- **Threat Hunting Methodologies** :

  - **Proactive Threat Hunting** : It can be defined as a security measure where the analysts search for potential threats even before the alarms are generated in the security tools. Proactive hunting is hypothesis driven.
    Hypothesis can be made based on known attacking techniques or attacking behaviours or Tactics, techniques and Procedures (TTPs) defined in MITRE framework.
    For eg : An analyst may hypothesize that the attacker may misuse valid accounts (T1078) to get the authorized access to the servers or network.
    Based on this hypothesis the analysts would examine the authentication logs to identify the anomalies like :
    a) Logins from different locations
    b) Attempts of login after business hours.

  - **Threat Hunting Frameworks** :
    a) SQRR (Search, Query, Retrieve, Respond) : This framework provides a structured approach for threat hunting.
    - Search : Identify areas where users suspicious behaviour is reported or unusual spikes in traffic are detected or observed.
    - Query : Use search queries to examine the logs or databases.
    - Retrieve : Collect related evidences from the queries.
    - Respond : Take immediate action depending on evidences collected, escalate to higher level if needed.

    b) **TaHiTi Framework** (Targeted Hunting Integrating Threat Intelligence) : It works with integrating threat intelligence with the hunting activities. It mainly focuses on :
    - Understanding the attacker and his behaviour of attacking
    - Mapping intelligence with data.

    c) **Data Sources for Hunting** : Effective Threat hunting requires analyzing or studying the data from multiple sources like EDR (Endpoint Detections and Response), Network traffic logs, Authentication logs, Threat intelligence reports. Studying logs or data from all sources helps to get a clear view of attack.

The key objective of threat hunting methodologies is to make security teams to proactively monitor threats and logs, improve the time taken for incident response and improve structured analysis.

- **Advanced SOAR Automation** :

  - **SOAR Components** : SOAR stands for Security Orchestration, Automation & Response and it consists of three components:
    a) **Orchestration** : It integrates multiple security tools like EDR, SIEM, firewalls into one workflow.

b) **Automation** : It is used to automatically do repetitive tasks like ticket creation, generation of alerts or messages.
c) **Responses** : It consists of immediate actions taken for resolution of incident like isolating affected machines, blocking of IP.

- **Playbook Development** : A playbook is predefined set of actions defined to respond to specific incidents that occur like phishing or malware infections.
  For Eg : A phishing playbook may consist of
  - Extracting URLs from emails
  - Checking of URLS against threat intelligence
  - Blocking malicious domains
  - Notifying users and SOC analysts.

- **Integration with SIEM / EDR** : SOAR platforms integrate with SIEM / EDR tools for :
  - Receiving threat alerts automatically
  - To generate alerts with additional contents.
  - Trigger response actions based on predefined rules.

The objective of the SOAR automation is to reduce manual work, reduce the time taken to handle the incidents and improve the efficiency.

- **Post Incident Analysis and Continuous Improvement** :

  - **Root Cause Analysis** : It identifies the base or root cause of the incident.
    Common RCA techniques include :
    - 5 Why's : Asking why this incident is occurring repeatedly and what is the root cause of this incident.
    - Fishbone Diagram : Categorizing the causes into people, processes, technology and environment.
      For Eg : An RCA of phishing breach may reveal lack of email filtering and lack of user training.

  - **Lesson Learned Processes** : The lesson learned processes include conducting a post incident review to evaluate :
    - What all happened
    - What worked well
    - What all failed
    - What improvements are needed.

  - **Metrics and KPIs** : They are used to measure the SOC performance it includes the following.
    - **Mean Time to Detect (MTTD)** : Time taken to identify the incident
    - **Mean Time to Respond (MTTR)** : Time taken to resolve the incident.

Tracking these metrics help organizations track their efficiency and fields in which they can improve.

The objective of the Post incident analysis is to continuously improve the security techniques so as to prevent the incident that occur often or similar incidents.

- **Adversary Emulation Techniques** :

  - **Adversary Emulation** : It can be defined as simulating real world attackers behaviour to test the detection and response capabilities.
    Eg : T1566 – Phishing
    T1210 – Exploitation of Remote services
    This helps organizations to test and monitor the security controls and response techniques.

  - **Emulation Frameworks** : Tools such as MITRE Caldera are used to automate adversary emulation by executing the predefined attacking techniques that are mapped to MITRE Attack.

  - **Red – Blue Team Collaboration** : Adversary emulation helps collaboration between red and blue teams of SOC.

    - Red Team : It simulates the attacker behaviour
    - Blue team : It detects and responds to attacks.

The key objective is to enhance readiness of SOC for the attacks and strengthen the defence systems.

- **Security Metrics and Executive Reporting** :

  - **Advanced SOC Metrics** : It provides insights in SOC performance which include:
    - **Dwell Time** : Time between the compromise and detection
    - **False Positive Rate** : Percentage of incorrect or false alerts.
    - **Incident Resolution Rate** : Percentage of incidents that re resolved successfully.

