

## Cyber Challenge ACAD CSIRT 2025 #3



Nama Tim : Fay The Demon King

Yusuf Darmawan

Dhio Zahwan Aryasetyo

Ahmad Fayaadh Baisa

## Tugas #1 (25 Point) attack.json

### Attack Pattern yang ditemukan :

#### 1. Initial Compromise

Pelaku mendapatkan akses awal ke jaringan korban dengan teknik paling sering digunakan yaitu spear phishing. Spear phishing adalah teknik yang paling umum digunakan oleh APT1. Email ini berisi lampiran berbahaya atau tautan ke file berbahaya untuk menginstal *backdoor*.

```
(kali@kali)-[~/acadef/File Pendukung Challenge 3]
$ jq '.objects[] | select(.type = "attack-pattern") | {name, kill_chain_phases, description}' attack.json
{
  "name": "Initial Compromise",
  "kill_chain_phases": [
    {
      "kill_chain_name": "mandiant-attack-lifecycle-model",
      "phase_name": "initial-compromise"
    }
  ],
  "description": "As with most other APT groups, spear phishing is APT1's most commonly used technique. The spear phishing emails contain either a malicious attachment or a hyperlink to a malicious file. The subject line and the text in the email body are usually relevant to the recipient. APT1 also creates webmail accounts using real peoples' names - names that are familiar to the recipient, such as a colleague, a company executive, an IT department employee, or company counsel. The files they use contain malicious executables that install a custom APT1 backdoor that we call WEBC2-TABLE."
}
```

Gambar 1.1 Initial Compromise

#### 2. Establish Foothold

Setelah korban membuka file berbahaya dan malware berhasil dijalankan, pelaku menginstal *backdoor* untuk mempertahankan akses. Mereka sering menggunakan *backdoor* kustom seperti WEBC2 dan BISCUIT, yang berkomunikasi melalui HTTP atau SSL untuk menyamarkan lalu lintas. Backdoor ini akan terus melakukan koneksi keluar (outbound) ke command and control (C2) server milik pelaku. Pelaku bisa memberikan perintah melalui halaman web yang telah disiapkan, dan WEBC2 akan membaca perintah tersebut melalui tag HTML tersembunyi.



```
{
  "name": "Establishing a Foothold",
  "kill_chain_phases": [
    {
      "kill_chain_name": "mandiant-attack-lifecycle-model",
      "phase_name": "establish-foothold"
    }
  ],
  "description": "APT1 establishes a foothold once email recipients open a malicious file and a backdoor is subsequently installed. In almost every case, APT backdoors initiate outbound connections to the intruder's 'command and control' (C2) server. While APT1 intruders occasionally use publicly available backdoors such as Poison Ivy and Gh0st RAT, the vast majority of the time they use what appear to be their own custom backdoors. APT1's backdoors are in two categories: 'Beachhead Backdoors' and 'Standard Backdoors.' Beachhead Backdoors offer the attacker a toe-hold to perform simple tasks like retrieve files, gather basic system information and trigger the execution of other more significant capabilities such as a standard backdoor. APT1's beachhead backdoors are usually what we call WEBC2 backdoors. WEBC2 backdoors are probably the most well-known kind of APT1 backdoor, and are the reason why some security companies refer to APT1 as the Comment Crew. A WEBC2 backdoor is designed to retrieve a webpage from a C2 server. It expects the webpage to contain special HTML tags; the backdoor will attempt to interpret the data between the tags as commands. WEBC2 backdoors are often packaged with spear phishing emails. Once installed, APT1 intruders have the option to tell victim systems to download and execute additional malicious software of their choice. The standard, non-WEBC2 APT1 backdoor typically communicates using the HTTP protocol (to blend in with legitimate web traffic) or a custom protocol that the malware authors designed themselves. The BISCUIT backdoor (so named for the command "bdkzt") is an illustrative example of the range of commands that APT1 has built into its "standard" backdoors. APT1 has used and steadily modified BISCUIT since as early as 2007 and continues to use it presently. Some APT backdoors attempt to mimic legitimate Internet traffic other than the HTTP protocol. When network defenders see the communications between these backdoors and their C2 servers, they might easily dismiss them as legitimate network traffic. Additionally, many of APT1's backdoors use SSL encryption so that communications are hidden in an encrypted SSL tunnel."
```

Gambar 1.2 Establish Foothold

### 3. Privilege Escalation

Untuk memperluas akses, pelaku kemudian melakukan escalation of privileges. Ini dilakukan dengan mengambil password hash dari sistem korban, menggunakan berbagai tools publik seperti mimikatz, fgdump, dan lainnya. Dengan hash ini, pelaku dapat mengakses akun pengguna lain di jaringan tanpa harus mengetahui passwordnya secara langsung.

```
{
  "name": "Privilege Escalation",
  "kill_chain_phases": [
    {
      "kill_chain_name": "mandiant-attack-lifecycle-model",
      "phase_name": "escalate-privileges"
    }
  ],
  "description": "Escalating privileges involves acquiring items (most often usernames and passwords) that will allow access to more resources within the network. APT1 predominantly uses publicly available tools to dump password hashes from victim systems in order to obtain legitimate user credentials."
```

Gambar 1.3 Privilege Escalation

### 4. Internal Reconnaissance

Setelah mendapatkan akses lebih tinggi, pelaku mulai menjelajahi jaringan internal korban. Mereka menggunakan perintah bawaan sistem operasi, seperti ipconfig, netstat, atau dir, untuk mencari tahu struktur jaringan, user lain, dan file penting. Aktivitas ini dilakukan manual atau melalui skrip batch agar lebih cepat dan efisien.

```
{
  "name": "Internal Reconnaissance",
  "kill_chain_phases": [
    {
      "kill_chain_name": "mandiant-attack-lifecycle-model",
      "phase_name": "internal-recon"
    }
  ],
  "description": "In the Internal Reconnaissance stage, the intruder collects information about the victim environment. Like most APT (and non-APT) intruders, APT1 primarily uses built-in operating system commands to explore a compromised system and its networked environment. Although they usually simply type these commands into a command shell, sometimes intruders may use batch scripts to speed up the process."
}
```

Gambar 1.4 Internal Reconnaissance

## 5. Lateral Movement

Dengan kredensial sah di tangan, pelaku mulai berpindah ke sistem lain dalam jaringan. Mereka menggunakan tool Windows seperti psexec (dari Sysinternals) atau scheduler seperti at.exe untuk menjalankan perintah di sistem lain. Karena menggunakan akun resmi, aktivitas ini seringkali luput dari pemantauan sistem keamanan.

```
{
  "name": "Lateral Movement",
  "kill_chain_phases": [
    {
      "kill_chain_name": "mandiant-attack-lifecycle-model",
      "phase_name": "move-laterally"
    }
  ],
  "description": "Once an APT intruder has a foothold inside the network and a set of legitimate credentials, it is simple for the intruder to move around the network undetected. They can connect to shared resources on other systems. They can execute commands on other systems using the publicly available 'psexec' tool from Microsoft Sysinternals or the built-in Windows Task Scheduler ('at.exe')."
}
```

Gambar 1.5 Lateral Movement

## 6. Maintain Presence

Untuk memastikan mereka tetap bisa mengakses jaringan meskipun sistem dibersihkan, pelaku akan melakukan persistensi. Ini dilakukan dengan menginstal backdoor tambahan di berbagai titik, menggunakan kredensial VPN sah, dan bahkan menyusup ke portal web internal. Tujuannya adalah tetap bisa masuk meski jalur awal sudah ditutup.

```
{
  "name": "Maintain Presence",
  "kill_chain_phases": [
    {
      "kill_chain_name": "mandiant-attack-lifecycle-model",
      "phase_name": "maintain-presence"
    }
  ],
  "description": "In this stage, the intruder takes actions to ensure continued, long-term control over key systems in the network environment from outside of the network. APT1 does this in three ways: Install new backdoors on multiple systems, use legitimate VPN credentials, and log in to web portals."
}
```

Gambar 1.6 Maintain Presence

## 7. Completing the Mission

Setelah menemukan file atau data target, pelaku akan mengarsipkan data tersebut dengan tool RAR, lalu membaginya ke dalam potongan-potongan agar mudah ditransfer. Data kemudian diambil melalui FTP atau backdoor yang sudah dipasang. Untuk email, pelaku menggunakan tool khusus seperti GETMAIL dan MAPIGET. Mereka bisa mencuri email mingguan secara teratur, menunjukkan serangan yang berkelanjutan dan terencana.

```
{
  "name": "Completing the Mission",
  "kill_chain_phases": [
    {
      "kill_chain_name": "mandiant-attack-lifecycle-model",
      "phase_name": "complete-mission"
    }
  ],
  "description": "Similar to other APT groups we track, once APT1 finds files of interest they pack them into archive files before stealing them. APT intruders most commonly use the RAR archiving utility for this task and ensure that the archives are password protected. Sometimes APT1 intruders use batch scripts to assist them in the process. After creating files compressed via RAR, the APT1 attackers will transfer files out of the network in ways that are consistent with other APT groups, including using the File Transfer Protocol (FTP) or their existing backdoors. Many times their RAR files are so large that the attacker splits them into chunks before transferring them. Unlike most other APT groups we track, APT1 uses two email-stealing utilities that we believe are unique to APT1. The first, GETMAIL, was designed specifically to extract email messages, attachments, and folders from within Microsoft Outlook archive ('PST') files. The GETMAIL utility allows APT1 intruders the flexibility to take only the emails between dates of their choice. In one case, we observed an APT1 intruder return to a compromised system once a week for four weeks in a row to steal only the past week's emails. Whereas GETMAIL steals email in Outlook archive files, the second utility, MAPIGET, was designed specifically to steal email that has not yet been archived and still resides on a Microsoft Exchange Server. In order to operate successfully, MAPIGET requires username/password combinations that the Exchange server will accept. MAPIGET extracts email from specified accounts into text files (for the email body) and separate attachments, if there are any."
}
```

Gambar 1.7 Completing the Mission

Dari data attack-pattern yang dilakukan oleh penyerang tersebut, ada 8 macam tools yang digunakan untuk melakukan credential exploitation, berikut tools-tools yang digunakannya :

#### 1. Cachedump

Mengekstrak hash kata sandi yang tersimpan dalam cache dari registry sistem.

```
(kali@kali)-[~/acadef/File Pendukung Challenge 3]
$ jq '.objects[] | select(.type == "tool") | {name,tool_types, kill_chain_phases,external_references, description}' attack.json
{
  "name": "cachedump",
  "tool_types": [
    "credential-exploitation"
  ],
  "kill_chain_phases": [
    {
      "kill_chain_name": "mandiant-attack-lifecycle-model",
      "phase_name": "escalate-privileges"
    }
  ],
  "external_references": null,
  "description": "This program extracts cached password hashes from a system's registry"
}
```

Gambar 1.8 Tools Cachedump

#### 2. Fgdump

Sebuah tool untuk melakukan dump (ekstraksi) hash kata sandi Windows.



```

"name": "fgdump",
"tool_types": [
  "credential-exploitation"
],
"kill_chain_phases": [
  {
    "kill_chain_name": "mandiant-attack-lifecycle-model",
    "phase_name": "escalate-privileges"
  }
],
"external_references": [
  {
    "source_name": "fgdump",
    "url": "http://www.foofus.net/fizzgig/fgdump/"
  }
],
"description": "Windows password hash dumper"

```

Gambar 1.9 Tools Fgdump

### 3. Gsecdump

Mendapatkan hash kata sandi dari registry Windows, termasuk file SAM, kredensial domain yang di-cache, dan rahasia LSA.

```

"name": "gsecdump",
"tool_types": [
  "credential-exploitation"
],
"kill_chain_phases": [
  {
    "kill_chain_name": "mandiant-attack-lifecycle-model",
    "phase_name": "escalate-privileges"
  }
],
"external_references": [
  {
    "source_name": "gsecdump",
    "url": "http://www.truesec.se"
  }
],
"description": "Obtains password hashes from the Windows registry, including the SAM file, cached domain credentials, and LSA secrets"

```

Gambar 1.10 Tools Gsecdump

### 4. Mimikatz

Sebuah utilitas yang utamanya digunakan untuk melakukan dump hash kata sandi.

```

"name": "mimikatz",
"tool_types": [
  "credential-exploitation"
],
"kill_chain_phases": [
  {
    "kill_chain_name": "mandiant-attack-lifecycle-model",
    "phase_name": "escalate-privileges"
  }
],
"external_references": [
  {
    "source_name": "mimikatz",
    "url": "http://blog.gentilkiwi.com/mimikatz"
  }
],
"description": "A utility primarily used for dumping password hashes"

