Penutupan Celah Keamanan Menggunakan Metode Hardening Studi Kasus: Cloudfri

Closing Security Vocations Using The Hardening Method Case Study: Cloudfri

1st Muhammad Farizqi Setiawan
Fakultas Rekayasa Industri
Universitas Telkom
Bandung, Indonesia
frzqsetiawan@telkomuniversity.ac.id

2nd Rd. Rohmat Saedudin
Fakultas Rekayasa Industri
Universitas Telkom
Bandung, Indonesia
rdrohmat@telkomuniversity.ac.id

3rd Umar Yunan K.s. Herdianto
Fakultas Rekayasa Industri
Universitas Telkom
Bandung, Indonesia
umaryunan@telkomuniversity.ac.id

Abstrak

***Security Hardening* merupakan metode yang ditujukan untuk meningkatkan keamanan suatu sistem agar suatu sistem tidak mudah dieksploitasi atau diserang oleh pihak yang tidak bertanggung jawab. Penelitian ini bertujuan untuk melakukan identifikasi kerentanan yang ada pada aplikasi berbasis *website* tap2go.cloudfri sekaligus menganalisis dan melakukan prosedur *hardening* pada *website.* Penelitian ini menggunakan metode *Security Hardening* sebagai panduan penelitian. Saat menggunakan metode *security hardening* dilakukan juga pengujian *vulnerability scanning* dan *penetration testing.* Hasil penelitian ini berupa analisis *vulnerability scanning* dan *penetration testing*. Kerentanan yang ditemukan yaitu kerentanan terhadap serangan DDoS, komunikasi yang tidak terenkripsi, tidak terpasangnya *Anti-ClickJacking* dan penggunaan SSl/TLS yang telah usang. *Penetration testing* yang dilakukan yaitu simulasi penyerangan DDos, Drupalgeddon, *Interception* dan SQL *Ijection.* Hasil *penetration testing* didapati sistem tersebut telah aman dari serangan SQL *injection* karena sudah terdapatnya *firewall* untuk menahan serangan tersebut sebaliknya untuk jenis serangan lainnya sistem tersebut belum aman dan perlu dilakukan konfigurasi ulang pada *web server* untuk meminialisir celah keamanan yang terdapat pada aplikasi berbasis *website* tap2go.cloudfri.**

Kata Kunci : *Security Hardening, vulnerability, Penetration Testing.*

Abstract

Security Hardening is a method aimed at increasing the security of a system so that a system is not easily exploited or attacked by irresponsible parties. This study aims to identify vulnerabilities that exist in the tap2go.cloudfri website-based application as well as analyze and perform hardening procedures on the website. This study uses the Security Hardening method as a research guide. When using the security hardening method, vulnerability scanning and penetration testing are also carried out. The results of this study are analysis of vulnerability scanning and penetration testing. The vulnerabilities found were vulnerability to DDoS attacks, unencrypted communications, non-installation of Anti-ClickJacking and outdated use of SSL/TLS. Penetration testing carried out is a simulation of DDos, Drupalgeddon, Interception and SQL Injection attacks. The results of the penetration testing found that the system was safe from SQL injection attacks because there was already a firewall to withstand these attacks. On the other hand, for other types of attacks, the system was not secure and needed to be reconfigured on the web server to minimize security holes in the tap2go.cloudfri website-based application.

Keywords: Security Hardening, vulnerability, Penetration Testing.

