



## SOC Alert Management and Incident Response

### Alert Priority Levels

In a Security Operations Center, alerts are generated in the range of hundreds to thousands per day and are produced by security tools like SIEM, EDR, IDS, and firewalls. It is not possible to work on all the alerts simultaneously. The way to manage all these alerts is to prioritize them according to their level of criticalness. Prioritizing the alerts enables the security team to respond to them in less time and prevents serious security events.

Alerts have been categorized into four general levels based on their impact and level of urgency:

**1.Critical:** Highly dangerous with active exploitation or immediate threat to the organization.

Example: Exploitation of the Log4Shell vulnerability on internet-facing servers.

**2.High:** High-priority alerts indicate serious security threats that could lead to major damage if not handled quickly.

Example: Unauthorized administrative access, privilege escalation.

**3.Medium:** Suspicious activity but with little impact or partial compromise.

Example: repeated failed login attempts.

**4.Low:** These are minimal alerts or information alerts.

Examples: Port scans, policy violations without exploit.

### Criteria Used to Assign Alert Priority

#### *1.Asset Criticality*

The criticality and importance of the impacted system are basically very important.

High value targets: Production servers, Databases, Domain Controllers

Low value assets: Test machines, Lab systems



An alert appearing on a production server is considered to be far more significant than a test environment alert for the same incident.

## ***2. Exploit Likelihood***

Alerts related to vulnerabilities with public exploits or active attacks are prioritized higher.

CVEs with PoC exploits publicly available

Examples:

Log4Shell (CVE-2021-44228) allows remote code execution and has publicly available exploits, making it highly dangerous.

## ***3. Business Impact***

SOC teams evaluate how an alert may impact an organization:

Financial loss, Service downtime, Legal or Compliance Issues, Reputation Damage

## **Scoring Systems in Alert Prioritization**

### ***CVSS (Common Vulnerability Scoring System):***

CVSS is a standardized scoring system used to measure vulnerability severity.

#### ***Base Score:***

This shows how dangerous a vulnerability is in general. It looks at how easy it is to exploit and how much damage it can cause, without considering time or environment.

#### ***Temporal Score:***

This shows how the risk changes over time. If an exploit is easily available and no patch exists, the risk is higher. When a fix is released, the risk becomes lower.

#### ***Environmental Score:***

This adjusts the risk based on the organization. The same vulnerability is more serious on important systems like production servers than on test systems.



## Incident Classification

Incident classification involves the identification, categorization, and labelling of incidents in accordance with their nature, method of attack, and impact. Proper incident classification in a SOC environment will help analysts understand what actually happened during an attack, assign the appropriate response action accordingly, and report on consistent outputs across teams.

Security incidents normally are categorized depending on the threat type involved. Some of the major categories of incidents include the following:

### ***1.Malware incidents***

Events related to malicious software: virus, worm, trojan, spyware, ransomware.

Examples:

- Ransomware encrypting files
- Spyware stealing credentials

### ***2.Phishing Incidents***

Social engineering attacks: These are attacks in which attackers trick users into giving away sensitive information.

Examples:

- Fake login emails stealing credentials
- Malicious email attachments

### ***3.DDoS - Distributed Denial of Service***

A point of attack used to flood systems or networks with traffic in order to make services unavailable.

Examples:

- Website outage due to traffic flooding
- Network bandwidth exhaustion



## ***4.Data Exfiltration***

Unapproved data transfer of sensitive information outside the organization.

Examples:

Sensitive files uploaded on to external cloud storage

Data sent to attacker-controlled servers

## **Incident Classification Using Taxonomies**

In a SOC, taxonomies serve as means of standardization for incident classification. There exist certain frameworks such as MITRE ATT&CK, ENISA Incident Taxonomy, and VERIS. They assist incident response teams in formally articulating incidents so that everyone speaks the same language. They eliminate confusion and allow for effective communication with respect to incident response analysis and management.

Example:

When an employee is sent a fraudulent email with the intention of clicking a link to provide login information, the attack is considered a phishing incident. Using the MITRE ATT&CK framework, this attack corresponds to T1566 – Phishing, which illustrates how the attacker obtained initial access. When the ENISA taxonomy system is used, this particular incident would fall under the category of a social engineering incident with potential impact on confidentiality. When the VERIS system is used, the incident would be logged as an external party employing social action on a user account.

