Integrasi End-point Security Berbasis Agent dan Bot Messenger untuk Deteksi dan Monitoring Serangan pada Web Server secara Real-time

Muhammad Alfian Fahrudi[1], I Made Suartana[2]

[1,3] Jurusan Teknik Informatika Fakultas Teknik Universitas Negeri Surabaya
[1]muhammadalfian.18045@mhs.unesa.ac.id
[2]imadesuartana@unesa.ac.id

***Abstrak*— Perkembangan teknologi mempengaruhi perusahaan atau instansi untuk memaksimalkan kinerjanya. Perusahaan menggunakan media internet untuk memberikan informasi, layanan, dan menyimpan data melalui web server. Mudahnya mendapatkan informasi pada media internet menimbulkan kejahatan siber dalam upaya untuk mengambil data pada web server. Salah satu pihak yang menangani dan melindungi keamanan jaringan sebuah perusahaan yaitu *Security Operation Center (SOC).* sangat berperan penting dalam kondisi ini. Pada penelitian ini mengusulkan sebuah sistem integrasi antara *end-point security* dengan *bot messenger* Telegram. Sistem integrasi akan membantu pengeluaran finansial perusahaan dan membantu kinerja SOC dalam memantau web server. Wazuh sebagai aplikasi end-point security yang diintegrasikan dengan bot Telegram. Wazuh merupakan aplikasi open source yang didirikan pada tahun 2015. Sistem integrasi Wazuh dengan bot Telegram mampu mengirim pesan dengan format penulisan sesuai kondisi aktivitas yang terjadi pada web server. Sistem integrasi juga mampu mengurangi pesan yang terkirim ketika terjadi aktivitas yang sama secara terus-menerus. Berdasarkan hasil penelitian yang dilakukan dapat disimpulkan bahwa integrasi sistem monitoring Wazuh dengan bot messenger Telegram berhasil mengirim pesan secara real-time.**

***Kata Kunci*— *End-point Security, Wazuh, Monitoring Server, Integration, Bot Messenger Telegram*.**

I. PENDAHULUAN

Perkembangan teknologi saat ini mengalami peningkatan yang cukup pesat. Menganalisis tren pasar saat ini dengan menyebarkan informasi atau data melalui internet dapat meningkatkan peluang sukses dalam bisnis. [1]. Perusahaan atau instansi memanfaatkan perkembangan teknologi untuk memaksimalkan kinerjanya agar dapat mencapai tujuan tertentu. Perusahan juga menggunakan media internet sebagai media penyimpanan yang mudah diakses dan tidak membutuhkan banyak biaya. Perusahaan mulai menggunakan *web server* sebagai media internet untuk memberikan informasi, memberi layanan, dan menyimpan data. Mudahnya akses dalam mendapat informasi menimbulkan masalah baru yaitu dapat dimanfaatkan data atau informasi penting yang dilakukan oleh pihak yang tidak bertanggung jawab. Data atau informasi penting yang sering didapatkan oleh peretas melalui *web server* yang memiliki celah keamanan yang cukup tinggi. Pentingnya memperkuat keamanan jaringan untuk mencegah penyalahgunaan data secara ilegal. [2]

Keamanan *web server* merupakan faktor penting yang harus diperhatikan dalam perancangan sebuah *website* maupun menyimpan data atau informasi. Infrastruktur jaringan yang mendukung dalam memainkan peran penting keamanan *web server* yaitu firewall, router, dan sistem deteksi intrusi.[3] Namun banyak developer *website* yang masih kurang teliti dalam meningkatkan keamanan *web server*, hal ini dapat memicu terjadinya tindak kejahatan *cyber* oleh orang yang tidak bertanggung jawab. Keamanan *web server* dapat diartikan juga seperti upaya untuk monitoring dan mencegah perilaku mencurigakan pada *web server* yang ilegal ataupun serangan yang berasal dari eksternal maupun internal. [4] Selain itu developer *website* juga harus mengetahui jenis serangan terbaru agar mereka dapat mempertahankan dan memperbaiki *website* mereka dari hal-hal yang tidak diinginkan. Data Breach Investigations Report (DBIR) 2021 dari Verizon mencatat, sekitar 29.207 insiden kasus kebocoran data di seluruh dunia selama tahun lalu. Laporan ini mencatat, industri hiburan merupakan industri yang paling sering alami kebocoran data pada 2021. Jumlahnya mencapai 7.065. Walaupun jumlah kasus turun dari kasus tahun sebelumnya developer *website* tetap harus waspada. Melakukan penilaian pada kerentanan *website* dengan mempertimbangkan perbedaan faktor – faktor yang terkait dengan *website* akan memberikan penjelasan yang lebih dan mengutamakan untuk mengamankan *website* lebih baik lagi. [5].