```

Gambar 1.11 Tools Mimikatz

### 5. Pwdump7

Melakukan dump hash kata sandi dari registry Windows.

```
{
  "name": "pwdump7",
  "tool_types": [
    "credential-exploitation"
  ],
  "kill_chain_phases": [
    {
      "kill_chain_name": "mandiant-attack-lifecycle-model",
      "phase_name": "escalate-privileges"
    }
  ],
  "external_references": [
    {
      "source_name": "pwdump7",
      "url": "http://www.tarasco.org/security/pwdump_7/"
    }
  ],
  "description": "Dumps password hashes from the Windows registry"
}
```

Gambar 1.12 Tools Pwdump7

#### 6. Pwdumpx

Melakukan dump hash kata sandi dari registry Windows.

```
{
  "name": "pwdumpX",
  "tool_types": [
    "credential-exploitation"
  ],
  "kill_chain_phases": [
    {
      "kill_chain_name": "mandiant-attack-lifecycle-model",
      "phase_name": "escalate-privileges"
    }
  ],
  "external_references": null,
  "description": "Dumps password hashes from the Windows registry"
}
```

Gambar 1.13 Tools Pwdumpx

#### 7. Lslass

Melakukan dump terhadap hash kata sandi dari sesi login yang aktif pada proses lsass

```
{
  "name": "lslass",
  "tool_types": [
    "credential-exploitation"
  ],
  "kill_chain_phases": [
    {
      "kill_chain_name": "mandiant-attack-lifecycle-model",
      "phase_name": "escalate-privileges"
    }
  ],
  "external_references": [
    {
      "source_name": "lslass",
      "url": "http://www.truesec.se"
    }
  ],
  "description": "Dump active logon session password hashes from the lsass process"
}
```

Gambar 1.14 Tool Lslass

#### 8. Pass-the-hash toolkit

Memungkinkan penyusup untuk menggunakan hash kata sandi (tanpa mengetahui kata sandi aslinya) untuk login ke sistem lain.

```
{
  "name": "pass-the-hash toolkit",
  "tool_types": [
    "credential-exploitation"
  ],
  "kill_chain_phases": [
    {
      "kill_chain_name": "mandiant-attack-lifecycle-model",
      "phase_name": "escalate-privileges"
    }
  ],
  "external_references": [
    {
      "source_name": "pass-the-hash toolkit",
      "url": "http://oss.coresecurity.com/projects/pshtoolkit.htm"
    }
  ],
  "description": "Allows an intruder to \"pass\" a password hash (without knowing the original password) to log in to systems"
}
```

Gambar 1.15 Tools Pass-the-hash toolkit

Didalam file attack.json juga ditemukan 2 threat-actor yang diduga milih penyerang. Keduanya memiliki 3 nama alias

```
(kali@kali)-[~/acadev/File Pendukung Challenge 3]
$ jq '.objects[] | select(.type == "threat-actor") | {aliases}' attack.json
{
  "aliases": [
    "Greenfield",
    "JackWang",
    "Wang Dong"
  ]
}
{
  "aliases": [
    "dota",
    "Rodney",
    "Raith"
  ]
}
```

Gambar 1.16 Threat-actor

Sekarang kita analisis filenya kembali untuk mencari email yang digunakan dari Wang dong.



```
{
  "type": "identity",
  "spec_version": "2.1",
  "id": "identity--e88ab115-7768-4630-baa3-3d49a7d946ea",
  "created": "2015-05-15T09:12:16.432Z",
  "modified": "2015-05-15T09:12:16.432Z",
  "name": "Wang Dong",
  "identity_class": "individual",
  "sectors": [
    "government-national"
  ],
  "contact_information": "uglygorilla@163.com"
}
{
  "type": "identity",
  "spec_version": "2.1",
  "id": "identity--0e9d20d9-fb11-42e3-94bc-b89fb5b007ca",
  "created": "2015-05-15T09:12:16.432Z",
  "modified": "2015-05-15T09:12:16.432Z",
  "name": "dota",
  "identity_class": "individual",
  "sectors": [
    "government-national"
  ],
  "contact_information": "dota.d013@gmail.com"
}
```

Gambar 1.17 Identity

Disitu terlihat bahwa user wangdong memiliki alamat email [uglygorilla@163.com](mailto:uglygorilla@163.com)

Di dalam file attack.json juga terdapat 2 FQDN yang berakhiran .org.  
Berikut domain namenya :

1. [hugesoft.org](http://hugesoft.org)
2. [msnhome.org](http://msnhome.org)

```
(kali@kali)-[~/acadeff/File Pendukung Challenge 3]
$ jq '.objects[] | select(.type="indicator" and (.pattern | test(".org"))) | .name, .pattern' attack.json

"FQDN hugesoft.org"
"[domain-name:value = 'hugesoft.org']"
"FQDN msnhome.org"
"[domain-name:value = 'msnhome.org']"
```

Gambar 1.18 FQDN akhiran .org

Berdasarkan serangan-serangan yang berasal dari file attack.json dapat kita analisis untuk mitigasi dari serangan tersebut. Berikut beberapa mitigasi yang dapat dilakukan :

1. Implementasi Email Filtering & Anti-Phishing

Blokir lampiran berbahaya, URL mencurigakan, dan gunakan sandbox untuk attachment.  
<https://attack.mitre.org/techniques/T1566/001/>

2. Penggunaan Multi-Factor Authentication (MFA) Mengurangi risiko akses akun

walau kredensial bocor.  
<https://attack.mitre.org/techniques/T1556/001/>

3. Batasi Penggunaan Tools Lateral Movement (psexec, at.exe) Mencegah penyebaran malware ke sistem lain dalam jaringan.  
<https://attack.mitre.org/techniques/T1021/002/>

4. Deteksi dan Blokir Credential Dumping Tools Mencegah tool seperti mimikatz, fgdump, pwdump dijalankan.  
<https://attack.mitre.org/techniques/T1003/>

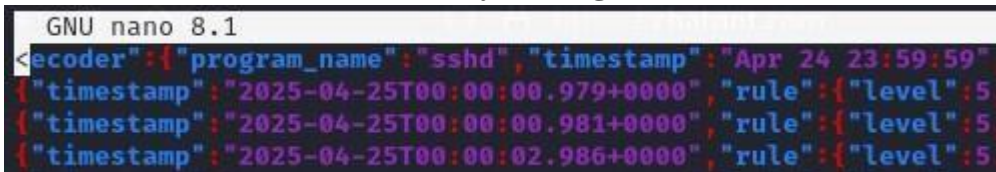
5. Patch Manajemen & Update Sistem Berkala  
Mencegah eksploitasi dari kerentanan yang sudah diketahui.

### **Tools-tools yang digunakan untuk menyelesaikan Tugas#1 yaitu memakai :**

1. Less untuk membaca dan menganalisis awal cara kerja file attack.json
2. JQ (Json Query) untuk mencari spesifik dari object type yang spesifik dalam file attack.json
3. Mitre Attack Untuk mencari teknik serangan yang sesuai untuk mitigasi serangan
4. Chat Gpt Membantu merapikan laporan dan tahapan proses analisa file.
5. Gemini Membantu merapikan laporan dan tahapan proses analisa file.

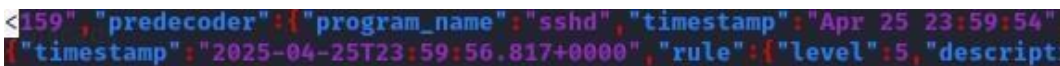
### **Tugas #2 (50 Point)**

#### **Tanggal dan Waktu dimulai dan berakhirnya serangan**



```
GNU nano 8.1
<ecoder":{"program_name":"sshd","timestamp":"Apr 24 23:59:59",
{"timestamp":"2025-04-25T00:00:00.979+0000","rule":{"level":5,
{"timestamp":"2025-04-25T00:00:00.981+0000","rule":{"level":5,
{"timestamp":"2025-04-25T00:00:02.986+0000","rule":{"level":5,
```

Gambar 2.1 Awal dari serangan



```
<159","predecoder":{"program_name":"sshd","timestamp":"Apr 25 23:59:54",
{"timestamp":"2025-04-25T23:59:56.817+0000","rule":{"level":5,"descript:
```

Gambar 2.2 Akhir dari serangan

berdasarkan analisa kami menggunakan nano pada log file ossec-25.json serangan dimulai tanggal 24 april 2025 pada jam 23:59:59 WIB hingga 25 april 2025 pada jam 23:59:56 WIB.

**Jabarkan semua jenis serangan yang tercatat dan urutkan jumlah serangan dari setiap insiden tersebut.**

```
(kali@kali)-[~/Downloads/acadefense/chall3/file2]
$ grep -o "description": "[^"]*" ossec-25.txt \
| cut -d ':' -f2- \
| tr -d '"' \
| sort | uniq -c | sort -nr
24032 sshd: Attempt to login using a non-existent user
13712 PAM: User login failed.
1674 sshd: authentication failed.
1387 Web server 400 error code.
267 Host-based anomaly detection event (rootcheck).
245 PHP Warning message.
140 System running out of memory. Availability of the system is in risk.
139 Integrity checksum changed.
129 Dpkg (Debian Package) half configured.
88 New dpkg (Debian Package) installed.
80 Multiple web server 400 error codes from same source ip.
59 Listened ports status (netstat) changed (new port opened or closed).
56 Agent event queue is full. Events may be lost.
43 sshd: brute force trying to get access to the system. Non existent user.
33 Agent event queue is back to normal load.
33 Agent event queue is 90% full.
32 Web server 500 error code (server error).
32 Nginx error message.
27 High amount of POST requests in a small period of time (likely bot).
18 Interface entered in promiscuous(sniffing) mode.
17 Nginx critical message.
17 Log file rotated.
16 Possible kernel level rootkit
14 PAM: Login session opened.
13 Systemd: Service exited due to a failure.
11 Suspicious URL access.
10 Dpkg (Debian Package) removed.
8 Web server 500 error code (Internal Error).
8 PAM: Login session closed.
8 File deleted.
6 sshd: authentication success.
4 Successful sudo to ROOT executed.
4 sshd: connection reset
4 Common web attack.
3 Docker: Error message
2 Wazuh agent stopped.
2 Wazuh agent started.
```

Gambar 2.3 Jenis dan jumlah serangan yang tercatat

Jenis serangan yang terjadi berdasarkan hasil analisa menggunakan grep pada log file ossec-25.txt yang diurutkan berdasarkan jumlah serangan yang dilakukan menggunakan teknik tersebut.

Jumlah	Rule Description	Jenis Serangan	Penjelasan
24032	sshd: Attempt to login using a non-existent user	Brute Force / Password Guessing	Upaya menebak akun dengan mencoba login ke user secara random.
13712	PAM: User login failed.	Brute Force / Password Guessing	Gagal login berulang bisa menunjukkan brute-force untuk membobol akun.
1674	sshd: authentication failed.	Credential Access	Gagal login SSH, kemungkinan bagian dari serangan brute-force.



1387	Web server 400 error code.	Web Attack Recon / Fuzzing	Banyak permintaan invalid ke server bisa menunjukkan scanning atau fuzzing.
139	Integrity checksum changed	File Tampering/ Persistence	Modifikasi file sistem, dapat berupa file penting.
80	Multiple web server 400 error codes from the same source ip.	Web Attack Recon / Fuzzing	IP mencurigakan mengirim banyak permintaan salah, mengindikasikan probing.
43	sshd: brute force trying to get access to the system. Non existent user.	Brute Force	Deteksi eksplisit dari upaya brute force SSH.
32	Web server 500 error code (server error).	Web Exploitation Attempt	Indikasi serangan mencoba membuat server error (DoS atau exploit logic).
32	Nginx error message.	Web Application Attack	Bisa terkait kesalahan akses atau exploit terhadap aplikasi web.
27	High amount of POST requests in a small period of time (likely bot).	DoS / Web Bot Attack	Bot yang mencoba spam POST request, bisa DoS atau exploit testing.
18	Interface entered in promiscuous(sniffing) mode.	Network Sniffing	Sistem dimodifikasi untuk menangkap semua lalu lintas jaringan indikasi sniffing.