## **Contextual Metadata for Incident Categorization**

Contextual metadata provides additional relevant details about an incident. It assists analysts in carrying out an investigation involving these details. The categories include affected systems, date and time of the incident, IP address of source, as well as indicators indicating compromise. For instance, in a phishing event, some of these pieces of metadata may include the target mail address, phishing link, and source IP address. Additional details provide clarity to enable a comprehensive investigation.



## **Incident Response**

Basic Incident Response is a process that teams within SOC follow to handle security incidents in a systematic and efficient way. This is because it ensures that incidents are detected early and resolved in a way that does not affect business. An incident response process is also effective in ensuring that damage is minimized and evidence is retained.

## **Incident Response Lifecycle**

The incident response lifecycle offers an incident response step-by-step approach when dealing with incidents in an organizational setting to ensure that no critical activity is left behind in incident response efforts.

### ***1.Preparation:***

This is the phase that concentrates on preparation before any breach. Incident response plans, playbooks, escalation matrices, and communication plans are developed by the SOC team. Security solutions such as SIEM systems, EDR solutions, and forensic tools are set up and tested. The final area is that of training and awareness. This trains analysts to act accordingly during actual breach situations.

### ***2.Identification:***

During this stage, security alerts and malicious activities are evaluated to identify if they actually correspond to a real incident. SOC Analysts carry out alert triage, log analysis, event correlation, and checks for indicators of compromise. This is a critical process because incorrect identification can lead to false alerts and missed true threats.

### ***3.Containment:***

Containment is a process of trying to control the effects of an incident. Short-term containment includes steps such as isolating an infected machine, disabling an account that has been compromised, or blocking an IP address associated with an incident. Longer-term containment requires implementing temporary measures and improving controls for further investigation.



#### ***4.Eradication:***

Eradication involves dealing with the root cause of the attack. Some of these processes include removing malware, patching vulnerabilities that have been taken advantage of, removing any attack-related mechanisms that allow persistence on the system, and revoking any compromised authentication credentials.

#### ***5. Recovery:***

During the process of recovery, systems return to normal operations. The infected systems can be rebuilt and reset using clean backups and put back on the network. A monitoring process occurs to ensure that the malicious presence has been removed and that systems are working with secure operations.

#### ***6.Lessons Learned:***

This is the final stage that involves analyzing incidents to see if there is a gap in incident detection, response, as well as communications. SOC teams record lessons learned and implement actions to ensure that such incidents do not occur in the future.

### **Incident Response Procedures**

Incident response encompasses a number of operation procedures that aid in handling different phases. System isolation entails preventing attackers or hackers from having lateral movement within the network. Preservation of evidence entails a number of procedures such as creating memory dumps, disk images, as well as gathering logs. The techniques that can be used to preserve evidence include cryptographic methods such as SHA-256.

### **Communication and Escalation**

Communication is a critical aspect of effective incident response. Teams at the SOC have structured communication channels to alert the incident response team and other stakeholders as needed. Communication is important for effective response to incidents and to ensure that all parties involved act in line with the guidelines of different regulations. This is a best practice.



## Role of SOAR Tools in Incident Response

Security Orchestration, Automation, and Response solutions assist in automating repetitive incident response activities and enforcing best-of-breed workflows. Such solutions include Splunk Phantom, which enables automation between SIEM solutions, endpoint security software, and threat intelligence solutions for alert enrichment, containment activities, and ticketing.

## Practical Application

### 1. Alert Management Practice

To practice alert classification, prioritization, incident documentation, visualization, and escalation using standard SOC tools and workflows.

### Alert Classification System

An alert classification table was created in Google Sheets to map security alerts to their priority levels and corresponding MITRE ATT&CK techniques.

	A	B	C	D	E	F
1	Alert ID	Type	Priority	Mitre attack tactic		
2		1 Phishing Email	High	T1566 - Phishing		
3		2 Ransomware	Critical	T1486 - Data Encrypted for Impact		
4		3 Brute Force SSH	Medium	T1110 - Brute Force		
5		4 Port Scan	Low	T1046 - Network Service Discovery		
6		5 Log4Shell Exploit	Critical	T1190 - Exploit Public-Facing Application		
7						
8						
9						

Fig1: Alert classification table



## Alert Prioritization Using CVSS

Alerts were prioritized using CVSS scoring based on impact and exploitability.

