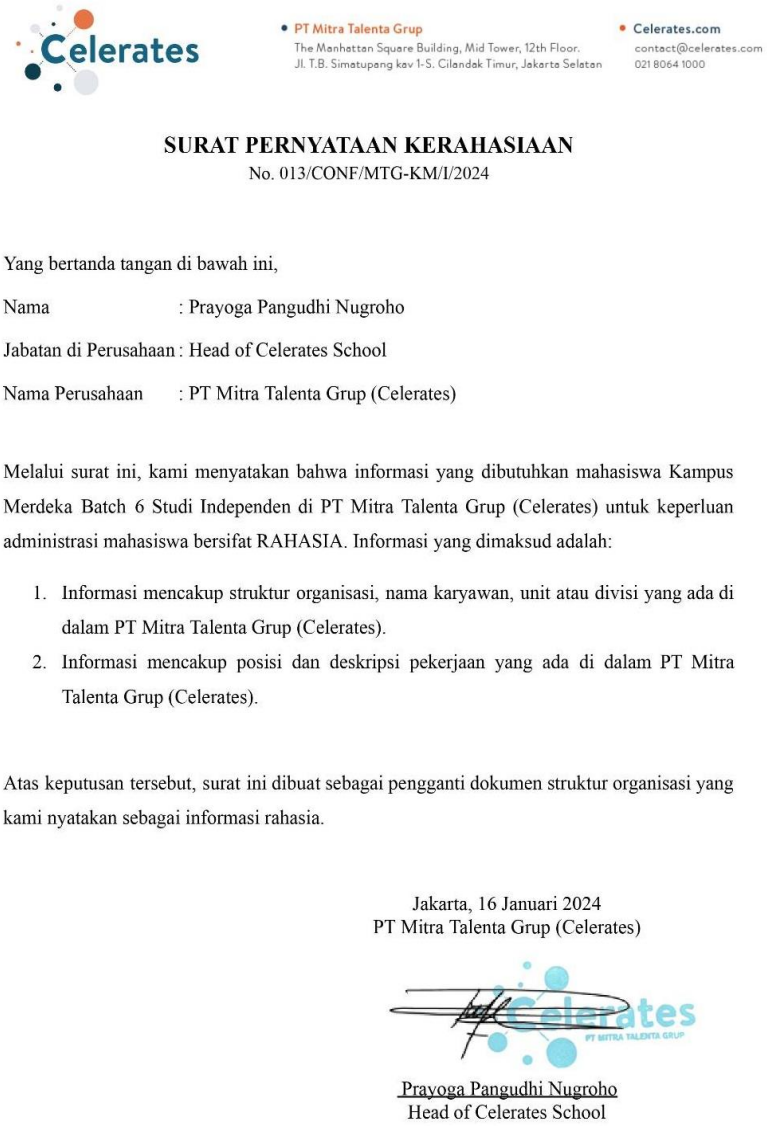
BAB I
GAMBARAN UMUM

1.1 Profil Perusahaan

Informasi yang mencakup struktur organisasi, nama karyawan, divisi, posisi dan deskripsi pekerjaan yang ada di dalam PT. Mitra Talenta Group (Celerates) adalah rahasia. Seperti yang ditunjukkan pada gambar 1.1 bahwa surat tersebut dibuat sebagai pengganti dokumen struktur organisasi.



Celerates

PT Mitra Talenta Grup
The Manhattan Square Building, Mid Tower, 12th Floor.
Jl. T.B. Simatupang kav 1-S. Cilandak Timur, Jakarta Selatan

Celerates.com
contact@celerates.com
021 8064 1000

SURAT PERNYATAAN KERAHASIAAN
No. 013/CONF/MTG-KM/I/2024

Yang bertanda tangan di bawah ini,

Nama : Prayoga Pangudhi Nugroho

Jabatan di Perusahaan : Head of Celerates School

Nama Perusahaan : PT Mitra Talenta Grup (Celerates)

Melalui surat ini, kami menyatakan bahwa informasi yang dibutuhkan mahasiswa Kampus Merdeka Batch 6 Studi Independen di PT Mitra Talenta Grup (Celerates) untuk keperluan administrasi mahasiswa bersifat RAHASIA. Informasi yang dimaksud adalah:

1. Informasi mencakup struktur organisasi, nama karyawan, unit atau divisi yang ada di dalam PT Mitra Talenta Grup (Celerates).
2. Informasi mencakup posisi dan deskripsi pekerjaan yang ada di dalam PT Mitra Talenta Grup (Celerates).

Atas keputusan tersebut, surat ini dibuat sebagai pengganti dokumen struktur organisasi yang kami nyatakan sebagai informasi rahasia.