16	Possible kernel level rootkit	Rootkit	Indikasi malware tingkat kernel untuk menyembunyikan aktivitas.
11	Suspicious URL access.	Web Exploitation Attempt	Mengakses URL mencurigakan bisa menunjukkan exploit atau scan.
8	File deleted.	Data Destruction / Anti-Forensic	Menghapus file bisa menunjukkan upaya menghapus jejak atau merusak data.
6	sshd: authentication success	Akses Kredensial Berhasil	Login SSH sukses, berpotensi hasil brute force sebelumnya.
4	Successful sudo to ROOT executed	Privilege Escalation	User berhasil menjalankan perintah dengan hak root.
4	Common web attack.	Web Exploitation Attempt	Deteksi umum terhadap serangan pada aplikasi web.

Tabel 2.1 Urutan Jenis Serangan dan Jumlahnya

**Daftarkan semua IP address penyerang (original source IP address) dan urutkan jumlah serangan berdasarkan IP address tersebut.**

```
(kali@kali)-[~/Downloads/acadefense/chall3/file2]
$ grep -Eo '([0-9]{1,3}\.){3}[0-9]{1,3}' ossec-25.txt \
| grep -v -E '^((10\.|192\.168\.|172\.(1[6-9]|2[0-9]|3[0-1])\.)' \
| sort | uniq -c | sort -nr
10106 195.211.191.176
9453 195.211.191.212
7810 195.211.191.229
7546 195.211.191.199
6622 195.211.191.159
5390 195.211.191.125
5386 195.211.191.189
5324 195.211.191.205
4688 195.211.191.207
4548 195.211.191.194
4288 45.144.212.139
4048 195.211.191.201
4016 195.211.191.210
```

Gambar 2.4 Daftar Ip address penyerang dan jumlahnya

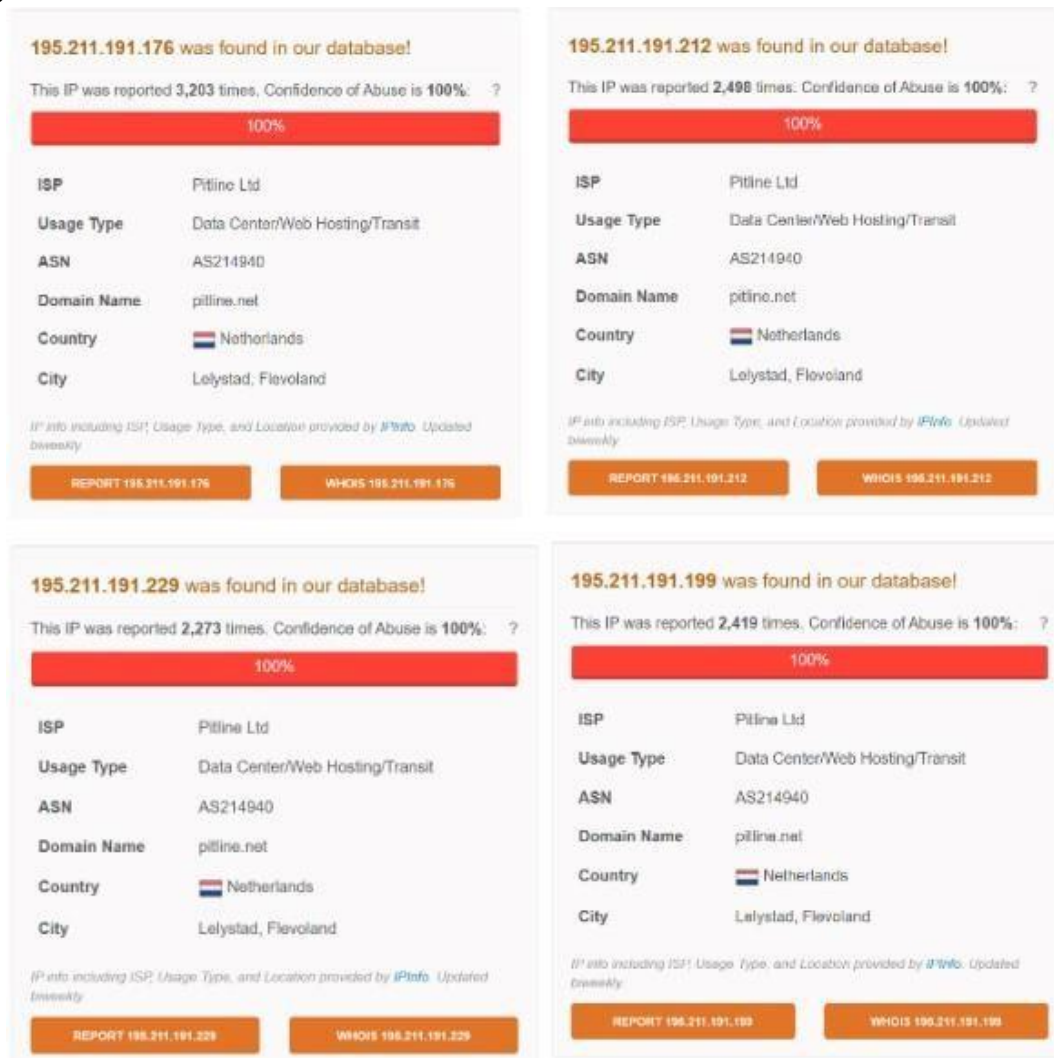
Daftar Ip address penyerang dan jumlah serangannya yang sudah diurutkan berdasarkan jumlah serangan yang dilakukan dari ip addressnya.

No.	Ip Address	Jumlah Serangan
1	195.211.191.176	10106
2	195.211.191.212	9453
3	195.211.191.229	7810
4	195.211.191.199	7546
5	195.211.191.159	6622
6	195.211.191.125	5390
7	195.211.191.189	5386
8	195.211.191.205	5324
9	195.211.191.207	4688
10	195.211.191.194	4548
11	45.144.212.139	4288
12	195.211.191.201	4048
13	195.211.191.210	4016

Tabel 2.2 Urutan Ip address penyerang dan Jumlahnya












Analisa lebih lanjut semua ip address yang tercatat dengan mengecek apakah mereka berbahaya



Gambar 2.5 Hasil cek Ip address pada abuseipdb

Berdasarkan hasil analisa pada ip address yang berawalan 195.211.191 menggunakan abuseipdb ditemukan bahwa semuanya merupakan site berbahaya dengan nama domain pitline.net berasal dari belanda kota lelystad, flevoland yang dimana semua ip addressnya sudah memiliki reputasi buruk jika dicek pada website abuseipdb rata-rata memiliki lebih dari 2000 orang melaporkan bahwa ip address tersebut berbahaya.

Reporter	IoA Timestamp in UTC	Comment	Categories
✓  <a href="#">shadow</a>	2025-05-12 13:04:04 (4 weeks ago)	[2025 May 12 01:04:13] DoS / DDoS detected from 195.211.191.207 () SYN=111 x / 24 Hours   ACTIVITY: ... <a href="#">show more</a>	DDoS Attack
✓  <a href="#">BSG Webmaster</a>	2025-05-12 07:35:15 (4 weeks ago)	Port scanning (Port 265)	Port Scan Hacking
✓  <a href="#">ATV</a>	2025-05-12 03:03:33 (4 weeks ago)	Unsolicited connection attempts to ports 265, 266	Port Scan
✓  <a href="#">shadow</a>	2025-05-11 22:26:16 (4 weeks ago)	[2025 May 12 00:24:57] TCP Port Scanning detected from 195.211.191.207 () SPT=21524 -> DPT=222	Port Scan
✓  <a href="#">Dadelinux</a>	2025-05-11 22:19:20 (4 weeks ago)	May 12 00:10:19 dadelinux sshd[233685]: Failed password for invalid user ansible from 195.211.191.20 ... <a href="#">show more</a>	Brute-Force SSH
✓  <a href="#">24-seven.io</a>	2025-05-11 22:11:50 (4 weeks ago)	May 12 00:03:09 cow sshd[3578095]: Failed password for invalid user ansible from 195.211.191.207 por ... <a href="#">show more</a>	Brute-Force SSH
✓  <a href="#">24-seven.io</a>	2025-05-11 21:39:23 (4 weeks ago)	May 11 23:34:13 cow sshd[3555158]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid= ... <a href="#">show more</a>	Brute-Force SSH
 <a href="#">noumea electronico</a>	2025-05-11 21:37:36 (4 weeks ago)	2025-05-12T08:31:53.658727 11:00 samba.electronico.nc sshd[170761]: pam_unix(sshd:auth): authenticat ... <a href="#">show more</a>	Brute-Force SSH
✓  <a href="#">Dadelinux</a>	2025-05-11 21:37:30 (4 weeks ago)	May 11 23:31:47 dadelinux sshd[229688]: pam_unix(sshd:auth): authentication failure; logname= uid=0 ...	Brute-Force SSH


Gambar 2.6 Laporan terkini pada abuseipdb

Berdasarkan laporan terbaru pada semua ip address berawalan 195.211.191 diketahui kalau ip address-ip address tersebut biasa digunakan untuk melakukan serangan brute-force, dan SSH walaupun terkadang melakukan DDoS dan juga port scanning.

**45.144.212.139** was found in our database!

This IP was reported **8,154** times. Confidence of Abuse is **100%**: ?

100%

ISP	Kprohost
Usage Type	Data Center/Web Hosting/Transit
ASN	AS214940
Domain Name	pitline.net
Country	 Netherlands
City	Heerlen, Limburg

IP info including ISP, Usage Type, and Location provided by [IPInfo](#). Updated biweekly.









REPORT 45.144.212.139

WHOIS 45.144.212.139

Gambar 2.7 Hasil cek Ip address 45.144.212.139 pada abuseipdb

hasil analisa pada Ip address 45.144.212.139 juga menunjukkan bahwa ip tersebut berbahaya yang berasal dari nama domain yang sama yaitu pitline.net di negara belanda namun kotanya berbeda dengan yang sebelumnya, yang ini berasal dari heerlen, limburg

memiliki laporan sebanyak 8.154 kali dengan keyakinan 100% kalau ip address tersebut berbahaya. Berdasarkan laporan terbaru ip address ini biasa melakukan Port Scanning.

Reporter	IoA Timestamp in UTC	Comment	Categories
✓  Study Bitcoin 🤖	2025-06-10 03:58:52 (21 minutes ago)	Port probe to tcp/30473 [srv126]	Port Scan
✓  Study Bitcoin 🤖	2025-06-10 03:48:22 (32 minutes ago)	Port probe to tcp/30473 [srv128]	Port Scan
✓  Study Bitcoin 🤖	2025-06-10 03:24:29 (56 minutes ago)	2 port probes: 2x tcp/30472 [srv130,srv129]	Port Scan
✓  Study Bitcoin 🤖	2025-06-10 03:02:50 (1 hour ago)	Port probe to tcp/30472 [srv126]	Port Scan
✓  Study Bitcoin 🤖	2025-06-10 02:36:48 (1 hour ago)	2 port probes: tcp/30471, tcp/30472 [srv128,srv130]	Port Scan
✓  Study Bitcoin 🤖	2025-06-10 02:32:28 (1 hour ago)	Port probe to tcp/30471 [srv129]	Port Scan
✓  Study Bitcoin 🤖	2025-06-10 02:08:57 (2 hours ago)	Port probe to tcp/30471 [srv126]	Port Scan
✓  Study Bitcoin 🤖	2025-06-10 01:52:21 (2 hours ago)	Port probe to tcp/30471 [srv126]	Port Scan

Gambar 2.8 Laporan terkini untuk Ip address 45.144.212.139

Berdasarkan laporan terbaru ip address 45.144.212.139 biasa melakukan Port Scanning, dengan frekuensi 3 hingga 4 kali dalam 1 jam

**Daftarkan dan urutkan (menurut jumlah) semua user ID yang dipakai menyerang untuk IP address yang paling sedikit menyerang.**

Karena berdasarkan analisa saya IP address yang paling sedikit menyerang adalah 195.211.191.210 dengan jumlah serangan sebanyak 4016 serangan maka kita cek semua user ID pada ip address tersebut.

```
(kali㉿kali)-[~/Downloads/acadefense/chall3/file2]
$ grep 'srcip':"195.211.191.210"' ossec-25.txt \
| grep -o '"srcuser": "[^"]*"' \
| cut -d ':' -f2 \
| tr -d '"' \
| sort | uniq -c | sort -nr \
| awk '{printf "%s:%s ", $2, $1} END {print "}'
```