I. PENDAHULUAN

Perkembangan teknologi informasi khususnya jaringan komputer pada saat ini telah menjadi hal yang mendasar. Teknologi informasi ditandai dengan lahirnya komputer dan perkembangannya yang sangat cepat Andrew R Molnar (1997:63). Seiring berjalannya waktu, teknologi telah berkembang menjadi lebih canggih. Ini dapat membantu dalam setiap kehidupan sehari-hari, baik dalam kegiatan industri maupun rumah tangga. Teknologi

Informasi memiliki banyak manfaat ada untuk transportasi, komunikasi, pendidikan, dll. Semakin cepatnya perkembangan suatu informasi menyebabkan banyaknya kemungkinan aksi *hacking* dan *data-sniffing*. Serangan–serangan tersebut perlu dihindari untuk menghindari kerusakan ataupun kehilangan data pada *website*. Salah satu cara untuk membangun keamanan data dalam jaringan internet adalah dengan menggunakan Metode *Hardening*

Penggunaan *Hardening* dibutuhkan untuk pengamanan sistem. *Hardening* berguna untuk menutup celah–celah yang rentan diserang oleh para *hacker*. Penutupan celah–celah inilah yang membuat sistem jadi sulit untuk diserang. *Hardening* bisa digunakan pada semua sistem, termasuk sistem *cloud*fri Telkom University.

*Cloud*fri merupakan sistem yang berisi kumpulan aplikasi yang digunakan oleh seluruh entitas Fakultas Teknik Industri Universitas Telkom yang mana didalamnya terdapat beberapa *web applications* seperti administrasi.cloudfri.id, ingram.cloudfri.id, labrecruitment.cloudfri.id, tap2go.cloudfri.id, dst. Penggunaan metode *Hardening* pada *cloudfri* dinilai bisa untuk menghindari kerusakan, pengubahan, atau pencurian data pada aplikasi yang terdapat pada sistem *cloudfri.* Hal ini dilakukan untuk menjaga stabilitas dan kinerja sistem *cloudfri.*

Metode *hardening* yang dilakukan pada *cloud*fri menggunakan metode *security hardening*. Dimana *security hardening* ini memiliki empat tahapan, yaitu *access*, *analyze*, *remediate*, dan *manage*. Tahapan ***access*** berguna untuk mencari celah-celah keamanan yang masih terdapat pada sistem, tahap ***analyze*** berguna untuk mencari tingkat keamanan dan menganalisis dampak yang terjadi pada celah keamanan tersebut, kemudian mengklasifikasikan tingkat kerusakan yang diakibatkan oleh celah keamanan tersebut. tahap ***remediate*** berguna untuk menemukan celah keamanan pada sistem yang diuji dan mencari cara untuk menutup celah keamanan tersebut untuk mengamankan sistem, dan tahap ***manage*** yang berguna untuk menutup lubang keamanan dan mencegah serangan masuk, serta mencegah terbukanya celah keamanan lain.

Tujuan dari metode *hardening* yang dilakukan pada c*loud*fri adalah untuk menegakkan keamanan data, membersihkan file sampah, menutup *port* yang tidak digunakan, dan menutup lubang keamanan lainnya. maka sistem tersebut akan sulit diserang karena celah–celah yang ada pada sistem sudah ditutup. Hal ini membuat sistem ***cloudfri*** akan lebih stabil karena tidak ada gangguan terhadap sistem tersebut.

II. KAJIAN TEORI

A. *Security Hardening*

Security Hardening didefinisikan sebagai setiap proses, metodologi, produk atau kombinasinya yang digunakan untuk menambah fungsionalitas keamanan dan/atau menghapus kerentanan atau mencegah eksploitasinya dalam perangkat lunak yang ada. Definisi ini berfokus pada pemecahan kerentanan, bukan pada pendeteksiannya. Pada konteks ini, berikut ini merupakan klasifikasi metode pengerasan keamanan:

a. Code-Level Hardening

Code-Level Hardening adalah perubahan dalam kode sumber dengan cara yang mencegah kerentanan tanpa mengubah desain. Beberapa kerentanan adalah akibat langsung dari aktivitas pemrograman. Pengerasan tingkat kode merupakan penghapusan kerentanan ini secara sistematis.

b. Software Process Hardening

Software Process Hardening adalah penggantian alat pengembangan dan compiler untuk penambahan fitur keamanan pada aplikasi, tanpa mengubah *source code* aslinya. Penggunaan implementasi *library* yang lebih kuat dan penggunaan kode di luar yang tidak mengubah kode asli namun menghasilkan peningkatan keamanan.

