HARDENING CLOUDFRI DENGAN METODE SECURITY HARDENING PADA WEBSITE SAP.CLOUDFRI.ID

CLOUDFRI HARDENING WITH SECURITY HARDENING METHOD ON SAP.CLOUDFRI.ID WEBSITE

Muhammad Rifki Oktova[1], Umar Yunan kurnia Septo Herdianto[2], Muhammad Fathinuddin[3]

[1,2,3] Universitas Telkom, Bandung

[1]mrifkioktova@student.telkomuniversity.ac.id, [2]umaryunan@telkomuniversity.ac.id,

[3]muhammadfathinuddin@telkomuniversity.ac.id

Abstrak

Hardening merupakan upaya untuk mengurangi risiko keamanan dengan menghilangkan potensi serangan dan menguatkan permukaan serangan sistem. Penelitian ini bertujuan untuk melakukan identifikasi kerentanan yang ada pada *website* sap.cloudfri.id yang selanjutnya akan dilakukan prosedur hardening untuk meminimalkan ancaman pada *website*. Dengan menggunakan metode *security hardening* dilakukan juga pengujian *vulnerability scanning* dan *exploit*. Pengujian yang dilakukan untuk menemukan *vulnerability* yang terdapat pada *website* sap.cloudfri.id sebagai objek penelitian dan memanfaatkan hasil analisis dari *vulnerability scanning* untuk melakukan *exploit* pada objek. *Tools vulnerability scanning* yang digunakan meliputi Nmap, Nessus, Vega dan Nikto untuk menghasilkan hasil analisis untuk dapat dilakukannya exploitasi yang bertujuan untuk membenarkan bahwa *website* memiliki *vulnerability* yang dijabarkan oleh *scanning tools*. Untuk melakukan exploitasi pada penelitian ini dibutuhkan tools yaitu Metasploit *Framework*, SQLMap, dan BurpSuite. Hasil analisis dari hasil *vulnerability* dan hasil *exploit website* dapat menerima serangan berupa *Session Hijacking*, *Man-in-the-Middle* dan DoS *attack*.

Kata Kunci: Hardening, Vulnerability Scanning, Exploit

Abstract

Hardening is an attempt to reduce security risks by eliminating potential attacks and strengthening the system's attack surface. This study aims to identify vulnerabilities that exist on the website sap. Cloudfri. Id which will then be carried out with a hardening procedure to minimize threats to the website. By using the security hardening method, vulnerability scanning and exploit testing are also carried out. Tests are carried out to find vulnerabilities found on the sap.cloudfri.id website as research objects and take advantage of the analysis results from vulnerability scanning to exploit objects. The vulnerability scanning tools used include Nmap, Nessus, Vega and Nikto to generate analysis results for exploiting to be carried out which aims to confirm that the website has a vulnerability described by the scanning tools. To perform the exploitation in this research, tools are needed, namely Metasploit Framework, SQLMap, and Burp Suite. The results of the analysis of the results of the vulnerability and exploit results of the website can receive attacks in the form of Session Hijacking, Man-in-the-Middle and DoS attacks.

Keywords: Hardening, Vulnerability Scanning, Exploit

1. Pendahuluan

Penggunaan komputer meningkat dari hari ke hari, kompleksitas sistem juga semakin meningkat. Semua aktivitas ini meningkatkan kerntanan dalam sistem. Terutama dalam bidang jaringan menuntut meningkatnya suatu kebutuhan akan kualitas keamanan jaringan. Dengan mengurangi kerentanan dalam sebuah sistem dan jaringan sebanyak mungkin dapat meningkatkan keamanan sistem dan jaringan[1]. *CloudFRI* merupakan sebuah sistem yang berisikan

kumpulan aplikasi yang digunakan oleh keseluruhan entitas Fakultas Rekayasa Industri Universitas Telkom, yang dimana didalamnya terdapat beberapa *web applications* seperti administrasi.cloudfri.id, ingram.cloudfri.id, labrecruitment.cloudfri.id dst. Melakukan *hardening* pada cloudFRI diperlukan karena untuk menghindari dan meminimalkan ancaman dengan cara instalasi *firewall*, *antivirus*, menghapus *cookie*, membuat *password* dan menghapus program yang tidak diperlukan[2]. Tujuan dari melakukan *hardening* pada *cloudFRI* adalah untuk melakukan pengamanan data, jika terjadi sebuah serangan pada sistem yang ada di *cloudFRI*, maka dapat berpengaruh terhadap kestabilan dan kinerja dari beberapa aplikasi yang terdapat didalam *cloudFRI*. *Hardening* juga untuk menghilangkan resiko dari ancaman yang bisa terjadi pada komputer, hal ini biasanya dilakukan dengan menghapus semua program/*file* maupun port-port (TCP/UDP) yang tidak diperlukan[1]. Permaslahan pada penelitian ini adalah bagaimana mengamankan sebuah *cloudFRI* dengan melakukan prosedur *hardening*, guna menghindari kerugian-kerugian yang tidak diinginkan oleh pihak pengelola dan *client*. Melakukan *security hardening* pada *cloudFRI* juga dapat membuat aplikasi yang tersedia pada *cloudFRI* lebih terjaga karena sudah dilakukannya penutupan celah keamanan dan konfigurasi ulang untuk mencegah terjadinya pencurian data atau serangan yang lainnya.

2. Landasan Teori

2.1 *Security Hardening*

Hardening adalah Prosedur yang meminimalkan ancaman yang datang dengan mengatur konfigurasi dan menonaktifkan aplikasi dan layanan yang tidak diperlukan. Instalasi *firewall*, instalasi antivirus, menghapus *cookie*, membuat password, menghapus program yang tidak diperlukan[1].

Dalam *Security Hardening* terdapat empat tahapan yaitu[3]:

1. *Access*
   Mengidentifikasi celah-celah keamanan yang masih terdapat dalam sistem dan mengeksploitasi sistem untuk mengetahui lubang keamanan pada sistem tersebut.
2. *Analyze*
   Memperkirakan tingkat keamanan dan menganalisis dampak yang terjadi pada celah keamanan tersebut, kemudian mengklasifikasikan tingkat kerusakan yang diakibatkan oleh celah keamanan tersebut.
3. *Remediate*
   Menemukan celah keamanan pada sistem yang diuji dan mencari cara untuk menutup celah keamanan tersebut untuk mengamankan sistem.
4. *Manage*

   Mengintegrasikan patch pada celah keamanan untuk menutup lubang keamanan dan mencegah serangan masuk, serta mencegah terbukanya celah keamanan lain.