Saat ini, terdapat berbagai aplikasi monitoring dan keamanan *web server* apabila ada suatu aktivitas mencurigakan yang sedang terjadi. Pemilihan aplikasi yang tepat mempengaruhi pada proses peningkatan keamanan *web server*. Pengumpulan log dilakukan untuk mengetahui dan memperbaiki celah keamanan yang dapat dieksploitasi oleh orang yang tidak bertanggung jawab. Semakin bertambahnya pengguna internet namun tidak diimbangi dengan adanya administrator jaringan yang handal di bidangnya maka ancaman – ancaman kejahatan *cyber* tetap akan muncul. [6]. *IDS* (*Intrusion Detection System*) adalah sistem yang memantau lalu lintas jaringan untuk aktivitas yang mencurigakan dan mengeluarkan peringatan ketika aktivitas tersebut ditemukan. *IDS* akan memberikan *alert* jika terdapat serangan dan memungkinkan untuk memblokir alamat *IP* agar tidak dapat mengakses kembali server yang diserang. [7]. Terdapat dua metode yang digunakan dalam *IDS* yaitu *HIDS* dan *NIDS*. *HIDS* (*Host-based Intrusion Detection System*) memiliki keunggulan dalam mendeteksi dan merespon serangan jangka panjang. *HIDS* juga dapat melakukan pemeriksaan sistem seperti *log analysis*, *file integrity checking*, *rootkit detection*, *active response* dan *registry monitoring*

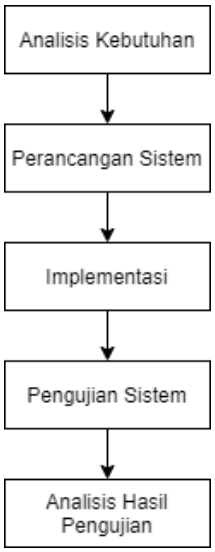
apabila pada *host* sudah terpasang aplikasi IDS. Monitoring merupakan kegiatan untuk mengidentifikasi masalah dan melakukan manajemen serta penilaian pola kerja untuk mencapai tujuan. [8]. Monitoring yang dilakukan untuk mengetahui *alert* yang diterima dari setiap *agent* (*end-point*). *End-point* merupakan komunikasi jarak jauh melalui protokol *HTTP* dengan aplikasi yang sudah di host. Tindakan monitoring diharuskan berada ditempat *security server* agar dapat mengetahui kendala yang sedang dialami oleh setiap *end-point* secara *real-time*. Namun, beberapa perusahaan memiliki sedikit orang yang bertugas untuk melakukan monitoring dan berakibat tidak bisa berada di tempat *security server* secara maksimal, sehingga jika terdapat kendala pada *end-point* akan membutuhkan waktu cukup lama untuk menganalisa sumber masalah. Oleh karena itu dibutuhkan sebuah aplikasi monitoring yang dapat diintegrasikan dan mengirim pesan secara langsung saat *end-point* sedang mengalami kendala.

Aplikasi *open source* untuk monitoring *web server* tersedia cukup banyak, salah satunya wazuh. Wazuh merupakan aplikasi yang memiliki sistem *HIDS* yang memiliki banyak fitur dibandingkan dari aplikasi *open source* sejenisnya. Wazuh didirikan pada tahun 2015 oleh Santiago Basset sebagai cabang dari *OSSEC*, dan saat ini memiliki lebih dari 10.000 pengguna. Beberapa penelitian yang menggunakan wazuh antara lain: *Intrusion Detection and Anomaly* Menggunakan Wazuh Pada Universitas Muhammadiyah Palembang [9]. Wazuh Sebagai Log Event Management Dan Deteksi Celah Keamanan Pada Server Dari Serangan Dos [10]. Implementasi Wazuh 4.0 Untuk Perlindungan Keamanan Integritas File [11]. Sedangkan untuk penggunaan media sosial seperti Telegram mengalami peningkatan yang cukup pesat, hal ini banyak dimanfaatkan sebagai mekanisme untuk menyampaikan pesan secara *real-time* termasuk *alert* dalam monitoring jaringan. Telegram merupakan aplikasi layanan pengirim pesan gratis yang cukup ringan dan memiliki banyak fitur yang menarik. Telegram tidak hanya sebagai media komunikasi antar pengguna aplikasi Telegram, namun Telegram juga memiliki fitur Telegram *bot* API yang berfungsi sebagai media komunikasi antara mesin dengan pengguna aplikasi Telegram. [12].