c. Design-Level Hardening

Design-Level Hardening terdiri dari rekayasa ulang aplikasi untuk mengintegrasikan fitur keamanan yang tidak ada atau tidak mencukupi. Beberapa kerentanan keamanan tidak dapat diatasi dengan perubahan sederhana dalam kode atau dengan lingkungan yang lebih baik, tetapi disebabkan oleh desain yang cacat secara fundamental.

d. *Operating environment hardening*

Operating environment hardening terdiri dari beberapa konteks (jaringan, *system* operasi, *library*, utilitas, dan lain sebagainya) yang diandalkan oleh perangkat lunak. Perubahan tersebut membuat eksploitasi kerentanan biasanya lebih sulit, meskipun mereka tidak memperbaikinya.

B. *Vulnerability*

Vulnerability adalah kelemahan sebuah sistem akibat dari berbagai jenis pola serangan. Pada setiap sistem dan jaringan tentu akan mempunyai *vulnerability* (kerentanan) dan dapat mengakibatkan kerusakan pada sistem bahkan data perusahaan sehingga menimbulkan kerugian. *Vulnerability* dapat terjadi pada perangkat keras(*hardware*), perangkat lunak (*software*), aplikasi yang dikembangkan perusahaan bahkan kelemahan dari sisi user itu sendiri. *Vulnerability* pada hardware misalnya terdapat lubang keamanan pada processor, *vulnerability* pada aplikasi misalnya dapat menembus ke *user* administrasi, dan serangan SQL *injection*, *kelemahan* pada *user* misalnya penggunaan password yang mudah.

Vulnerability atau kelemahan ini kemudian memiliki peluang digunakan sebagai pintu masuk di mana penyerang akan menyerang sistem yang terdapat *vulnerability* tersebut. Dengan melakukan *penetration testing* setelah dilakukannya *vulnerability scanning* dapat mengidentifikasi *vulnerability* yang ada, mencoba masuk melalui *vulnerability* dan memberikan rekomendasi untuk menutup *vulnerability* yang ada sehingga sistem menjadi aman.

C. *Penetration testing*

Penetration testing merupakan metode evaluasi keamanan sistem komputer atau jaringan dengan mensimulasikan serangan dari sumber yang berbahaya dan

merupakan kegitan *security* audit. Simulasi serangan yang dibuat seperti kasus yang bisa dibuat oleh *black hat hacker, cracker*, dan sebagainya. Tujuannya adalah menentukan dan mengetahui macam – macam serangan yang mungkin terjadi karena kelemahan sistem.

Jenis atau metode *Penetration Testing* terbagi menjadi tiga yakni:

a. Metode *Black-box*, merupakan pengujian yang dilakukan berdasarkan detail aplikasi, seperti tampilan aplikasi, fungsi-fungsi yang terdapat pada aplikasi,serta penyesuaian alur fungsi pada aplikasi dengan bisnis yang diinginkan oleh pelanggan. Pengujian ini dilakukan tanpa melihat dan menguji *source code* program yang ada pada aplikasi.

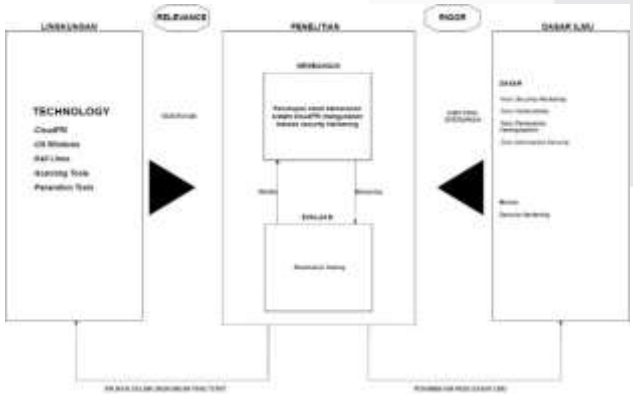
b. Metode *White-box* merupakan pengujian yang dilakukan berdasarkan detail prosedurserta alur logika dari sebuah kode program. Pada metode ini, tester akan melihat keseluruhan *source code* sebuah program untuk menemukan *bugs* dari kode program tersebut.