A	B	C	D	E
Alert Name	Description	CVSS Score	Severity	
Log4Shell Exploit	Remote code execution vulnerability	9.8	Critical	
Ransomware Activity	File encryption detected	9.5	Critical	
Brute Force SSH	Multiple failed login attempts	6.5	Medium	
Port Scan	Network reconnaissance	3.2	Low	

Fig2: Alert Prioritization Table

### Priority Logic

- CVSS  $\geq 9.0$  → Critical
- CVSS 7.0 – 8.9 → High
- CVSS 4.0 – 6.9 → Medium
- CVSS  $< 4.0$  → Low

## Dashboard Creation in Wazuh

A custom Wazuh dashboard was created using the wazuh-alerts-\* index to visualize alert priorities.

### Dashboard Details

- Visualization Type: Pie Chart
- Data Source: wazuh-alerts-\*
- Severity Field Used: rule.level

### Severity Mapping

- Critical: rule.level  $\geq 12$  and rule.level  $\leq 15$
- High: rule.level  $\geq 8$  and rule.level  $\leq 11$



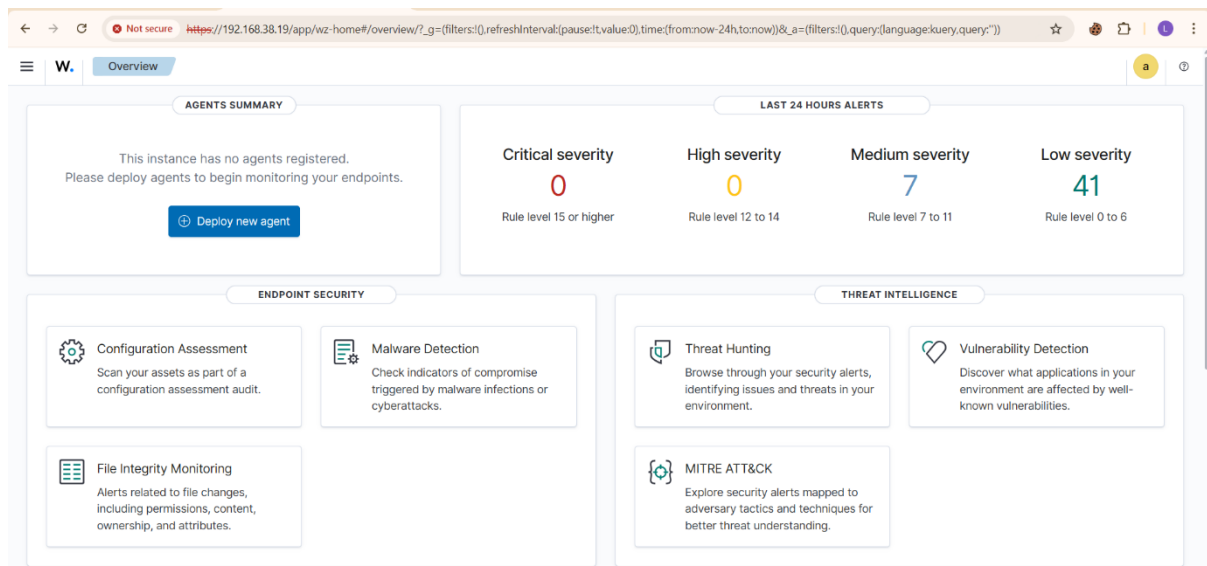


Fig 3: Wazuh home page showing severity levels

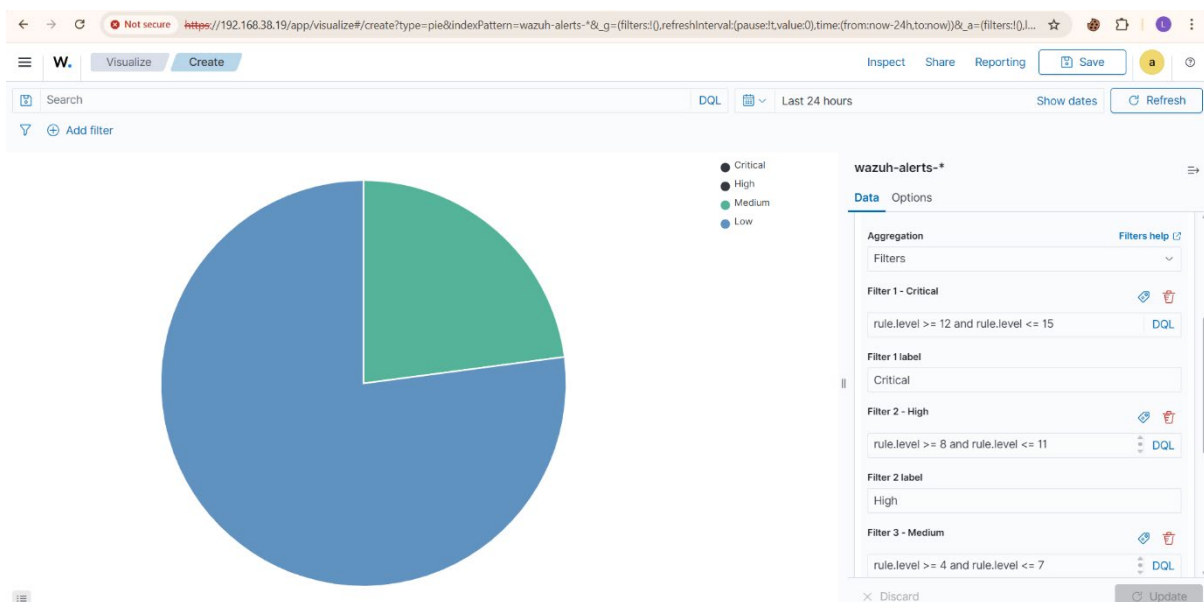


Fig4: Pie chart representing the severity levels

This dashboard allows SOC analysts to quickly assess the distribution of high-severity alerts for effective.



## Escalation Role-Play

Subject: [Critical Escalation] Ransomware Activity Detected on Server-X

Hello Tier 2 Team,

Critical ransomware activity has been noticed on Server-X, which demands an urgent escalation. Wazuh has noticed a suspicious executable process (crypto\_locker.exe) and malicious network activity from IP address 192.168.1.50. The threat corresponds to MITRE ATT&CK technique T1486 (Data Encrypted for Impact). The system is already isolated for containment purposes. No data exfiltration has been noticed at present. Requesting an urgent analysis and a verification of the ransomware activity and procedures on how to remove and reclaim the system.

