Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK) DOI: 10.25126/jtiik.2023105872
Vol. 10, No. 1, Februari 2023, hlm. 51-58 p-ISSN: 2355-7699
Akreditasi KEMENRISTEKDIKTI, No. 36/E/KPT/2019 e-ISSN: 2528-6579

OPTIMASI KEAMANAN *WEB SERVER* TERHADAP SERANGAN *BRUTE-FORCE* MENGGUNAKAN *PENETRATION TESTING*

Fahmi Fachri*[1]

[1]Universitas Ma'arif Nahdlatul Ulama, Kebumen
Email: [1]fahmifachriumnu@gmail.com
*Penulis Korespondensi

(Naskah masuk: 25 November 2021, diterima untuk diterbitkan: 27 Februari 2023)

Abstrak

Peningkatan serangan siber dan pencurian data *sensitive* menjadi topik utama yang sering dibahas saat ini, karena semakin banyak aplikasi berorientasi pengguna dengan memberikan semua informasinya dikerahkan ke *web*. Uji coba penetrasi diartikan sebagai upaya resmi dalam mengeksploitasi *system* dengan tujuan mencari kelemahan yang ada pada *web server* serta meningkatkan keamanan *system*. Pengujian penetrasi ini dilakukan pada *web server* yang merupakan Sistem Informasi Akademik pada perguruan tinggi. Metode yang digunakan dalam penelitian ini mencakup *Intelligence Gathering, Vulnerability Analysis, Exploitation, Post Exploitation, Reporting*. Hasil Penelitian menampilkan terdapat tiga kategori kelemahan, 5 kerentanan dalam level *High*, 164 kerentanan dalam level *Medium*, 52 kerentanan di level *Low*. Terbukanya beberapa *port* yang masih terbuka dan menyebabkan penyusup dengan mudah masuk kedalam *system* untuk melakukan serangan *Brute Force* atau yang lainnya. Hasil uji coba simulasi serangan pada *server* berhasil dilakukan dengan mendapatkan *username* dan *password*, hal ini tentunya berbahaya *system* dapat diambil alih oleh penyusup. Optimalisasi keamanan pada *system* dilakukan perbaikan dengan mengkonfigurasi *File2ban* yang ada pada *server* untuk mencegah dan menutup akses penyusup agar tidak bisa masuk kedalam *system*, hal tersebut sudah dilakukan dan berhasil menolak *attacker* untuk masuk kedalam *sytem*. Berdasarkan perolehan data pada perbaikan *web server* ini telah sesuai dengan harapan yang diinginkan peneliti.

Kata kunci: *Web Server, Penetration Testing, Brute Force, Log.*

OPTIMIZING WEB SERVER SECURITY FOR BRUTE-FORCE ATTACKS USING PENETRATION TESTING

Abstract

The increase in cyber attacks and theft of sensitive data is a major topic that is often discussed today, as more and more user-oriented applications by providing all their information are deployed to the web. Penetration testing is defined as an official attempt to exploit the system with the aim of finding weaknesses in the web server and improving system security. This penetration test is carried out on a web server which is an Academic Information System at a university. The methods used in this research include Intelligence Gathering, Vulnerability Analysis, Exploitation, Post Exploitation, Reporting. The results of the study show that there are three categories of weaknesses, 5 vulnerabilities at the High level, 164 vulnerabilities at the Medium level, 52 vulnerabilities at the Low level. The opening of several ports that are still open and causes intruders to easily enter the system to carry out Brute Force attacks or others. The results of the simulation trial of the attack on the server were successfully carried out by obtaining a username and password, this is of course dangerous that the system can be taken over by intruders. Optimization of security on the system was repaired by configuring File2ban on the server to prevent and close access to intruders so that they could not enter the system, this has been done and succeeded in refusing the attacker to enter the system. Based on the data obtained on the repair of this web server, it is in accordance with the expectations of the researchers.

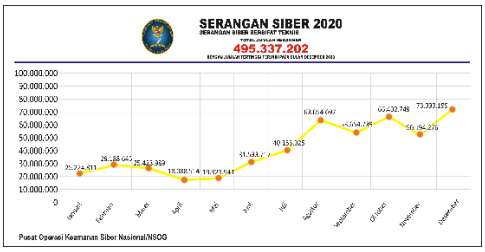
Keywords: *Web Server, Penetration Testing, Brute Force, Log.*

1. PENDAHULUAN

Kerentanan halaman *web* telah dieksploitasi sejak awal tahun 90-an, peningkatan serangan siber dan pencurian data sensitive menjadi topik utama yang sering dibahas saat ini, karena semakin banyak aplikasi-aplikasi berorientasi pengguna dengan memberikan semua informasinya ke halaman *web*

seperti toko *online*, *internet banking*, *facebook* dan *twitter* (Bin Ibrahim & Kant, 2018).