Gambar 2.9 Command grep untuk user id pada ip tertentu

Menggunakan command grep sebagai berikut untuk mendapatkan daftar dan urutan yang sudah terurut berdasarkan jumlah serangan pada ip adress 195.211.191.210.



```

user:8 ubuntu:8 tj:6 test:6 oracle:6 guest:6 deploy:6 demo:6 yuelongLi2:4 xiao:4 userftp:4 teamspeak3:4 postgres:4 penghao:4
odoo:4 nvidia:4 lyz:4 kyweb:4 hby:4 gc:4 docker:4 dell:4 debian:4 apache:4 zyxing:2 zychen:2 zwx2:2 zv:2 zuoying:2 zt:2 zst
an:2 zrwu:2 zoupeng:2 zoujiangrui:2 zctest:2 zjime:2 zhuyue:2 zhupeng:2 zhuguoqing:2 zhoushaomin:2 zhoufanghua:2 zhouchaoyan
g:2 zhouchao:2 zhongren1:2 zhongcx:2 zhenxin:2 zhengqiuhua:2 zhaozj:2 zhaoxu:2 zhaoxt:2 zhaowei:2 zhaop:2 zhanhong:2 zhangzx
:2 zhangyz:2 zhangliming:2 zhangjun:2 zhaibo:2 zengle:2 zdd:2 zcfep:2 z1:2 yzh:2 yysheng:2 yxgu:2 ywj:2 yuxiaolin:2 yupeiyao
:2 yulv:2 yulin_envs:2 yulei:2 yujiakun:2 yuh:2 yuguangzeng:2 yueyuan:2 yuexz:2 yuanmin:2 yrliu:2 yrrj:2 ymLiu_Wang:2 ylving:
2 yjs2:2 yhyb:2 yhy:2 yfw:2 yfliu:2 yeneng:2 yelj:2 yding:2 ycx:2 yc_bai:2 ybk:2 yanyang:2 yanxt:2 yangy:2 yangrui:2 yanggua
ngkai:2 yangfanqi:2 yangdongxiao:2 yajunwang:2 xyren:2 xxzhang:2 xwnie:2 xweng:2 xw:2 xview:2 xumeng:2 xuhx:2 xuchaozhi:2 xs
y:2 xsrao:2 xli:2 xiaxinyi:2 xiaofei:2 xiaochuang:2 xh:2 xgy18:2 xgwang:2 xchu:2 wz:2 wyk:2 wxx:2 wuhaoyang:2 wuhaoming:2 ws
:2 wqii:2 wpyan:2 wkx:2 wjhao:2 wjh:2 wj:2 wilmar:2 white:2 wenyuanma:2 wenjun.liu:2 wenb:2 weiyulin:2 weixin:2 weblogic:2 w
eb:2 wcl:2 wbf:2 wanlu:2 wangzihao:2 wangyuheng:2 wangxuyang:2 wangwenjie:2 wangwb:2 wangshenling:2 wangpengfei:2 wangfangfa
ng:2 wangdx:2 wangdai:2 wangbo:2 wang:2 wan:2 vo:2 vm-shilei:2 v95agentfep:2 usr:2 user7:2 user62:2 user4:2 user1:2 unisound
:2 ubuntu123:2 uat:2 tzy:2 txs:2 txiang4:2 tw_admin:2 tony:2 tomcat:2 tokyo:2 tiny:2 tianyuan:2 tianxin:2 tiansheng:2 thqh:2
tfma:2 textile:2 testenv:2 testdev:2 test4:2 test_0:2 teri:2 tech:2 taoyl18:2 tanping:2 takazawa:2 szy:2 syx:2 sysadm:2 syl
iu:2 sxu:2 swb:2 suxubo:2 supp:2 sunshuai:2 suen:2 subversion:2 sty:2 steven:2 squirrel:2 sqm:2 songjianing:2 softadmin:2 sl
x:2 slm:2 skxu:2 shomaserv:2 shl:2 shc:2 sharge:2 share:2 shaoshangxiao:2 shangxinyi:2 sf:2 ses:2 server:2 sdunwh:2 sdu:2
sbwang:2 sbsadmin:2 saurabh:2 samouris:2 sambal:2 salter:2 salt:2 rxYu:2 ruth:2 rubin:2 rstudio-server:2 rr:2 rou:2 roo:2 ri
p:2 rh:2 renkai:2 recovery:2 raj:2 qxh:2 qwertyasdfghzxcvbn:2 quqingtao:2 qtb:2 qsyao:2 qsczse:2 Qiao-Zhang:2 qhdhpu:2 qchua
ng:2 pymu:2 py:2 public:2 psma:2 pro01:2 precious:2 pocket:2 platform:2 pjx:2 phonedownload:2 pengqiwei:2 penelope:2 peace:2
pb:2 patrol:2 patrick:2 patest:2 overberg:2 oujc:2 ouctest:2 otelie:2 opusmonk:2 Operator:2 oo:2 Onrain:2 omm:2 old:2 olap:
2 okuno:2 ohae02:2 ods:2 oct:2 ntadmin:2 node:2 nlp:2 netserver:2 ncafact:2 nc:2 nagios:2 mythtv:2 mysqlserver:2 mx:2 msx:2
msl:2 mrkonm:2 mpadmin:2 mobilej:2 mike:2 mercury:2 menu:2 mendoza:2 may:2 marcia:2 maoyj:2 mak:2 mahout:2 lzt:2 lz5:2 lz112
6:2 lxx:2 lxy:2 lwj:2 lvzheqi:2 lvmx:2 luoyuxuan:2 lulvqun:2 lu:2 ltr:2 lsc:2 lrdong:2 lp19:2 loukun:2 longrj:2 longr:2 lzm:
2 llama:2 lxiize:2 liuzuozen:2 liuxueliang:2 liuxiaolong:2 liutt:2 liujc:2 liuhui:2 liuhaoyuan:2 liuhan:2 liuc:2 lisiyi:2 l
ishengjie:2 lirun:2 linxiaolin:2 linq:2 linjiayi:2 lilq123:2 lilang:2 lilijin:2 lijiawei:2 lifengjin:2 liangshu:2 liangbo:2
lhb:2 lg:2 lfs:2 leah:2 ldy:2 ldggzxc:2 lcy:2 lava:2 kxfeng:2 kvhost:2 kunlun:2 kun2:2 kolla:2 kj:2 kg:2 kevin:2 jyz:2 jvx:
2 justin:2 junit:2 juliana:2 jorgec:2 jones:2 joefagan:2 jmuai002:2 jjwang:2 JingZhang1:2 jinchengzhi:2 jhl:2 jiazhan:2 j
iaqi:2 jiaojinlong:2 jiangyh:2 jiangwp:2 jhz:2 jhweb:2 jhuang:2 jgr:2 jfan:2 jesse:2 jacky:2 jackie:2 jabber:2 itpc:2 isdel
n:2 iraish:2 internship:2 internet:2 intgle:2 insta:2 info:2 ines:2 ikn:2 iiii:2 ie:2 hyunmoonsa:2 hyh:2 hxy:2 huxiaobin:2 h
umaer:2 hujiajie:2 huhuiqi:2 huhual:2 huhaonan:2 huawei:2 huangzq:2 huangyuhan:2 huangr:2 huanghaorong:2 hongu
z:2 hjf:2 hive:2 hippo:2 herojoe:2 help:2 hcx:2 hcftp:2 hcat:2 hc:2 hae:2 hadoopusertest:2 hadoop:2 hadi:2 gxzcfepv11:2 gsla
1:2 guoxianzhuang:2 guoqiwei:2 guolili:2 guochao.ma:2 gujionghong:2 guest4:2 guest3:2 guaranty:2 gqshu:2 gongzhan:2 goldie:2
goatherd:2 gml:2 gjy:2 gitread:2 gitlab-prometheus:2 geyouming:2 geeko:2 geek:2 gdh:2 gb:2 gaqx17:2 gansufep:2 gambam:2 fy
f:2 fzy:2 ftpadmin:2 frpuser:2 df:2 fanhuan:2 fangkebang:2 exporter:2 es:2 eq:2 eoffice:2 elsearch:2 ejn:2 ectrip:2 eaglew
i:2 dzq818:2 dumy:2 dum:2 duh:2 drupal:2 doni:2 dongy:2 docs:2 docker2:2 dmall:2 dlk:2 dli:2 dlyb:2 dkhdndn:2 disk1:2 dirk
2 dzb:2 dg:2 developer:2 depth:2 dengpengyong:2 dbuser:2 db2inst1:2 david:2 cyh:2 cxl:2 cxc:2 cwendelues:2 curo2:2 css:2 c
sh3766:2 csbcs:2 cps:2 codyy:2 cnsdptweb:2 cmsadmin:2 client005:2 cjm:2 cirros:2 cipriano:2 chq:2 chenongsheng:2 chenshf:
2 chennm:2 chenjingrui:2 chenjianfei:2 chenjh:2 chenhuo:2 chenchan:2 chena:2 chef:2 changxu:2 cgy:2 cfguser:2 cersz:2 cernetr
t:2 cer2113:2 cebfc:2 cax:2 cas:2 caren:2 caoting:2 caja:2 caiyangyang:2 caffe:2 buzhidao:2 brg:2 bot:2 bobo:2 bna:2 bingli:
2 bharat:2 bgsc:2 bestellung:2 beida:2 bbserver:2 azure:2 awu:2 avt:2 auditadm:2 asis3:2 arena:2 arbiter:2 ara:2 aprl
l:2 appMgr:2 app:2 apetro:2 ansible:2 anne:2 amelita:2 alibaba:2 ailu:2 aft:2 admintemp:2 adminsecurity:2 administrator:2 a
dmin:2 adidas:2 acunetix:2 access:2 aax:2

```

Gambar 2.10 Hasil query grep 2.7

user yang digunakan untuk menyerang 8 kali

= user, ubuntu

user yang digunakan untuk menyerang 6 kali = tj, test,

oracle, guest, deploy, demo

user yang digunakan untuk menyerang 4 kali

= yuelongLi2, xiao, userftp, teamspeak3, postgres, penghao, odoo, nvidia, lyz, kyweb, hby, gc, docker, dell, debian, apache

user yang digunakan untuk menyerang 2 kali

= zyxing, zychen, zwx2, zv, zuoying, zt, zstan, zrwu, zoupeng, zoujiangrui, zctest, zjime, zhuyue, zhupeng, zhuguoqing, zhoushaomin, zhoufanghua, zhouchaoyang, zhouchao, zhongren1, zhongcx, zhenxin, zhengqiuhua, zhaozj, zhaoxu, zhaoxt, zhaowei, zhaop, zhantong, zhangzx, zhangyz, zhangliming, zhangjun, zhaibo, zengle, zdd, zcfep, z1, yzh, yysheng, yxgu, ywj, yuxiaolin, yupeiyao, yulv, yulin\_envs, yulei, yujiakun, yuh, yuguangzeng, yueyuan, yuexz, yuanmin, yrliu, yrrj, ymLiu\_Wang, ylving, yjs2, yhyb, yhy, yfw, yfliu, yeneng, yelj, yding, ycx, yc\_bai, ybk, yanyang, yanxt, yangy, yangrui, yangguangkai, yangfanqi, yangdongxiao, yajunwang, xyren, xxzhang, xwnie, xweng, xw, xview, xumeng, xuhx, xuchaozhi, xsy, xsrao, xli, xiaxinyi, xiaofei, xiaochuang, xh, xgy18, xgwang, xchu, wz, wyk, wxx, wuhaoyang, wuhaoming, ws, wqii, wpyan, wkx, wjhao, wjh, wj, wilmar, white, wenyuanma, wenjun.liu, wenb, weiyulin, weixin, weblogic, web, wcl, wbf, wanlu, wangzihao, wangyuheng, wangxuyang, wangwenjie, wangwb, wangshenling, wangpengfei, wangfangfang, wangdx, wangdai, wangbo, wang, wan, vo, vm-shilei, v95agentfep, usr, user7, user62, user4, user1, unisound, ubuntu123, uat, tzy, txs, txiang4, tw\_admin, tony, tomcat, tokyo, tiny, tianyuan, tianxin, tiansheng, thqh, tfma, textile, testenv, testdev, test4, test\_0, teri, tech, taoyl18, tanping, takazawa, szy, syx, sysadm, syliu,