c. Metode *Grey-box*, merupakan metode pengujian yang berasal dari kombinasi *Black Box* dan *White Box*. Dimana pentester melakukan pengujian aplikasi berdasarkan spesifikasi namun menggunakan cara kerja dari dalam aplikasi tersebut alias *source code* program.

III. METODE

A. Model Konseptual

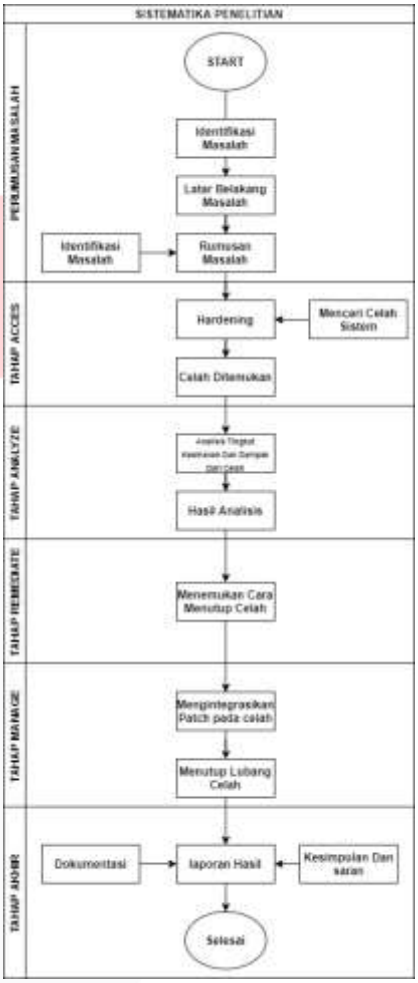
Model konseptual adalah suatu diagram dari satu set hubungan antara faktor – faktor tertentu yang memberi dampak terhadap suatu kondisi target. Tujuan dari model ini adalah untuk mewujudkan sebuah kerangka terstruktur yang digunakan untuk memahami tujuan dari sebuah penelitian. Model konseptual yang digunakan dalam penelitian ini adalah sebagai berikut:



GAMBAR 3. 1 *Model Konseptual*

B. Sistematika Penulisan

Sistematika penyelesaian masalah merupakan pemecahan masalah, sebuah proses terencana yang dilakukanuntuk mencapai tujuan penelitian. Berikut sistematika pemecahan masalah yang dilakukan



GAMBAR 3.2 Sistematika penelitian

a. Perumusan Masalah

Pada tahap ini dimulai dengan melakukan identifikasi latar belakang yang bertujuan untuk menggambarkan masalah yang akan diteliti. Setelah mendapatkan perumusan masalah, maka selanjutkan akan mendapatkan batasan masalah. Batasan masalah ini bertujuan agar penelitian tidak menyimpang dari topik penelitian.

b. Tahap *Access*

Pada tahap ini, dilakukan proses awal *security hardening*. yaitu melakukan *Vulnerability Scanning* dan *Penetration testing*. Dimana Proses ini bertujuan untuk mencari celah-celah keamanan pada aplikasi berbasis *website* tap2go.cloudfri dan mengetahui jalur–jalur yang kemungkinan akan dilakukannya penyerangan terhadap sistem. Setelah data ditemukan, kemudian akan dilakukan pengolahan pada tahap–tahap selanjutnya.

c. Tahap *Analyze*

Pada tahap ini, dilakukan analisis tingkat keamanan dan dampak dari celah yang sudah ditemukan. Proses ini dilakukan untuk mengetahui cara menutup celah yang ditemukan sehingga tingkat keamanan sistem menjadi