2.2 Keamanan Data

Keamanan data telah menjadi bagian dari pengembangan teknologi informasi mengingat bahwa berjuta-juta bit informasi telah dipertukarkan dalam jaringan komputer terutama di internet. Masalah keamanan data dapat diklasifikasi ke dalam beberapa dimensi [12].

2.3 *Exploit*

Exploit adalah sebuah kode yang menyerang keamanan computer secara spesifik. *Exploit* banyak digunakan untuk melakukan penetrasi baik secara legal ataupun illegal untuk mencari kelemahan (*vulnerability*) pada computer tujuan [13]. Penetrasi yang dilakukan tidak hanya berlaku pada computer tetapi dapat juga menyerang sebuah perangkat lunak. Tetapi *exploit* tidak hanya digunakan untuk penyerangan, banyak peneliti yang menggunakan *exploit* untuk melakukan demonstrasi bahwa sesuatu sistem memiliki celah keamanan.

2.4 *Vulnerability Scanninng*

Vulnerability adalah kelemahan yang dapat menyebabkan mesin dapat dihentikan, dimofikasi, atau diambil alih oleh pihak ketiga. Vulnerability scanning adalah teknik yang menggunakan computer untuk mencari kelemahan dalam

keamanan komputer atau sistem lainnya yang memungkinkan penyerang mengurangi informasi sistem. Proses vulnerability scanning adalah langkah-langkah untuk mencari, menemukan dan mengeksploitasi kelemahan aplikasi web dengan menggunakan teknik atau alat tertentu [14].

Vulnerability scanning juga dapat diartikan proses dimana melakukan deteksi terhadap suatu kelemahan yang mengancam nilai integrity, confidentiality, dan availability dari suatu asset. Proses ini juga melakukan sebuah identifikasi ancaman dan risiko yang ditimbulkan biasanya melibatkan penggunaan tools pengujian yang menjalankan proses ini secara otomatis. Beberapa jenis vulnerability scanning yaitu[15]:

1. *Network-based scanning*
   Digunakan untuk mengidentifikasi kemungkinan serangan keamanan jaringan. Jenis scan ini juga dapat mendeteksi system yang rentan pada jaringan kabel atau nirkabel
2. *Host-based scanning*
   Digunakan untuk menemukan dan mengidentifikasi kerentanan di *server*, *workstation* atau *host* jaringan lainnya. Jenis scan ini biasanya memeriksa *port* dan layanan yang mungkin juga terlihat oleh *Network-based scanning*, tetapi ini menawarkan visibilitas yang lebih besar ke pengaturan konfigurasi dan menutup sejarah system yang di *scan*.
3. *App scan*
   Digunakan untuk menguji *website* untuk mendeteksi kerentanan perangkat lunak yang diketahui dan konfigurasi yang salah dalam aplikasi jariangn atau *web*.

2.5 *Virtual Hosting*

Virtual Hosting merupakan kumpulan dari beberapa server yang menyediakan serviceservice guna memberikan kemudahan bagi pengguna untuk membuat *website*. Dengan kata lain *Virtual Hosting* (Web Hosting) adalah penyedia layanan untuk menampung data-data yang diperlukan oleh sebuah *website* dan sehingga dapat diakses lewat internet. Data disini dapat berupa file, gambar, email, aplikasi/program/script dan database. *Virtual Hosting* menyediakan hardware, jaringan (infrastruktur), email (telepon), dan sebagainya agar anda dapat membuka/membuat *website*. Server (gedung Mall) dihuni oleh banyak pelanggan, masing-masing pelanggan mempunyai batas penggunaan diskspace (batasan ruangan) dan tentu saja setiap pelanggan mengoperasikan *website*nya masingmasing[4].

2.6 OWASP

Open Web Application Security Project (OWASP) adalah sebuah organisasi internasional yang bersifat non-profit, didirikan oleh OWASP *foundation* pada 21 April 2004 di Ameriaka Serikat. Keanggotaan OWASP berasal dari para ilmuwan, peneliti, dan sektor swasta yang menerbitkan laporan artikel, Penerapan *Framework* OWASP dan *Network Forensics* untuk Analisis, Deteksi, dan Pencegahan Serangan Injeksi di Sisi *Host-Based* 10 alat/peralatan, dan dokumen yang bersifat *open source*. OWASP merupakan vendor netral yang tidak berafiliasi dengan perusahaan teknologi manapun, tidak mendukung atau merekomendasikan produk atau layanan komersial. Proyek yang sudah dibuat dan dipublikasikan ada 363 proyek dan semua berkaitan dengan keamaan aplikasi, diantara proyek tersebut yaitu OWASP *Top Ten Project*, OWASP ASVS *Assessment tool*, OWASP Zed *Attack Proxy Project*, OWASP *Testing Guide*[14].

2.7 *Cloud Computing*

Cloud computing merupakan sebuah model untuk memberikan kenyamanan, pada akses jaringan permintaan beberapa pengguna untuk berbagi sumber daya komputasi yang dikonfigurasi (seperti, jaringan server, penyimpanan, aplikasi, dan jasa) yang dapat dengan cepat ditetapkan dan dirilis dengan usaha pengelolaan yang minimal atau interaksi penyedia layanan[5].

Terdapat beberapa model layanan yang terdapat pada cloud computing yaitu[6]:

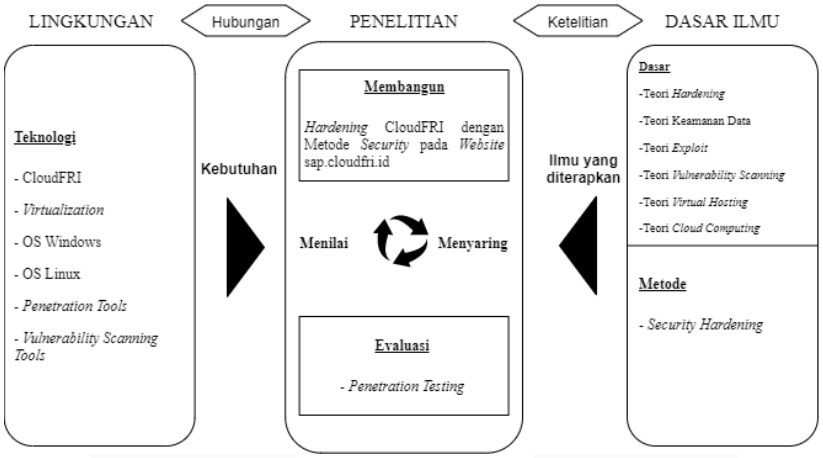
1. *Software as a Service* (SaaS)
   Kemampuan yang diberikan kepada konsumen untuk menggunakan aplikasi penyedia dapat beroperasi pada infrastriktur *cloud*. Aplikasi dapat diakses dari berbagai perangkat klien melalui antarmuka seperti *web browser* (misalnya, email berbasis web)..
2. *Platform as a Service* (PaaS)
   Kemampuan yang diberikan kepada konsumen untuk menyebarkan aplikasi yang dibuat konsumen atau diperoleh ke infrastruktur *cloud computing* menggunakan bahasa pemrograman dan peralatan yang didukung oleh *provider*.
3. *Infrastructure as a Service* (IaaS)

