yani, K.M. 33,5 - Kampus STMIK Banjarbaru
Loktabat – Banjarbaru (Tlp. 0511 4782881), e-mail: puslit.stmikbjb@gmail.com
e-ISSN: 2685-0893
p-ISSN: 2089-3787

Analisis keamanan *Content Delivery Network* (CDN) *Cloudflare* (Studi kasus: Web *Hakazon)*

Eko Putra Rahmadyanto[1*], Dian Widiyanto Chandra [2]
Teknik Informatika, Universitas Kristen Satya Wacana, Salatiga, Indonesia
*e-mail *Corresponding Author*: ekoputrarahmadyanto@gmail.com

Abstract

Content Delivery Network (CDN) is a server network system that has the function of distributing servers globally. On the Cloudflare CDN, there is a Web Application Firewall (WAF) feature that is used to secure servers and websites so that Cloudflare can take preventive actions against hacker attacks. The purpose of this study is to analyze and compare the security of websites using Cloudflare CDN and without Cloudflare CDN. The method used in this research is research and comparison. The results show that Cloudflare's CDN security is effective if it is used to strengthen the security of a website in the form of preventive actions using the CDN and WAF contained in it. In addition, requests that are considered dangerous will be blocked by WAF so that website security becomes stronger on the server-side.
Keywords: *Content Delivery Network; Hacker; Website Security; Web Application Firewall*

Abstrak

Content Delivery Network (CDN) merupakan sistem jaringan server yang memiliki fungsi mendistribusikan server secara global. Pada CDN *Cloudflare* terdapat fitur *Web Aplication Firewall* (WAF) yang dimanfaatkan untuk melakukan pengamanan pada server dan *website* sehingga *Cloudflare* dapat melakukan tindakan *preventif* terhadap serangan peretas. Tujuan penelitian ini adalah menganalisis dan membandingkan keamanan *website* yang menggunakan CDN *Cloudflare* dan tanpa CDN *Cloudflare*. Metode yang digunakan dalam penelitian ini adalah penelitian dan perbandingan. Hasil penelitian menunjukkan bahwa keamanan CDN *Cloudflare* efektif jika digunakan untuk memperkuat keamanan suatu *website berupa* tindakan *preventif* menggunakan CDN dan WAF yang ada di dalamnya. Selain itu, *request* yang dianggap berbahaya akan dilakukan *block* oleh WAF sehingga keamanan *website* menjadi lebih kuat melalui sisi server.
Kata kunci: *Content Delivery Network; Peretas; Keamanan Website; Web Application Firewall*

1. Pendahuluan

Pada tahun 2021 sudah terdapat total lebih dari 1,9 triliun *website* dan lebih dari 5,1 triliun pengguna internet yang angkanya akan semakin bertambah seiring berjalannya waktu [1]. Sejak *website* pertama kali dibuat pada tahun 1991 oleh Tim Berners-Lee hingga saat ini pada tahun 2021 *website* berserta Teknologi Jaringan telah mengalami banyak perkembangan, namun jumlah pengguna internet-pun juga meningkat dengan sangat cepat, oleh karena itu suatu server harus memiliki kapasitas untuk dapat melayani jutaan *user* dari seluruh dunia secara bersamaan tanpa mengurangi latensi yang ada atau yang terparah yaitu *down* [2], [3]. Selain memperhatikan kemampuan dari suatu server tentunya perlu juga untuk memperhatikan keamanan atau *security* dari suatu jaringan maupun server agar tidak terdapat celah atau *vulnerability* yang dapat dimanfaatkan oleh *threat actor* atau *hacker* yang tentu saja hal tersebut akan menimbulkan kerugian bagi korban [4].

Content Delivery Network (CDN) merupakan sistem jaringan server yang memiliki fungsi mendistribusikan server secara global sehingga *client* tidak harus melakukan *request* secara langsung ke server utama yang jaraknya bisa saja sangat jauh dan membutuhkan waktu yang sangat lama untuk melakukan *download* konten, CDN akan melakukan *intercept* pada *request* dan akan mengalihkan *request* tersebut ke server CDN terdekat yang menyebabkan *latency* menjadi lebih rendah [5]. Hal ini juga sangat efektif untuk mengurangi jumlah tugas yang datang pada server utama karena sebagian atau keseluruhan konten yang ada pada

server utama telah di-replikasi pada server CDN, sehingga *request* untuk melakukan *download* konten tidak harus dilayani oleh server utama [6].

Pada *Content Delivery Network* (CDN) *Cloudflare* di dalamnya juga terdapat fitur *Web Aplication Firewall* yang dapat dimanfaatkan untuk melakukan pengamanan pada server dan *website* sehingga *Cloudflare* dapat melakukan tindakan *preventif* terhadap serangan yang dilakukan oleh peretas [7], [8].

Setiap hari terdapat lebih dari ratusan *website* maupun infrastruktur jaringan yang terkena aksi usil *hacker* yang dengan sengaja merusak, mengambil, mengubah, maupun menghapus data dari suatu *website*, server maupun infrastruktur jaringan yang tentunya dilakukan secara ilegal, hal tersebut dapat dilihat melalui *website Honey-Net* milik Badan Siber dan Sandi Negara (honeynet.bssn.go.id) dan *website* arsip *defacement* (zone-h.org) [9], [10], Maka dari itu diperlukan suatu konfigurasi maupun teknologi yang dapat mencegah *hacker* untuk dapat melakukan serangan ke suatu *website*, server, maupun jaringan [11].