Tujuan penelitian ini membuat sistem deteksi, monitoring dan reporting yang terintegrasi sehingga bisa melakukan monitoring pada *end-point* dan melakukan *reporting* secara *real-time*. Dengan sistem integrasi antara Wazuh dengan Telegram diharapkan dapat membantu dalam mengawasi *web server* dan dapat dengan cepat untuk mengetahui sumber masalah ketika *end-point* sedang terjadi gangguan atau masalah.

II. METODOLOGI PENELITIAN

Metode yang diterapkan pada penelitian ini adalah metode *experimental design*. Penerapan metode bertujuan untuk mengintegrasi Wazuh sebagai *end-point security* dengan *bot messenger* pada Telegram yang akan difungsikan sebagai alat monitoring. Berikut merupakan alur tahapan yang dilakukan dalam penelitian ini:



Gambar. 1 Diagram Alur Metode Penelitian

A. Analisis Kebutuhan

Analisis Kebutuhan merupakan tahap pertama dalam penelitian yang dilakukan untuk menentukan detail kebutuhan agar penelitian ini berjalan sesuai tujuan. Berikut merupakan kebutuhan yang dibagi menjadi beberapa bagian, yaitu :

1. Kebutuhan Data

   Penggunaan data dalam penelitian ini diambil dari beberapa sumber referensi, yaitu:

   a. Studi Literatur

      Pada penelitian ini referensi yang diambil dari jurnal nasional, jurnal international, makalah, buku, video dari youtube, web resmi dan sumber lainnya.

   b. Observasi

      Penelitian ini juga melakukan observasi terhadap beberapa website referensi tentang integrasi, monitoring server, dan bot messenger.

2. Kebutuhan Alat

   Kebutuhan perangkat yang digunakan dalam penelitian ini terbagi menjadi dua jenis, yaitu:

   a. Kebutuhan perangkat keras (Hardware)

      Perangkat keras yang diperlukan guna tujuan penelitian yaitu Laptop sebagai uji coba dengan spesifikasi berikut :

      1) Processor Intel Core i5-7200U
      2) RAM 12 GB
      3) Harddisk 1 TB
      4) Sistem Operasi Windows 10 Pro 64-bit

   b. Kebutuhan perangkat lunak (Software)

      Perangkat lunak berfungsi untuk pengoperasian sistem pada penelitian ini. Pada penelitian ini virtual machine yang digunakan adalah *VMware Workstation.*

Tabel. 1 Kebutuhan Perangkat Lunak

No.	Nama Perangkat Lunak	Keterangan
1.	Sistem Operasi Ubuntu Server	Digunakan sebagai tempat Wazuh server
2.	VMware	Aplikasi yang dipergunakan untuk memenuhi kebutuhan banyak sistem operasi
3.	Wazuh	Aplikasi yang dipergunakan untuk monitoring security dalam server
4.	Telegram	Sebuah social media yang digunakan untuk menampilkan notifikasi *alert* menggunakan API
5.	MobaXterm	Aplikasi yang dipergunakan untuk menghubungkan dengan server melalui SSH server
6.	Sistem Operasi Fedora Server	Digunakan sebagai tempat *web server*

B. *Perancangan Sistem*

Perancangan sistem dilakukan untuk membuat desain perencanaan arsitektur sistem yang akan dibangun dapat berjalan sesuai dengan tujuan penelitian.

1. Perancangan Monitoring Server

   Monitoring server merupakan sistem yang digunakan untuk memantau aktivitas yang terjadi pada setiap *end-point*. Perancangan monitoring server pada penelitian ini menggunakan dua *web server* sebagai *end-point* yang akan di monitor oleh wazuh dan satu perangkat *penetration testing*.

2. Perancangan Integrasi Bot Messenger

   Integrasi dengan *bot messenger* berfungsi untuk menampilkan *alert* hasil analisis dari data wazuh pada Telegram. Sistem integrasi pada penelitian ini menggunakan *API* yang disediakan pada *bot* Telegram agar dapat menghubungkan antara Wazuh dengan Telegram.