sxu, swh, suxubo, supp, sunshuai, suen, subversion, sty, steven, squirrel, sqm, songjianing, softadmin, slx, slm, skxu, shomaserv, shl, shcac, sharge, share, shaochangxiao, shangxinyi, sf, ses, server, sdunwh, sdu, sbwang, sbsadmin, saurabh, samouris, samba1, salter, salt, rxYu, ruth, rubin, rstudio-server, rr, rou, roo, rip, rh, renkai, recovery, raj, qxh, qwertyasdfghzxcvbn, quqingtao, qtb, qsyao, qsczse, Qiao-Zhang, qdhdp, qchuang, pymu, py, public, psma, pro01, precious, pocket, platform, pjx, phonedownload, pengqiwei, penelope, peace, pb, patrol, patrick, patest, overberg, oujc, ouctest, otelie, opusmonk, Operator, oo, Onrain, omm, old, olap, okuno, ohae02, ods, oct, ntadmin, node, nlp, netserver, ncafact, nc, nagios, mythtv, mysqlserver, mx, msx, msl, mrkonm, mpadmin, mobilej, mike, mercury, menu, mendoza, may, marcia, maoyj, mak, mahout, lzt, lz5, lz1126, lxz, lxy, lwj, lvzheqi, lvmx, luoyuxuan, lulvqun, lu, ltr, lsc, lrdong, lp19, loukun, longrj, longr, lmz, llama, lixize, liuzuozen, liuxueliang, liuxiaolong, liutt, liujc, liuhui, liuhaoyuan, liuhan, liuc, lisiyi, lishengjie, lirun, linxiaolin, linq, linjiayi, lilq\_123, liling, lilijin, lijiawei, lifengjin, liangshu, liangbo, lhb, lg, lfs, Leah, ldy, ldggzxc, lcy, lava, kxfeng, kvhost, kunlun, kun2, kolla, kj, kg, kevin, jyz, jvx, justin, junit, juliana, jorgec, jones, joefagan, jmuai002, jjwang, JingZhang\_1, jinchengzhi, jhl, jiazhan, jiaqi, jiaojinlong, jiangyh, jiangwp, jhz, jhweb, jhuang, jgr, jfan, jesse, jacky, jackie, jabber, itpc, isdeln, iraish, internship, internet, integre, insta, info, ines, ikn, iii, ie, hyunmoonsa, hyh, hxy, huxiaobin, humaer, hujiajie, huhuiqi, huhuali, huhaonan, huawehong, huawei, huangzq, huangyuan, huangr, huanghaorong, honguz, hjf, hive, hippo, herojoe, help, hc, hcftp, hcat, hc, hae, hadoopusertest, hadoop, hadi, gxzcfev11, gxlai, guoxianzhuang, guoqiwei, guolili, guochao.ma, gujionghong, guest4, guest3, guaranty, gqshu, gongzhan, goldie, goatherd, gml, gjy, gitread, gitlab-prometheus, geyouming, geeko, geek, gdh, gb, gaoxq17, gansufep, gambam, fyf, fxy, ftpadmin, frpuser, fdf, fanhuan, fangkebang, exporter, es, eq, eoffice, elsearch, ejn, ectrip, eaglewiz, dzq818, dummy, dum, duh, drupal, doni, dongy, docs, docker2, dmail, dlk, dli, dlfb, dkhcdndn, disk1, dirk, dnb, dg, developer, depth, dengpengyong, dbuser, db2inst1, davida, cyh, cxi, cxc, cwendelues, curo2, css, csh3766, csbs, cpss, codyy, cnsdptweb, cmsadmin, client005, cjm, cirros, cipriano2, chq, chenyongsheng, chenshf, chenm, chenjingrui, chenjianfei, chenjh, chenhao, chenchuan, chena, chef, changxu, cgy, cfguser, cersz, cernetnt, cer2113, cebfc, cax, cas, caren, caoting, caja, caiyangyang, caffe, buzhidao, brg, bot, bobo, bna, bingli, bhara, bgsc, bestellung, beida, bbserver, azure, ayan, awu, avt, auditadm, asis3, arena, arbiter, ara, april, applmgr, app, apetro, ansible, anne, amelita, alibaba, ailu, aft, admintemp, adminsecurity, administrator, admin, adidas, acunetix, access, aax.

**Kesimpulan dari apa saja yang penyerang lakukan dan metode penyerangannya serta layanan apa yang terpengaruh**

```

access sshd: Attempt to login using a non-existent user Password Guessing SSH sshd
acunetix sshd: Attempt to login using a non-existent user Password Guessing SSH sshd
adidas sshd: Attempt to login using a non-existent user Password Guessing SSH sshd
administrator sshd: Attempt to login using a non-existent user Password Guessing SSH sshd
adminsecurity sshd: Attempt to login using a non-existent user Password Guessing SSH sshd
admin sshd: Attempt to login using a non-existent user Password Guessing SSH sshd
admintemp sshd: Attempt to login using a non-existent user Password Guessing SSH sshd
aft sshd: Attempt to login using a non-existent user Password Guessing SSH sshd
ailu sshd: Attempt to login using a non-existent user Password Guessing SSH sshd
alibaba sshd: Attempt to login using a non-existent user Password Guessing SSH sshd
amelita sshd: Attempt to login using a non-existent user Password Guessing SSH sshd
anne sshd: Attempt to login using a non-existent user Password Guessing SSH sshd
ansible sshd: Attempt to login using a non-existent user Password Guessing SSH sshd
apache sshd: Attempt to login using a non-existent user Password Guessing SSH sshd
apetro sshd: Attempt to login using a non-existent user Password Guessing SSH sshd
applmgr sshd: Attempt to login using a non-existent user Password Guessing SSH sshd
app sshd: Attempt to login using a non-existent user Password Guessing SSH sshd
april sshd: Attempt to login using a non-existent user Password Guessing SSH sshd
ara sshd: Attempt to login using a non-existent user Password Guessing SSH sshd
arbiter sshd: Attempt to login using a non-existent user Password Guessing SSH sshd
arena sshd: Attempt to login using a non-existent user Password Guessing SSH sshd
asis3 sshd: Attempt to login using a non-existent user Password Guessing SSH sshd
auditadm sshd: Attempt to login using a non-existent user Password Guessing SSH sshd
avt sshd: Attempt to login using a non-existent user Password Guessing SSH sshd
awu sshd: Attempt to login using a non-existent user Password Guessing SSH sshd
ayan sshd: Attempt to login using a non-existent user Password Guessing SSH sshd
azure sshd: Attempt to login using a non-existent user Password Guessing SSH sshd
bbserver sshd: Attempt to login using a non-existent user Password Guessing SSH sshd
beida sshd: Attempt to login using a non-existent user Password Guessing SSH sshd
bestellung sshd: Attempt to login using a non-existent user Password Guessing SSH sshd
bgsc sshd: Attempt to login using a non-existent user Password Guessing SSH sshd
bharat sshd: Attempt to login using a non-existent user Password Guessing SSH sshd
bingli sshd: Attempt to login using a non-existent user Password Guessing SSH sshd
bna sshd: Attempt to login using a non-existent user Password Guessing SSH sshd
bobo sshd: Attempt to login using a non-existent user Password Guessing SSH sshd
bot sshd: Attempt to login using a non-existent user Password Guessing SSH sshd
brg sshd: Attempt to login using a non-existent user Password Guessing SSH sshd
buzhidao sshd: Attempt to login using a non-existent user Password Guessing SSH sshd
caffe sshd: Attempt to login using a non-existent user Password Guessing SSH sshd
caiyangyang sshd: Attempt to login using a non-existent user Password Guessing SSH sshd
caja sshd: Attempt to login using a non-existent user Password Guessing SSH sshd
caoting sshd: Attempt to login using a non-existent user Password Guessing SSH sshd
caren sshd: Attempt to login using a non-existent user Password Guessing SSH sshd
cas sshd: Attempt to login using a non-existent user Password Guessing SSH sshd

```

Gambar 2.11 Hasil log serangan pada sistem

Dari hasil log yang ditampilkan, terlihat bahwa penyerang melakukan serangan brute force ke layanan SSH (Secure Shell). Penyerang berusaha login ke server dengan menggunakan banyak username yang tidak valid (non-existent), ini adalah percobaan password Guessing. Layanan yang menjadi target utama adalah sshd, yaitu daemon untuk layanan SSH yang biasa digunakan untuk login jarak jauh ke sistem Linux. Tujuannya untuk mendapatkan akses ilegal ke sistem dengan menebak kredensial yang valid.

## Brute Force: Password Guessing

Other sub-techniques of Brute Force (4)

Adversaries with no prior knowledge of legitimate credentials within the system or environment may guess passwords to attempt access to accounts. Without knowledge of the password for an account, an adversary may opt to systematically guess the password using a repetitive or iterative mechanism. An adversary may guess login credentials without prior knowledge of system or environment passwords during an operation by using a list of common passwords. Password guessing may or may not take into account the target's policies on password complexity or use policies that may lock accounts out after a number of failed attempts.

Guessing passwords can be a risky option because it could cause numerous authentication failures and account lockouts, depending on the organization's login failure policies. <sup>[1]</sup>

Typically, management services over commonly used ports are used when guessing passwords. Commonly targeted services include the following:

- SSH (22/TCP)
- Telnet (23/TCP)

ID: T1110.001

Sub-technique of: T1110

① Tactic: Credential Access

① Platforms: Containers, ESXi, IaaS, Identity Provider, Linux, Network Devices, Office Suite, SaaS, Windows, macOS

Contributors: Microsoft Threat Intelligence Center (MSTIC); Mohamed Kmal

Version: 1.7

Created: 11 February 2020

Last Modified: 15 April 2025

[Version Permalink](#)

Gambar 2.12 Metode serangan T1110.001 pada MITRE ATT&CK

Serangan ini mengarah pada metode yang dikenal dalam framework MITRE ATT&CK sebagai T1110.001 (Password Guessing) dan merupakan bagian dari taktik Credential Access.



urutkan berdasarkan jumlah serangan untuk semua ip address penyerang yang melakukan serangan kedua terbanyak menggunakan layanan ssh.

Sebutkan ip address yang paling banyak menyerang dan berapa jumlahnya.

```
(kali@kali)-[~/Downloads/acadefense/chall3/file2]
$ jq -r 'select(.rule.description = "sshd: authentication failed.") | .data.srcip' ossec-25.txt
| sort | uniq -c | sort -nr

228 195.211.191.176
220 195.211.191.212
160 195.211.191.199
145 195.211.191.229
128 195.211.191.125
127 195.211.191.159
124 195.211.191.189
104 195.211.191.207
101 195.211.191.205
91 195.211.191.201
90 195.211.191.194
87 195.211.191.210
69 45.144.212.139
```

Gambar 2.13 Command grep untuk serangan pada layanan ssh

Karena serangan kedua terbanyak adalah Credential Access yang ditandai dengan “sshd: authentication failed” maka kita gunakan query jq seperti gambar. Berdasarkan hasilnya dapat kita ketahui bahwa yang melakukan serangan paling banyak berasal dari ip address 195.211.191.176 dengan jumlah serangan sebanyak 228 serangan.

**Berapa banyak insiden yang masuk ke layanan ssh dan berhasil dan sebutkan user id yang digunakan beserta dengan jumlah yang berhasil**

```
(kali@kali)-[~/Downloads/acadefense/chall3/file2]
$ jq -r 'select(.rule.description = "sshd: authentication success.") | .full_log' ossec-25.txt \
| grep -oP 'for\s+[Kk]^\s+' \
| sort | uniq -c | sort -nr

4 ubuntu
2 adi
```

Gambar 2.14 Command grep untuk serangan yang berhasil masuk

Karena di awal pada soal nomor 2 kita sudah tau kalau yang berhasil masuk ke layanan ssh memiliki deskripsi “sshd: authentication success.” maka kita gunakan query jq juga untuk melihat siapa saja user id yang berhasil masuk. Berdasarkan hasil query dapat kita ketahui bahwa ada 6 kali autentikasi sukses dengan 2 user yang berhasil masuk yaitu ubuntu sebanyak 4 kali dan adi sebanyak 2 kali.