Jakarta, 16 Januari 2024
PT Mitra Talenta Grup (Celerates)

Celerates
PT MITRA TALENTA GRUP

Prayoga Pangudhi Nugroho
Head of Celerates School

Gambar 1. 1 Surat pernyataan kerahasiaan

Nama Perusahaan	:	PT. Mitra Talenta Grup
Lokasi Perusahaan	:	The Manhattan Square Mid Tower 12th Floor. Jl. TB Simatupang Kav 1-S, Jakarta, 12560.
Tipe Industri	:	Tekno*log*i Informasi
Profil Mitra	:	Celerates merupakan wadah pelatihan dan akselerasi karir yang fokus pada bidang *Software Development, Data Science, Business Intelligence,* dan *Data Engineer*. Sejak berdiri pada tahun 2013 di bawah naungan Cybertrend Intrabuana selaku partner Tableau di Indonesia, Celerates memiliki tenaga pengajar dengan pengalaman lebih dari 20 tahun di bidang Data dan *Software Development*.

1.2 Deskripsi Kegiatan

Posisi	:	Celerates *Acceleration Program - Cyber security*
Deskripsi	:	Program ini dibentuk untuk memberikan pengalaman praktis hingga tingkat lanjut dalam melindungi jaringan dari serangan peretas. Program ini juga menekankan pada konsep pendampingan atau *mentorship* yang terdiri dari dua jenis *mentoring*. Yang pertama ada *Individual mentoring* dan *Group mentoring*. *Individual mentoring* berfokus pada kinerja individu, keterampilan, motivasi pribadi, strategi belajar, dan panduan masing-masing individu. *Grouping mentor* fokus pada kinerja para kelompok, panduan penelitian, manajemen *project*. *Mentoring* dilakukan setiap hari Senin hingga Jumat pada pukul 09.00-13.00, dengan

tambahan pendampingan hingga pukul 21.00 jika diperlukan. Pendampingan akhir pekan juga tersedia pada hari Sabtu atau Minggu dengan mentor.

Hasil akhir dari program ini diharapkan bukan hanya mencetak *cyber security* yang siap diserap industri, tetapi juga bisa menciptakan pengembangan yang profesional dengan keterampilan kolaborasi, adaptasi, dan komunikasi yang baik.

Table 1. 1 *Learning Guidance Cyber security*

Cyber security		
Soft Skill	*Security Governance*	Kebijakan dan prosedur yang memerlukan kemampuan manajemen, komunikasi, dan kepemimpinan
Hard Skill	*Cyber security Foundations*	Pengetahuan dasar tentang prinsip dan praktik dalam *cyber security*
	Cryptography	Pengetahuan seputar enkripsi untuk melindungi data
	Network security	Pengetahuan untuk melindungi jaringan komputer dari serangan dan akses yang tidak sah
	Web Application Security	Pengetahuan untuk melindungi aplikasi *web* dari serangan
	Pengenalan *Malware*	Pengetahuan tentang berbagai jenis *malware* dan cara mendeteksinya serta melindungi sistem dari *malware*

1.2.1 Deskripsi *Capstone Project*

Project ini dirancang sebagai puncak dari proses pembelajaran, di mana penulis menerapkan pengetahuan dan keterampilan yang telah penulis pelajari selama program berlangsung.

a. Penyelesaian

Semakin berkembangnya internet, banyak perusahaan yang semakin mengandalkan komunikasi internal. Namun, hal tersebut dapat membawa risiko terhadap data sensitif yang dapat dieksploitasi oleh pihak yang tak berwenang. Tanpa kesadaran keamanan yang memadai, akses data bisa menjadi rentan terhadap serangan siber yang semakin canggih.

Salah satu masalah yang sering terjadi dalam keamanan siber adalah mendeteksi dan me-*respons* dengan cepat adanya perilaku mencurigakan. Misalnya, serangan *Shellshock*, *Brute Force*, dan *SQL Injection* memiliki risiko besar. Serangan *Brute Force* dengan menebak sandi secara berulang, seringkali berhasil memanfaatkan kata sandi yang lemah, yang bisa membuka akses dan memperoleh data secara tidak sah.

Pada serangan *Shellshock*, penyerang memanfaatkan celah dengan cara memasukkan nilai spesifik ke dalam variabel, memaksa *Bash* untuk menjalankan perintah tertentu saat variabel tersebut diproses.

Beberapa server telah menerapkan langkah-langkah keamanan untuk mencegah serangan *bruteforce*, *Shellshock,* dan *SQL Injection.* Seperti pengguanaan mekanisme *lockout* untuk memblokir alamat IP setelah sejumlah upaya *login* yang gagal, implementasi pembaruan keamanan terkini, dan penggunaan *firewall* aplikasi web (WAF) untuk memblokir serangan berbasis *SQL Injection*. Meskipun demikian, langkah-langkah ini tidak selalu cukup untuk mengatasi semua jenis ancaman yang semakin kompleks dan berkembang. Oleh karena itu, diperlukan sistem *monitoring* atau pemantauan secara *real-time* yang dapat membuktikan efektivitasnya dalam meningkatkan keamanan jaringan.