C. *Implementasi Sistem*

Pada tahap ini hasil dari rancangan akan diimplementasikan sesuai alur kerja sistem yang telah dibuat pada lingkungan virtual agar lebih efisien. Instalasi dan konfigurasi dilakukan sesuai referensi dokumentasi dari masing-masing *software* yang digunakan.

D. *Pengujian Sistem*

Pada tahap ini hasil dari implementasi sistem akan diuji sesuai skenario pengujian yang telah ditentukan. Parameter pengujian yang akan dilakukan adalah menguji *log event* dari *agent* yang diterima *end-point security* dan dapat dijadikan sebagai *alert notification* pada *bot messenger* yang telah diintegrasikan.

E. *Analisis Hasil Kebutuhan*

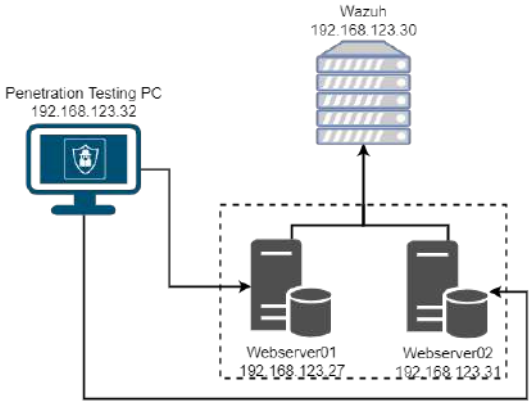
Kesimpulan yang diambil dari hasil pengujian sistem dapat menjadi data untuk dijadikan sebagai kualitas hasil pengujian. Apabila terdapat kendala pada pengujian sistem maka akan langsung untuk mencari solusi agar kendala dapat teratasi.

III. HASIL DAN PEMBAHASAN

Hasil penelitian dan pembahasan penelitian menjelaskan tentang hasil implementasi sistem yang dirancang berdasarkan masalah dan tujuan penelitian yang telah dirumuskan.

A. *Implementasi Sistem*

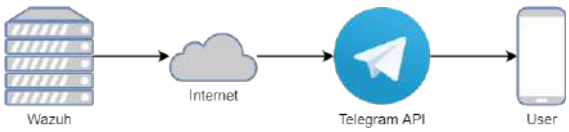
1. Deployment Wazuh in Server



Gambar. 2 Sistem Monitoring Server

Aplikasi monitoring yang digunakan pada penelitian ini adalah Wazuh. Wazuh ditugaskan untuk memantau aktivitas yang terjadi pada web server. Rancangan sistem yang dibuat seperti pada gambar 2 dimana server Wazuh memonitoring aktivitas yang terjadi pada 2 web server. Aktivitas yang terjadi pada web server akibat dari serangan yang dilakukan dari *Penetration Testing PC.* Pada web server terinstal Wazuh Agent yang berfungsi untuk membantu mengirimkan log atau aktivitas ke server Wazuh.

2. Integrasi Bot Telegram



Gambar. 3 Sistem Integrasi Bot Messenger

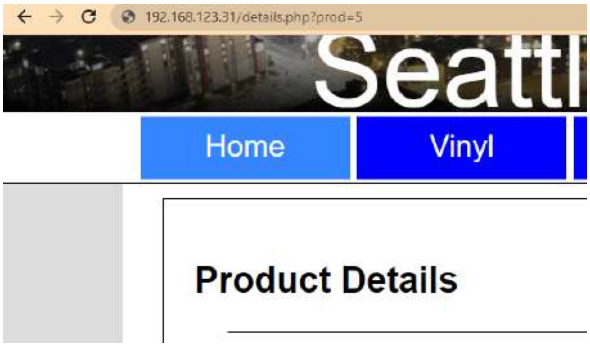
Pada gambar 3 Wazuh mengirimkan deskripsi *alert* menuju Telegram dengan bantuan pemrograman python. *Alert* dari wazuh dikirim ke bot messenger melalui Telegram API yang di setting pada python

script. Selanjutnya akan muncul deskripsi *alert* pada Telegram *user*.

B. *Uji Coba Sistem*

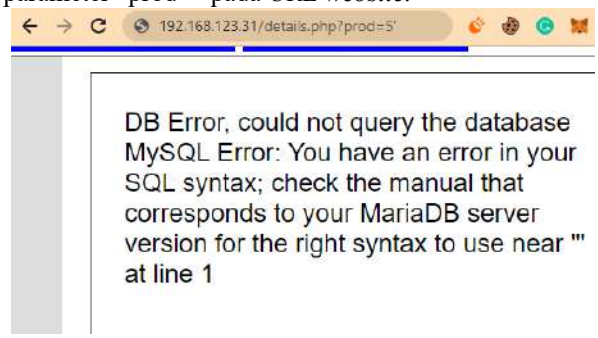
1. Vulnerability Assessment

Melakukan analisis kerentanan terhadap website maupun server merupakan tindakan yang harus dilakukan ketika ingin menguji dengan beberapa metode serangan. Pada penelitian ini analisis kerentanan dilakukan pada *website* dan *port* pada web server.