Kemampuan yang diberikan kepada konsumen untuk memproses, menyimpan, berjaringan, dan sumber komputasi penting yang lain, dimana konsumen dapat menyebarkan dan menjalankan perangkat lunak secara bebas, yang dapat mencakup sistem operasian aplikasi.

3. Metodologi Penelitian

3.1 Model Konseptual

Model konseptual adalah gambaran untuk memahami, melaksanakan dan mengevaluasi penelitian sistem infomasi. Model ini berisikan tujuan, tugas, masalah sekaligus oportunitas. Tujuan utama dari model ini adalah untuk mewujudkan sebuah kerangka terstruktur yang digunakan untuk memahami tujuan dari sebuah penelitian itu sendiri. Model konseptual yang digunakan dalam penelitiaan ini dibuat berdasarkan sumber dari metode Alan R. Hevner adalah sebagai berikut:



Gambar 3. 1 Model Konseptual

3.2 Sistematika Penelitian

Proses hardening pada cloudfri ini menggunakan metode security hardening dimana didalam security hardening terdapat tiga tahapan yang sudah menjadi prosedur dari melakukan hardening. Pada penelitian ini tahapan security hardening berhenti sampai tahap Remediate dimana pada tahapan itu diberikan rekomendasi untuk melakukan penutupan celah yang terdapat pada website sap.cloudfri.id

1. Tahap Awal
   Pada tahap awal dimulai dengan melakukan identifikasi masalah lalu dilanjutkan dengan latar belakang masalah yang bertujuan untuk menggambarkan masalah yang hendak diselesaikan, dengan itu solusi yang mengacu pada studi literatur akan diperoleh. Penulis melakukan tahap *hardening* yang pertama yaitu tahap *Access*. Dimana proses ini bertujuan untuk mencari celah-celah keamanan yang masih terdapat dalam sistem dan menganalisis dampak yang dapat terjadi pada celah keamanan yang ada.
2. Tahap *Hardening* (Tahap *Acces*s)
   Pada tahap *hardening* yang pertama yaitu tahap *Access*. Dimana proses ini bertujuan untuk mencari celah-celah keamanan yang masih terdapat dalam sistem dan menganalisis dampak yang dapat terjadi pada celah keamanan yang ada.
3. Tahap *Hardening* (Tahap Analyze)

   Pada tahap hardening yang kedua yaitu tahap *Analyze*. Dimana pada proses ini dilakukan perkiraan tingkat keamanan dan menganalisis dampak yang terjadi pada celah keamanan yang ada, kemudian melakukan eksploitasi untuk mengetahui serangan berupa apa yang dapat diterima oleh *website* dan melakukan analisis dari percobaan exploit yang sudah dilakukan sebelumnya.

4. Tahap *Hardening* (Tahap *Remediate*)

   Pada tahap selanjutnya dalam hardening yaitu tahap *Remediate*. Proses ini dilakukan pencarian cara untuk menutup celah yang sudah didapat dari tahap *Analyze*. Pada tahap ini juga diberikan rekomendasi tentang hal apa yang dapat dilakukan untuk menutup celah yang kurang optimal untuk keamanan *website.*

5. Tahap Akhir

   Tahap akhir ini adalah tahapan terakhir dari penelitian yang dilakukan. Pada tahap ini, menghasilkan laporan hasil penelitian. Laporan hasil penelitian didukung dengan dokumentasi dari penelitian yang dilakukan serta memberikan kesimpulan dan saran.

4. Pengujian dan Analisis

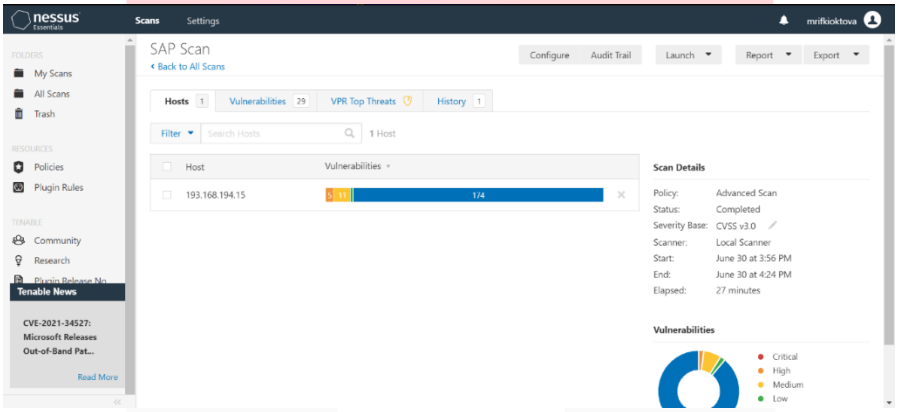
4.1 Pengujian

4.1.1 Pengujian *vulnerability scan* dari Nmap

Scan yang dilakukan Nmap menggunakan versi GUI (Graphical User Interface) pada sistem operasi windows 10 melalui IP yang di *scan* 193.168.194.15 menunjukan *port*, *state*, *service* dan *version* dari IP yang di *scan*.

4.1.2 Pengujian *vulnerability scan* dari Nessus

Scan jaringan menggunakan Nessus dapat menampilkan *vulnerability* yang terdapat pada IP target yang di *scan*.