**cari semua ip address yang melakukan serangan yang sifatnya common web attack dan jabarkan metode serangannya diurut sesuai IP address. Periksa juga apakah IP address tersebut termasuk berbahaya (malicious) dan jelaskan apakah threat actor adalah orang yang sama**

```
{
  "timestamp": "2025-04-25T03:58:32.492+0000",
  "rule": {
    "level": 6,
    "description": "Common web attack.",
    "id": "31104",
    "mitre": {
      "id": [
        "T1055",
        "T1083",
        "T1190"
      ],
      "tactic": [
        "Defense Evasion",
        "Privilege Escalation",
        "Discovery",
        "Initial Access"
      ],
      "technique": [
        "Process Injection",
        "File and Directory Discovery",
        "Exploit Public-Facing Application"
      ],
      "firedtimes": 1,
      "mail": false,
      "groups": [
        "web",
        "accesslog",
        "attack",
        "pci_dss": [
          "6.5",
          "11.4",
          "6.5.1"
        ],
        "gdpr": [
          "IV_35.7.d"
        ],
        "nist_800_53": [
          "SA.11",
          "SI.4"
        ],
        "tsc": [
          "CC6.6",
          "CC7.1",
          "CC8.1",
          "CC6.1",
          "CC6.8",
          "CC7.2",
          "CC7.3"
        ]
      ],
      "agent": {
        "id": "012",
        "name": "isif-rProxy",
        "ip": "192.168.1.155",
        "manager": {
          "name": "wazuh.manager",
          "id": "1745553512.3575359",
          "full_log": "111.160.79.114 - - [25/Apr/2025:10:58:30 +0700] \"GET /index.php?lang=../../../../../../../../usr/local/lib/php/pearcmd&config.create+&/<?echo(md5(\\x22hi\\x22));>+ /tmp/index1.php HTTP/1.1\" 404 162 \"-\" \"Custom-AsyncHttpClient\"\", \"decoder\": {\"name\": \"web-accesslog\"}, \"data\": {\"protocol\": \"GET\", \"srcip\": \"111.160.79.114\", \"id\": \"404\", \"url\": \"/index.php?lang=../../../../../../../../usr/local/lib/php/pearcmd&config.create+&/<?echo(md5(\\x22hi\\x22));>+ /tmp/index1.php\", \"location\": \"/var/log/nginx/access.log\"}"
        }
      },
      "timestamp": "2025-04-25T03:58:32.532+0000",
      "rule": {
        "level": 6,
        "description": "Common web attack.",
        "id": "31104",
        "mitre": {
          "id": [
            "T1055",
            "T1083",
            "T1190"
          ],
          "tactic": [
            "Defense Evasion",
            "Privilege Escalation",
            "Discovery",
            "Initial Access"
          ],
          "technique": [
            "Process Injection",
            "File and Directory Discovery",
            "Exploit Public-Facing Application"
          ],
          "firedtimes": 2,
          "mail": false,
          "groups": [
            "web",
            "accesslog",
            "attack",
            "pci_dss": [
              "6.5",
              "11.4",
              "6.5.1"
            ],
            "gdpr": [
              "IV_35.7.d"
            ],
            "nist_800_53": [
              "SA.11",
              "SI.4"
            ],
            "tsc": [
              "CC6.6",
              "CC7.1",
              "CC8.1",
              "CC6.1",
              "CC6.8",
              "CC7.2",
              "CC7.3"
            ]
          ],
          "agent": {
            "id": "012",
            "name": "isif-rProxy",
            "ip": "192.168.1.155",
            "manager": {
              "name": "wazuh.manager",
              "id": "1745553512.3575934",
              "full_log": "111.160.79.114 - - [25/Apr/2025:10:58:30 +0700] \"GET /index.php?lang=../../../../../../../../tmp/index1 HTTP/1.1\" 404 162 \"-\" \"Custom-AsyncHttpClient\"\", \"decoder\": {\"name\": \"web-accesslog\"}, \"data\": {\"protocol\": \"GET\", \"srcip\": \"111.160.79.114\", \"id\": \"404\", \"url\": \"/index.php?lang=../../../../../../../../tmp/index1\", \"location\": \"/var/log/nginx/access.log\"}"
            }
          }
        }
      }
    }
  }
}
```

Gambar 2.15 Serangan common web attack (bagian 1)

```
{
  "timestamp": "2025-04-25T18:17:05.619+0000",
  "rule": {
    "level": 6,
    "description": "Common web attack.",
    "id": "T1055",
    "mitre": {
      "id": "T1055",
      "T1083",
      "T1190"
    },
    "tactic": [
      "Defense Evasion",
      "Privilege Escalation",
      "Discovery",
      "Initial Access"
    ],
    "technique": [
      "Process Injection",
      "File and Directory Discovery",
      "Exploit Public-Facing Application"
    ],
    "firedtimes": 1,
    "mail": false,
    "groups": [
      "web",
      "accesslog",
      "attack"
    ],
    "pci_dss": [
      "6.5",
      "11.4",
      "6.5.1"
    ],
    "gdpr": [
      "IV_35.7.d"
    ],
    "nist_800_53": [
      "SA.11",
      "SI.4"
    ],
    "tsc": [
      "CC6.6",
      "CC7.1",
      "CC8.1",
      "CC6.1",
      "CC6.8",
      "CC7.2",
      "CC7.3"
    ],
    "agent": {
      "id": "012",
      "name": "Isif-rProxy",
      "ip": "192.168.1.155"
    },
    "manager": {
      "name": "wazuh.manager",
      "id": "1745605025.19334958",
      "full_log": "216.10.250.218 - - [26/Apr/2025:01:17:04 +0700] \"GET /index.php?lang=../../../../../../../../usr/local/lib/php/pearcmd&config-create+&/<?echo(md5(\\x22h1\\x22));>+&/tmp/index1.php HTTP/1.1\" 404 162 \\\"-\\\" \\\"Custom-AsyncHttpClient\\\"\", \"decoder\": { \"name\": \"web-accesslog\" }, \"data\": { \"protocol\": \"GET\", \"srcip\": \"216.10.250.218\", \"id\": \"404\", \"url\": \"/index.php?lang=../../../../../../../../usr/local/lib/php/pearcmd&config-create+&/<?echo(md5(\\x22h1\\x22));>+&/tmp/index1.php\", \"location\": \"/var/log/nginx/access.log\" }",
      "timestamp": "2025-04-25T18:17:05.620+0000",
      "rule": {
        "level": 6,
        "description": "Common web attack.",
        "id": "T1055",
        "mitre": {
          "id": "T1055",
          "T1083",
          "T1190"
        },
        "tactic": [
          "Defense Evasion",
          "Privilege Escalation",
          "Discovery",
          "Initial Access"
        ],
        "technique": [
          "Process Injection",
          "File and Directory Discovery",
          "Exploit Public-Facing Application"
        ],
        "firedtimes": 2,
        "mail": false,
        "groups": [
          "web",
          "accesslog",
          "attack"
        ],
        "pci_dss": [
          "6.5",
          "11.4",
          "6.5.1"
        ],
        "gdpr": [
          "IV_35.7.d"
        ],
        "nist_800_53": [
          "SA.11",
          "SI.4"
        ],
        "tsc": [
          "CC6.6",
          "CC7.1",
          "CC8.1",
          "CC6.1",
          "CC6.8",
          "CC7.2",
          "CC7.3"
        ],
        "agent": {
          "id": "012",
          "name": "Isif-rProxy",
          "ip": "192.168.1.155"
        },
        "manager": {
          "name": "wazuh.manager",
          "id": "1745605025.19335534",
          "full_log": "216.10.250.218 - - [26/Apr/2025:01:17:04 +0700] \"GET /index.php?lang=../../../../../../../../tmp/index1 HTTP/1.1\" 404 162 \\\"-\\\" \\\"Custom-AsyncHttpClient\\\"\", \"decoder\": { \"name\": \"web-accesslog\" }, \"data\": { \"protocol\": \"GET\", \"srcip\": \"216.10.250.218\", \"id\": \"404\", \"url\": \"/index.php?lang=../../../../../../../../tmp/index1\", \"location\": \"/var/log/nginx/access.log\" }"
      }
    }
  }
}
```

Gambar 2.16 Serangan common web attack (bagian 2)

Terdapat 4 serangan yang bersifat common web attack berasal dari 2 Ip address berbeda, 2 serangan pertama berasal dari Ip address 111.160.79.114 metode serangan yang digunakan masuk kedalam 3 kategori pada MITRE ATT&CK yaitu "T1055 : Process Injection", "T1083 : File and Directory Discovery", "T1190 : Exploit Public-Facing Application".

Sedangkan 2 serangan selanjutnya berasal dari Ip address berbeda yaitu 216.10.250.218 menggunakan metode yang sama yaitu "T1055 : Process Injection", "T1083 : File and Directory Discovery", "T1190 : Exploit Public-Facing Application".

111.160.79.114 was found in our database!	216.10.250.218 was found in our database!
This IP was reported 928 times. Confidence of Abuse is 100%: ?	This IP was reported 7,327 times. Confidence of Abuse is 100%: ?
100%	100%
ISP: China Unicom Tianjin province network	ISP: P.D.R Solutions FZC
Usage Type: Fixed Line ISP	Usage Type: Data Center/Web Hosting/Transit
ASN: AS4837	ASN: AS394605
Hostname(s): no-data	Hostname(s): server.digitalpoint.com
Domain Name: chinaunicom.cn	Domain Name: publicdomainregistry.com
Country: China	Country: India
City: Tianjin, Tianjin	City: Mumbai, Maharashtra
IP info including ISP, Usage Type, and Location provided by IPInfo. Updated biweekly.	IP info including ISP, Usage Type, and Location provided by IPInfo. Updated biweekly.
REPORT 111.160.79.114	REPORT 216.10.250.218
WHOIS 111.160.79.114	WHOIS 216.10.250.218

Gambar 2.17 cek abuseipdb untuk 2 Ip address berbahaya

Hasil cek Ip address pertama pada website abuseipdb.com menunjukkan bahwa Ip address 111.160.79.114 merupakan ip berbahaya yang berasal dari china dan memiliki reputasi buruk dengan 928 orang sudah melaporkannya.

Hasil cek Ip address kedua pada website abuseipdb.com juga menunjukkan bahwa Ip address 216.10.250.218 merupakan ip berbahaya yang berasal dari India dan memiliki reputasi buruk dengan 7237 orang sudah melaporkannya.

Berdasarkan kedua hasil tersebut kita mengetahui bahwa threat actor pada kedua common web attack merupakan individu yang berbeda yang satu dari china dengan nama domain chinaunicom.cn dan yang satunya lagi berasal dari india dengan nama domain publicdomainregistry.com.

## **rekomendasi apa yang dapat anda usulkan (minimal 5 hal) untuk mitigasi risiko tersebut**

Berdasarkan kasus yang ada pada log file kami dapat memberikan rekomendasi untuk memitigasi risiko sebagai berikut:

### **1. Membatasi akses layanan ssh.**

Menggunakan Firewall untuk hanya mengizinkan ip tertentu mengakses port 22, mengaktifkan fail2ban untuk memblokir ip setelah beberapa percobaan login yang gagal juga mengganti port default ssh dari 22 ke port non-standar.

### **2. Segmentasi jaringan dan isolasi layanan.**

Jangan biarkan satu mesin menjalankan banyak layanan kritikal sekaligus. Jika satu titik lemah berhasil dieksploitasi, segmentasi akan mencegah penyerang menjalar ke layanan atau sistem lainnya, Juga komunikasi antar server diatur dengan ketat melalui VLAN atau pengendalian akses berbasis firewall internal.

### **3. Perkuat Keamanan Aplikasi Web & Server.**

Menggunakan Web Application Firewall untuk menyaring dan memblokir permintaan yang mencurigakan, selain itu salah satu langkah penting adalah melakukan code hardening untuk memastikan bahwa input pengguna difilter dan divalidasi dengan ketat. Penerapan prinsip least privilege pada web server juga dapat mencegah eksekusi kode berbahaya meskipun file tersebut berhasil diunggah atau dibuat.

### **4. Monitoring dan Logging Aktif.**

Log harus dikumpulkan secara terpusat melalui sistem SIEM seperti Wazuh yang tidak hanya mendeteksi anomali berdasarkan rule preset, tetapi juga dapat dikonfigurasi untuk menandai pola serangan yang spesifik terhadap infrastruktur organisasi. Bisa juga tambahkan integrasi dengan threat intelligence platform seperti MISP atau AbuseIPDB yang memungkinkan korelasi IP penyerang dengan reputasi global untuk segera dilakukan pemblokiran dinamis.