Gambar. 4 parameter pada URL

Analisis kerentanan website dilakukan pada parameter *URL website*. Bermacam jenis serangan dilakukan melalui parameter dengan celah keamanan yang cukup tinggi. Seperti pada gambar 4 terdapat parameter "prod=" pada *URL website.*



Gambar. 5 identifikasi kerentanan

Memuat kembali URL dengan penambahan simbol petik satu setelah parameter. Seperti pada gambar 5 setelah dimuat kembali website akan muncul pesan error yang dimana merupakan ciri dari website yang memiliki kerentanan.

Gambar. 6 mencari celah kolom database

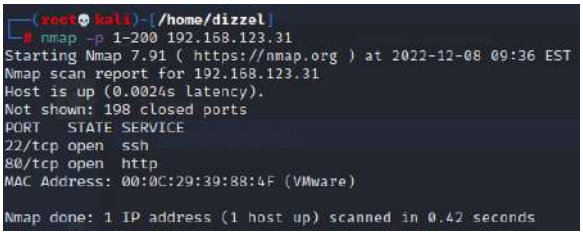
Mencari celah kolom pada *database* dapat dilakukan dengan mengganti simbol petik satu dengan *script* "order+by+1+--+". Seperti pada gambar 6 setelah dimuat ulang maka tampilan website akan kembali seperti semula.

Gambar. 7 tampilan database error

Mengubah angka 1 pada *script* dengan mengurutkan angka normal untuk mendapatkan total kolom yang terdapat pada *database*. Seperti pada gambar 7 pada angka 6 website menampilkan pesan error yang berarti jumlah kolom pada website kurang dari 6.

Gambar. 8 scanning port

Selanjutnya menganalisis kerentanan pada web server. Analisis pada server untuk mengetahui port yang terbuka. Menggunakan *tools* Nmap dalam scanning port yang sedang terbuka pada IP web server. Seperti pada gambar 8 *scanning port* dilakukan dari 1 sampai 200 dengan menggunakan fungsi -p pada Nmap.

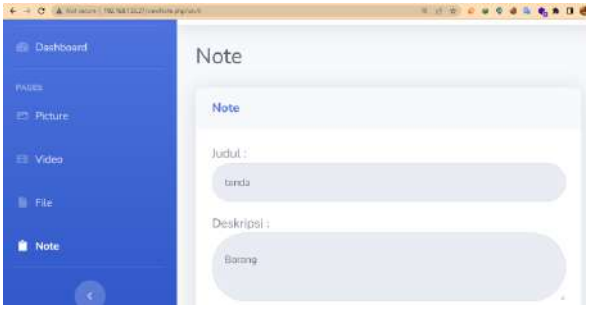


Gambar. 9 status port

Setelah proses *scanning port* selesai, maka akan muncul *port* yang sedang terbuka. Seperti pada gambar 9 port 22 pada 192.168.123.31 terbuka untuk *SSH*.

2. Metode Serangan *Injection*

Pada penelitian ini metode serangan *injection* dilakukan pada *URL webpage*. Serangan yang dilakukan pada penelitian ini bertujuan untuk mencari kerentanan terhadap *database* pada web server.



Gambar.10 Webpage note web server 01

Pada gambar 10 merupakan tampilan *web page note* dari web server 01. *URL* pada *web page* produk adalah *http://192.168.123.27/viewNote.php?id=9*.