Sebuah aplikasi *web* adalah aplikasi berorientasi tugas yang digunakan pada *web server*, hal ini menjadi bagian penting karena *web server* dituntut harus menjaga integritas informasi yang disampaikan kepada pengguna *web*. Menurut data dari Badan Siber dan Sandi Negara Tahun 2020, bahwa serangan *web* atau *cyber attack* mendeteksi jumlah serangan naik 4 kali lipat jumlah serangan dari tahun sebelumnya yaitu 2019 dengan jumlah serangan hanya 98 juta. Data tersebut disajikan pada Gambar 1. Jumlah *Cyber attack* 2020, BSSN.



Gambar 1. Jumlah *cyber attack* 2020, BSSN

Berdasarkan Gambar 1. menampilkan data dari Pusat Operasi Keamanan Siber Nasional tentang serangan siber selama tahun 2020, jumlah serangan yang terjadi pada periode Januari sampai Desember 2020 adalah sebanyak 495.337.202. Hal ini dimaksudkan agar keamanan *system web* perlu ditingkatkan.

Penggunaan *web* yang semakin banyak tidak menjamin akan keamanan, dapat disimpulkan dari data diatas bahwa banyaknya sebaran *hacker* yang berdampak pada resiko adanya serangan atau eksploitasi. Keamanan, menjadi penanganan penting pada *web server* akan terjaganya data dan informasi untuk disampaikan kepada pengunjung *web* agar terhindar dari kerentanan yang dapat diambil alih oleh peretas (Divya, 2019).

Banyak kerentanan keamanan pada halaman web yang dihasilkan dari permasalahan yang telah dibuktikan, salah satunya adalah tindakan *brute force*. *Brute force* adalah teknik serangan atau tindakan *hacker* secara paksa pada sistem keamanan *web* dengan menggunakan percobaan menebak *username* dan *password* (Sadasivam, 2018).

Peretas menggunakan serangan *brute force* untuk mendapatkan kerentanan kode dan halaman *web* tersembunyi yang dapat dieksploitasi, setelah teridentifikasi penyerang menggunakan informasi itu untuk menyusup kedalam sistem dan membahayakan data, tujuan akhir mereka adalah menyebabkan beberapa penolakan layanan pada halaman *web* dan mengeluarkan data dari system untuk ditujukan ke pihak ketiga (Hossain, 2020).

Beberapa penelitian telah melakukan pengujian keamanan *web server* terhadap serangan *Brute Force* dengan berbagai metode, diantaranya dengan metode *Failtoban* (Prasetyo, 2020); Graphic *Processing Power* (Pramaditya, 2016); Metode *K-Means* dan *Naive Bayes* (Sandra, 2016). Selain itu terdapat pula riset yang menggunakan metode *Penetration Testing* (Stiawan, 2016).

Cara paling akurat dalam mengevaluasi sikap keamanan informasi di *web* adalah dengan mengamati bagaimana organisasi, perusahaan dan yang lain untuk berdiri melawan serangan, dengan pengujian penetrasi sering kali ditemukan kerentanan baru dalam menganalisis kemanan sistem (Zeebaree, 2020)

Penelitian ini menggunakan metode *penetration testing* yaitu pengujian kerentanan pada *web server* sebelum *hacker* melakukan serangan dan memperbaikinya tepat waktu tanpa jejak. Pengujian penetrasi membantu manajemen untuk mendapatkan tampilan keamanan halaman *web* mereka dari sudut pandang penyerang (Krishnan & Wei, 2019).

Dalam studi kasus keamanan, riset ini bertujuan untuk mencari *Vulnerability* atau celah keamanan pada *web server* terhadap serangan *Brute Force* agar segera diperbaiki, (Mutemwa, 2019).

2. METODE PENELITIAN

2.1. Penetration Testing

Penelitian ini menggunakan metode *Penetration*. Penetration testing adalah tindakan untuk mengamankan suatu organisasi dengan meniru yang dilakukan penyerang, hal ini membantu dalam menentukan berbagai tingkat kerentanan yang dapat merusak system dan memperbaikinya sebelum benar-benar terjadi (Riadi, 2020)

Metode penelitian *Penetration testing* memiliki lima tahapan, yaitu *Intelligence Gathering, Vulnerability Analysis, Exploitation, Post Exploitation, Reporting* (Chipher, 2020).

Gambar 2. *Methodology Penetrtion Testing*

Berikut deskripsi dari alur metode penelitian *Penetration Testing* yang terdiri dari lima bagian utama dan saling terhubung,

1. *Intelligence Gathering*, pengumpulan informasi mengenai web server.
2. *Vulnerability Analysis*, tahapan ini dilakukan pemindaian kerentanan dan menentukan jenis serangan yang bisa di eksploitasi.
3. *Exploitation*, dilakukan serangan terhadap kerentanan yang ditemukan dan menguji apakah benar bisa di eksploitasi.

4. *Post Exploitation*, tahap ini dilakukan perbaikan dan menerapkan solusi atas kerentanan, selanjutnya dilakukan pengujian kembali.
5. *Reporting*, ini adalah bagian pembuatan laporan berdasarkan hasil analisis berupa hasil pengujian kerentanan sebelum dan sesudah perbaikan.