Berdasarkan penelitian sebelumnya yang relevan dan terkait dengan CDN dan WAF belum ada yang membahas tentang Analisis keamanan *Content Delivery Network* (CDN) *Cloudflare* studi kasus web Hakazon oleh karena itu penulis melakukan penelitian tentang Analisis Keamanan *Content Delivery Network* (CDN) *Cloudflare* Studi Kasus Web Hakazon dan cara mengimplementasikan *Content Delivery Networks* (CDN) *Cloudflare* pada *website* dengan tujuan untuk meneliti seberapa efektif *Cloudflare* CDN dan *Web Aplication Firewall* (WAF) untuk menjaga keamanan pada suatu *website*. Penelitian dilakukan dengan cara membandingkan keamanan pada *website* yang telah terimplementasi *Cloudflare* CDN dan WAF dengan *website* yang belum diimplementasikan. Penulis melakukan serangan seperti *Sql Injection, Cross Site Scripting* (XSS), *BruteForce*, dan serangan-serangan lainnya pada *website* yang belum diimplementasikan *Cloudflare* CDN dan WAF dan pada *website* yang telah diimplementasikan *Cloudflare* CDN dan WAF sehingga nantinya dapat dilakukan perbandingan keamanan pada *website* tersebut [12]. Penelitian ini diharapkan akan bermanfaat bagi *system admin, developer,* dan profesi yang lainnya untuk mengetahui tentang seberapa efektif *Cloudflare* CDN dan *Web Aplication Firewall* dalam menjaga keamanan suatu *website*, sehingga nantinya penelitian ini dapat menjadi bahan pertimbangan apabila akan melakukan implementasi *Cloudflare* CDN dan *Web Aplcation Firewall* (WAF). Tujuan penelitian ini adalah menganalisis keamanan *website* yang menggunakan CDN *Cloudflare* dan melakukan perbandingan dengan *website* yang tidak menggunakan CDN *Cloudflare*.

2. Tinjauan Pustaka

Penelitian terkait sebelumnya dilakukan oleh Dewi Laksmiati [13] meneliti tentang implementasi *Content Delivery Network* untuk mengatasi meningkatnya jumlah pengguna internet yang membuat *traffic* internet menjadi lebih padat dan berat untuk server, Penelitian ini menyimpulkan bahwa penerapan *Cloudflare Content Delivery Network* dapat mengatasi masalah jumlah *traffic* yang banyak, karena request yang datang tidak harus dilayani oleh server utama namun dapat dilayani oleh server CDN terdekat sehingga membuat *latency* menjadi lebih kecil.

Ramadhan [14] melakukan penelitian yang bertujuan untuk membahas strategi yang paling tepat dalam menjaga keamanan *cyber* di Kawasan Asia Tenggara. Penelitian ini juga membuktikan bahwa *Cybersecurity* perlu mendapatkan prioritas dalam studi keamanan karena pada saat ini banyak kebutuhan yang tidak dapat terlepas dari dunia maya bahkan kebutuhan negara-bangsa tidak dapat terlepas dari peranan dunia maya.

I Gede Putu Krisna Juliharta [15] menyatakan bahwa suatu *website* akan diakses dari berbagai negara yang berbeda dan tentu saja di setiap negara akan memiliki kecepatan *download* konten yang berbeda-beda tergantung seberapa jauh dari lokasi server utama atau *origin* server, maka *Content Delivery Network* dapat mengatasi masalah tersebut. Server CDN mendistribusikan konten yang ada dalam sebuah aplikasi/web ke berbagai pengakses/pengguna di berbagai belahan dunia agar data/konten yang dikirim dapat diterima lebih cepat selain itu Server CDN *Cloudflare* terdapat di seluruh belahan dunia, sehingga *request* yang datang akan dialihkan ke server CDN Terdekat.

Paulus Miki Resa Gumilang dan Dian Widiyanto Chandra [9] menyatakan bahwa setiap hari ada belasan sampai puluhan *website* yang diretas, aksi *hacking* saat ini bisa dibilang masih sangat marak dan sangat banyak dilakukan, kurang perhatiannya pemerintah Indonesia tentang hal ini dirasa sangat miris dan patut dipertanyakan karena tidak sedikit *website* milik

pemerintahan Indonesia yang diretas oleh *hacker* dan dapat dilihat di situs arsip zone-h.org. Oleh karena itu sangat penting untuk memperhatikan faktor keamanan pada suatu sistem maupun aplikasi agar nantinya tidak menimbulkan kerugian yang besar.

Dalam konsep penelitian ini yang membedakan dari penelitian-penelitian sebelumnya yaitu penelitian ini bertujuan untuk menginvestigasi sejauh mana efektivitas CDN Cloudflare dan Web Application Firewall (WAF) dalam menjaga keamanan sebuah situs web. Metodologi penelitian melibatkan perbandingan keamanan antara situs web yang telah mengimplementasikan CDN Cloudflare dan WAF dengan situs web yang belum mengadopsinya. Penulis melakukan serangan seperti Sql Injection, Cross Site Scripting (XSS), BruteForce, dan serangan-serangan lainnya terhadap kedua jenis situs web tersebut. Hasil penelitian ini diharapkan memberikan wawasan yang bermanfaat bagi administrator sistem, pengembang, dan profesional lainnya untuk memahami efektivitas CDN Cloudflare dan Web Application Firewall dalam melindungi keamanan situs web. Dengan demikian, penelitian ini dapat menjadi acuan penting ketika akan mengimplementasikan CDN Cloudflare dan Web Application Firewall (WAF) dalam pengelolaan keamanan suatu situs web.

3. Metodologi

Dalam pelaksanaan penelitian, penulis memilih penelitian dan perbandingan sebagai metode penelitian, dikarenakan penulis akan melakukan implementasi CDN dan WAF kemudian akan membandingkan keamanan antara *website* yang telah diimplementasi dan *website* yang belum diimplementasi.