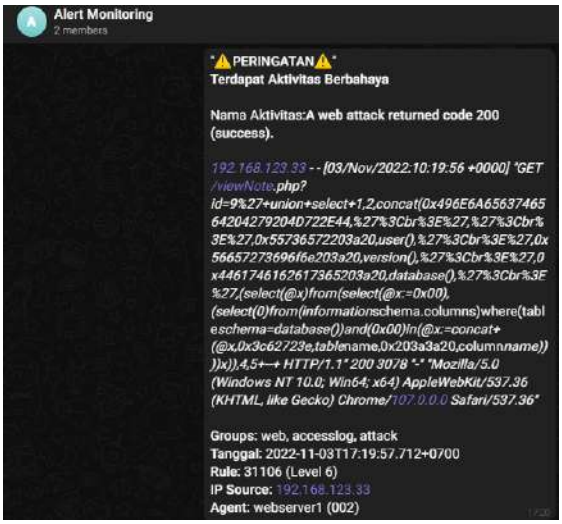
Tabel. 2 Sql injection script

```
http://192.168.123.27/viewNote.php
?id=9%27+union+select+1,2,concat(0
x496E6A6563746564204279204D722E44,
%27%3Cbr%3E%27,%27%3Cbr%3E%27,0x55
736572203a20,user(),%27%3Cbr%3E%27
,0x56657273696f6e203a20,version(),
%27%3Cbr%3E%27,0x44617461626173652
03a20,database(),%27%3Cbr%3E%27,(s
elect(@x)from(select(@x:=0x00),(se
lect(0)from(information_schema.col
umns)where(table_schema=database()
)and(0x00)in(@x:=concat+(@x,0x3c62
723e,table_name,0x203a3a20,column_
name))))x)),4,5+--+
```

Menggunakan *payloads SQL injection* pada *URL webpage* dilakukan setelah parameter *id=*. Pada tabel 2 merupakan *script SQL injection* untuk membuka daftar tabel yang tersedia di *database*.

Gambar.11 Data tabel database pada web page note

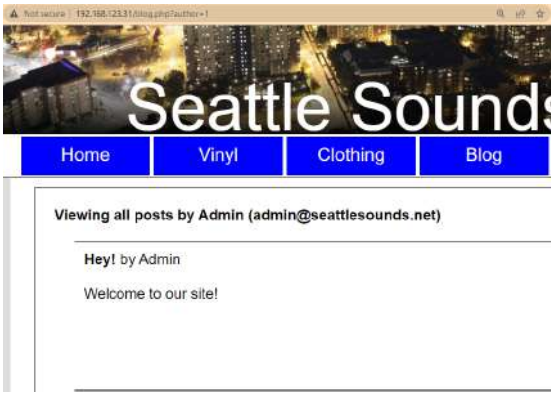
Setelah berhasil membuka *URL* dengan tambahan *payload SQL injection*, maka pada *webpage* akan muncul nama dari setiap tabel pada *database*. Seperti pada gambar 11 merupakan tampilan *webpage* setelah berhasil dilakukan serangan *injection*.



Gambar.12 Pesan bot hasil serangan SQL injection

Bot yang telah diintegrasikan akan merespon dengan mengirimkan pesan ke Telegram. Seperti pada gambar 12 *bot* mengirim pesan sesuai dengan *alert* yang diterima Wazuh dan telah dirubah sesuai dengan format pesan pada *source code* custom-telegram.py.

Selanjutnya jenis serangan injection yang akan digunakan adalah *XSS* (*Cross Site Scripting*). Jenis serangan *XSS* sering digunakan untuk mendapatkan *cookie* dari *user* lain atau mengirimkan program yang dapat merusak *user*. Serangan *XSS* akan diuji pada *website* dari web server 02.



Gambar.13 Webpage blog pada web server 02

Pada gambar 13 merupakan webpage dari blog dengan *URL http:192.168.123.31/blog.php?author=1*. Penerapan XSS dilakukan pada *parameter author=*.

Tabel.3 Script XSS injection

http://192.168.123.31/blog.php?author"><script>alert("Injection")</script>

Script pada tahap ini bertujuan untuk mengetahui apakah terdapat celah *XSS injection* pada *website*. Penambahan fungsi alert untuk memunculkan pesan berbentuk jendela dialog (*pop-up*) seperti pada tabel 3.

Gambar.14 Pop-up hasil XSS injection

Jendela dialog akan muncul ketika *URL* pada *web page* yang telah ditambahkan dengan *script XSS* dimuat kembali. Seperti pada gambar 14 jendela dialog terdapat pesan *"Injection"* sesuai dengan script XSS yang digunakan.

Gambar.15 Pesan bot hasil serangan XSS injection

Pada gambar 15 terdapat pesan dari hasil serangan *XSS injection.* Bot yang telah dibuat berhasil mengirim pesan dengan rinci beserta dengan *script XSS injection* yang digunakan.