No	*Threat level*	*Vulnerability*	CVSS	*Service* (*Port*)
1	*High*	SSL *Medium Strength Cipher Suites Supported* (SWEET32)	7.5	443 / tcp / www 110 / tcp / pop3 143 / tcp / imap 993 / tcp / imap 995 / tcp / pop3
		DNS *Server Spoofed Request Amplification* DDoS	7.5	53 / udp / dns
2	*Medium*	TLS *Version* 1.0 *Protocol Detection*	6.5	443 / tcp / www 110 / tcp / pop3 143 / tcp / imap 993 / tcp / imap 465 / tcp / smtp 995 / tcp / pop3
		HSTS *Missing from* HTTPS *Server* (RFC 6797)	6.5	2078 / tcp / www 443 / tcp / www 2096 / tcp / www 2083 / tcp / www
		DNS *Server Recursive Query Cache Poisoning Weakness*	5.0	53 / udp / dns
		SSL *Anonymous Cipher Suites Supported*	5.9	21 / tcp / ftp
3	*Low*	POP3 *Cleartext Logins Permitted*	2.6	110 / tcp /pop3

lebih kuat dan tidak mudah untuk dilakukan serangan. Setelah mendapatkan hasil dari analisis maka selanjutnya adalah melakukan penutupan pada celah.

d. Tahap *Remediate*

Pada tahap ini, dilakukan analisis untuk penutupan celah yang terdapat pada sistem. kemudian mencari solusi

dan rekomendasi apa yang harus dilakukan untuk menutup celah keamanan tersebut guna mengamankan sistem.

e. Tahap *Manage*

Pada tahap ini Mengintegrasikan *patch* pada celah keamanan kemudian melakukan penutupan lubang keamanan dan mencegah serangan masuk, serta mencegah terbukanya celah keamanan lainnya.

f. Tahap Akhir

Pada tahap akhir, hasil dari simulasi dan hasil analisis dapat dijadikan sebagai referensi untuk laporan hasil penelitian. Dokumentasi dari penelitian dilakukan untuk mendukung laporan hasil penelitian serta dapat dijadikan sebagai referensi untuk memberikan kesimpulan

IV. HASIL DAN PEMBAHASAN

A. Analisis *Vulnerability* Berdasarkan Data dari Nessus

Scanning yang dilakukan menggunakan Nessus pada objek tap2go.cloudfri.id menunjukan bahwa tap2.cloudfri.id memiliki beberapa vulnerability. Data yang diambil dari Nessus antara lain *vulnerability*, CVSS, VPR (*Vulnerability Priority Rating)*, *services*, dan *threat level.* Berikut hasil scan dari Nessus:

GAMBAR 4. 1 Hasil Scanning dari Nessus

Hasil *vulnerability scanning* menggunakan aplikasi Nessus pada *website* tap2go.cloudfri dengan *ip* 193.168.194.15 terdeteksi bahwa terdapat *vulnerability* terdiri dari berbagai kategori yakni 7 *high risk*, 12 *medium risk* dan 1 *low risk*.

TABEL 4.1 Hasil Vulnerability pada Nessus

B. Analisis *Vulnerability* Berdasarkan Data dari Arachni

hasil *scanner* menggunakan aplikasi Arachni. Scanner akan diperoleh hasil berupa detail dari *threat level* target yaitu aplikasi berbasis *website* tap2go.cloudfri. Berikut hasil dari *scanning* pada *website* tap2go.cloudfri:

GAMBAR 4. 2 Hasil Scanning Dari Arachni

Hasil dari *scanning* tersebut dapat dilihat *threat level* pada *website* tap2go.cloudfri adalah sebagai berikut:

TABEL 5. 2 Hasil Scanning pada Arachni

Threat level	***Vulnerability***
Low	*X-Frame-Options header missing*
Informational	*Interesting Response*