Gambar 1. Metode Penelitian

Tahapan – tahapan dari gambar 1 adalah sebagai berikut:

1) Identifikasi Masalah
   Pada tahap Identifikasi adalah melakukan identifikasi pada masalah yang menjadi topik penelitian di lapangan.
2) Melakukan Studi
   Pada tahap ini penulis melakukan studi terhadap permasalahan yang ada sekaligus mempelajari cara mengatasi permasalahan yang ditemukan
3) Implementasi dan Pengujian
   Pada tahap ini dilakukan implementasi CDN & WAF untuk kemudian dilakukan pengujian pada keamanan webite yang telah diimplementasi.
4) Penulisan Laporan Penelitian
   Pada tahap akhir ini akan dilakukan penulisan laporan dari penelitian yang telah dilakukan sebagai bentuk dari dokumentasi serta *summary* rampungnya sebuah *project* penelitian.

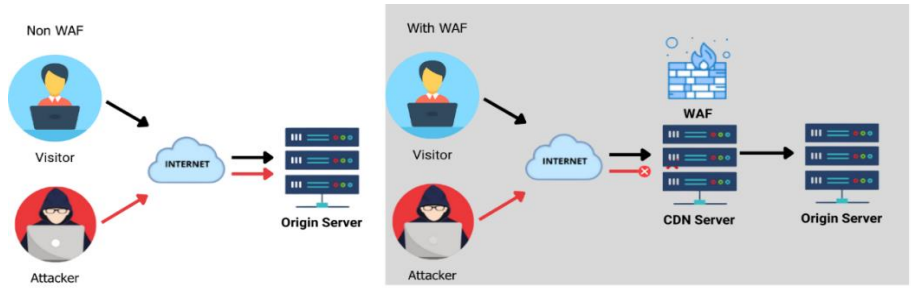
Alat-alat yang digunakan dalam melakukan penelitian ini adalah sebagai berikut:

1) Cloudflare
   Cloudflare digunakan untuk melakukan implementasi *Content Delivery Network* dan *Web Application Firewall.*

2) *Browser*
   Browser digunakan untuk membuka *Cloudflare* dan melakukan konfigurasi CDN dan WAF, selain itu *browser* juga digunakan pada saat proses testing aplikasi
3) Laptop
   Dalam proses Implementasi CDN dan WAF dibutuhkan perangkat komputer yang nantinya digunakan untuk melakukan konfigurasi.
4) Burpsuite
   Burpsuite merupakan *tool* yang digunakan untuk membantu proses testing keamanan pada aplikasi.

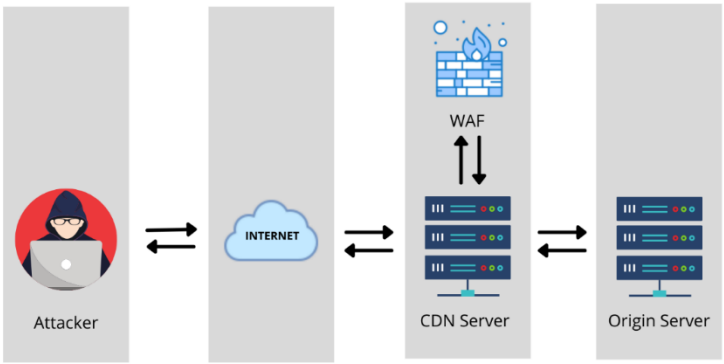
4. Hasil dan Pembahasan

Sebelum melakukan penelitian terkait dengan implementasi CDN & WAF, terlebih dahulu akan dilakukan desain topologi dalam lingkungan *environment system* atau infrastruktur yang akan diteliti. Topologi ini sekaligus menjadi alur yang akan dilewati *request* yang dikirimkan oleh *attacker.* Adapun topologi jaringan yang akan digunakan adalah sebagai berikut :



Gambar 2. Topologi *Non* WAF dan Menggunakan WAF

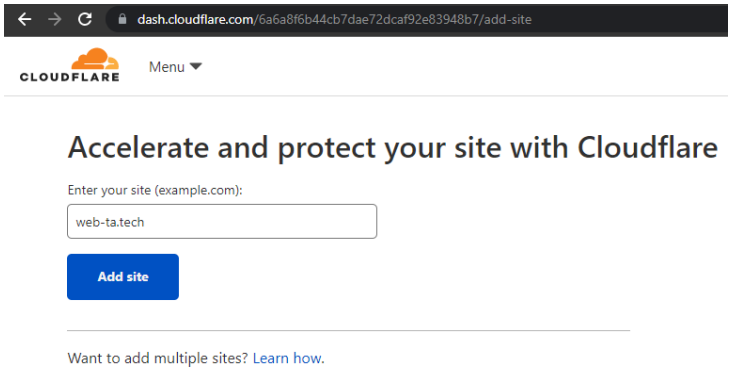
Berdasarkan gambar 2 pada sisi kiri dapat dilihat bahwa *user* maupun *attacker* mengkases langsung menuju *website* tanpa melalui CDN & WAF, Sedangkan pada sisi kanan, sebelum *user* maupun *attacker* mengakses *website* maka akan melewati CDN yang memiliki WAF.



Gambar 3 *Alur Request Attacker*

Attacker akan mengirimkan *payload* maupun *malicious request* melalui internet yang kemudian *request* akan dialihkan ke server CDN terdekat, CDN memiliki *Web Application Firewall* yang dapat melakukan filter pada *request* yang datang, apabila WAF mendeteksi adanya *malicius request* maka akan langsung dilakukan *block.*

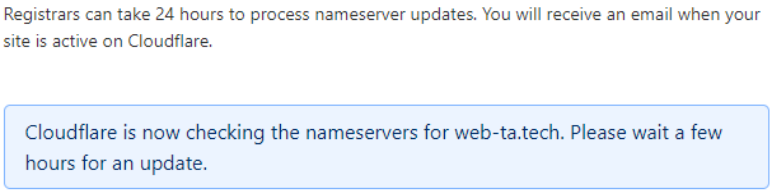
Gambar 4 merupakan *form* penambahan domain yang digunakan untuk menambahkan nama domain yang akan diimplementasikan CDN dan WAF pada *Cloudflare*. Hal pertama yang harus dilakukan yaitu membuka halaman *website Cloudflare* dan melakukan pendaftaran akun terlebih dahulu agar dapat menggunakan fitur yang dimiliki *Cloudflare*. Mencantumkan nama domain yang akan dilakukan implementasi pada *form Enter Your Site* kemudian dilanjutkan dengan melakukan klik pada tombol *Add Site*.