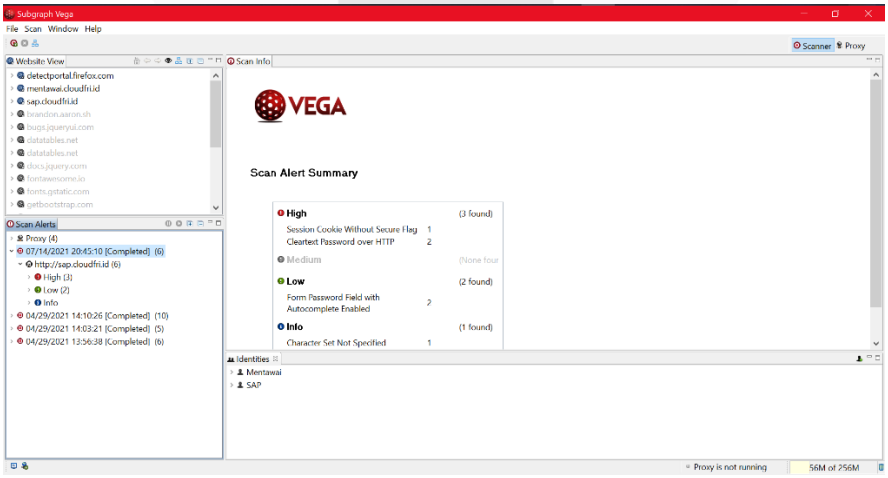


Gambar 4. 1 *Vulnerability scan* dengan Nessus

Dari hasil *scan* IP 193.168.194.15 yang dilakukan Nessus didapatkan *vulnerability*, pada *icon* yang ditandai warna terdapat keterangan dari setiap informasi.

4.1.3 Pengujian *vulnerability scan* dari Vega

Scan jaringan menggunakan Vega dapat menampilkan *vulnerability* yang terdapat pada IP target yang di *scan*, Vega juga dapat berfungsi sebagai *proxy*.



Gambar 4. 2 *Vulnerability scan* dengan Vega

Dari hasil *scan* sap.cloudfri.id yang dilakukan Vega didapatkan *vulnerability*, pada icon yang ditandai warna terdapat keterangan dari setiap informasi, tanda yang berwarna orange menunjukan *High*, tanda yang berwarna hijau menunjukan *Low*, dan tanda berwarna biru menunjukan Info. Selain mendapatkan informasi yang sudah bertanda dengan warna, Vega juga memberikan informasi deskripsi data yang berupa tabel untuk mempermudah dalam menganalisis hasil dari pengolahan data.

4.1.4 Pengujian *vulnerability scan* dari Nikto

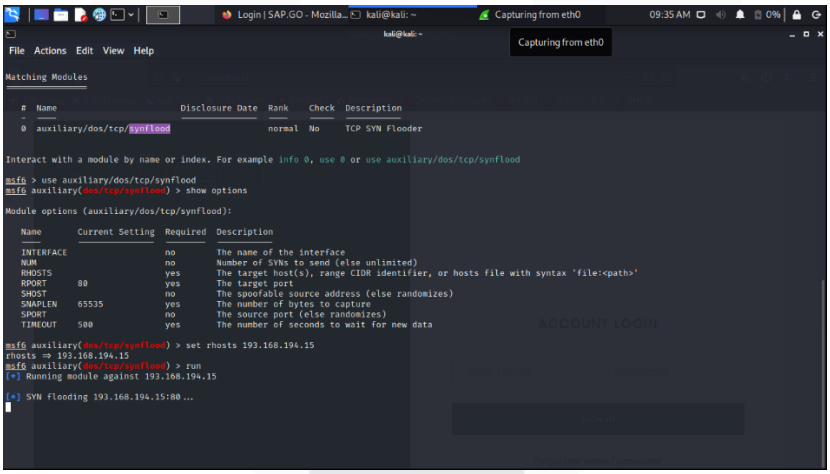
Scan *vulnerability* menggunakan Nikto dapat menampilkan celah keamanan yang terdapat pada URL atau IP target yang di *scan*.

Gambar 4. 3 *Vulnerability scan* dengan Nikto

Perintah yang diketik untuk menguji kerentanan dalam *website* adalah “nikto –h sap.cloudfri.id –p 80 –o nikto_result –f txt”. Perintah ini dilakukan untuk melakukan *scan* kepada *website* sap.cloudfri.id melalui *port* 80, setelah melakukan *scan* maka hasil *scan* akan tersimpan otomatis dengan nama file nikto_result.txt.

4.1.5 Pengujian *exploit* menggunakan Metasploit

pengujian *exploit* yang sudah dirancang sebelumnya, pada pengujian *scan*ning *vulnerability* didapatkan informasi celah keamanan yang terdapat pada *website* sap.cloudfri.id.

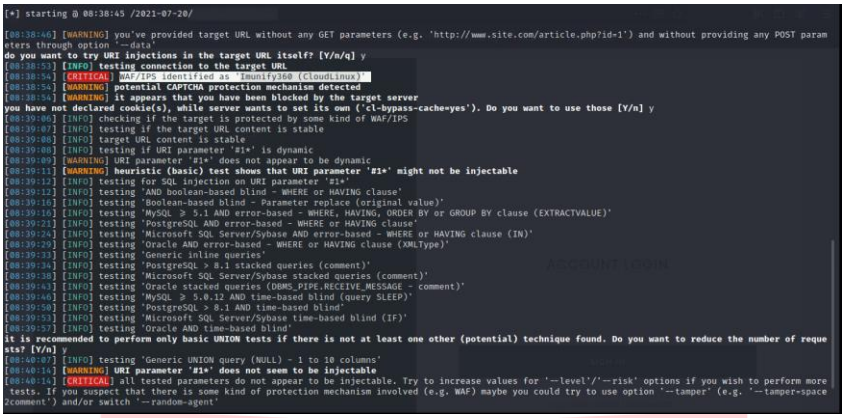


Gambar 4. 4 *Exploit* menggunakan Metasploit

Pada prosess pengujian *DoS attack* disini Metasploit *framework* sudah menyediakan modul *auxiliary* untuk penyerangan tersebut. Sebelum menjalankan program tersebut dibutuhkan adanya konfigurasi “set RHOSTS 193.168.194.15” terhadap program tersebut untuk memasukan alamat IP dari target dan juga memasukan melalui *port* mana penyerangan ini akan dilakukan.