Untuk mengatasi masalah ini, maka dilakukan *Proof of Concept* dengan menggunakan Wazuh, *platform* pemantauan keamanan *open-source* yang memadai. PoC ini difokuskan dalam mendeteksi serangan

Shellshock, *Brute Force*, dan *SQL Injection*. Hasil PoC diharapkan dapat menunjukkan bahwa Wazuh mampu mendeteksi intrusi, analisis *log*, dan mengenali ancaman secara *real-time* yang membuktikan efektivitasnya dalam meningkatkan keamanan jaringan.

Project ini menggunakan Wazuh untuk memperkuat keamanan jaringan dengan mengidentifikasi serangan-serangan tersebut. Dengan kemampuan Wazuh dalam mendeteksi serangan dan analisis *log* secara *real-time*, penulis dapat segera merespons insiden keamanan dan mengatasi serangan yang memanfaatkan celah keamanan. Dari penerapan Wazuh, dapat memperkuat posisi keamanan siber, meningkatkan visibilitas terhadap ancaman, dan melindungi jaringan mereka dari serangan siber yang semakin canggih[1].

Meskipun beberapa server sudah menerapkan berbagai mekanisme *anti-attack* untuk mencegah serangan seperti *bruteforce, Shellshock*, dan *SQL Injection*, solusi-solusi ini tidak selalu sempurna atau mencalup semua vektor serangan yang mungkin. Wazuh menambahkan lapisan keamanan tambahan dengan memberikan pemantauan dan deteksi *real-time* yang dapat mengidentifikasi perilaku mencurigakan atau anomali yang mungkin tidak terdeteksi oleh mekanisme keamanan yang sudah ada. Dengan integrasi Wazuh dapat memperoleh visibilitas yang lebih baik terhadap aktivitas jaringan, memungkinkan deteksi dini terhadap serangan yang lolos dari mekanisme pertahanan yang ada, dan menyediakan alat untuk analisis mendalam serta respon insiden yang cepat dan efektif.

b. Anggota Tim

Berikut merupakan nama - nama anggota kelompok bersamaan dengan peran-nya. Semua nama di bawah telah disetujui oleh mitra, sementara peran masing - masing dipilih berdasarkan keahlian anggota.

Table 1. 2 Anggota Tim *Capstone Project*

Nama	Role	Asal Universitas
Dimas Vidillah Christian	Penyerang *Scenario Shellshock*	Sekolah Tinggi Manajemen Informatika dan Komputer Indo Daya Suvana
Salsabil Syifa Arinda	Penyerang *Scenario Brute Force*	Politeknik Negeri Cilacap
Irfan Wijaya	Penyerang *Scenario SQL Injection*	Universitas 17 Agustus 1945 Surabaya

c. Gambaran *Project*

Project ini menggunakan *platform* pemantauan keamanan *open-source* Wazuh, *project* ini melakukan *Proof of Concept* (PoC) yang menunjukkan kemampuan Wazuh dalam mendeteksi intrusi secara *real-time*, analisis *log* dan identifikasi dengan akurat.

PoC ini fokus pada simulasi serangan-serangan tersebut untuk menguji keandalan Wazuh dalam menangkap aktivitas mencurigakan dan mengelola insiden keamanan. Dengan implementasi Wazuh, *project* ini tidak hanya bertujuan pada peningkatan respon terhadap serangan, tetapi juga memberikan visibilitas yang lebih baik terhadap keamanan jaringan.

Project ini juga melibatkan penggunaan VirtualBox sebagai platform virtualisasi untuk pengujian yang aman dan terisolasi. Tiga mesin virtual digunakan dalam project ini:

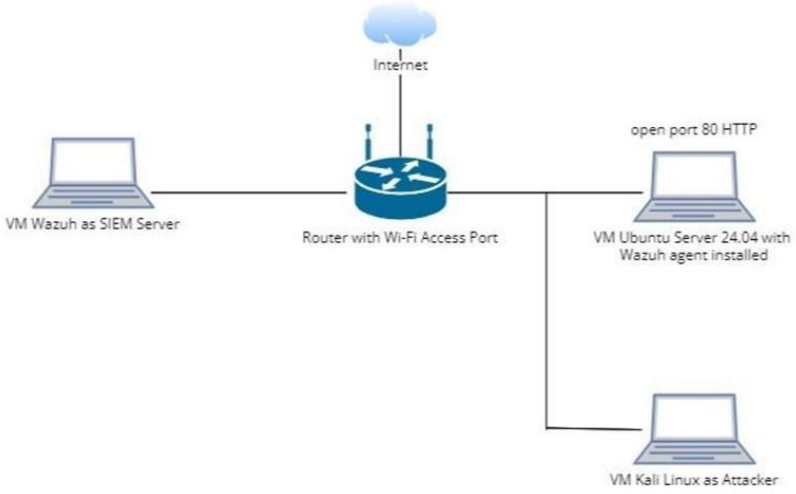
- *Ubuntu Server*. Mesin ini digunakan sebagai server utama yang menjalankan Wazuh *Manager*, Wazuh *Indexer* dan Wazuh *Dashboard*. Server ini bertanggung jawab untuk mengumpulkan, menganalisis serta mengelola data keamanan dari Wazuh *Agent* yang diinstal pada mesin lain.