Gambar 4. *Form* Penambahan Domain

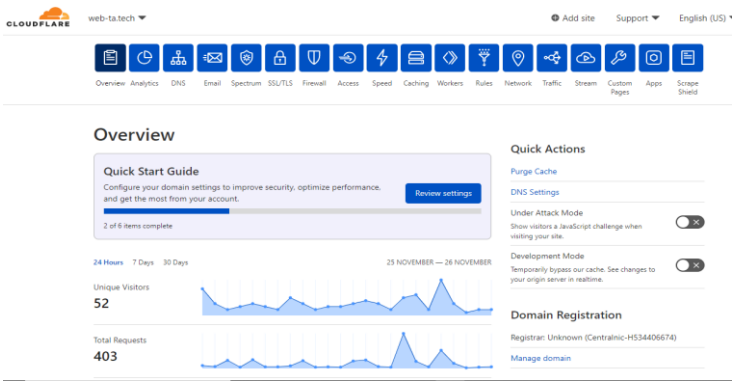
Gambar 5. *Cloudflare Nameserver*

Gambar 5 merupakan *Cloudflare nameserver* yang harus digunakan pada domain. Mengubah dan melihat *nameserver* dapat dilakukan dengan cara login pada *administrator account* domain *registrar,* apabila domain masih baru dan belum pernah diubah maka akan terdapat 4 *nameservers default,* hapus ke 4 *nameservers* tersebut kemudian ganti dengan *nameservers Cloudflare*.



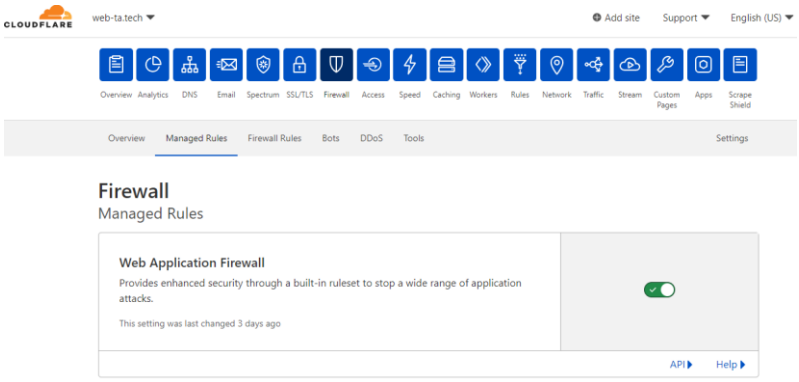
Gambar 6. Proses Pengecekan *Nameserver* pada Domain *Web-Ta.Tech*

Gambar 6 merupakan pemberitahuan yang akan didapatkan setalah selesai melakukan perubahan pada *nameserver,* pesan ini memberitahukan bahwa proses pengecekan *nameserver* pada domain *web-ta.tech* akan memakan waktu beberapa jam hingga *nameserver* berhasil diverifikasi dan CDN berhasil diimplementasikan.



Gambar 7. *Dashboard Cloudflare*

Gambar 7 merupakan tampilan dari *Dashboard Cloudflare* yang akan muncul setelah CDN berhasil diimplementasi. Pada saat CDN berhasil diimplementasi, Status *Web Application Firewall* pada CDN tidak akan langsung *Running,* perlu diaktifkan secara manual agar *Web Application Firewall* dapat menyala.

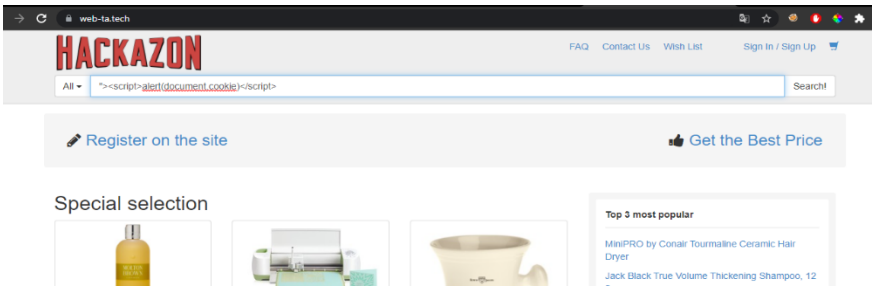


Gambar 8. *Firewall Managed Rules*

Gambar 8 merupakan tampilan dari *Dashboard* pada bagian *Firewall manage Rules*, bagian ini digunakan untuk mengaktifkan *Web Application Firewall* dan *Ruleset* yang telah disediakan oleh *Cloudflare* untuk melindungi *website* dengan cepat dan efektif tanpa harus membuat *rule* secara manual. Melakukan pengaktifan *Web Application Firewall* dapat dilakukan dengan cara menekan radio *button* seperti yang ada pada Gambar 5.

Vulnerability Assessment dilakukan secara *Grey Box, Grey Box testing* merupakan metode pengujian perangkat lunak yang digunakan untuk menguji keamanan dari suatu perangkat lunak dimana tester telah mengetahui sebagian dari struktur *internal* kode atau program. *Website* yang akan dilakukan *penetration testing* belum diimplementasikan CDN dan WAF, oleh karena itu *website* akan sangat mungkin memiliki celah keamanan atau *vulnerability,* maka dari itu penulis akan melakukan beberapa *Vulnerability Assessment* supaya nanti dapat dibandingkan keamanannya dengan *website* yang telah terimplementasi CDN dan WAF.