4.1.6 Pengujian *exploit* menggunakan SQLMap

Pada exploit ini digunakan *tools* SQLMap untuk melakukan percobaan penyerangan SQL *Injection*

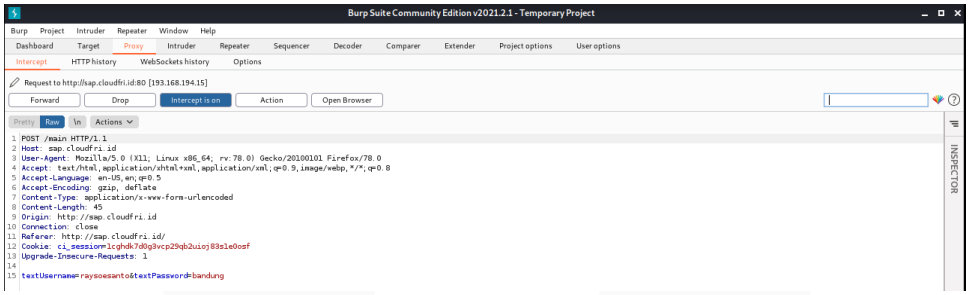


Gambar 4. 5 *Exploit* menggunakan SQLMap

SQLMap menjalankan programnya dan proses ini terhamabat karena *website* target terlindungi oleh WAF.

4.1.7 Pengujian *exploit* menggunakan BurpSuite

Exploitasi ini digunakan *tools* yaitu BurpSuite untuk melakukan percobaan *intercept* komunikasi *website* kepada *server*. Serangan *Intercept* merupakan serangan yang sulit dilihat dan merupakan kategori serangan pasif.



Gambar 4. 6 *Exploit* menggunakan BurpSuite

Pada kasus ini *intercept* yang dilakukan oleh burpsuite adalah melakukan *intercept* pada saat request *login* kedalam *website* dan dengan ini maka pada kasus ini *session cookie* yang diberikan oleh *website* dapat dipergunakan kembali untuk melakukan upaya *login* yang tidak sah.

4.2 Hasil Analisis

4.2.1 Hasil Analisis *vulnerability scan* Nmap

Terdapat beberapa port yang terbuka pada hasil scan yang dilakukan Nmap yaitu port 21, port 25, port 80, port 110, port 143, port 443, port 465, port 587, port 993, port 995, dan port 3306.

- *Port* 21

Port ini merupakan *port* yang digunakan untuk mengkoneksikan kedalam FTP *server*. FTP (*File Transsmission Protocol*) sederhananya berfungsi untuk dapat saling menghubungkan komputer satu dengan lainnya

- *Port* 25

Port ini merupakan *port* yang digunakan untuk *Simple Mail Transfer Protocol* (SMTP).SMTP digunakan untuk mengrimkan data dari komputer pengirim email ke *server* penerima email.

- *Port* 80

Port ini merupakan *port* yang digunakan untuk web *server*, paling umum digunakan untuk mengakses internet atau bisa disebut HTTP *port server*

- *Port* 110

Port ini merupakan *port* yang digunakan untuk menjalankan protokol Post Office version 3 (POP3). POP3 merupakan protokol email klien yang paling sering digunakan untuk mengumpulkan email dari *server*.

- *Port* 143

Port ini merupakan *port* yang digunakan untuk menjalankan Internet Message Access Protocol (IMAP). Protokol IMAP adalah protokol biasa untuk mengakses email ke dari *server*.

- *Port* 443

Port ini merupakan *port* yang digunakan untuk *Hypertext Transfer Protocol Securre* (HTTPS) yang bisa disebut juga protokol valid dan aman.

- *Port* 465

Port ini merupakan *port* yang digunakan untuk *Secure Socket Layer* (SSL). SSL adalah cara dari *website* untuk membuat sambungan aman dengan web *browser* pengguna.

- *Port* 587

Port ini merupakan *port* yang digunakan untuk *Mail Submission Agent* (MSA). MSA adalah *software* yang menerima email dari *Mail User Agent* (MUA) dan bekerja sama dengan *Mail Transfer Agent* (MTA) untuk mengirimkan sebuah email.

- *Port* 993

Port ini merupakan *port* yang digunakan untuk *Secure Internet Mail Access Protocol* (SIMAP). SIMAP adalah protokol yang berfungsi untuk pengamanan dalam mengakses email dari *server* dengan komunikasi yang lebih aman.

- *Port* 995

Port ini merupakan *port* yang digunakan POP3 untuk menjalankan SSL/TLS. Sederhananya adalah *port* ini menjalankan layanan POP3 yang terenkripsi, dan digunakan oleh *server* email.

- *Port* 3306

Port ini merupakan *port* yang digunakan untuk *port default* dari protokol MySQL. *Port* ini digunakan untuk terhubung dengan klien MySQL dan utilitas seperti mysqldump.

4.2.2 Hasil Analisis *vulnerability* Nessus

Hasil analisis *vulnerability* yang didapatkan dari Nessus dapat diketahui deskripsi dari *vulnerability* yang ditemukan oleh Nessus dan kategori dari ancaman tersebut.

1. *SSL Medium Strength Cipher Suites Supported* (SWEET32)

Host jarak jauh mendukung penggunaan cipher SSL yang menawarkan enkripsi kekuatan sedang. Nessus menganggap kekuatan sedang sebagai enkripsi apa pun yang menggunakan panjang kunci setidaknya 64 bit dan kurang dari 112 bit, atau yang menggunakan paket enkripsi 3DES. Perhatikan bahwa jauh lebih mudah untuk menghindari enkripsi kekuatan menengah jika penyerang berada di jaringan fisik yang sama. Karena DES dan triple-DES hanya memiliki ukuran blok 64-bit, maka dapat berpotensi mendapati *Birthday attack* menjadi perhatian nyata. Dengan kemampuan untuk menjalankan Javascript di *browser*, dimungkinkan untuk mengirim lalu lintas yang cukup untuk menyebabkan tabrakan, dan kemudian menggunakan informasi itu untuk memulihkan sesuatu seperti *Session Cookie*.

Karena DES dan triple-DES hanya memiliki ukuran blok 64-bit, maka dapat berpotensi mendapati *Birthday attack* menjadi perhatian nyata. *Birthday attack* adalah salah satu jenis serangan kriptografi yang diberinama berdasarkan teori yang melatarbelakangi serangan ini yaitu teori *birthday paradox*. Serangan ini digunakan kriptanalis untuk menyerang fungsi *hash* atau kode dari hasil enkripsi[8].

2. TLS *Version 1.0 Protocol Detection*

Layanan jarak jauh menerima koneksi yang dienkripsi menggunakan TLS 1.0. TLS 1.0 memiliki sejumlah kekurangan desain kriptografi. Implementasi modern TLS 1.0 mengurangi masalah ini, tetapi versi TLS yang lebih baru seperti 1.2 dan 1.3 dirancang untuk mengatasi kekurangan ini dan harus digunakan jika memungkinkan. Mulai tanggal 31

Maret 2020, *Endpoint* yang tidak diaktifkan untuk TLS 1.2 dan yang lebih tinggi tidak akan berfungsi lagi dengan baik dengan *website* utama dan vendor besar.