Tabel.4 Tabel pengujian serangan SQL injection

No.	Deskripsi Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	Membuat serangan Injection pada web server	Integrasi berhasil dan alert dikirim ke bot Telegram	Sesuai Harapan	Normal
2.	Menggunakan script SQL Injection	Script berhasil muncul pada format pesan	Sesuai harapan	Normal
3.	Menggunakan script XSS Injection	Script berhasil muncul pada format pesan	Sesuai harapan	Normal

Pada penelitian ini proses untuk mengetahui hasil integrasi berjalan sesuai dengan fungsinya menggunakan metode *blackbox*. Pada tabel 4 dapat diketahui hasil dari pengujian terhadap serangan *injection* pada web server berjalan normal dan format penulisan sesuai dengan tujuan penelitian.

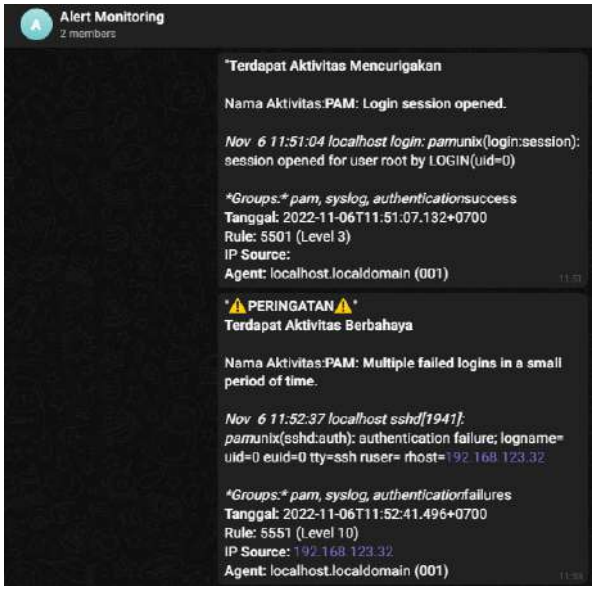
3. Metode Serangan Identification and Authentication Failures

Serangan yang dipakai dalam metode *identification and authentication failures* yaitu *brute force*. Serangan *brute force* merupakan metode peretasan untuk mengetahui kata sandi, direktori, dan kunci enkripsi.

Pada pengujian *brute force*, peneliti menerapkan serangan pada akses *SSH* server. *Brute force* dilakukan dengan bantuan *tools* Nmap. Nmap adalah *tools* yang sering digunakan untuk eksplorasi dan audit keamanan jaringan.

Gambar.16 Command brute force SSH web server 02

Pengujian dilakukan pada web server 02 dengan *IP* 192.168.123.31 pada *port* 22. Pada gambar 16 serangan *brute force* menggunakan *word list user* dan *password* acak. *Word list* disimpan pada file users10.txt dan passwords10.txt. Setiap file *word list* berisi 10 kata yang sering digunakan sebagai *user* dan *password*.



Gambar.17 Pesan bot hasil serangan brute force SSH

Seperti pada gambar 17 merupakan pesan hasil dari serangan *brute force*. Pesan yang berisi aktivitas *PAM Multiple failed logins in a small period of time* akan muncul ketika terjadi gagal login sebanyak 25 kali.

Tabel.5 Pengujian brute force SSH

No.	Jumlah Kombinasi User dan Password	Jumlah Alert	Waktu Telegram pada pesan terakhir
1..	50	100 Request	1 menit
2.	500	1000 Request	11 menit
3.	3000	6000 Request	50 menit

Pengujian juga dilakukan dengan kombinasi jumlah *word list* sampai 3000. Seperti pada tabel 5 jeda waktu pengiriman *alert* cukup bervarian. Dengan jumlah kombinasi *word list* 3000 didapatkan jeda waktu pengiriman selama 50 menit. Jeda waktu diakibatkan dari performa perangkat peneliti yang terbagi pada sistem *Virtual Machine*.

Tabel. 6 Keterangan jenis serangan penetration testing PC

	Jenis Serangan		
	Pentest PC1	Pentest PC2	Pentest PC3
Pengujian 1	Sql Injection	Sql Injection	Sql Injection
Pengujian 2	Bruteforce	Sql Injection	Sql Injection
Pengujian 3	Bruteforce	Bruteforce	Sql Injection
Pengujian 4	Bruteforce	Bruteforce	Bruteforce

Pada penelitian ini melakukan pengujian dengan menerima serangan secara bersamaan dari 3 *penetration testing* PC. Seperti pada tabel 6 merupakan jenis serangan yang dilakukan dari setiap *penetration testing* PC untuk masing-masing pengujian.