Pada tabel 5.2 membahas mengenai hasil yang didapatkan pada *vulnerability* dari *website* tap2go.cloudfri dengan menggunakan aplikasi Arachni. Pada daftar *vulnerability* yang didapat, masing – masing memiliki penjelasan sebagai berikut:

a. low risk

a. X-Frame-Options Header Missing

Deskripsi: *Clickjacking (User Interface redress attack, UI redress attack, UI redressing)* adalah teknik berbahaya untuk menipu pengguna Web agar mengklik sesuatu yang berbeda dari apa yang pengguna anggap sedang mereka klik, sehingga berpotensi mengungkapkan informasi rahasia atau mengambil kendali komputer mereka saat mengklik halaman web yang tampaknya tidak berbahaya.

b. Informational

a. *Interesting Response*

Deskripsi: Server merespons dengan kode status bukan 200 (OK) atau 404 (Tidak Ditemukan). Ini bukan masalah, namun kode status respons HTTP eksotis dapat memberikan wawasan yang berguna tentang perilaku aplikasi web dan membantu uji penetrasi.

C. Hasil Pengujian Eksploitasi DoS

Pada subbab ini menjelaskan implementasi eksploitasi DoS *SynFlood*. eksploitasi DoS Syn*Flood* merupakan sebuah serangan dimana sebuah jaringan akan dibanjiri dengan *fake traffic* yang sangat banyak. Server atau jaringan yang diserang tidak mampu mengakomodasi lalu lintas lintas tersebut, sehingga menyebabkan *website*/layanan *down* dan tidak bisa beroperasi. Eksploitasi DoS *SynFlood* ini sendiri menggunakan aplikasi yang dijalankan pada sistem operasi Kali Linux yaitu Metasploit *Framework* yang digunakan sebagai alat untuk melakukan serangan DoS Syn*Flood* dan Wireshark sebagai alat untuk menganalisis detail lalu lintas jaringan pada saat proses eksploitasi DoS

GAMBAR 4. 3 Pengujian Eksploitasi DoS

Pada proses pengujian DoS di sini Metasploit Framework sendiri sudah menyediakan *module auxiliary* untuk melakukan penyerangan tersebut dengan memasukan perintah "auxiliary/dos/tcp/synflood" untuk dapat mengakses ke dalam *module* tersebut. Sebelum menjalankan proses eksploitasi perlu dilakukannya konfigurasi alamat IP target dan *port* yang dituju dengan memasukan perintah "set RHOSTS 193.168.194.15" untuk konfigurasi alamat IP target, dan "set RPORT 53" untuk konfigurasi *port* target yang dituju. Selain itu diperlukan juga adanya aplikasi wireshark untuk merekam semua paket yang melewati *interface* yang dipilih.

D. Hasil Pengujian *Penetration Testing* SQl Injection

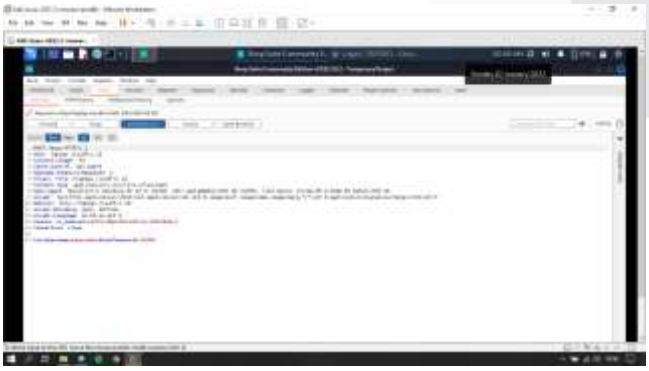
Penyerangan SQL *Injection* menggunakan SQLMap terhadap aplikasi berbasis *website* tap2go.cloudfri. SQL *Injection* sendiri merupakan sebuah tindakan ekploitasi terhadap suatu *website* dengan memasukkan perintah-perintah SQL melalui url untuk dieksekusi oleh database, dimana penyerang dapat mengambil alih database