3. HSTS *Missing From* HTTPS *Server* (RFC 6797)

Server web jarak jauh tidak memberlakukan HSTS, seperti yang didefinisikan oleh RFC 6797. HSTS adalah *header* respons opsional yang dapat dikonfigurasi di *server* untuk menginstruksikan *browser* agar hanya berkomunikasi melalui HTTPS. Kurangnya HSTS memungkinkan serangan *downgrade*, serangan *man-in-the-middle* SSL-*stripping*, dan melemahkan perlindungan *cookie-hijacking*.

4. SSL Anonymous Cipher Suites *Supported*

Host jarak jauh mendukung penggunaan cipher SSL anonim. Sebuah 'cipher suite' mendefinisikan algoritma kriptografi yang digunakan selama koneksi SSL/TLS, dan biasanya berisi informasi tentang primitif untuk pertukaran kunci, otentikasi, enkripsi simetris, dan perlindungan integritas. Contoh suite, diberikan dalam format digunakan oleh OpenSSL, akan menjadi DHE-RSA-AES128-SHA[7]. Meskipun hal ini memungkinkan administrator untuk menyiapkan layanan yang mengenkripsi lalu lintas tanpa harus membuat dan mengkonfigurasi sertifikat SSL, ini tidak menawarkan cara untuk memverifikasi identitas host jarak jauh dan menjadikan layanan rentan terhadap serangan *man-in-the-middle*.

5. POP3 Cleartext Logins Permitted

Host jarak jauh menjalankan daemon POP3 yang memungkinkan *login cleartext* melalui koneksi yang tidak terenkripsi. Penyerang dapat menemukan nama pengguna dan sandi dengan melakukan *sniffing* lalu lintas ke *daemon* POP3 jika mekanisme otentikasi yang kurang aman (misalnya, perintah USER, AUTH PLAIN, AUTH LOGIN) digunakan. jika penyerang dapat menemukan nama pengguna dan kata sandi maka penyerang akan sangat mudah untuk melancarkan serangan yang lebih aktif atau serangan jenis lainnya dengan informasi yang sudah didapat.

4.2.3 Hasil Analisis *vulnerability* menggunakan vega

Hasil analisis *vulnerability* yang didapatkan dari Vega dapat diketahui deskripsi dari *vulnerability* yang ditemukan oleh Vega dan kategori dari ancaman tersebut.

1. *Cleartext Password over* HTTP

Vega mendeteksi formulir dengan bidang masukan kata sandi yang dikirimkan ke target tidak aman (HTTP). Nilai kata sandi tidak boleh dikirim dengan jelas melalui saluran yang tidak aman. Kerentanan ini dapat mengakibatkan pengungkapan kata sandi yang tidak sah kepada penyerang jaringan pasif. Hal tersebut akan berdampak pengiriman kata sandi melalui saluran yang tidak aman dan Hal ini dapat mengakibatkan pengungkapan kata sandi kepada penyadap jaringan *Website* ini mengirimkan kata sandi melalui koneksi yang tidak terenkripsi, membuatnya rentan terhadap intersepsi.

2. Session Cookie Without Secure Flag

Vega telah mendeteksi bahwa *session cookie* yang dikenal mungkin telah disetel tanpa tanda aman. Dengan demikian, akan ada risiko bahwa penyerang akan mencegat komunikasi *clear-text* antara *browser* dan *server* dan akan mencuri *cookie* dari pengguna. Hal tersebut akan berdampak *Cookie* dapat diekspos ke penyadap jaringan dan *Session cookies* adalah kredensial otentikasi, penyerang yang mendapatkannya bisa mendapatkan akses tidak sah ke *website* yang terpengaruh. *Session Cookie* adalah waktu sebuah sesi koneksi yang diperkenankan *server* kepada klien untuk mengakses suatu halaman *website* [9].

3. Form Password Field with Autocomplete Enabled

Vega mendeteksi formulir yang menyertakan bidang masukan kata sandi. Atribut pelengkapan otomatis tidak disetel ke nonaktif. Hal ini dapat mengakibatkan beberapa *browser* menyimpan nilai yang dimasukkan oleh pengguna secara lokal, yang dapat diambil oleh pihak ketiga. Hal ini dapat berdampak Nilai kata sandi dapat disimpan di sistem *file* lokal klien dan Kata sandi yang disimpan secara lokal pada *browser* dapat diambil oleh pengguna lain karena tidak semua *browser* mengenkripsi yang memungkin kan tersimpan di komputer korban dalam bentuk teks yang jelas.

4. Character Set Not Specified

Vega telah mendeteksi bahwa web belum menentukan set karakter dalam respons. Jika kumpulan karakter tidak ditentukan, *browser* dapat membuat asumsi tentang kumpulan karakter berdasarkan konten web. Ini mungkin menimbulkan masalah keamanan jika web yang terpengaruh berisi konten yang dibuat secara dinamis yang berasal

dari pengguna. Dalam kasus seperti itu, pengguna jahat berpotensi memanfaatkan cara *browser* tertentu menafsirkan karakter yang menyebabkan konten berbahaya dirender.

4.2.4 Hasil Analisis *vulnerability* menggunakan Nikto

Hasil analisis *vulnerability* yang didapatkan dari Nikto dapat diketahui deskripsi dari *vulnerability* yang ditemukan oleh Nikto dan kategori dari ancaman tersebut.

1. *The anti-clickjacking X-Frame-Options header is not present*

Nikto mendeteksi header *X-Frame-Options* yang hilang yang berarti bahwa situs web ini berisiko terkena serangan *clickjacking*. *Clickjacking* (*User Interface redress attack*, UI *redress attack*, UI *redressing*) adalah Serangan yang membangun halaman web berbahaya untuk mengelabui pengguna agar melakukan klik yang tidak diinginkan yang menguntungkan penyerangnya menyebarkan *worm*, mencuri kata sandi informasi rahasia, *cookie*, mengirim *spam*, menghapus surat pribadi, dll.[10].

2. *The* X-XSS-*Protection header is not defined*

Nikto mendeteksi header X-XSS-*Protection* yang hilang yang berarti bahwa situs web ini berisiko terkena serangan *Cross-site Scripting* (XSS). Serangan *Cross-Site Scripting* dapat terjadi ketika data memasuki aplikasi Web melalui sumber yang tidak tepercaya, paling sering melalui permintaan web. Data disertakan dalam konten dinamis yang dikirim ke pengguna web tanpa divalidasi untuk kode berbahaya. Konten berbahaya yang dikirim ke *browser* web sering kali berbentuk segmen *JavaScript*, tetapi juga dapat mencakup HTML, *Flash*, atau jenis kode lain yang dapat dijalankan oleh *browser*.