Sincerely

SOC Tier 1 Analyst

## Alert Triage Practice

The objective of this task is to practice alert triage using Wazuh SIEM, validate alerts, identify false positives, and enrich alerts using threat intelligence platforms such as VirusTotal and AlienVault OTX.

Wazuh Server: wazuh ova (IP: 192.168.94.108)

Wazuh Agent: Ubuntu Linux (IP: 192.168.94.104)

Tools Used:

- Wazuh
- VirusTotal (online)
- AlienVault OTX (online)

A brute-force SSH attack was simulated by generating multiple failed SSH login attempts on the monitored Ubuntu agent. Wazuh detected and generated an alert based on predefined SSH brute-force rules.



```
Activities Terminal Jan 9 11:36
lisa@Ubuntu: ~
Process: 710 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
Main PID: 738 (sshd)
Tasks: 1 (limit: 4598)
Memory: 668.0M
CPU: 98ms
CGroup: /system.slice/ssh.service
--738 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Jan 09 10:20:35 Ubuntu systemd[1]: Starting OpenSSH Secure Shell server...
Jan 09 10:20:35 Ubuntu sshd[738]: Server listening on 0.0.0.0 port 22.
Jan 09 10:20:35 Ubuntu systemd[1]: Started OpenSSH Secure Shell service.
Jan 09 10:20:35 Ubuntu sshd[738]: Server listening on :: port 22.
lisa@Ubuntu: ~$ ssh fakeuser@192.168.94.104
The authenticity of host '192.168.94.104 (192.168.94.104)' can't be established.
ED25519 key fingerprint is SHA256:EGIsJr/7DGA/0tkDgcw7bVYVUqHMuvaTcZnlt3hfc.
This host key is known by the following other names/addresses:
- /usr/known_hosts:11 (hashed name)
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '192.168.94.104' (ED25519) to the list of known hosts.
fakeuser@192.168.94.104's password:
Permission denied, please try again.
fakeuser@192.168.94.104's password:
fakeuser@192.168.94.104: Permission denied (publickey,password).
```