XSS *Reflected (Cross site Scripting)* adalah kerentanan yang memungkinkan penyerang mengirim kode berbahaya (biasanya dalam bentuk Javascript) dan disisipkan ke dalam *form* atau *link* pada *website.* XSS *Reflected* biasanya memunculkan *notifikasi error* dan juga merusak tampilan *website.*



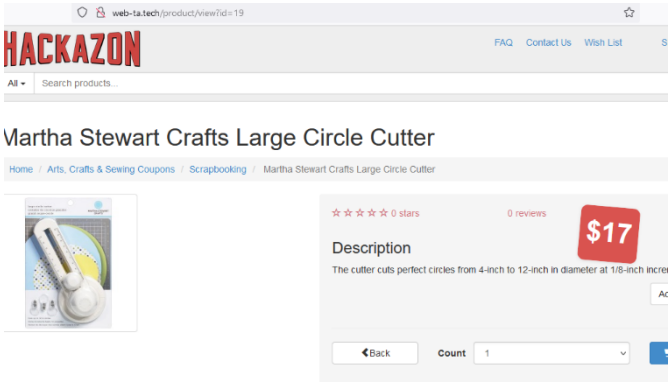
Gambar 9. Input *Payload* pada Kolom Pencarian

Gambar 9 merupakan penginputan *Payload* XSS pada kolom pencarian, berikut adalah *payload* yang penulis inputkan "><sctipt>alert(document.cookie)</script> *payload* tersebut dibuat dengan tujuan untuk memunculkan *pop-up* yang berisi *cookie* dari *user* yang sedang aktif.

Gambar 10 merupakan tampilan *popup payload* XSS yang telah berhasil tereksekusi, pada *pop-up* tersebut terdapat *cookie* dari *user* yang sedang aktif. *Parameter input* tidak menggunakan *filter* metakarakter sehingga memungkinkan untuk mengeksekusi Skrip XSS, Pengguna berbahaya dapat menyuntikkan JavaScript, VBScript, ActiveX, HTML, atau Flash ke aplikasi yang rentan untuk menipu pengguna agar mengumpulkan data dari mereka. Penyerang

dapat mencuri *cookie* sesi dan mengambil alih akun, menyamar sebagai pengguna. Dimungkinkan juga untuk memodifikasi konten halaman yang disajikan kepada pengguna.

Gambar 10. *Payload* berhasil tereksekusi

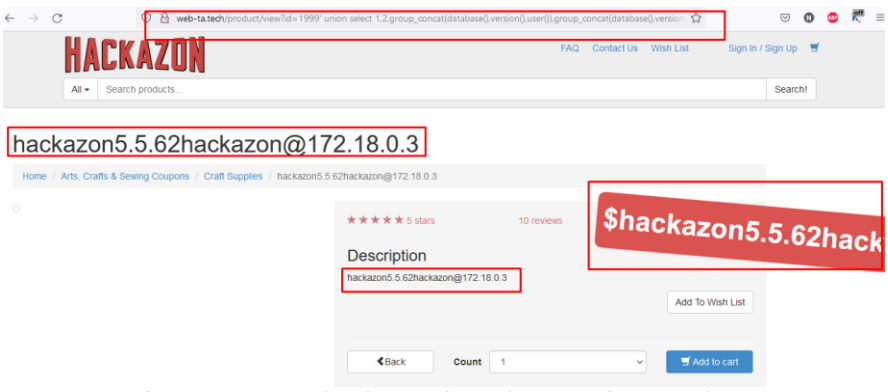


Gambar 11. Tampilan Halaman *View Product*

Gambar 11 merupakan tampilan dari halaman *View Product,* pada halaman ini penulis menemukan *vulnerability* berupa SQL *Injection* pada *parameter* ID yang muncul karena belum adanya sanitasi inputan pada parameter tersebut, SQL *Injection* merupakan teknik eksploitasi dengan cara memodifikasi perintah sql pada *form* input aplikasi yang memungkinkan penyerang untuk dapat mengirimkan *sintaks* ke *database* aplikasi. SQL *injection* juga dapat didefinisikan sebagai teknik eksploitasi celah keamanan pada layer *database* untuk mendapatkan *query* data pada sebuah aplikasi.

```
http://web-
ta.tech/product/view?id=1999%27%20union%20select%201,2,group_concat(database(),
version(),user()),group_concat(database(),version(),user()),5,group_concat(database()
version(),user()),7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27--%20
```

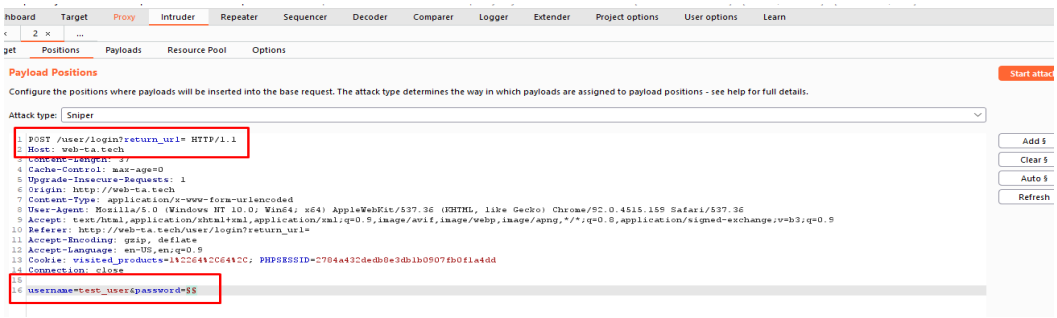
Kode di atas merupakan URL dari *website* yang sudah ditambahkan dengan *payload* SQL *injection* yang penulis gunakan untuk melakukan injeksi pada parameter ID.