3. *The X-Content-Type-Options header is not set*

Header respons HTTP 'X-*Content-Type-Options*' mencegah *browser* dari *'MIME-sniffing*' respons dari tipe konten yang dideklarasikan. MIME sniffing adalah fungsi standar di *browser* untuk menemukan cara yang tepat untuk merender data di mana header HTTP yang dikirim oleh *server* tidak meyakinkan atau hilang.

Masalah akan muncul setelah sebuah situs web memungkinkan pengguna untuk mengunggah konten yang kemudian dipublikasikan di *server* web. Jika penyerang dapat melakukan serangan XSS (*Cross-site Scripting*) dengan memanipulasi konten sedemikian rupa agar dapat diterima oleh aplikasi web dan dirender sebagai HTML oleh *browser*, dimungkinkan untuk menyuntikkan kode dalam bentuk file gambar dan membuat korban mengeksekusinya dengan melihat gambar.

4. *Server banner has changed from 'LiteSpeed' to 'Imunify360-webshield/1.16'*

Kalimat tersebut menjelaskan bahwa banner dari Web *Server* yang semula LiteSpeed yang digunakan *website* berubah menjadi Imunify360 yang mungkin mengacu kepada WAF (*Web Aplication Firewall*). Dengan demikian telah diketahui bahwa *website* sap.cloudfri.id yang menggunakan jasa hosting dari niagahoster itu sudah dilindungin WAF untuk melakukan perlindungan dari akses yang tidak sah di internet. Konsep dibalik *Web Application Firewall* sangat mirip dengan bagaimana *firewall* bekerja. *Firewall* bekerja berdasarkan suatu set aturan yang dikonfigurasi pada *firewall* atau yang disebut dengan *rule* [11]. Aturan ini yang mengamankan suatu web dengan cara, memonitor, dan memblokir lalu lintas HTTP dan dari *website*.

4.2.5 Hasil analisis *exploit* menggunakan Metasploit

Pada prosess pengujian *DoS attack* disini Metasploit *framework* sudah menyediakan modul *auxiliary* untuk penyerangan tersebut. Sebelum menjalankan program tersebut dibutuhkan adanya konfigurasi "set RHOSTS 193.168.194.15" terhadap program tersebut untuk memasukan alamat IP dari target dan juga memasukan melalui *port* mana penyerangan ini akan dilakukan. Selain itu diperlukan juga adanya aplikasi wireshark untuk merekam semua paket yang melewati *interface* yang dipilih.

Pengujian ini menghasilkan jumlah paket per detiknya mencapai 1000-1250 yang dapat menyebabkan membuat hampir mustahil *website* ini diakses oleh penggunanya karena *traffic* yang berjalan pada *website* tersebut sangat sibuk. *Website* dapat sangat sibuk karena Metasploit sudah membanjiri *traffic* jaringan dengan banyak data sehingga pengguna lain tidak bisa dilayani selama serangan tersebut berlangsung. Serangan ini melakukan exploitasi TCP dengan cara mengrimkan paket SYN dengan *spoof* alamat IP dalam jumlah yang sangat besar. Setiap koneksi yang

masuk akan ditanggapi oleh *server* yang menunggu proses koneksi berjalan, namun tidak pernah terjadi. Hal ini mengakibatkan proses yang terus berjalan pada *server* menjadi melebihi kapasitas yang bisa ditangani *server*.

4.2.6 Hasil analisis *exploit* menggunakan SQLMap

Hasil pengujian melakukan penyerangan SQL *Injection* menggunakan SQLMap terhadap *website* sap.cloudfri.id dengan command "sqlmap –threads 10 –url http://sap.cloudfri.id/data/payment~" Pengujian ini menghasilkan bahwa SQLMap menjalankan programnya dan proses ini terhamabat karena *website* target terlindungi oleh WAF. Terdapat pernyataan bahwa "WAF/IPS identified as Imunify360 (CloudLinux)" yang berarti Imunify360 adalah sebuah WAF yang digunakan oleh *server* untuk melakukan pertahanan dari penyerangan seperti ini. Dengan demikian maka proses untuk melakukan penyerangan akan gagal karena *server* akan segera melakukan blok terhadap proses yang sedang berjalan. Oleh sebab itu *website* sap.cloudfri.id tidak bisa menerima serangan SQL *Injection* karena Imunify360 telah mendeteksi bahwa ada permintaan yang tidak wajar yang diterima oleh *server*.

aplikasi SQLMap yang sudah cukup handal dalam membobol keamanan dengan menggunakan beberapa teknik yang dimilikinya masih dapat dicegah oleh WAF yang dimiliki oleh *server* penyedia layanan *website*. Hal ini sudah baik karena tidak perlu adanya lagi tindakan untuk mencegah atas penyerangan SQL *Injection*, sebab WAF disini sudah dapat mempertahankan serangan berupa SQL *Injection* untuk menghindari pencurian data oleh pihak yang tidak bertanggung jawab.

4.2.7 Hasil analisis *exploit* menggunakan BurpSuite

Pada kasus *intercept* yang dilakukan oleh burpsuite adalah melakukan *intercept* pada saat request *login* kedalam *website*. Penyerangan dilakukan untuk mengetahui apakah terdapat komunikasi yang tidak terenkripsi dengan baik sehingga dapat menimbulkan masalah karena dapat terjadinya kebocoran data yang dikarenakan komunikasi yang tidak terjaga keamanannya. Pengujian ini mendapati bahwa pada saat *website* sap.cloudfri.id melakukan request *login*, terdapat komunikasi atribut untuk *login* yaitu *username* dan *password* yang tidak terenkripsi. Dengan demikian maka penyerang dapat dengan mudah mekeksplor *website* karena sudah mendapatkan informasi berupa username dan password dari pengguna. Hal ini dapat terjadi karena *website* tidak menggunakan enkripsi pada tingkat *transport* seperti SSL/TLS untuk melindungi semua komunikasi sensitif yang lewat diantara klien dengan *server*.