Tabel.7 Pengujian 3 penetration testing PC

No.	Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1.	1,2,3&4	Format pesan muncul dengan IP penetration testing PC.	Sesuai Harapan	Normal
2.	1,2,3&4	Semua pesan terkirim	Sesuai Harapan	Normal
3.	1,2,3&4	Jeda waktu ± 1 menit	Jeda waktu lebih dari 10 menit	Tidak Sesuai Harapan

Pengujian mendapatkan hasil yang variatif. Kendala yang didapatkan dari tabel 7 akibat dari penerimaan bot untuk memproses setiap *alert* yang diterima.

IV. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan peneliti berhasil menerapkan sistem monitoring security web server berbasis *agent* yang diintegrasikan dengan *bot messenger*. Kesimpulan yang diperoleh berdasarkan rumusan masalah, pembahasan, pengujian, dan hasil penelitian bahwa. Metode integrasi yang diterapkan pada sistem berhasil mengirim pesan dengan format sesuai kondisi *alert* yang ditentukan dan penggunaan kondisi dalam *source code* berhasil membatasi pengiriman pesan serangan *brute force* menggunakan penyimpanan *temporary data* untuk menyimpan batas perulangan. Pengembangan penelitian yang serupa kedepannya pada rules yang tersedia pada Wazuh dan memisahkan perangkat server untuk meminimalisir jeda waktu yang terkirim ke Telegram.

REFERENSI

[1] Prismana, I. G. L. P. E., Prehanto, D. R., & Nuryana, I. K. D. (2020, November). The Design and Implementation of Web Crawler Distributed News Domain Detection System. In *International Joint Conference on Science and Engineering (IJCSE 2020)* (pp. 92-97). Atlantis Press.

[2] Suartana, I. M., Putra, R. E., Bisma, R., & Prapanca, A. (2022). Pengenalan Pentingnya *Cyber* Security Awareness pada UMKM. *Jurnal Abadimas Adi Buana*, *5*(02), 197-204.

[3] Nam H Nguyen. (2018). Buku Panduan Keamanan Cyber Penting Di Bahasa Indonesia: Essential Cyber Security Handbook In Indonesian.

[4] Marta, I. K. K. A., Hartawan, I. N. B., & Satwika, I. K. S. (2020). Analisis Sistem Monitoring Keamanan Server Dengan Sms *Alert* Berbasis Snort. *INSERT: Information System and Emerging Technology Journal*, *1*(1), 25-40.

[5] Elanda, A., & Buana, R. L. (2020). Analisis Keamanan Sistem Informasi Berbasis *Website* Dengan Metode Open Web Application Security Project (OWASP) Versi 4: Systematic Review. *CESS (Journal of Computer Engineering, System and Science)*, *5*(2), 185-191.

[6] Maulana, S. A. (2021). Analisis Keamanan *Website* dengan Information System Security Assessment Framework (Issaf) dan Open

Web Application Security Project (Owasp) di Rumah Sakit Xyz. *Jurnal Indonesia Sosial Teknologi*, *2*(4), 506-519.

[7] Syani, M. (2019). Analisis Dan Implementasi Network Security System Menggunakan Teknik Host-Based Intrusion Detection System (Hids) Berbasis Cloud Computing.

[8] Megawaty, D. A. (2020). Sistem Monitoring Kegiatan Akademik Siswa Menggunakan *Website*. *Jurnal Tekno Kompak*, *14*(2), 98-101

[9] Harahap, A. G. S., & Hutrianto, H. (2021, November). INTRUSION DETECTION AND ANOMALY MENGGUNAKAN WAZUH PADA UNIVERSITAS MUHAMMADIYAH PALEMBANG. In *Bina Darma Conference on Computer Science (BDCCS)* (Vol. 3, No. 2, pp. 324-328).

[10] Nova, F., Pratama, M. D., & Prayama, D. (2022). Wazuh sebagai Log Event Management dan Deteksi Celah Keamanan pada Server dari Serangan Dos. *JITSI: Jurnal Ilmiah Teknologi Sistem Informasi*, *3*(1), 1-7.

[11] Laksmiati, D. (2021). IMPLEMENTASI WAZUH 4.0 UNTUK PERLINDUNGAN KEAMANAN INTEGRITAS FILE. *Jurnal Akrab Juara*, *6*(3), 164-174.

[12] Pradana, D. O. (2020). Implementasi Notifikasi Menggunakan Telegram Messenger Pada Software The Dude Network Monitoring. *Jurnal Manajemen Informatika*, *11*(1).