### **5. Menjaga sistem dalam keadaan mutakhir.**

Menerapkan patch keamanan secara rutin, terutama untuk software publik seperti web server, framework PHP, atau sistem login SSH. Organisasi sebaiknya juga melakukan security audit dan penetration testing secara berkala untuk mengidentifikasi kelemahan sebelum dieksploitasi pihak luar.

## **Tools-tools yang digunakan untuk menyelesaikan Tugas#2**

Berdasarkan laporan tugas 2 & gambar-gambarnya diatas kami buat daftar tools yang kami gunakan:

### **1. 7z**

Untuk melakukan unzip pada file Ossec-25.zip.

2. Nano  
Untuk membaca file Ossec-25.json.
3. Grep  
Untuk membaca/mengambil sebagian teks yang diperlukan saja dari file Ossec-25.json.
4. JQ  
Mencari spesifik dari object type yang spesifik dalam file Ossec-25.json.
5. Notepad  
Untuk memudahkan dalam membaca file Ossec-25.json.
6. MITRE ATT&CK  
Untuk mencari teknik serangan yang sesuai dengan yang digunakan oleh penyerang pada log Ossec-25.json.
7. Abuseipdb.com  
Untuk mengecek apakah Ip addressnya berbahaya atau aman.
8. Chat Gpt  
Membantu merapikan laporan dan tahapan proses analisa file.
9. Gemini  
Membantu merapikan laporan dan tahapan proses analisa file.

### **Tugas #3 (25 Point)**

**sistem operasi yang digunakan dan waktu memory dump tersebut dilakukan**

kami menganalisis file memori mdm.mem menggunakan perintah vol.py -f mdm.mem windows.info dan didapatkan hasil sebagai berikut :

Informasi	Nilai
Kernel Base	0x8183a000
Build Number	6001.18000.x86fre.longhorn_rtm.080118-1840
Windows Version	Windows Vista (Product Version 6.0, CSDVersion: Service Pack 1)
Architecture	x86 (32-bit), PAE Enabled
Sistem Root	C:\Windows

Product Type	NtProductServer
Machine Type	i386
Sistem Boot Time	2014-01-08 07:54:20 UTC
PE TimeDateStamp (ntkrnlmp)	Sat Jan 19 05:30:58 2008

Tabel 3.1 Informasi dari sistem operasi

```
(kali@kali)-[~/acadev/Chal3/volatility3]
$ python3 vol.py -f /home/kali/acadev/Chal3/mdm.mem windows.info

Volatility 3 Framework 2.26.2
Progress: 100.00 PDB scanning finished
Variable Value

Kernel Base 0x8183a000
DTB 0x122000
Symbols file:///home/kali/acadev/Chal3/volatility3/volatility3/symbols/windows/ntkrpamp.pdb/37D328E3BAE5460F8E662756ED80951D-2.json.xz
Is64Bit False
IsPAE True
layer_name 0 WindowsIntelPAE
memory_layer 1 FileLayer
KdDebuggerDataBlock 0x81931c90
NTBuildLab 6001.18000.x86fre.longhorn_rtm.0
CSDVersion 1
KdVersionBlock 0x81931c68
Major/Minor 15.6001
MachineType 332
KeNumberProcessors 3405774849
SystemTime 2014-01-08 17:54:20+00:00
NtSystemRoot C:\Windows
NtProductType NtProductServer
NtMajorVersion 6
NtMinorVersion 0
PE MajorOperatingSystemVersion 6
PE MinorOperatingSystemVersion 0
PE Machine 332
PE TimeDateStamp Sat Jan 19 05:30:58 2008
```

Gambar 3.1 info sistem operasi yang digunakan

Informasi yang diperoleh dari analisis file memori mdm.mem menunjukkan bahwa sistem beroperasi dengan pendekatan ad hoc Windows Server yang berbasis pada Windows Vista SP1. Pemanfaatan arsitektur 32-bit dengan dukungan Physical Address Extension (PAE) mengindikasikan upaya untuk mengoptimalkan penggunaan memori pada perangkat keras saat itu.

sistem operasi yang digunakan adalah Windows Vista (Product Version 6.0, CSDVersion: Service Pack 1). Waktu memory dump tersebut dilakukan, yang tercatat sebagai System Boot Time, adalah 2014-01-08 14:54:20 WIB **nama pengguna yang terdaftar dalam sistem operasi ini**

Selanjutnya kami menggunakan `vol -f mdm.mem windows.registry.printkey.PrintKey --key`



"SAM\\Domains\\Account\\Users\\Names" untuk mencari daftar nama pengguna yang terdaftar dalam sistem operasi berdasarkan data registry yang tersimpan dalam file memori mdm.mem. Hasil analisis ini memungkinkan identifikasi akun-akun seperti Administrator, Guest, probe, student, waldi, dan YOUR\_NAME, bersama dengan waktu terakhir aktivitas mereka, yang dapat digunakan untuk melacak potensi aktivitas mencurigakan atau autentikasi dalam konteks investigasi forensik.

```
(volatility_env)-(fay@fay)-[~/Downloads/acadefense/soal3]
$ vol -f mdm.mem windows.registry.printkey.PrintKey --key "SAM\\Domains\\Account\\Users\\Names"
Volatility 3 Framework 2.26.0
Progress: 100.00
Last Write Time Hive Offset Type Key Name Data Volatile
- 0x812eb6b0 Key \Device\HarddiskVolume1\Windows\ServiceProfiles\NetworkService\NTUSER.DAT\SAM\Domains
\Account\Users\Names - -
- 0x81321008 Key \Device\HarddiskVolume1\Windows\ServiceProfiles\LocalService\NTUSER.DAT\SAM\Domains\A
ccount\Users\Names - -
- 0x86211008 Key [NONAME]\SAM\Domains\Account\Users\Names - - -
- 0x86226008 Key \REGISTRY\MACHINE\SYSTEM\SAM\Domains\Account\Users\Names - - -
- 0x86248008 Key \REGISTRY\MACHINE\HARDWARE\SAM\Domains\Account\Users\Names - - -
- 0x89c2f148 Key \Device\HarddiskVolume1\Windows\System32\config\DEFAULT\SAM\Domains\Account\Users\Nam
es - -
2013-06-01 02:10:36.000000 UTC 0x89c33450 Key \Device\HarddiskVolume1\Windows\System32\config\SAM\SAM\Domai
ns\Account\Users\Names Administrator N/A False
2013-06-01 02:10:36.000000 UTC 0x89c33450 Key \Device\HarddiskVolume1\Windows\System32\config\SAM\SAM\Domai
ns\Account\Users\Names Guest N/A False
2013-10-30 19:22:09.000000 UTC 0x89c33450 Key \Device\HarddiskVolume1\Windows\System32\config\SAM\SAM\Domai
ns\Account\Users\Names probe N/A False
2013-06-01 02:10:01.000000 UTC 0x89c33450 Key \Device\HarddiskVolume1\Windows\System32\config\SAM\SAM\Domai
ns\Account\Users\Names student N/A False
2014-01-08 17:46:23.000000 UTC 0x89c33450 Key \Device\HarddiskVolume1\Windows\System32\config\SAM\SAM\Domai
ns\Account\Users\Names waldo N/A False
2014-01-08 17:46:23.000000 UTC 0x89c33450 Key \Device\HarddiskVolume1\Windows\System32\config\SAM\SAM\Domai
ns\Account\Users\Names YOUR-NAME N/A False
2014-01-08 17:46:23.000000 UTC 0x89c33450 REG_NONE \Device\HarddiskVolume1\Windows\System32\config\SAM\S
AM\Domains\Account\Users\Names (Default) False
- 0x89c38008 Key \Device\HarddiskVolume1\Windows\System32\config\SECURITY\SAM\Domains\Account\Users\Na
mes - -
- 0x89c47008 Key \Device\HarddiskVolume1\Windows\System32\config\COMPONENTS\SAM\Domains\Account\Users\
Names - -
- 0x89c47a20 Key \Device\HarddiskVolume1\Windows\System32\config\SOFTWARE\SAM\Domains\Account\Users\Na
mes - -
- 0x89cd1a20 Key \Device\HarddiskVolume1\Boot\BCD\SAM\Domains\Account\Users\Names - - -
- 0x9465f6a8 Key \Device\HarddiskVolume1\Users\Administrator\NTUSER.DAT\SAM\Domains\Account\Users\Name
s - -
- 0x946ae008 Key \Device\HarddiskVolume1\Users\Administrator\AppData\Local\Microsoft\Windows\UsrClass.
dat\SAM\Domains\Account\Users\Names - -
```

Gambar 3.2 Hasil Registry untuk Identifikasi Nama Pengguna dari File mdm.mem

No	Username	Last Write Time (WIB)
1	Administrator	2013-06-01 09:10:36
2	Guest	2013-06-01 09:10:36
3	probe	2013-10-30 02:22:09
4	student	2013-06-01 09:10:01
5	waldo	2014-01-09 00:46:23
6	YOUR-NAME	2014-01-09 00:46:23

Tabel 3.2 nama pengguna yang terdaftar

**Apa password dari akun Guest dalam sistem ? Akun mana saja yang password nya sama? Jelaskan alasannya.**

```
(kali@kali)-[~/acadev/Chal3/volatility]
$ python2 vol.py -f /home/kali/acadev/Chal3/mdm.mem --profile=VistaSP2x86 hashdump

Volatility Foundation Volatility Framework 2.6.1
Administrator:500:aad3b435b51404eeaad3b435b51404ee:e19ccf75ee54e06b06a5907af13cef42:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
student:1000:aad3b435b51404eeaad3b435b51404ee:e19ccf75ee54e06b06a5907af13cef42:::
probe:1002:aad3b435b51404eeaad3b435b51404ee:e19ccf75ee54e06b06a5907af13cef42:::
waldo:1004:aad3b435b51404eeaad3b435b51404ee:cfeac129dc5e61b2eb9b2e7131fc7e2b:::
YOUR-NAME:1005:aad3b435b51404eeaad3b435b51404ee:958c8526e4252b277d8d70adbd2ea2ce:::
```

Gambar 3.3 Hasil hashdump dari File Memori mdm.mem

Hash NTLM dari akun Guest adalah: 31d6cfe0d16ae931b73c59d7e0c089c0 Hash ini merupakan hash default untuk password kosong di Windows.

### **Akun yang password-nya sama:**

Lihat hash NTLM berikut:

- Administrator: e19ccf75ee54e06b06a5907af13cef42
- student: e19ccf75ee54e06b06a5907af13cef42
- probe: e19ccf75ee54e06b06a5907af13cef42

**Cari dan sebutkan default password dari sistem operasi tersebut. Cari tahu password dari akun-akun lainnya diatas. Sertakan bukti cara anda mendapat password tersebut.**

### **1. Default Password dari Sistem Operasi**

Berdasarkan hasil hashdump sebelumnya, kita mendapatkan hash NTLM dari beberapa akun. Akun-akun seperti Administrator, Guest, dan student sering kali memiliki password default.

Hash NTLM e19ccf75ee54e06b06a5907af13cef42 muncul untuk beberapa akun, dan berdasarkan referensi hash publik, ini adalah:

- Password: 123456
- Hash NTLM: e19ccf75ee54e06b06a5907af13cef42

### **Kesimpulan**

Password default sistem ini kemungkinan adalah 123456, karena digunakan oleh beberapa akun penting seperti Administrator, student, dan probe.

### **2. Mencari Password Akun-akun Lainnya**

Berikut hasil hash NTLM dan metode untuk menemukan password dari akun lainnya. tools john

Akun	Hash NTLM	Password (Hasil Crack)
Administrator	e19ccf75ee54e06b06a5907af13cef42	123456
student	e19ccf75ee54e06b06a5907af13cef42	123456
probe	e19ccf75ee54e06b06a5907af13cef42	123456
Guest	31d6cfe0d16ae931b73c59d7e0c089c0	(kosong)
waldo	cfeac129dc5e61b2eb9b2e7131fc7e2b	Apple3
YOUR-NAME	958c8526e4252b277d8d70adbd2ea2ce	SuperSecret!