2.2. Bahan Penelitian

Bagian ini menentukan alat dan bahan yang memainkan peran penting dalam simulasi serangan terhadap *web server*, seperti pada Tabel 1. Bahan penelitian

Tabel 1. Bahan Penelitian

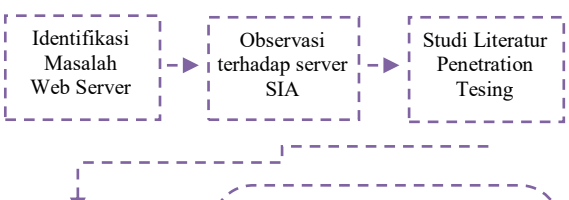
No	Bahan	Status	Informasi
1	Dell 7373	Perangkat Keras	Attacker
		Perangkat Keras	
2	Dell 7450	Perangkat	Server
		Lunak	
3	Parrot	Perangkat	OS Attacker
		Lunak	
4	Ubuntu	Perangkat	OS Server
		Lunak	
5	NMAP	Perangkat	Information Gathering
		Lunak	Information Gathering
6	Whois	Perangkat	Vulnerabilty Analys
		Lunak	Vulnerabilty Analys
7	Nikto	Perangkat	Exploitation
		Lunak	
8	Acunetix	Perangkat	Exploitation
		Lunak	
9	Metasploit	Perangkat	Post Exploitation
		Lunak	
10	Medusa	Perangkat	
		Lunak	
11	File2ban		

Pada Tabel 1. menyajikan daftar data dari alat dan bahan untuk melakukan serangkaian simulasi. Proses serangan dilakukan oleh *attacker* menggunakan *Operating System Parrot Security*. *Parrot OS* menyediakan *Tools* lengkap untuk melakukan *Penetration Testing* yaitu *NMAP, Whois, Nikto, Acunetix, Metasploit, Medusa* (Meng, 2015).

Pada serangan kasus *brute force* dilakukan kegiatan simulasi dengan melakukan perbaikan dan evaluasi Sistem Informasi Akademik pada *server* yang dibangun sendiri menggunakan sistem operasi *Ubuntu server* 20.05.

2.3. Pengujian Sistem

Pengujian sistem pada penelitian ini menjelaskan alur atau tahapan yang dibutuhkan peneliti agar penyampaian isi laporan runtut dan mudah dipahami, sehingga nantinya kerangka penelitian bisa membuat kedalaman penelitian tetap terjaga (Pohan, 2020), seperti pada Gambar 3. Tahapan *Penetration Testing.*



Gambar 3. Tahapan *Penetration Testing*

3. Hasil dan Pembahasan

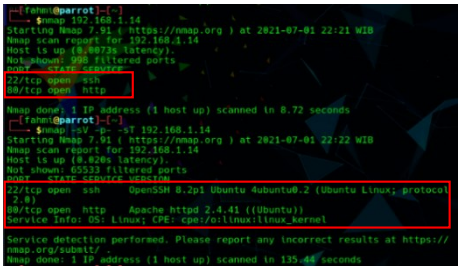
Pada simulasi kasus dilakukan kegiatan simulasi serangan dengan melakukan peninjauan dan evaluasi Sistem Informasi Akademik pada *server*.

3.1. Intelligence Gathering

Langkah pertama adalah *Intelligence Gathering,* yaitu salah satu proses yang paling penting dari pengujian penetrasi, karena tahap pertama di mana tindakan langsung terhadap target diambil (Kothia, 2019). *Intelligence gathering* bertujuan menyelidiki masalah dan menentukan *tools* yang tepat untuk fase berikutnya.

Proses pengumpulan informasi sendiri terbagi menjadi dua, yaitu *Active information gathering* dan *passive information gathering* (Sadigh, 2018). Pengumpulan informasi teknik *active information gathering* menggunakan *tools NMAP (Network Exploration or Security Auditing)* sedangankan pengumpulan information *passive information gathering* menggunakan *tools WHOIS.*

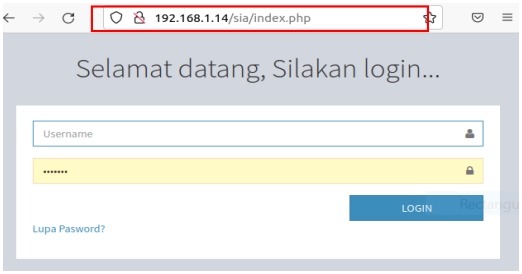
1. *Nmap (Network exploration or security auditing)*



Gambar 4. *Scanning NMAP*

Gambar 4 yaitu hasil pemindaian menggunakan *Nmap* pada *server* Sistem Informasi Akademik, menampilkan informasi mengenai *port* pada *system* masih terbuka, salah satunya port 22 *SSH (Secure Shell)* dan *port* 80 *HTTP (Hypertext Transfer Protocol)* yang berjalan pada ip 192.168.1.14.