- *Ubuntu Desktop*. Mesin ini berperan sebagai target serangan dalam skenario pengujian. Ubuntu Desktop diinstal dengan Wazuh Agent untuk memantau sistem dan aplikasi, mengirimkan data log dan keamanan ke Wazuh *Manager*.
- *Kali Linux*. Mesin ini berfungsi sebagai platform penyerang dalam simulasi serangan. *Kali Linux* digunakan untuk menjalankan berbagai alat dan teknik *hacking*, seperti *bruteforce* menggunakan Hydra, untuk menguji keandalan dan ketahanan Wazuh dalam mendeteksi dan merespon aktivitas mencurigakan.

1) Deteksi *Shellshock Attack*

a) Perancangan Infrastruktur

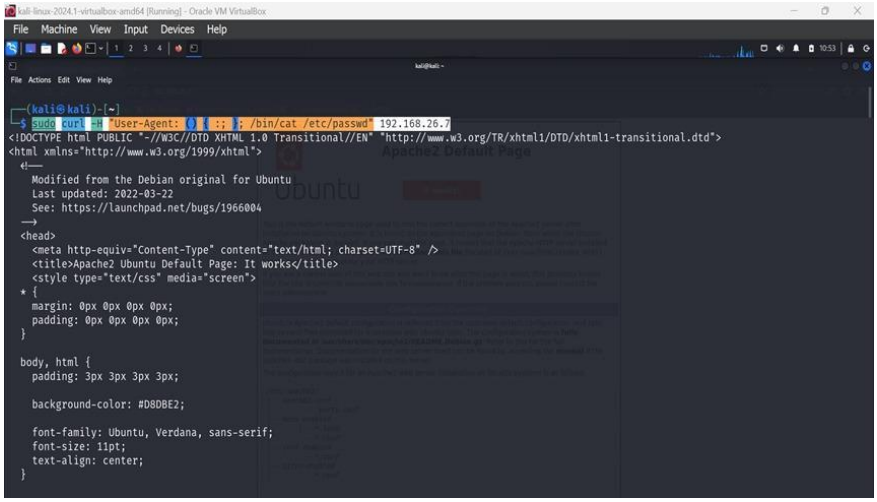
Penulis menggunakan Wazuh untuk mendeteksi serangan *Shellshock* dengan menganalisis *log server web* yang dikumpulkan dari *endpoint* yang dimonitor[2]. Dalam penggunaan ini, penulis telah mengatur *server web Apache* pada *endpoint Ubuntu Server* dan mensimulasikan serangan *Shellshock* dengan *Kali Linux*.



Gambar 1. 2 Perancangan Infrastruktur (*Shellshock Attack*)

b) Simulasi Serangan

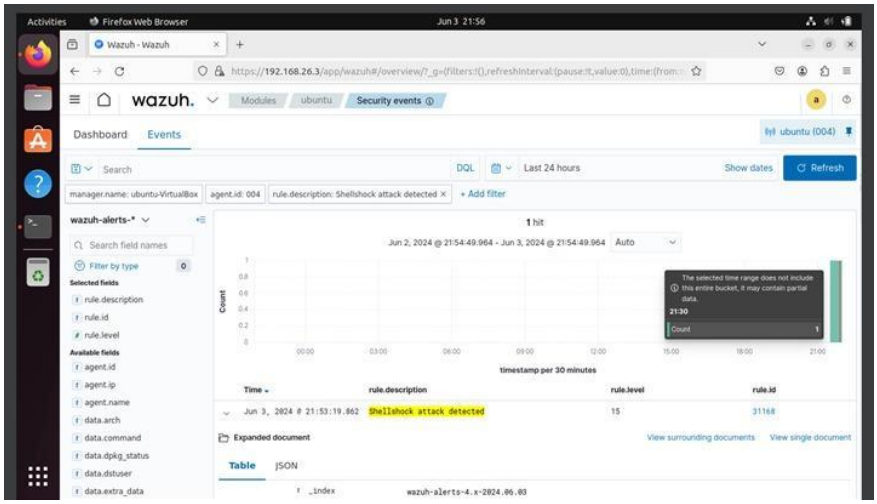
Pada Gambar 1.3 merupakan proses menjalankan simulasi serangan *Shellshock* dengan mengganti <*WEBSERVER_IP_ADRESS*> dengan alamat IP *Ubuntu Server*.