GAMBAR 4. 4 pengujian *Penetration Testing* SQl Injection

Pada Gambar 4. 4 menunjukan bahwa SQLMap menjalankan programnya dan proses ini mengalamani penghambatan karena *website* tersebut terlindungi oleh WAF. mengalami pemblokiran. Pada Gambar V,12 terdapat pernyataan "[CRITICAL] WAF/IPS identified as Imunify360 (CloudLinux)". Hal ini menandakan bahwa aplikasi berbasis *website* tap2go.cloudfri sudah terpasang sebuah WAF (*Web Application Firewall*) dengan nama Imunify360 sehingga serangan *SQL Injection* yang dilakukan terhadap aplikasi berbasis *website* tap2go.cloudfri terblokir. Maka 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*.Dari hasil tersebut dapat disimpulkan bahwa untuk saat ini aplikasi berbasis *website* tap2go.cloudfri tidak perlu menambahkan pengamanan untuk pencegahan terhadap serangan *SQL Injection* dikarenakan sudah terpasangnya WAF

E. Hasil Pengujian *Penetration Testing Interception*

Hasil pengujian *penetration testing Interception* menggunakan aplikasi Burp suite pada sistem operasi Kali Linux. *Interception* merupakan merupakan sebuah ancaman terhadap kerahasiaan dimana ada pihak yang tidak memiliki wewenang secara sah berhasil mendapatkan hak akses informasi dari sistem komputer.



GAMBAR 4. 5 *Pengujian Penetration Testing Interception*

Pada Gambar 4.5 dapat dilihat pengujian ini mendapati bahwa saat proses *request login* sedang berlangsung pada aplikasi berbasis *website* tap2go.cloudfri hasil dari interception dengan aplikasi Burp Suite menunjukan informasi mengenai *username* dan *password* yang digunakan untuk proses *request login* tidak terenkripsi. Dengan *username* dan *password* yang tidak terenkripsi, aplikasi berbasis *website* tap2go.cloudfri akan mudah terkena serangan. Hal ini dikarenakan attacker mendapatkan informasi yang bersifat *critical* dari *website* tap2go.cloudfri. *Username* serta *password* yang tidak terenkripsi bisa terjadi karena komunikasi dari tap2go.cloudfri tidak terenkripsi oleh *SSL/TLS*. Komunikasi yang tidak terenkripsi oleh *SSL/TLS*

Sebagai solusi untuk mengatasi serangan ini yaitu dengan menggunakan SSL pada *website* untuk dapat mencegah terjadinya penyerangan DNS *spoofing*, serta untuk melindungi komunikasi sensitif antara klien dan *server*, karena SSL memungkinkan enkripsi 256 bit yang susah ditembus oleh penyerang sekalipun.

V. KESIMPULAN

Setelah dilakukan pengujian terhadap aplikasi berbasis *website* tap2go.cloudfri dan mendapatkan hasil analisis *vulnerability dan* hasil *penetration testing* dengan menggunakan beberapa *tools*, dapat diambil kesimpulan sebagai berikut :

A. Terdapat beberapa kategori *vulnerability* yang ada pada aplikasi berbasis *website* tap2go.cloudfri setelah dilakukan analisis dan penilaian menggunakan beberapa aplikasi dengan terdeteksi bahwa terdapat *vulnerability* seperti SSL berkekuatan sedang yang kurang untuk melindungi sebuah website, penggunaan SSL/TLS yang telah usang, *X-Frame* yang tidak digunakan pada *header* sehingga berpotensi mendapatkan serangan *clickjacking*, dan terakhir tidak terdapat XSS Protection yang berpotensi membuat *website* dapat terserang XSS (*Cross site scripting)*.