2. Whois Lookup



Gambar 5. *Front End Web*

Pada Gambar 5. menampilkan Informasi bagian depan *web* yaitu pengguna dapat melihat dan berinteraksi langsung pada awal berkungjung sebelum masuk ke dalam system dan mendapatkan semua informasi yang ada didalamnya (Udjaja, 2018)

Gambar 6. *Back End Web*

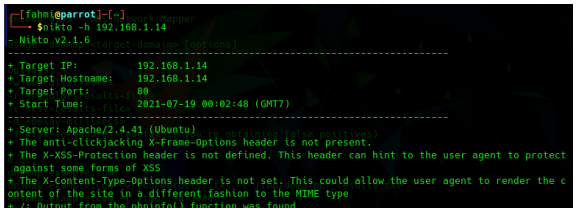
Gambar 6. yaitu halaman *web* pada bagian belakang atau *Back end web* yang menginformasikan mengenai sebuah *server* pada saat pertama kali dibangun, mencakup tanggal didaftarkannya *Domain Name Server*, menjelaskan informasi mengenai otoritatif zona *domain* yang mengasuh dan menampilkan waktu untuk *domain* kapan akan hangus dan *update* pada *server* (Wu, 2018).

3.2. Vulnerability Analysis

Langkah kedua yaitu *Vulnerability Analysis*. *Vulnerability Analysis* adalah proses identifikasi, mengklasifikasikan, dan memprioritaskan kerentanan dalam sistem komputer, aplikasi, dan infrastruktur jaringan dan memberikan organisasi melakukan penilaian dengan pengetahuan (Angelini, 2019). Pencarian kerentanan pada *server* menggunakan dua *tools* yaitu :

1. Nikto

Merupakan *tools* Pemindai keamanan aplikasi *web* berfungsi sebagai informasi lalulintas dan kerentanan *web server* secara *detail* yang dirancang sebagai alat yang tersembunyi dalam mengevaluasi system atau pencarian kelemahan didalam penelitian ini (Abdur Rahman, 2020). Seperti pada Gambar 7. *Scanning Nikto.*

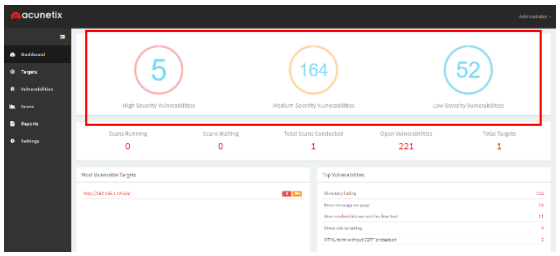


Gambar 7. Scanning *Nikto*

Gambar 7. memberikan Informasi mengenai hasil Scanning oleh *Nikto*, diperoleh beberapa kerentanan yaitu :

a. Anti-clickjacking
b. X-XSS (Cross-site Scripting)
c. X-Frame Header Options is Missing
d. Indikasi file /index.php
e. *HTML Form without CSRF Protection*

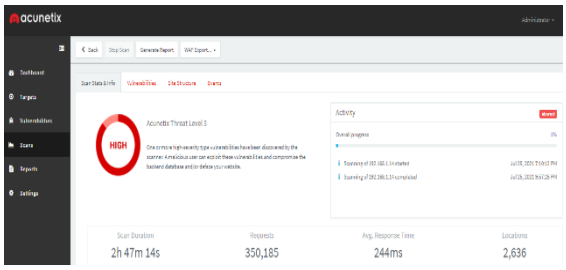
2. Acunetix



Gambar 8. *Scanning Acunnetix*

Pada Gambar 8. menampilkan hasil pemindaian kerentanan menggunakan *Acunetix*, menghasilkan beberapa kategori, yaitu :

a. 5 Kerentanan berada pada *level High Saverity Vulnerabilities*
b. 164 Kerentanan berada pada *level Medium Saverity Vulnerabilities*
c. 52 Kerentanan berada pada *level Low Saverity Vulnerabilities*



Gambar 9. *Scanning Acunnetix*

Gambar 9. memberikan Informasi bahwa tingkat keamanan *web server* pada *tools Acunetix* berada pada level berbahaya (*high*). Pada level ini peretas sangat mudah untuk mengeksploitasi yang mungkin dilakukan mulai dari, melihat, merubah, menghapus data, membuat akun baru dengan tujuan adalah membahayakan web dan mengambil alih fungsi *website* sepenuhnya (Amankwah, 2020).