Dengan memanfaatkan informasi yang didapat dari pengujian sebelumnya yaitu informasi yang diperoleh dari proses *intercept*. Pengujian penyerangan ini dilakukan untuk mengetahui bahwa *session cookie* yang didapat dapat dipergukanan kembali lebih dari satu kali. Karena berdasarkan hasil pengujian *scan* yang sudah dilakukan menyatakan bahwa *session cookie* yang disetel tidak menggunakan tanda aman. Tidak adanya tanda aman pada *session cookie* dapat berarti *cookie* yang diberikan sebelumnya dapat dipergunakan kembali oleh pengguna lain, yang mendapatkannya bisa mendapatkan akses tidak sah masuk kedalam *website*.

Pengujian ini mendapati bahwa percobaan *login* dengan tidak sah berhasil memasuki *website* sap.cloudfri.id dengan menggunakan *session cookie* sebelumnya pada upaya *login* dengan menggunakan akun admin yang diperoleh dari serangan *intercept*. Dengan ini maka pada kasus ini *session cookie* yang diberikan oleh *website* dapat dipergunakan kembali untuk melakukan upaya *login* yang tidak sah. Percobaan penyerangan ini biasa disebut *session hijacking*. *Session hijacking* adalah sebuah kerentanan yang dimanfaatkan oleh penyerang disebabkan oleh *cookie*s saat user *login* tidak dikonfigurasi ke *secure* atau konfigurasi *HttpOnly*.

5. Kesimpulan

1. Dalam melakukan indentifikasi kerentanan yang terdapat pada *website* sap.cloudfri.id menghasilkan beberapa celah kerentanan yang memungkinan untuk menjadi ancaman bagi *website*. Terdapat beberapa kerentanan seperti SSL berkekuatan sedang yang kurang untuk melindungi sebuah website, Versi TLS yang sudah lama dan membutuhkan perbaharuan, fitur HSTS yang tidak menyala pada HTTP sehingga tidak aman, komunikasi POP3 yang tidak terenkripsi yang menyebabkan komunikasi yang terjadi dapat dilihat oleh penyerang yang melakukan pencegatan komunikasi, *X-Frame* yang tidak digunakan pada *header* sehingga berpotensi mendapatkan serangan *clickjacking*, tidak adanya pertahanan untuk melakukan pencegahan penyerangan XSS yang berpotensi membuat *website* dapat terserang XSS dan yang terakhir tidak adanya *X-Content-Option* yang memungkinkan penyerang dapat melakukan XSS dengan mengubah beberapa fitur yang terdapat pada *website*.
2. Prosedur *hardening* yang dilakukan pada penelitian ini adalah dengan mencapai tahap *Remediate* dimana pada tahap ini merupakan tahap untuk melakukan analisis dan memberikan rekomendasi terhadap kerentanan yang ada pada *website* ini. Selain melakukan analisis dari kerentanan penelitian ini juga melakukan pengujian penyerangan

terhadap *website* dan penyerangan yang berhasil meliputi serangan DoS, *Intercept* dan *Session Hijacking*. Terdapat pula penyerangan SQL *Injection* yang tidak berhasil pada pengujian penyerangan yang dilakukan, karena terhalang dengan *Firewall* yang terdapat pada *server* penyedia layanan *website* yang sudah cukup kuat yang dapat melakukan pemblokiran IP jika pengguna terindikasi melakukan penyerangan dan hal ini sudah cukup baik untuk menahan serangan terhadap percobaan pencurian data.

Referensi

[1] Sirait, F., & Putra, M. S. (2018). Implementasi Metode *Vulnerability* Dan Hardening Pada Sistem Keamanan Jaringan. *Teknologi Elektro, Universitas Mercu Buana*, 16.

[2] FakultasRekayasaIndustri. (2020, October 22). *CloudFRI*. Diambil kembali dari CloudFRI: http://cloudfri.id/

[3] Retza, L. F., & Affandy. (2016). SECURITY HARDENING DENGAN CLOUD WEB SERVICE UNTUK PENGAMANAN WEBSITE BERBASIS WORDPRESS. 4.

[4] Abdullah, D. (2013). PERANCANGAN DAN IMPLEMENTASI VIRTUAL HOSTING. *CITACEE 1.1*, 2

[5] Wulansari, P. (2015). PERPUSTAKAAN BERBASIS CLOUD COMPUTING. *Jurnal Iqra'* .

[6] Ashari, A., & Setiawan, H. (2011). Cloud Computing : Solusi ICT ? *Jurnal Sistem Informasi (JSI), VOL. 3, NO. 2*.

[7] Koschuch, M., Fruhwirth, T., Glaser, A., Schmidt, S., & Hudler, M. (2015). Speaking in Tongues Practical Evaluation of TLS Cipher Suites Compatibility. *International Conference on Data Communication Networking*, 3.

[8] Patel, K., & Mahida, D. (2017). Birthday Attack on Cryptography Applications. *International Journal of Advance Engineering and Research Development*, 2.

[9] Sasongko, A. (2014). WEB-BUG DENGAN MEMANFAATKAN VARIABLE SERVER PHP UNTUK MENGUMPULKAN INFORMASI AKTIFITAS PENGUNJUNG WEBSITE. *JURNAL KHATULISTIWA INFORMATIKA, VOL. 2 NO. 1*, 2.

[10] Narayanan, A. (2012). Clickjacking Vulnerability and Countermeasures. *International Journal of Applied Information Systems (IJAIS)*, 7.

[11] Suartana, I. M., Wahanani, H. E., & Sandy, A. N. (2015). SISTEM PENGAMANAN WEB SERVER DENGAN WEB APPLICATION FIREWALL (WAF). *Jurnal Teknologi Informasi dan Komunikasi (SCAN VOL. X NOMOR 1)*, 40.

[12] Hasibuan, A. M. (2017). Rancang Bangun Aplikasi Keamanan Data Menggunakan Metode AESPada Smartphone. *MEANS (Media Informasi Analisa dan Sistem)*, 30.

[13] Muliawan, D., & R. R. (2018). Mencegah Exploit URL Pada Model Business to Customer Pada Toko Citra Ponsel Ketapang. *Jurnal ENTER Volume 1*, 25.

[14] Kurniawan, A. (2019). Penerapan Framework OWASP dan Network Forensics untuk Analisis, Deteksi, dan Pencegahan Serangan Injeksi di Sisi Host-Based . *Jurnal Telematika, vol. 14 no. 1*, 11.

[15] Laksmiati, D. (2020). VULNERABILITY ASSESSMENT PADA SITUS WWW.HATSEHAT.COM MENGGUNAKAN OPENVAS. *Jurnal AKRAB JUARA Volume 5 Nomor 3*, 241-242.