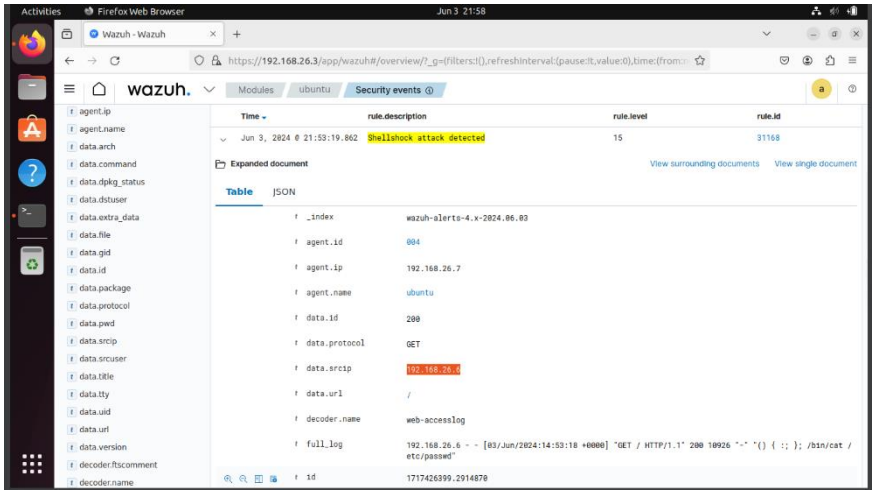
Gambar 1. 3 Perintah *Shellshock Attack* dengan *curl*

c) Hasil deteksi

Pada Gambar 1.4, dapat dilihat data peringatan pada Wazuh *Dashboard* dan menggunakan pencarian untuk menambahkan beberapa *filter*.



Gambar 1. 4 *Shellshock Attack detected rule level* 15

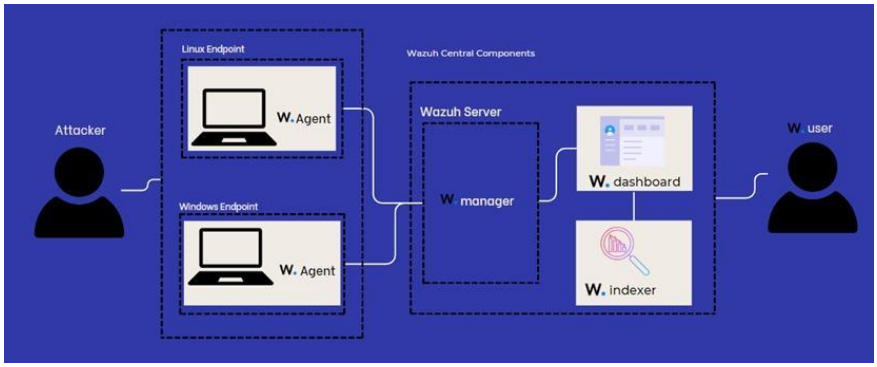


Gambar 1. 5 IP *address Attacker*

2) Deteksi *Brute force Attack*

a) Perancangan Infrastruktur

Untuk melaksanakan pengujian ini, penulis menggunakan *Linux endpoint* untuk meluncurkan serangan *Brute force*. Serangan ini melibatkan banyak percobaan *login* dengan kata sandi yang salah pada sistem *Windows* serta *Linux* untuk mensimulasikan situasi serangan *Brute force* di dunia nyata[3].



Gambar 1. 6 Perancangan Infrastruktur (*Brute force Attack*)

b) Simulasi Serangan

Simulasi serangan diluncurkan dengan menggunakan Hydra *tool*. Hydra dikenal sebagai salah satu *tool* tercepat dalam melakukan serangan *bruteforce*, yang memungkinkan pengujian yang efisien

dalam waktu yang lebih singkat. Salah satu alasan penggunaan Hydra pada *project* ini juga karena Hydra mendukung banyak protokol jaringan termasuk *RDP* dan *SSH*, yang merupakan target utama dalam simulasi ini. Hydra juga memudahkan konfigurasi dan peluncuran serangan, terutama bagi pengguna yang belum memiliki pengalaman mendalam dengan alat-alat *cracking*.