Hasil Pemindaian dari kedua *tools* dapat disimpulkan bahwa kerentanan terhadap serangan *Brute Force* berada di kategori *High*, selanjutnya akan di analisis dan ditingkatkan untuk menjaga tingkat keamanan sistem dari serangan luar. Jenis kerentanan lain dan kategori tingkat keamanan yang ditemukan dalam system dalam dilihat pada Tabel 2. Kerentanan *Web Server.*

Tabel 2. kerentanan Web Server

No	Indikasi File	Jenis Kerentanan	Kategori
1	/login.php	X-Frame Header Options is Missing	High
2		Cross Site Scripting	High
3	/login.php	HTML Form Without CSRF Protection	High
4	/login.php	Login *page-password-guessing attack*	High
		Open Port	
5	Port 22, 80		Medium

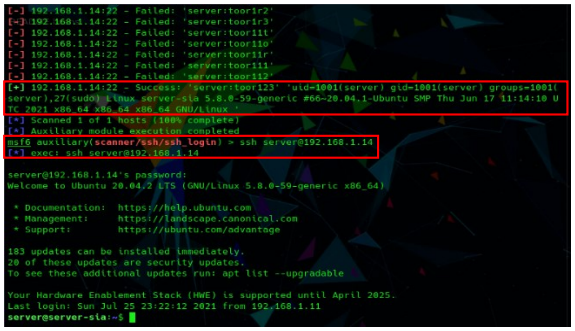
3.3. Exploitation

Lahkah ketiga setelah menemukan kerentanan yaitu melakukan eksploitasi, untuk melanggar semua jenis keamanan dan mengambil alih kendali jarak jauh akses jaringan, aplikasi atau sistem (McKinnel, 2019). Pada penelitian ini menggunakan Kerangka kerja *Metasploit* dan *Medusa* untuk mengeksploitasi kerentanan. Melalui eksploitasi, pentester bisa mendapatkan akses jarak jauh dari sistem.

Tujuan dari pen tester adalah seberapa jauh masuk ke infrastruktur untuk mengidentifikasi target dan menghindari deteksi (Seema & Ritu, 2019).

1. Metasploit

Tools yang pertama untuk menguji keamanan sytem dan meluncurkan serangan kepada *server* menggunakan kerangka kerja *Metasploit*, dengan perintah pada gambar berikut.

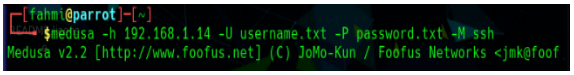


Gambar 10. Session Metasploit

Gambar 10. menyajikan *Session* pencarian untuk mengetahui *username* dan *password* dengan perintah *"Auxiliary (scanner/ssh/ssh_login) run*" pada *Parrot OS* sebagai *attacker*, didapat *username* "*server*" dan *password* "toor123". dengan demikian *attacker* bisa masuk kedalam *system* menggunakan perintah "ssh server@192.168.1.14" dan *password* yang terlah diketahui. sehingga *tools Metasploit* dapat berkerja sebagaimana mestinya.

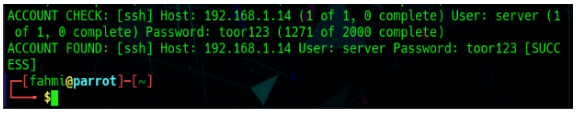
2. Medusa

Tools kedua untuk menguji keamanan *sytem* dan meluncurkan serangan kepada *server* menggunakan kerangka kerja *Medusa*. Seperti Gambar 11. *Session* serangan *Medusa.*



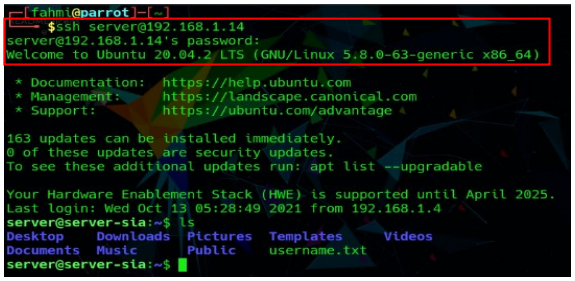
Gambar 11. Session serangan Medusa

Pada Gambar 11. memperlihatkan attacker menggunakan perintah Medusa dengan tujuan bisa mendapatkan akses agar bisa masuk kedalam system target. Session ini tentunya untuk mencari username dan password dengan memasukan beberapa kemungkinan user secara acak yang telah dibuat pada tahap sebelumnya.



Gambar 12. Session Medusa

Gambar 12. menampilkan *ACCOUNT FOUND* pada *ssh host* 192.168.1.14 dengan mendapatkan *username* dan *password*, pada tahap ini *attacker* berhasil dengan mudah masuk kedalam *system* target.



Gambar 13. Masuk kedalam System

Gambar 13. memberikan informasi bahwa *attacker* mencoba *login* ssh ke *system* target dengan dimasukannya *username* dan *password* yang telah didapat, dan berhasil masuk ke dalam *system.* Setelah masuk kedalam system attacker dapat mengambil alih *web* dan mendapatkan semua informasi yang ada pada *system*, pada tahap ini peretas dapat sepenuhnya merubah data atau informasi yang ada pada *web server* (Gede, 2020).

3.4. Post Exploitation

Setelah tahapan *Exploitation,* maka langkah keempat yaitu *Post Exploitation.* Tahapan ini dilakukan proses perbaikan pada *web server*

berdasarkan solusi yang tepat untuk mengatasi kerentanan dan serangan yang terjadi (Rapley, 2018).