Fig 5: SSH in ubuntu terminal

The /var/ossec/logs/alerts/alerts.log file was analyzed using keyword filtering for SSH events. The logs confirmed repeated failed login attempts for invalid users from the same source IP, which is characteristic of a brute-force attack.

```
wazuh-user@wazuh-server: ~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:27:32:3b:77 brd ff:ff:ff:ff:ff:ff
    altname enp0s8
    inet 10.0.2.15/24 metric 1024 brd 10.0.2.255 scope global dynamic eth0
        valid_lft 86230sec preferred_lft 86230sec
    inet6 fe80::20c:27ff:fe32:3b77/64 scope link proto kernel ll
        valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:27:37:03:ca brd ff:ff:ff:ff:ff:ff
    altname enp0s17
    inet 192.168.94.100/24 metric 1024 brd 192.168.94.255 scope global dynamic eth1
        valid_lft 413sec preferred_lft 413sec
    inet6 fe80::20c:27ff:fe37:03ca/64 scope link proto kernel ll
        valid_lft forever preferred_lft forever
wazuh-user@wazuh-server: ~$ sudo grep ssh
wazuh-user@wazuh-server: ~$ sudo grep "ss" /var/ossec/logs/alerts/alerts.log | tail -n 20
Rule: 5710 (level 5) -> "sshd: Attempt to login using a non-existent user"
Jan 09 05:32:26 Ubuntu sshd(8091): Failed password for invalid user fakeuser from 192.168.94.104 port 59440 ssh2
Jan 09 05:32:26 Ubuntu sshd(8091): PAM 2 more authentication failures; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.94.104
--Alert 170736765: 1852991: - syslog,sshd,authentication_failed,gdpr_IV_35.7.d,gdpr_IV_32.2,pp13.7.1,hipaa_164.312.b,imoid_login,nist_800_53.01.4,nist_800_53.06.7,nist_800_53.06.6,pci_dss_10.2.4,pci_dss_10.2.5,pci_dss_10.6.1,tsc_OC6.1,tsc_OC6.8,tsc_OC7.2,tsc_OC7.3,
Rule: 5710 (level 5) -> "sshd: Attempt to login using a non-existent user"
Jan 09 05:32:33 wazuh-server sshd(8208): Invalid user fakeuser from 192.168.94.104 port 44448
--Alert 170736759: 1861025: - syslog,sshd,authentication_failed,gdpr_IV_35.7.d,gdpr_IV_32.2,pp13.7.1,hipaa_164.312.b,imoid_login,nist_800_53.01.4,nist_800_53.06.7,nist_800_53.06.6,pci_dss_10.2.4,pci_dss_10.2.5,pci_dss_10.6.1,tsc_OC6.1,tsc_OC6.8,tsc_OC7.2,tsc_OC7.3,
Rule: 5710 (level 5) -> "sshd: Attempt to login using a non-existent user"
Jan 09 05:32:37 wazuh-server sshd(8208): Failed password for invalid user fakeuser from 192.168.94.104 port 44448 ssh2
--Alert 170736761: 1861550: - syslog,sshd,authentication_failed,gdpr_IV_35.7.d,gdpr_IV_32.2,pp13.7.1,hipaa_164.312.b,imoid_login,nist_800_53.01.4,nist_800_53.06.7,nist_800_53.06.6,pci_dss_10.2.4,pci_dss_10.2.5,pci_dss_10.6.1,tsc_OC6.1,tsc_OC6.8,tsc_OC7.2,tsc_OC7.3,
Rule: 5710 (level 5) -> "sshd: Attempt to login using a non-existent user"
Jan 09 05:32:41 wazuh-server sshd(8208): Failed password for invalid user fakeuser from 192.168.94.104 port 44448 ssh2
--Alert 170736765: 1862498: - syslog,sshd,authentication_failed,gdpr_IV_35.7.d,gdpr_IV_32.2,pp13.7.1,hipaa_164.312.b,imoid_login,nist_800_53.01.4,nist_800_53.06.7,nist_800_53.06.6,pci_dss_10.2.4,pci_dss_10.2.5,pci_dss_10.6.1,tsc_OC6.1,tsc_OC6.8,tsc_OC7.2,tsc_OC7.3,
Rule: 5710 (level 5) -> "sshd: Attempt to login using a non-existent user"
Jan 09 05:32:45 wazuh-server sshd(8208): Failed password for invalid user fakeuser from 192.168.94.104 port 44448 ssh2
Jan 09 05:32:45 wazuh-server sshd(8208): PAM 2 more authentication failures; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.94.104
Jan 09 05:32:48 wazuh-server sudo(8240): wazuh-user - tty=pty : 000:home/wazuh-user : SSH-root : 000000-wazuh/bin/grep ssh
command: /usr/bin/grep ssh
wazuh-user@wazuh-server: ~$
```