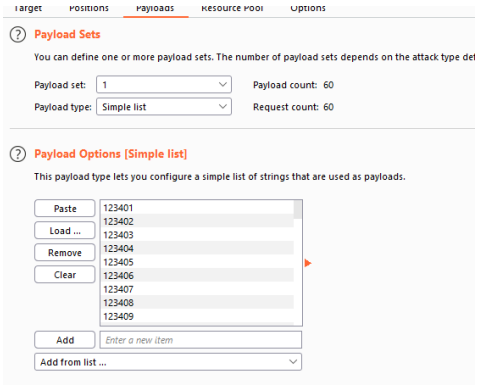
Gambar 12. SQL *Injection* pada Halaman *View Product*

Gambar 12 merupakan tampilan halaman *View Product* yang telah berhasil dilakukan injeksi SQL. Penulis berhasil mendapatkan nama *database*, versi dari *database* yang digunakan, serta *username* dari *admin database* hal ini membuktikan bahwa pada *parameter* ID dapat dilakukan injeksi SQL.

Gambar 13 merupakan tampilan *Intruder* Burpsuite, penulis mencoba untuk melakukan *bruteforce* pada halaman *login*, *brute force* merupakan upaya yang dilakukan peretas untuk bisa masuk ke dalam suatu sistem dengan cara mencoba-coba kata sandi sampai menemukan kode yang tepat.

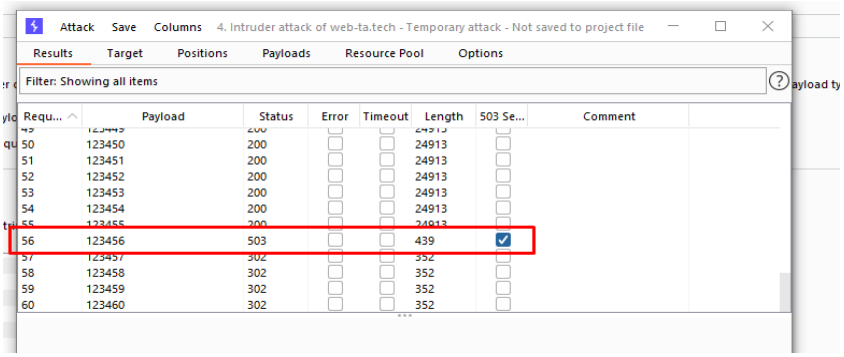


Gambar 13. Tampilan *Intruder* Burpsuite



Gambar 14. Tampilan *Tab Set Payload* pada *Intruder* Burpsuite

Gambar 14 merupakan *tab* yang digunakan untuk memasukkan *payload* atau *word list* yang akan digunakan untuk melakukan Bruteforce pada *login page*, *payload* berupa kombinasi huruf dan angka acak yang nantinya digunakan untuk menebak *password* dari suatu *user.*



Gambar 15. Hasil dari *Bruteforce*

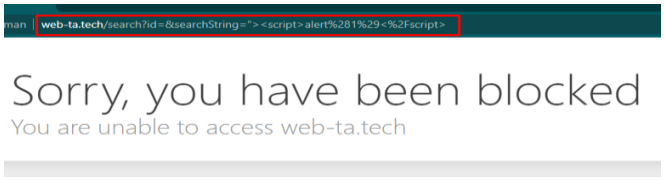
Gambar 15 merupakan hasil dari *bruteforce*, setelah melakukan proses *brutefore* selama kurang dari 1 menit dengan lebih dari 50 *request login* dikirimkan, penulis berhasil menemukan *password* menggunakan metode *bruteforce*. *Website* tidak membatasi berapa jumlah maksimal *request* yang dapat dikirimkan dalam suatu waktu.

Tabel 1 merupakan rekapan dari *vulnerability* yang telah ditemukan, tabel berisikan nama *vulnerability*, *path* dimana *vulnerability* berada, dan juga *status* sebagai tanda apakah *vulnerability* masih terbuka atau tidak. Dari hasil yang sudah penulis temukan, terdapat 3 *vulnerability* yang masih memiliki status *Open.*

Tabel 1. *VA* pada *Website* yang Telah Diimplementasi CDN dan WAF

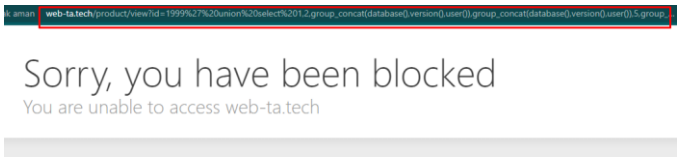
No.	*Vulnerability*	*Path*	Status
1	*Cross Site Scripting* (XSS)	http://web-ta.tech/search	*Open*
2	SQL *Injection*	http://web-ta.tech/product/view	*Open*
3	*No Rate Limiting - Login*	http://web-ta.tech/user/login	*Open*

Vulnerability Assessment pada *website* yang telah diimplementasi dan WAF dilakukan secara *Grey Box* sama seperti sebelumnya. *Grey Box testing* merupakan metode pengujian perangkat lunak yang digunakan untuk menguji keamanan dari suatu perangkat lunak dimana tester telah mengetahui sebagian dari struktur *internal* kode atau program. *Website* telah diimplementasikan CDN dan WAF yang membuat keamanan *website* akan menjadi lebih kuat dari pada yang belum terimplementasi CDN dan WAF, oleh karena itu penulis akan melakukan *Vulnerability Assessment* menggunakan metode serangan yang sama dengan yang digunakan pada *Website* yang belum terimplementasi WAF dan CDN agar dapat dilakukan perbandingan pada keduanya.