Penyelesaian permasalahan password cracking dengan menggunakan algoritma brute force akan menempatkan dan mencari semua kemungkinan password dengan masukan karakter dan panjang password tertentu tentunya dengan banyak sekali kombinasi password (Lustick & Tetlock, 2021)

Cara terbaik untuk mengatasi permasalahan tersebut adalah dengan sistem untuk mencegah dari serangan atau penyusupan. *Fail2ban* merupakan paket program untuk mendeteksi usaha *login* yang gagal dan kemudian memblokir alamat *IP host* asal (Ibnu Muakhori, Sunardi, 2020).

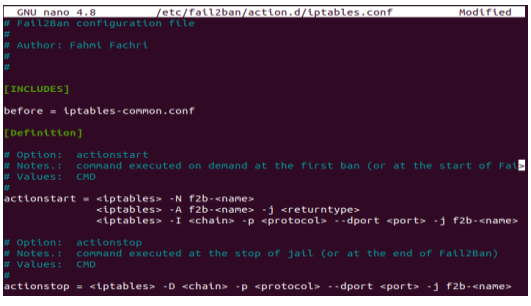
1. Perbaikan pada *Web Server*

   - Konfigurasi *ssh* pada *rule fail2ban* dengan perintah "*sudo nano /etc/fail2ban/jail.local*"

Gambar 14. konfigurasi ssh pada file2ban

Gambar 14. merupakan isi file jail.local untuk mengatur *rule* dan membatasi jumlah kegagalan hingga batas waktu *ip addres* terblokir dan deteksi pencegahan pada *service ssh.*

   - Konfigurasi banaction berfungsi mengirimkan informasi serangan kedalam database, dengan perintah "*sudo nano /etc/fail2ban/ action.d/iptables.conf*"



Gambar 15. konfigurasi banaction ssh pada file2ban

Pada tahap ini dilakukan agar file fail2ban yang bertanggung jawab untuk mengirimkan informasi ke database dapat berjalan dan membaca suatu serangan.

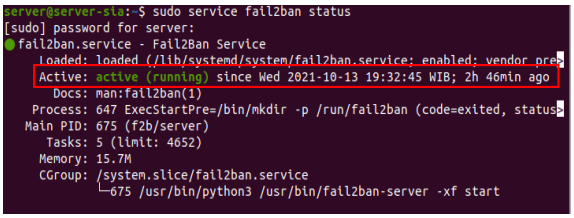
   - Konfigurasi otomatis ban adalah tahapan pada server agar ketika terjadi serangan, server lain dapat melakukan pencegahan, konfigurasi dilakukan dengan membuat file db2ban.php yang berada pada directory /etc/crontab seperti Gambar 16 Konfigurasi otomatis ban.

Gambar 16. konfigurasi banaction ssh pada file2ban

Gambar 16. menampilkan konfigurasi File fail2ban.php yang dapat melakukan pembacaan pada databe dan akan mengambil informasi serangan secara otomatis terhadap server dengan interval kurang dari 60 detik (Prasetyo, 2020)

2. Simulasi dan Uji coba serangan

Ujicoba serangan dilakukan dari host attacker dan simulasi srengan *brute force* terhadap *server* dilakukan saat *fail2ban* dalam keadaan aktif (*enabled)*. Serangan *brute force* dilakukan terhadap *SSH*. Perintah yang digunakan yaitu, *"sudo service fail2ban status"*.



Gambar 17. Status Fail2ban

Simulasi serangan *Brute force* menggunakan *tools medusa* dengan perintah *"medusa -h 192.168.1.14 -U username.txt –P password.txt –M ssh". Medusa* adalah pemaksa brute yang digunakan untuk memaksa kredensial agar mengarah pada eksekusi *dictionary* atau *list password* untuk mencoba masuk (Khormali, 2021).

Gambar 18. Serangan ssh setelah Fail2ban

Gambar 18. memberikan Informasi bahwa ketika *File2ban dinonaktifkan* maka semua serangan berhasil masuk kedalam *system* untuk menemukan *username* dan *password* yang valid, hal ini berbeda seperti gambar diatas bahwa *Fail2ban* dalam status *Active* maka semua serangan tidak berhasil masuk kedalam *system* untuk menemukan *username* dan *password*.

3.5. Reporting

Tahap kelima atau terakhir dari metode *Penetration testing* ini yaitu *reporting*. Seperti pada Tabel 3. reporting.

Tabel 3. reporting

No	Tools	Exploitasi	Setelah Perbaikan
1	Metasploit	Sukses	Was Not Open
2	Medusa	Sukses	Failed
3	File2ban	Aktive	Block
3			

Tabel 3. Menampilkan Laporan pengujian kerentanan menghasilkan penerapan solusi terhadap jenis serangan *brute force* yang dapat ditingkatkan dari yang sebelumnya bisa mendapatkan akses masuk kedalam *system* menjadi lebih baik dan mendapatkan penolakan saat penyusup mencoba masuk ke *server* dan penutupan *port 22 ssh* dengan status *was not open*. *Metode Penetration testing* berhasil diterapkan pada system web server.