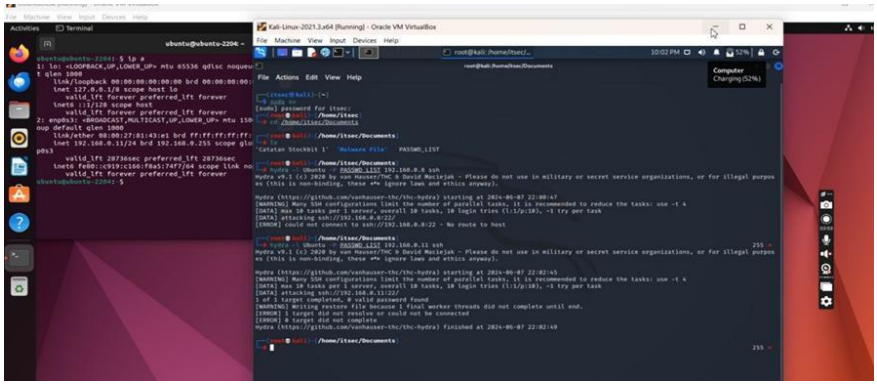
Dalam konteks simulasi serangan ini, Hydra memberikan keseimbangan yang baik antara kecepatan, kemudahan penggunaan, dan dukungan protokol yang dibutuhkan, menjadikannya pilihan yang ideal dibandingkan dengan *tools* lainnya untuk tujuan PoC ini. Berikut merupakan perbandingan *tools* Hydra dengan *tools Brute Force* lain.

Table 1. 3 Perbandingan *Tools BruteForce*

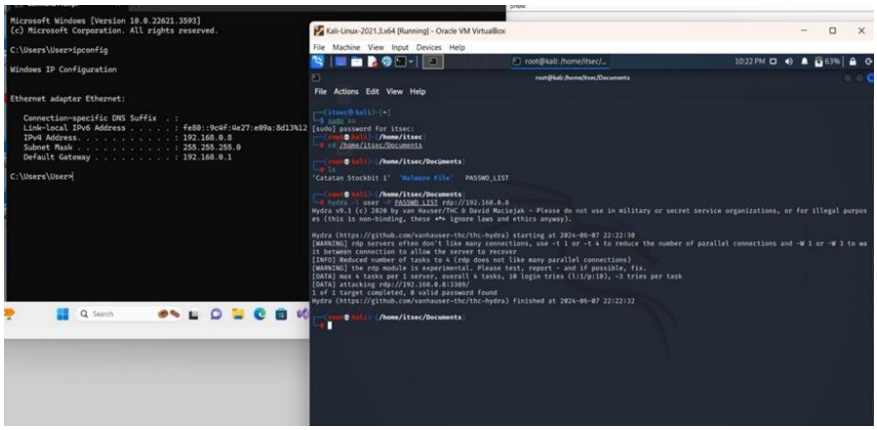
Tool	Kelebihan	Kekurangan
Hydra	- Cepat dan efisien - Mendukung banyak protokol (RDP, SSH, FTP, dll.) - Mudah digunakan	- Membutuhkan daftar kata sandi yang baik untuk hasil yang efektif - Terbatas pada serangan *bruteforce*
John the Ripper	- Sangat cepat pada mode *default* - Mendukung berbagai jenis *hash password* - Mendukung serangan *hybrid* (dictionary + bruteforce)	- Lebih kompleks dalam konfigurasi - Tidak mendukung langsung serangan terhadap layanan jaringan (memerlukan *dump hash*)
Medusa	- *Multi-threading* yang sangat baik - Mendukung berbagai layanan jaringan - Konfigurasi fleksibel	- Dokumentasi kurang lengkap - Lebih sulit untuk digunakan dibandingkan dengan Hydra
Ncrack	- Dirancang untuk kecepatan dan efisiensi	- Antarmuka pengguna tidak seintuitif Hydra

	- Mendukung banyak protokol jaringan - Fitur *recovery* dari serangan yang gagal	- Kurang populer sehingga komunitas dan dukungan lebih kecil

Targetnya diatur ke alamat IP *Windows* dan *Linux*, dengan nama pengguna '*Users*' dan '*Ubuntu'*. Lalu membuat daftar kata sandi yang digunakan untuk serangan yang disimpan pada mesin *Linux.*