B. Dari hasil *penetration testing* yang telah dilakukan terhadap *vulnerability* yang terdapat pada aplikasi berbasis *website* tap2go.cloudfri memiliki dampak yang cukup merugikan bagi *website.* Pada pengujian penyerangan DoS SYNFlood berhasil membuat *website* untuk sulit diakses sehingga layanan yang terdapat pada *website* menjadi terganggu, kemudian hasil dari *exploit* menggunakan metode *Drupal* pada aplikasi berbasis *website* tap2go.cloudfri untuk membuktikan *vulnerability Drupal Remote Command Execution* mengalami kegagalan, sehingga data yang diinginkan dari pengujian *exploit* tidak mendapatkan informasi apapun. Pada pengujian Interception mendapati bahwa pada saat *website* tap2go.cloudfri melakukan request *login*, terdapat komunikasi atribut untuk *login* yaitu *username* dan *password* yang tidak terenkripsi. Dan Terakhir terdapat pula pengujian penyerangan SQL *Injection* yang mengalami pemblokiran yang disebabkan sudah terpasangnya WAF (*Web Application Firewall*) pada *website* tap2go.cloudfri.

REFERENSI

[1] Anjar, P. (2006). Vulnerability Assessment untuk Meningkatkan Kesadaran Pentingnya Keamanan Informasi. Jurnal Sistem Informasi, 1(2), 73-83

[2] Antunes, N., & Vieira, M. (2012). *The Devils Behind Web Application Vulnerabilities*.

[3] Arbi, A. (2020). Penetration Testing Untuk Mengetahui Kerentanan Keamanan Aplikasi Web Menggunakan Standar OWASP 10 pada domain Web Perusahaan.

[4] Juardi, D. (2017). Kajian Vulnerability Keamanan Jaringan Internet Menggunakan Nessus. SyntAax Jurnal Informatika, 6(1), 11–19.

[5] Hardening Cookies in Web-based Systems for Better System Integrity (Penambahbaikan Penggunaan Cookie bagi Meningkatkan Integriti Sistem Berasaskan Web), *Seminar R&D 2012, 26 - 28 Sept 2012, Nuclear Malaysia*

[6] Mourad, A., Laverdi´ere, M., & Debbabi, M. (2007). A High-Level Aspect-Oriented Based Language For Software Security Hardening. *Computer Security Laboratory, Concordia Institute for Information Systems Engineering, Concordia University, Montreal (QC), Canada fmourad,ma laver,debbabig@ciise.concordia.ca*

[7] Mourad, A. Alhadidi, D., & Debbabi, M. (2008). Cross-LanguageWeaving Approach Targeting Software Security Hardening.

[8] Mourad, A., Laverdière, M., & Debbabi, M. (2006). *Security hardening of open source software*. 1. https://doi.org/10.1145/1501434.1501486

[9] Mourad, A., Laverdi´ere, M., & Debbabi, M. (2008). A Security Hardening Language Based on Aspect-Orientation.

[10] Putra, F. A., & Purwanto, J. (2015). Perancangan Pengamanan Jaringan Pada Perguruan Tinggi Xyz. Seminar Nasional Sistem Informasi Indonesia.

[11] Prabowo, M. A., Darusalam, U., & Ningsi, S. (2020). Perancangan Keamanan Server Linux Dengan Metode Hardening Pada Layer 1 dan Layer 7. Jurnal Media Informatika Budidarma, 4(3). 591-603. ISSN 2614-5278 (media cetak), ISSN 2548-8368 (media online).

[12] PT. Inovasi Informatika Indonesia. (2020, 1 2). *Security Hardening bersama Fikri Muhammad Arifin*. Retrieved from https://i-3.co.id/: https://i-3.co.id/sesi-qa-tentang-hardening-bersama-fikri-muhammad-arifin/

[13] Retza, L. F. G., & Affandy. (2016). Security Hardening Dengan Cloud Web Service Untuk Pengamanan Website Berbasis Wordpress. 4-7.