Fig 6: Result showing alerts in wazuh

The logs show multiple entries such as “Failed password for invalid user” and “authentication failure,” generated within a short time window from the same source IP. This behavior triggered Wazuh’s SSH brute-force detection mechanism.

The alert was assigned a medium priority because multiple authentication failures were detected without evidence of successful login. Continuous attempts indicate potential malicious intent, but no account compromise was observed.



No results found for "192.168.94.104"

Fig 7: Openvault result

The source IP was investigated using AlienVault OTX. No malicious pulses, threat reports, or indicators of compromise were associated with the IP address. This confirms that the activity originated from a controlled lab environment and is not linked to known external threats.

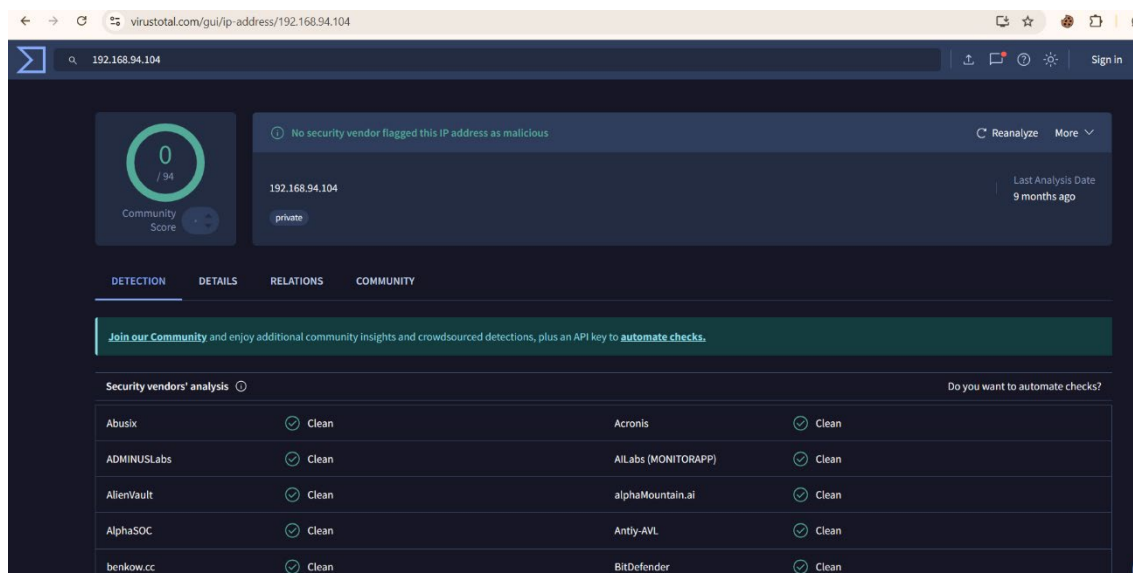


Fig8: Virustotal result



## Evidence Preservation

### Volatile Data Collection

To collect volatile network connection information from a Windows virtual machine for incident response analysis.

#### Procedure

- Velociraptor was executed in standalone GUI mode on the Windows VM.
- The artifact Windows.Network.Netstat was selected.
- The collector was executed and results were saved in CSV format.
- The CSV file was preserved without modification for analysis.