Gambar 1. 7 *Brute force* ke *Windows* dengan Hydra *tools*

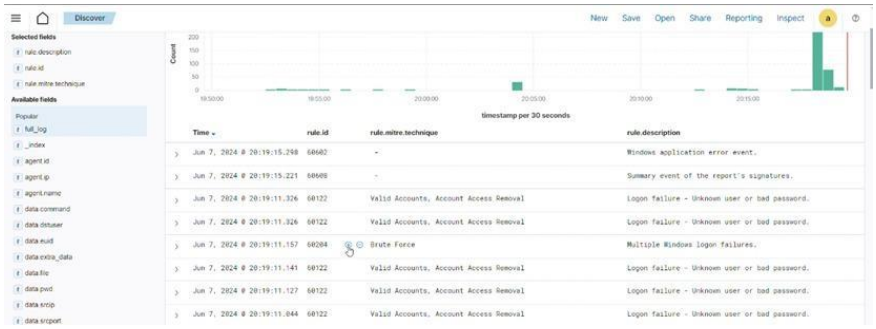


Gambar 1. 8 *Brute force* ke *Windows* dengan Hydra *tools*

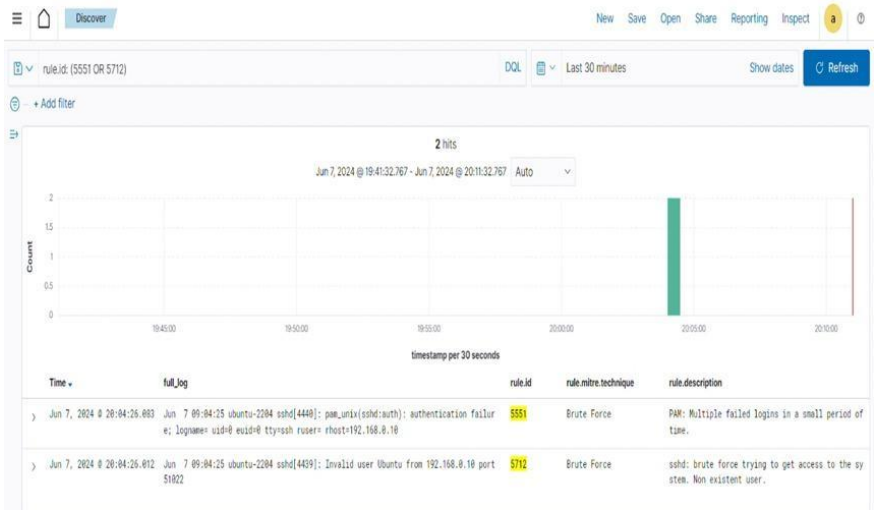
c) Hasil deteksi

Wazuh berhasil mengidentifikasi serangan *Brute force* dengan menangkap dan analisis *log event* yang relevan. Setiap upaya *login* yang gagal menghasilkan *event* yang sesuai dengan aturan Wazuh

yang dikonfigurasi (60122 dan 60204), yang segera terdeteksi dan ditandai oleh Wazuh sebagai aktivitas mencurigakan.



Gambar 1. 9 *Security Events* pada *Windows*

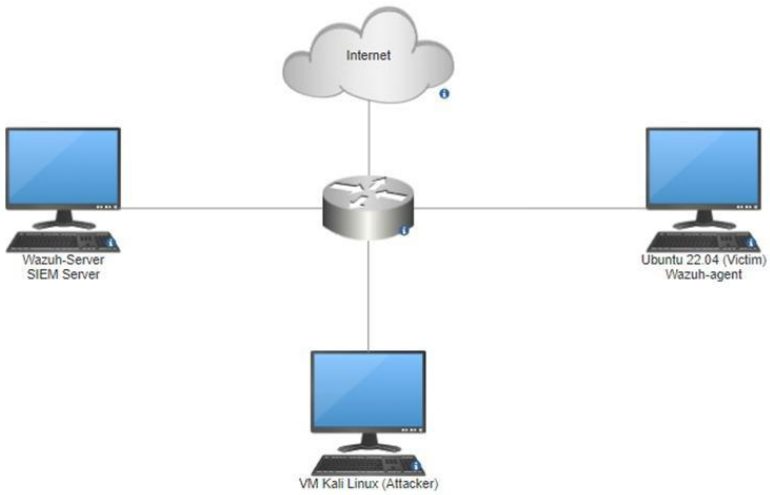


Gambar 1. 10 *Security Events* pada *Linux*

3) Deteksi *SQL Injection*

a) Perancangan Infrastruktur

Dalam serangan ini, pelaku ancaman memasukkan kode berbahaya ke dalam masukan yang kemudian diteruskan ke *database* untuk diproses dan dieksekusi[4]. Jika berhasil, serangan *SQL Injection* dapat memberikan akses tidak sah kepada penyerang untuk membaca data sensitif yag tersimpan dalam *database* tersebut. Penulis mensimulasikan serangan *SQL Injection* terhadap *endpoint Ubuntu* dan mendeteksinya dengan Wazuh.