4. KESIMPULAN

Berdasarkan implementasi perbaikan fail2ban yang diterapkan pada web server terbukti dapat mencegah serangan brute force dan melakukan pemblokiran attacker. Peneliti juga menemukan beberapa kerentanan yang terdapat pada server Sistem Informasi Akademik, Jenis kelemahan dikategorikan menjadi tiga kategori yaitu 5 *level high*, 164 *level medium* dan 52 *level low*. Simulasi serangan menggunakan Parrot OS dengan menggunakan dua *tools yaitu Metasploit dan Medusa* yang menghasilkan keberhasilan masuk kedalam *system*, dengan ditemukannya *username* dan *password* maka dapat *Login* ke target. Perbaikan *system* dilakukan dengan mengkonfigurasi *File2ban* yang terdapat pada *server* dan berhasil dijalankan untuk mencegah serangan dan menggagalkan akses untuk masuk kedalam *system* dengan menutup pintu untuk penyerang. Hasil simulasi serangan menggunakan metode *Penetration Testing* dapat digunakan untuk memberikan informasi celah kerentanan, mempercepat pihak *IT* dalam mengatasi serangan secara terstruktur dan sistematis pada Sistem Informasi Akademik.

Berdasarkan perolehan data pada perbaikan web server ini telah sesuai dengan harapan yang diinginkan peneliti.

DAFTAR PUSTAKA

ABDUR RAHMAN, M., AMJAD, M., AHMED, B., & SAEED SIDDIK, M. 2020. Analyzing web application vulnerabilities: An empirical study on e-commerce sector in Bangladesh. *PervasiveHealth: Pervasive Computing Technologies for Healthcare*, 5–10. https://doi.org/10.1145/3377049.3377107

AMANKWAH, R., CHEN, J., KUDJO, P. K., & TOWEY, D. 2020. An empirical comparison of commercial and open-source web vulnerability scanners. *Software - Practice and Experience*, *50*(9), 1842–1857. https://doi.org/10.1002/spe.2870

ANGELINI, M., BLASILLI, G., CATARCI, T., LENTI, S., & SANTUCCI, G. 2019. Vulnus: Visual vulnerability analysis for network security. *IEEE Transactions on Visualization and Computer Graphics*, *25*(1), 183–192. https://doi.org/10.1109/TVCG.2018.2865028

BIN IBRAHIM, A., & KANT, S. 2018. Penetration Testing Using SQL Injection to Recognize the Vulnerable Point on Web Pages. *International Journal of Applied Engineering Research*, *13*(8), 5935–5942. http://www.ripublication.com

CHIPHER. 2020, JULY. A Complete Guide to the Phases of Penetration Testing. *2020 5th International Conference on Computer and Communication Systems, ICCCS 2020*. https://cipher.com/blog/a-complete-guide-to-the-phases-of-penetration-testing/

DIVYA, K. V., JATTI, A., JOSHI, P. R., & KRISHNA, S. D. 2019. Progress in Advanced Computing and Intelligent Engineering. In *Progress in Advanced Computing and Intelligent Engineering* (Vol. 714). Springer Singapore. https://doi.org/10.1007/978-981-13-0224-4

GEDE, S. S. A. 2020. Evaluasi Keamanan Website Lembaga X Melalui Penetration Testing Menggunakan Framework ISSAF. *Jurnal Ilmiah Merpati*, *8*(2), 113–124.

HOSSAIN, M. D., OCHIAI, H., DOUDOU, F., & KADOBAYASHI, Y. 2020. SSH and FTP brute-force attacks detection in computer networks: Lstm and machine learning approaches. *2020 5th International Conference on Computer and Communication Systems, ICCCS 2020*, 491–497. https://doi.org/10.1109/ICCCS49078.2020.9118459

IBNU MUAKHORI, SUNARDI, A. F. 2020. *Jurnal Mantik Modules Jurnal Mantik*. *3*(4), 444–450.

KHORMALI, A., PARK, J., ALASMARY, H., ANWAR, A., SAAD, M., & MOHAISEN, D. 2021. Domain name system security and privacy: A contemporary survey. *Computer Networks*, *185*, 107699. https://doi.org/10.1016/j.comnet.2020.107699

KOTHIA, A., SWAR, B., & JAAFAR, F. 2019. Knowledge Extraction and Integration for Information Gathering in Penetration Testing. *Proceedings - Companion of the 19th IEEE International Conference on Software Quality, Reliability and Security, QRS-C 2019*, 330–335. https://doi.org/10.1109/QRS-C.2019.00068

KRISHNAN, S., & WEI, M. 2019. SCADA testbed

for vulnerability assessments, penetration testing and incident forensics. *7th International Symposium on Digital Forensics and Security, ISDFS 2019*, 1–6. https://doi.org/10.1109/ISDFS.2019.8757543