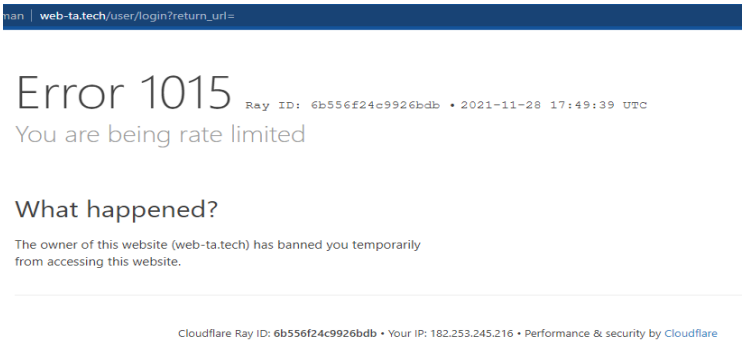
Gambar 16. *Blocked* XSS *Payload Request*

Gambar 16 merupakan tampilan dari *Website* yang dikirimkan XSS *Payload, request* yang ditujukan untuk menyuntikkan kode JavaScript, VBScript, ActiveX, HTML akan langsung di*block* oleh *Web Application Firewall*, maka dapat dikatakan bahwa *vulnerability Cross Site Scripting* telah dapat ditutup dari sisi server.



Gambar 17. *Blocked SQL Injection Payload Request*

Gambar 17 merupakan tampilan dari *website* yang dikirimkan SQL *Injection Payload, request* yang ditujukan untuk menyuntikkan *Payload* SQL akan langsung di-*block* oleh *Web Application Firewall,* maka dapat dikatakan bahwa *vulnerability SQL Injection* telah ditutup dari sisi server dengan menggunakan *Web Application Firewall*.



Gambar 18. *Rate Limit*

Gambar 18 merupakan tampilan *website* yang telah melakukan *block* pada *user* yang melakukan lebih dari 5 *request* dalam waktu yang singkat, *Web Application Firewall* dapat melakukan *rate limiting* sehingga *user* hanya dapat mengirimkan beberapa *request* dalam suatu waktu, hal ini tentu saja akan membuat *website* menjadi lebih aman dan *website* akan terhindar dari serangan seperti Brutefore, DOS, DDOS.

Tabel 2 *VA* pada Website yang Telah Diimplementasi CDN dan WAF

No.	*Vulnerability*	*Path*	Status
1	*Cross Site Scripting* (XSS)	http://web-ta.tech/search	*Close*
2	SQL *Injection*	http://web-ta.tech/product/view	*Close*
3	*No Rate Limiting* - Login	http://web-ta.tech/user/login	*Close*

Tabel 2 merupakan rekapan dari hasil *Vulnerability Assesment* pada *website* yang telah diimplementasikan CDN dan WAF. Dari hasil *Vulnerability Assessment* ditemukan bahwa *vulnerability* yang tadinya ditemukan telah berhasil dilakukan tindakan *preventif* dengan menggunakan *Web Application Firewall. Malicious request* yang dikirimkan oleh penulis dapat di-*block* oleh WAF sehingga serangan tidak lagi dapat dilakukan.

Tabel 3 Perbandingan Keamanan Antara *website* yang Telah Diimplementasi CDN & WAF dan sebelum Diimplementasi.

		Close	
No	***Vulnerability***	Tidak Menggunakan CDN	Menggunakan CDN
1	*Cross Site Scripting*	*Open*	*Close*
2	SQL *Injection*	*Open*	*Close*
3	*No Rate Limiting - Login*	*Open*	*Close*

Tabel 3 merupakan perbandingan keamanan antara *website* yang telah diimplementasi CDN & WAF dan *website* yang belum terimplementasi CDN & WAF, terdapat 3 *vulnerability* yang ditemukan pada *website* yang belum terimplementasi CDN & WAF, ke 3 *vulnerability* tersebut yaitu *Cross Site Scripting*, SQL *Injection*, *No Rate Limiting* pada halaman *Login*. Karena ke tiga *vulnerability* tersebut masih tersedia pada *website* yang belum terimplementasi CDN & WAF maka status dari *vulnerability* tersebut adalah *Open*. Pada *website* yang telah terimplementasi CDN & WAF ketiga *vulnerability* yang sebelumnya telah ditemukan menjadi tidak tersedia, dikarenakan *website* telah dilindungi dari sisi server, maka status *vulnerability* yang sebelumnya *Open* diubah menjadi *Close* karena *vulnerability* sudah tidak tersedia kembali.

$$\frac{\text{Serangan Terblokir}}{\text{Total Serangan}}\ x\ 100\%$$

$$\frac{3}{3}\ x\ 100\% = 100\%$$

Perhitungan secara kuantitatif berdasarkan serangan yang dilakukan pada *website* Hakazone yang telah terimplementasi CDN & WAF *Cloudflare* untuk mengukur keefektifan dari CDN & WAF yang diimplementasikan, didapatkan hasil 100% dari seluruh serangan dapat diblock oleh *Web Application Firewall*, namun perlu diperhatikan perhitungan tersebut diluar dari kemungkinan adanya *zero-day* v*ulnerability* maupun *vulnerability* lainnya yang mungkin saja akan muncul pada suatu versi *software, plugin*, atau perangkat lunak lainnya. Maka dari itu perlu dilakukan *Vulnerability Assessment* secara rutin minimal 1 tahun sekali dan pengecekan *update* apabila terdapat versi *software* yang baru minimal 1 bulan sekali. Karena serangan juga semakin berkembang seiring berjalannya waktu maka perhitungan diatas hanya akan valid dalam jangka waktu tertentu yaitu maksimal 1 tahun, bahkan dapat lebih cepat apabila pada suatu waktu ditemukan *zero-day vulnerability*, oleh karena itu sangat penting mengikuti kabar maupun *update* tentang teknologi, terutama dalam hal *IT Security* agar selalu mendapat *update* terutama ketika terdapat *zero-day vulnerability* baru yang muncul.