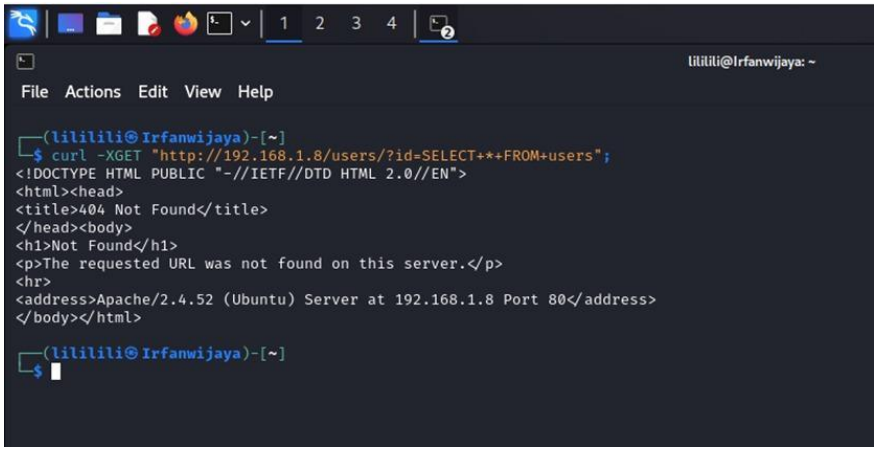
Gambar 1. 11 Perancangan Infrastruktur (*SQL Injection*)

b) Simulasi Serangan

Simulasi serangan dimulai dengan menjalankan perintah

```
$ curl -XGET "http://<UBUNTU_IP>/users/?id=SELECT+*+FROM+users";
```

dan menyesuaikan dengan IP *Address victim* masing-masing.

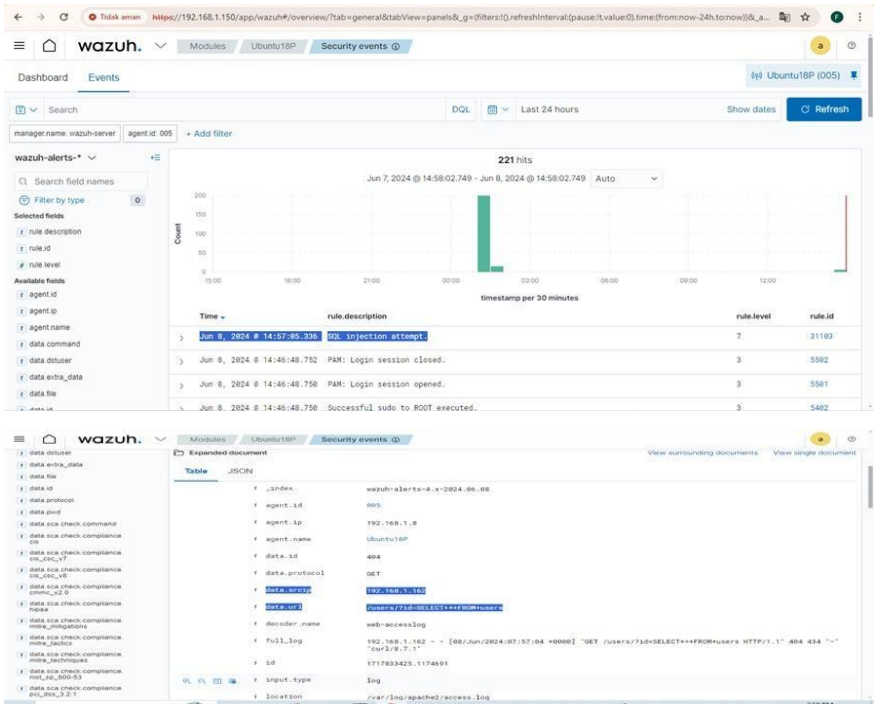


Gambar 1. 12 Perintah *SQL Injection* dengan *curl*

c) Hasil deteksi

Setelah *Attack* dieksekusi, terlihat pada Wazuh *Dashboard Log SQL Injection attempt* yang telah dilakukan sebelumnya. Wazuh juga

menampilkan detail IP *Address* dari *Attacker* dan *code* yang dilakukan oleh *Attacker*.



Gambar 1. 13 Deteksi *SQL Injection* oleh Wazuh

4) Mitigasi

Selain deteksi serangan, mitigasi sangat penting untuk meminimalkan dampak. Langkah-langkah mitigasi yang bisa dilakukan setelah terjadinya *attack* meliputi:

- *Patch Management*. Memastikan semua perangkat lunak dan sistem diperbarui dengan *patch* keamanan terbaru.
- *Firewalls* dan *IDS/IPS*. Menggunakan *firewall* dan sistem deteksi/preventif intrusi untuk membatasi akses dan memblokir serangan.
- *Backup* dan *Recovery*. Memiliki rencana cadangan dan pemulihan yang baik untuk meminimalkan dampak jika terjadi serangan.
- *Training* dan *Awareness*. Melatih karyawan untuk mengenali ancaman umum seperti *phishing* dan bagaimana melaporkannya.
- Segregasi Jaringan. Memisahkan jaringan untuk membatasi penyebaran serangan.

- *Access Control*. Menggunakan kontrol akses yang ketat untuk membatasi hak istimewa hanya kepada pengguna yang memerlukannya.