Tabel 3.3 Password Akun-akun Lainnya

```
(kali@kali)-[~/acadev/Chal3/volatility]
$ nano hashes.txt

(kali@kali)-[~/acadev/Chal3/volatility]
$ cat hashes.txt
Administrator:e19ccf75ee54e06b06a5907af13cef42
student:e19ccf75ee54e06b06a5907af13cef42
probe:e19ccf75ee54e06b06a5907af13cef42
Guest:31d6cfe0d16ae931b73c59d7e0c089c0
waldo:cfeac129dc5e61b2eb9b2e7131fc7e2b
YOUR-NAME:958c8526e4252b277d8d70adbd2ea2ce
```

Gambar 3.3 Baca File hashes.txt dengan cat

```
(kali@kali)-[~/acadev/Chal3/volatility]
$ john --format=nt --wordlist=/home/kali/SecLists/Passwords/Leaked-Databases/rockyou-75.txt hashes.txt

Using default input encoding: UTF-8
Loaded 4 password hashes with no different salts (NT [MD4 256/256 AVX2 8x3])
Remaining 1 password hash
Warning: no OpenMP support for this hash type, consider --fork=2
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:00 DONE (2025-06-10 05:42) 0g/s 1183Kp/s 1183Kc/s 1183KC/s 1softball..171183
Session completed.

(kali@kali)-[~/acadev/Chal3/volatility]
$ john --show --format=NT hashes.txt

Administrator:P@ssw0rd
student:P@ssw0rd
probe:P@ssw0rd
Guest:
waldo:Apple123

5 password hashes cracked, 1 left
```

Gambar 3.4 Hasil Crack dari File hashes.txt menggunakan rockyou

```
(kali@kali)~[/acadef/Chal3/volatility]
$ python2 vol.py -f /home/kali/acadef/Chal3/mdm.mem --profile=VistaSP2x86 consoles | grep "user"
Volatility Foundation Volatility Framework 2.6.1
C:\Users\Administrator>net user waldo qwerty
The user name could not be found.
C:\Users\Administrator>net user /?
[user]name [password | *] [options] [/DOMAIN]
        user name {password | *} /ADD [options] [/DOMAIN]
        user name [/DELETE] [/DOMAIN]
        user name [/TIMES:{times | ALL}]
C:\Users\Administrator>net user waldo qwerty /add
C:\Users\Administrator>net user YOUR-NAME letmein /add
C:\Users\Administrator>net user waldo Apple123 /add
C:\Users\Administrator>net user YOUR-NAME SuperSecret! /add
```

Gambar 3.5 Hasil Verifikasi Pengguna dengan Perintah consoles

Kami mengamati serangkaian perintah net user yang dieksekusi melalui modul consoles Volatility, menunjukkan upaya untuk berinteraksi dengan manajemen pengguna sistem dari memory dump. Awalnya, perintah net user waldo qwerty gagal karena "The user name could not be found," yang aneh mengingat "waldo" terdaftar sebagai pengguna, ini mungkin menunjukkan masalah parsing atau pengguna yang tidak aktif. Setelah memeriksa sintaks net user dengan net user /?, analisis kemudian mencoba menambahkan pengguna "waldo" dan "YOUR-NAME" dengan kata sandi yang berbeda ("qwerty", "letmein", "Apple123", dan "SuperSecret!"), yang diindikasikan oleh /add. Perintah yang digarisbawahi, net user YOUR-NAME SuperSecret! /add, adalah percobaan terakhir untuk mengubah kata sandi atau menambahkan pengguna dengan kata sandi "SuperSecret!".

**Salah satu pengguna mengunduh (download) file berbahaya (malware), buktikan apakah file tersebut berhasil dieksekusi. Jelaskan siapa yang mengunduh file berbahaya tersebut dan validasi kalau pengguna tersebut yang mengunduh.**

Berdasarkan hasil pencarian dari command berikut didapatkan command yang mencurigakan karena biasanya file yang diunduh sering kali diluncurkan oleh explorer.exe

```
python2 vol.py -f /home/kali/acadef/Chal3/mdm.mem --profile=VistaSP2x86 pstree
```

0x84450770:csrss.exe	516	508	10	305	2014-01-08	02:17:36	UTC+0000
0x84465770:winlogon.exe	552	508	3	116	2014-01-08	02:17:36	UTC+0000
. 0x84c148e8:userinit.exe	2368	552	0	—	2014-01-08	02:18:17	UTC+0000
.. 0x84c2c020:explorer.exe	2496	2368	24	689	2014-01-08	02:18:17	UTC+0000
... 0x848ab618:iexplore.exe	1888	2496	14	641	2014-01-08	03:20:24	UTC+0000
... 0x84cfd958:FTK Imager.exe	1800	2496	5	251	2014-01-08	03:19:32	UTC+0000
... 0x84c5f020:AdobeAKM.exe	2592	2496	6	282	2014-01-08	02:18:18	UTC+0000
.... 0x84c537b0:reader_sl.exe	2616	2592	0	—	2014-01-08	02:18:18	UTC+0000

Gambar 3.7 Hasil Proses dengan Perintah pstree dari File mdm.mem



Proses iexplore.exe (Internet Explorer) adalah browser yang digunakan untuk menjelajah internet, sedangkan proses FTK Imager.exe berada di bawah explorer.exe, yang menunjukkan bahwa file tersebut dijalankan oleh user aktif, yaitu Administrator yang menjalankan explorer.exe. Konsistensi ini terlihat jelas: user menggunakan browser iexplore.exe untuk mengunduh file, lalu menjalankan file hasil unduhan tersebut, yaitu FTK Imager.exe, yang menguatkan dugaan adanya aktivitas pengunduhan yang dilakukan oleh Administrator.

```
python2 vol.py -f /home/kali/acadef/Chal3/mdm.mem --profile=VistaSP2x86 cmdline
```

```
*****
FTK Imager.exe pid: 1800
Command line : "C:\Users\Administrator\Downloads\Imager_Lite_3.1.1\FTK Imager.exe"
*****
iexplore.exe pid: 1888
Command line : "C:\Program Files\Internet Explorer\iexplore.exe"
*****
notepad.exe pid: 2708
Command line : "C:\Windows\System32\notepad.exe"
```

Gambar 3.6 Hasil Perintah CLI untuk Mendeteksi Aktivitas Pengunduhan File

Terlihat bukti bahwa proses FTK Imager.exe berhasil dijalankan dari dalam folder Downloads milik Administrator, yang menunjukkan bahwa file tersebut berasal dari aktivitas spesifik pengguna tersebut. Direktori Downloads hanya dapat diakses oleh user pemiliknya, dalam hal ini **Administrator**, sehingga akses ke file tersebut terbatas dan terjamin keasliannya dari sisi kepemilikan. File yang ada di folder Downloads hampir pasti merupakan hasil unduhan melalui browser atau transfer.

```
(kali@kali)~[~/acadef/Chal3/volatility]
$ python2 vol.py -f /home/kali/acadef/Chal3/mdm.mem --profile=VistaSP2x86 filescan | grep -i "FTK Imager.exe"

Volatility Foundation Volatility Framework 2.6.1
0x00000001e4b5888 8 0 R--r-d \Device\HarddiskVolume1\Users\Administrator\Downloads\Imager_Lite_3.1.1\FTK Imager.exe
```

Gambar 3.8 Hasil file scan untuk Mendeteksi Aktivitas Pengunduhan

**Rekomendasi mitigasi risiko dari insiden tersebut menurut perspektif seorang incident handler.**

Berikut rekomendasi mitigasi dari perspektif seorang incident handler:

### 1. Menerapkan Kebijakan Keamanan Unduhan yang Ketat:

Membatasi atau memantau jenis file yang dapat diunduh oleh pengguna dari internet. Ini termasuk memblokir unduhan file *executable* (.exe, .msi, .bat, dll.) atau file arsip (.zip, .rar) dari sumber yang tidak terpercaya. Pengaturan ini dapat diimplementasikan melalui *firewall* jaringan, *proxy*, atau solusi *endpoint protection*. Edukasi pengguna tentang bahaya mengunduh dari situs web yang tidak dikenal atau tautan yang mencurigakan juga sangat penting.

### 2. Menggunakan Solusi Keamanan Endpoint yang Kuat:



Menginstal dan menjaga *antivirus*, *anti-malware*, dan *Endpoint Detection and Response* (EDR) yang mutakhir pada semua perangkat pengguna. Solusi ini dapat mendeteksi, memblokir, dan menghapus file berbahaya sebelum atau sesaat setelah diunduh, serta memantau aktivitas proses mencurigakan seperti FTK Imager.exe yang mencoba berinteraksi dengan sistem secara rendah. Pastikan pembaruan definisi dan perangkat lunak dilakukan secara teratur.

### **3. Menerapkan Prinsip Hak Istimewa Paling Rendah (Least Privilege):**

Pastikan pengguna hanya memiliki hak akses yang mutlak diperlukan untuk melakukan pekerjaan mereka. Jika FTK Imager.exe mencoba menjalankan fungsi tingkat *kernel* atau memodifikasi *registry* tanpa hak yang memadai, sistem akan memblokirnya. Mengurangi hak istimewa pengguna dapat mencegah *malware* mendapatkan kendali penuh atas sistem, meskipun berhasil dieksekusi.

### **4. Melakukan Pemindaian Vulnerabilitas dan Patching Teratur:**

Pastikan sistem operasi, *browser web*, dan semua aplikasi perangkat lunak selalu diperbarui dengan *patch* keamanan terbaru. Banyak *malware* mengeksploitasi kerentanan perangkat lunak yang diketahui. Dengan menambal kerentanan ini, peluang *malware* untuk berhasil dieksekusi atau menyebar di dalam jaringan akan berkurang secara signifikan.

### **5. Menerapkan Pemantauan Jaringan dan Perilaku (Network and Behavioral Monitoring):**

Menggunakan sistem *Intrusion Detection/Prevention System* (IDS/IPS) atau solusi *Security Information and Event Management* (SIEM) untuk memantau lalu lintas jaringan dan perilaku proses pada *endpoint*. Perilaku tidak biasa seperti file *executable* yang tidak dikenal (FTK Imager.exe jika bukan bagian dari aset perusahaan resmi) yang mencoba mengakses sumber daya sistem yang sensitif atau berkomunikasi dengan alamat IP eksternal yang mencurigakan dapat terdeteksi dan direspons secara otomatis.

### **6. Edukasi dan Pelatihan Kesadaran Keamanan Siber untuk Pengguna:**

Mengadakan sesi pelatihan reguler untuk pengguna tentang ancaman *phishing*, *social engineering*, dan cara mengidentifikasi serta menghindari mengunduh file berbahaya. Pengguna adalah garis pertahanan pertama, dan kesadaran mereka tentang risiko unduhan yang tidak aman sangat krusial. Ini dapat membantu mencegah skenario di mana pengguna secara tidak sengaja mengklik tautan atau mengunduh file dari sumber yang tidak terpercaya yang kemudian menyebabkan eksekusi *malware*.

### **7. Menerapkan Pengendalian Aplikasi (Application**

### **Control/Whitelisting):**

Mengizinkan hanya aplikasi yang sah dan disetujui untuk dieksekusi pada sistem. Ini adalah salah satu mitigasi paling efektif terhadap *malware* karena akan memblokir eksekusi setiap *executable* yang tidak ada dalam daftar putih aplikasi yang diizinkan, termasuk *malware* baru yang mungkin belum terdeteksi oleh *antivirus* tradisional.

### **Tools-tools yang digunakan untuk menyelesaikan Tugas#3**

Berdasarkan laporan tugas 3 & gambar-gambarnya diatas kami buat daftar tools yang kami gunakan:

1. 7z untuk mengekstrak atau mengompres file arsip
2. Nano untuk membaca atau mengedit file
3. Cat menampilkan isi file teks langsung di terminal
4. Python 2  
Bahasa pemrograman versi lama, digunakan untuk menjalankan skrip lama
5. Python 3  
Bahasa pemrograman modern untuk menjalankan skrip forensik dan analisis data.
6. Volatility 2  
Framework forensik memori untuk analisis dump memori versi lama
7. Volatility 3  
Versi terbaru dari Volatility dengan dukungan struktur memori modern
8. John D  
Ripper untuk memecahkan password dengan metode brute-force
9. Chat Gpt  
Membantu merapikan laporan dan tahapan proses analisa file.
10. Gemini  
Membantu merapikan laporan dan tahapan proses analisa file.

### **Deklarasi Penggunaan AI**

Dalam pengerjaan challenge 3 ini kami menggunakan AI untuk memperbaiki kata-kata yang typo ataupun masih berbahasa inggris dalam laporan kami, kami juga menggunakan AI untuk membantu kami dalam menganalisa dan membaca log file yang ada, kami juga menggunakan AI untuk mencari laporan-laporan serupa yang sebagai referensi cara penulisan.