Sebagai rekomendasi evaluasi agar mengantisipasi hal-hal yang tidak diinginkan terjadi pada web Hakazon yaitu, penting untuk menerapkan *secure code* diaplikasi serta secara rutin melakukan *Vulnerability Assessment* setidaknya setiap satu tahun sekali dan memeriksa pembaruan setiap bulan, jika terdapat versi perangkat lunak yang baru. Seiring dengan perkembangan serangan yang semakin canggih seiring berlalunya waktu, perhitungan di atas hanya berlaku dalam jangka waktu tertentu, yaitu paling lama satu tahun, dan bahkan bisa lebih cepat jika suatu saat ditemukan kerentanan zero-day. Oleh karena itu, sangat krusial untuk tetap mengikuti berita dan pembaruan teknologi, khususnya dalam bidang Keamanan Teknologi Informasi, agar selalu mendapatkan pembaruan, terutama saat muncul kerentanan zero-day yang baru.

5. Simpulan

Dari hasil penelitian yang dilakukan, dapat disimpulkan bahwa keamanan *Content Delivery Network* (CDN) *Cloudflare* efektif jika digunakan untuk memperkuat keamanan suatu *website.* Setiap metode serangan yang dilakukan penulis sudah dapat dilakukan tindakan preventif menggunakan CDN dan WAF yang ada di dalamnya. *Request* yang dianggap berbahaya akan dilakukan *block* oleh WAF, sehingga keamanan *website* menjadi lebih kuat melalui sisi server. Dengan menggunakan WAF dapat membuat *programmer* lebih fokus pada pengembangan sistem, tetapi keamanan memang perlu dilakukan secara berlapis yang akan memberikan efek lebih baik.

Daftar Referensi

[1] Internetlivestats, “Total number of Websites,” *www.internetlivestats.com*. https://www.internetlivestats.com/watch/websites/ (accessed Mar. 31, 2021).

[2] E. Nurhayati, R. S. Yudiantini, P. S. Informatika, U. Siliwangi, P. S. Informatika, and U. Siliwangi, “Sejarah Web Service Dalam Perkembangan Teknologi Informasi,” no. March, 2020.

[3] D. Stiawan, M. Y. Idris, A. H. Abdullah, M. AlQurashi, and R. Budiarto, “Penetration testing and mitigation of vulnerabilities windows server,” *Int. J. Netw. Secur.*, vol. 18, no. 3, pp. 501–513, 2016.

[4] A. M. Tania *et al.*, “Keamanan Website Menggunakan Vulnerability Assessment,” *Informatics Educ. Prof.*, vol. 2, no. 2, pp. 171–180, 2018.

[5] H. A. Tuara, N. Maridyah, and K. Khaerudin, “Implementasi CDN (Content Delivery Network) Menggunakan Cloudflare terintegrasi Dengan Docker Container,” *J. Mechatron. Electr. Eng.*, vol. 1, no. 1, pp. 42–51, 2021.

[6] S. P. Sitorus, “Analisis Kinerja Content Delivery Network Fakultas Ilmu Komputer Dan Teknologi Informasi,” 2017.

[7] E. Pantoulas, “Description, analysis and implementation of a Web Application Firewall (WAF). Creation of attack scenarios and threats prevention.”.

[8] V. Chubaievskyi, Y. Shestak, D. Tyshchenko, and R. Brzhanov, “Experimental Studies Of The Features Of Using Waf To Protect Internal Services In The Zero Trust Structure,” vol. 100, no. 3, 2022.

[9] P. M. R. Gumilang and D. W. Chandra, “Implementasi dan modifikasi WebShell untuk monitoring serangan berbasis website,” *Aiti*, vol. 18, no. 1, pp. 54–68, 2021, doi: 10.24246/aiti.v18i1.54-68.

[10] Agus Permana, “Indonesia’s Cyber Defense Strategy In Mitigating The Risk Of Cyber Warfare Threats,” vol. 3, no. 1, p. 6, 2021.

[11] R. Umar, I. Riadi, and E. Handoyo, “Analisis Keamanan Sistem Informasi Berdasarkan Framework COBIT 5 Menggunakan Capability Maturity Model Integration (CMMI),” *J. Sist. Inf. Bisnis*, vol. 9, no. 1, p. 47, 2019, doi: 10.21456/vol9iss1pp47-54.

[12] Bangkit Wiguna, W. Adi Prabowo, and R. Ananda, “Implementasi Web Application Firewall Dalam Mencegah Serangan SQL Injection Pada Website,” *Digit. Zo. J. Teknol. Inf. dan Komun.*, vol. 11, no. 2, pp. 245–256, 2020, doi: 10.31849/digitalzone.v11i2.4867.

[13] Dewi Laksmiati, “Implementasi Content Delivery Network (Cdn) Untuk Optimasi Kecepatan Akses Website,” *Akrab Juara*, vol. 5, pp. 49–56, 2020.

[14] I. Ramadhan, “Strategi Keamanan Cyber Security Di Kawasan Asia Tenggara,” *J. Asia Pacific Stud.*, vol. 3, no. 2, pp. 181–192, 2020, doi: 10.33541/japs.v3i1.1081.

[15] I Gede Putu Krisna Juliharta, “Distribusi Konten Web Server Menggunakan Motode

Content Delivery Network," *J. Sist. dan Inform.*, vol. 10, no. 1, pp. 159–169, 2015, [Online]. Available: https://jsi.stikom-bali.ac.id/index.php/jsi/article/view/16.