LUSTICK, I. S., & TETLOCK, P. E. 2021. The simulation manifesto: The limits of brute-force empiricism in geopolitical forecasting. *Futures & Foresight Science*, *3*(2), 1–22. https://doi.org/10.1002/ffo2.64

MCKINNEL, D. R., DARGAHI, T., DEHGHANTANHA, A., & CHOO, K. K. R. 2019. A systematic literature review and meta-analysis on artificial intelligence in penetration testing and vulnerability assessment. *Computers and Electrical Engineering*, *75*, 175–188. https://doi.org/10.1016/j.compeleceng.2019.02.022

MENG, H., WOLF, M., IVIE, P., WOODARD, A., HILDRETH, M., & THAIN, D. 2015. A case study in preserving a high energy physics application with Parrot. *Journal of Physics: Conference Series*, *664*(3). https://doi.org/10.1088/1742-6596/664/3/032022

MUTEMWA, M., MTSWENI, J., & ZIMBA, L. 2019. Integrating a security operations centre with an organization's existing procedures, policies and information technology systems. *2018 International Conference on Intelligent and Innovative Computing Applications, ICONIC 2018*, *December*. https://doi.org/10.1109/ICONIC.2018.8601251

POHAN, Y. A., YUNUS, Y., & SUMIJAN, S. 2020. Improving Webserver Security for Local Tax Reporting Applications Using Standard Penetration Testing Execution Methods. *Jurnal Sistim Informasi Dan Teknologi*, *3*, 7–10. https://doi.org/10.37034/jsisfotek.v3i1.83

PRAMADITYA, H. 2016. Brute Force Password Cracking Dengan Menggunakan Graphic Processing Power. *Jurnal Teknologi Dan Manajemen Informatika*, *2*(1). https://doi.org/10.26905/jtmi.v2i1.615

PRASETYO, K. A., IDHOM, M., & WAHANANI, H. E. 2020. *Pada Multiple Server Dengan Menggunakan*. *1*(3), 789–796.

RAPLEY, A., BELLEKENS, X., SHEPHERD, L. A., & MCLEAN, C. 2018. Mayall: A framework for desktop javascript auditing and post-exploitation analysis. *Informatics*, *5*(4). https://doi.org/10.3390/informatics5040046

RIADI, I., YUDHANA, A., & W, Y. 2020. Analisis Keamanan Website Open Journal System Menggunakan Metode Vulnerability Assessment. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, *7*(4), 853. https://doi.org/10.25126/jtiik.2020701928

SADASIVAM, G. K., HOTA, C., & ANAND, B. 2018. Towards Extensible and Adaptable Methods in Computing. In *Towards Extensible and Adaptable Methods in Computing*. Springer Singapore. https://doi.org/10.1007/978-981-13-2348-5

SADIGH, D., LANDOLFI, N., SASTRY, S. S., SESHIA, S. A., & DRAGAN, A. D. 2018. Planning for cars that coordinate with people: leveraging effects on human actions for planning and active information gathering over human internal state. *Autonomous Robots*, *42*(7), 1405–1426. https://doi.org/10.1007/s10514-018-9746-1

SANDRA, S., STIAWAN, D., & HERYANTO, A. 2016. Visualisasi Serangan Brute Force Menggunakan Metode K-Means dan Naïve Bayes. *Proceeding - Annual Research Seminar Proceeding*, *2*(1), 315–320.

SEEMA, R., & RITU, N. 2019. Penetration Testing Using Metasploit Framework : an Ethical Approach. *International Research Journal of Engineering and Technology(IRJET)*, *06*(08), 538–542. https://doi.org/2395-0056

STIAWAN, D., IDRIS, M. Y., ABDULLAH, A. H., ALQURASHI, M., & BUDIARTO, R. 2016. Penetration testing and mitigation of vulnerabilities windows server. *International Journal of Network Security*, *18*(3), 501–513. http://joiv.org/index.php/joiv/article/view/190

UDJAJA, Y. 2018. EKSPANPIXEL BLADSY STRANICA: Performance Efficiency Improvement of Making Front-End Website Using Computer Aided Software Engineering Tool. *Procedia Computer Science*, *135*, 292–301. https://doi.org/10.1016/j.procs.2018.08.177

WU, K. T., CHOU, S. H., CHEN, S. W., TSAI, C. T., & YUAN, S. M. 2018. Application of machine learning to identify Counterfeit Website. *ACM International Conference Proceeding Series*, 321–324. Https://doi.org/10.1145/3282373.3282407

ZEEBAREE, S. R. M., JACKSI, K., & ZEBARI, R. R. 2020. Impact analysis of SYN flood DDoS attack on HAProxy and NLB cluster-based web servers. *Indonesian Journal of Electrical Engineering and Computer Science*, *19*(1), 505–512. https://doi.org/10.11591/ijeecs.v19.i1.pp505-512.