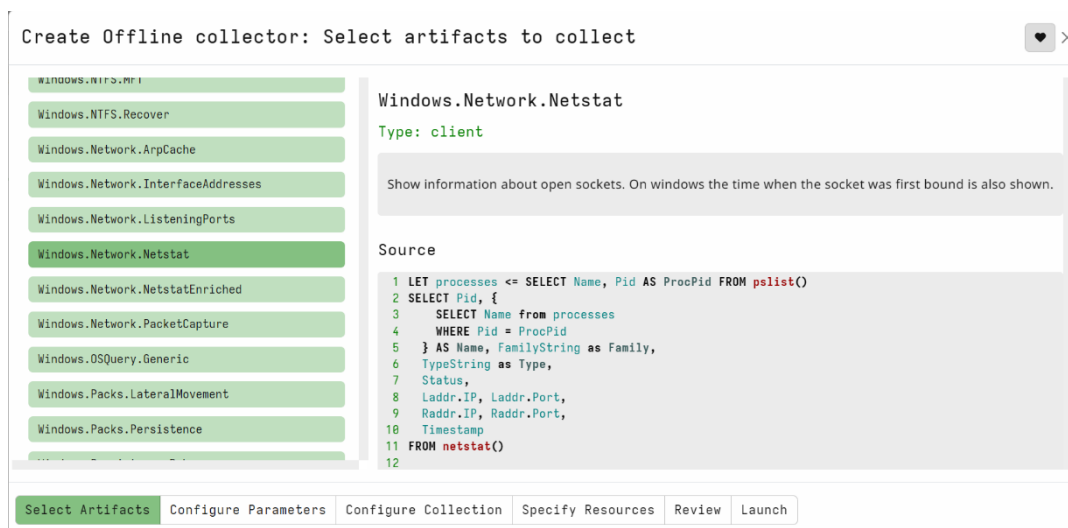


Fig9: Netstat result



State	FlowId	Artifacts	Created	Last Active	Creator	Mb	Rows
✓	F.D5DMID5EM2I80	Server.Utils.CreateCollector	2026-01-05T07:33:08.223Z	2026-01-05T07:33:08.916Z	admin	65 Mb	1

Artifact Names	Server.Utils.CreateCollector
Flow ID	F.D5DMID5EM2I80
Creator	admin
Create Time	2026-01-05T07:33:08.223Z
Start Time	2026-01-05T07:33:08.265Z
Last Active	2026-01-05T07:33:08.916Z
Duration	0.00 seconds
State	Completed
Ops/Sec	Unlimited
CPU Limit	Unlimited
IOPS Limit	Unlimited
Timeout	600 seconds
Max Rows	1m rows
Max Mb	1000.00 Mb
Parameters	

Artifacts with Results	Server.Utils.CreateCollector
Total Rows	1
Uploaded Bytes	68208112 / 68208112
Files uploaded	1
Download Results	
Available Downloads	
server-server-F.D5DMID5EM2I80	
Uncompressed	65 Mb
Compressed	24 Mb
Container Files	7
Started	2026-01-05T07:33:26Z
Duration (Sec)	4
SHA256 Hash	c563d46c328cbadb0c47d798b66cd8cad8a26b466b5322fd dd5b24d8d2270a381

Fig 10:Memory acquisition Result

## Evidence Collection

To acquire a full physical memory image from a Windows virtual machine while preserving evidence integrity.

### Procedure

- Velociraptor was run with administrative privileges.
- An offline collector was created using the artifact Windows.Memory.Acquisition.
- The memory acquisition process completed successfully.
- The resulting memory image was packaged in a ZIP container.
- The memory file was extracted and hashed using SHA-256 to ensure integrity.



## Evidence Collected

State	FlowId	Artifacts	Created	Last Active	Creator	Mb	Rows
✓	F.D5DN07RLUF0S4	Server.Utils.CreateCollector	2026-01-05T08:02:39.251Z	2026-01-05T08:02:39.859Z	admin	65 Mb	1
✓	F.D5DMID5EM2I00	Server.Utils.CreateCollector	2026-01-05T07:33:08.223Z	2026-01-05T07:33:08.916Z	admin	65 Mb	1

Artifact Names	
Flow ID	F.D5DN07RLUF0S4
Creator	admin
Create Time	2026-01-05T08:02:39.251Z
Start Time	2026-01-05T08:02:39.274Z
Last Active	2026-01-05T08:02:39.859Z
Duration	0.00 seconds
State	Completed
Ops/Sec	Unlimited
CPU Limit	Unlimited
IOPS Limit	Unlimited
Timeout	600 seconds
Max Rows	1m rows
Max Mb	1000.00 Mb

Artifacts with Results	
Total Rows	1
Uploaded Bytes	68208112 / 68208112
Files uploaded	1
Download Results	
Available Downloads	
server-server-F.D5DN07RLUF0S4	
Uncompressed	65 Mb
Compressed	24 Mb
Container Files	7
Started	2026-01-05T08:03:20Z
Duration (Sec)	4
SHA256 Hash	164840424b4efc56aedbb78526946a82044200286001a5e67297183d5ddf6d8c

Fig 11: Result with hash value

## Capstone Project

This testing project showcases a thorough alert-to-response process within a lab setting. This process began with a purposeful vulnerability in a Metasploitable2 target machine, exploited with Metasploit launched from a Kali Linux attack machine. The malicious event was successfully detected by the Wazuh Security Information and Event Management system, triggering a security alert associated with a MITRE ATT&CK technique. Subsequent steps for triaging the alert included a subsequent Containment step done using CrowdSec to showcase IP blocking. Verification of the response occurred from inspection of firewall rules and connectivity checks. This particular use case showcases a detection, response, and documentation cycle often implemented within Security Operations Center.

- Attack Type: Remote exploitation (VSFTPD backdoor)
- Victim System: Metasploitable2 (192.168.94.101)
- Attacker System: Kali Linux
- Detection Tool: Wazuh
- Response Tool: CrowdSec

Fig 12: Metasploit exploitation of VSFTPD backdoor on Metasploitable2

Fig 13: Wazuh alert generated for VSFTPD exploitation mapped to MITRE ATT&CK





```
root@kali: /home/kali
File Actions Edit View Help
ACCEPT 0 -- 0.0.0.0/0 0.0.0.0/0 ctstate RELATED,ESTABLISHED
DOCKER 0 -- 0.0.0.0/0 0.0.0.0/0
ACCEPT 0 -- 0.0.0.0/0 0.0.0.0/0
ACCEPT 0 -- 0.0.0.0/0 0.0.0.0/0
ACCEPT 0 -- 0.0.0.0/0 0.0.0.0/0 ctstate RELATED,ESTABLISHED
DOCKER 0 -- 0.0.0.0/0 0.0.0.0/0
ACCEPT 0 -- 0.0.0.0/0 0.0.0.0/0
ACCEPT 0 -- 0.0.0.0/0 0.0.0.0/0

Chain OUTPUT (policy ACCEPT)
target prot opt source destination

Chain DOCKER (2 references)
target prot opt source destination

Chain DOCKER-ISOLATION-STAGE-1 (1 references)
target prot opt source destination
DOCKER-ISOLATION-STAGE-2 0 -- 0.0.0.0/0 0.0.0.0/0
DOCKER-ISOLATION-STAGE-2 0 -- 0.0.0.0/0 0.0.0.0/0
RETURN 0 -- 0.0.0.0/0 0.0.0.0/0

Chain DOCKER-ISOLATION-STAGE-2 (2 references)
target prot opt source destination
DROP 0 -- 0.0.0.0/0 0.0.0.0/0
DROP 0 -- 0.0.0.0/0 0.0.0.0/0
RETURN 0 -- 0.0.0.0/0 0.0.0.0/0

Chain DOCKER-USER (1 references)
target prot opt source destination
RETURN 0 -- 0.0.0.0/0 0.0.0.0/0

root@kali: /home/kali
# ping 192.168.94.101
PING 192.168.94.101 (192.168.94.101) 56(84) bytes of data.
^C
--- 192.168.94.101 ping statistics ---
352 packets transmitted, 0 received, 100% packet loss, time 564457ms
```

Fig 14: CrowdSec blocking attacker IP as part of containment

## Learnings

- I became familiar with methods for categorizing security alerts based on severity levels, asset criticality, and CVSS scoring.
- Learned to evaluate alerts and do basic triage to differentiate true positives and false positives.
- Gained practical experience on documenting incidents through a SANS incident response template.
- Learned the whole life cycle process for incident response, including detection, containment, recovery, and lessons learned.
- Developed skills in preservation of evidence through memory collection and hashing for verification.
- Learnt how to use SOC solutions like Wazuh, TheHive, and CrowdSec for monitoring and response.
- Increased skills in the documenting clearly, reporting