  - **Executive Reporting** : It helps to translate the technical data into simple, clear business insights with the helps of :
    - Dashboards
    - Trend analysis
    - High level summaries

    This helps concerned people understand the data more clearly and helps them take further measures.

  - **Continuous Improvement** : Metrics help identify the gaps like false positives, late detection of incidents. This insights help improvement in

- Tools
- Processes
- Training of staff
- Measures taken for containment.

The key objective is to improve the SOC effectiveness in handling of tasks and incidents and communicate the report to concerned stakeholders in a understandable way.

- **Threat Hunting Practice** :

  For doing this task we have used tools like :
  - Elastic Security
  - Velociraptor
  - AlienVault OTX
  - Ubuntu Virtual Machine

  Below are the steps to be followed for the task.

  a) Start the Ubuntu virtual machine and verify the installations in terminal
  b) The installations of velociraptor, elasticsearch by checking the status.
  c) The next step after verification of installations is to open elastic search.
  d) After opening elasticsearch open elastic security dashboard.
  e) After opening it navigate to Discover/Security Events in that
  f) The next step is to apply a filter to the log events
  g) Apply the filter of event.code:4672 and review the results to identify users receiving special privileges.
  h) Identify a suspicious event like :

Timestamp	User	Event ID	Notes
2025-08-18 15:00:00	testuser	4672	Unexpected admin role

  i) The next step is to login into Alienvault OTX and search for MITRE ATTACK technique T1078.
  j) After reviewing no malicious logs or suspicious IOCs match were found in the AlienVault.
  k) The next step is to validate it using velociraptor.
  l) Open the vm terminal and execute the velociraptor query.
  m) The query gives output for all process running alongwith the name and user role.
  n) By anaylzing the output we can see that all the processes running were valid or legitimate and root level.
  o) Below is the output of the query.

```
 "pid": 3766,
 "name": "mutter-x11-frames",
 "username": "prasad"
},
{
 "pid": 4865,
 "name": "kworker/u9:2-events_power_efficient",
 "username": "root"
},
{
 "pid": 5787,
 "name": "kworker/1:0-cgroup_destroy",
 "username": "root"
},
{
 "pid": 6118,
 "name": "kworker/0:1-cgroup_destroy",
 "username": "root"
},
{
 "pid": 6169,
 "name": "kworker/u10:1-events_unbound",
 "username": "root"
},
{
 "pid": 6250,
```


p) The next step is to check the network connection query so as to identify the ports listening.
q) The ports were checked and it was confirmed that there were no unknown or external services running.
r) Below is output of the query.


```
"Family": 2,
"Type": 1,
"Laddr": {
 "IP": "0.0.0.0",
 "Port": 22
},
"Raddr": {
 "IP": "0.0.0.0",
 "Port": 0
},
"Status": "LISTEN",
"Pid": 995,
"FamilyString": "IPv4",
"TypeString": "TCP"
},
{
"Fd": 0,
"Family": 2,
"Type": 1,
"Laddr": {
 "IP": "127.0.0.1",
 "Port": 8080
},
"Raddr": {
 "IP": "0.0.0.0",
 "Port": 0
}
```

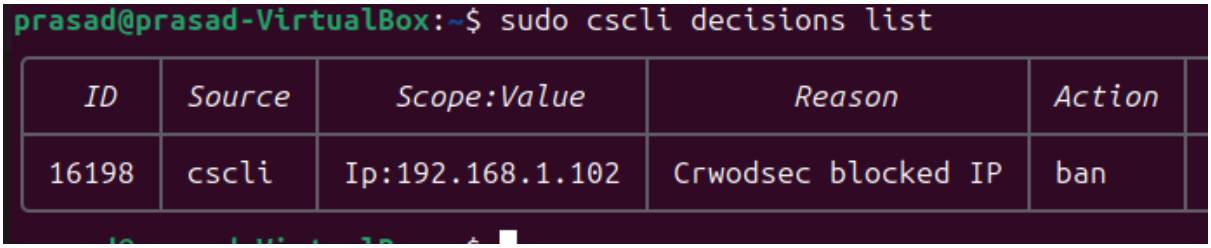
Report :

Threat Hunting exercise was conducted so as to investigate potential unauthorized privilege escalation aligned with MITRE ATTACK T1078 (valid Accounts). The elasticsearch was queried for Event ID 4078 so as to identify the privilege role assignments. A suspicious event was detected but when it was further validated with AlienVault threat intelligence no matching suspicious indicators were found. It was also validated through Velociraptor and it also confirmed that all processes running were legitimate and valid. The validation from tools concluded that there was no privilege escalation or compromise.

- **SOAR Playbook Development** :
  For doing this task we have used following tools :
  - Splunk Phantom (SOAR) – Playbook automation
  - Wazuh SIEM – Phishing alert generation
  - Crowdsec – For IP blocking
  - Hive – Incident case management
  - Google Docs – Documentation

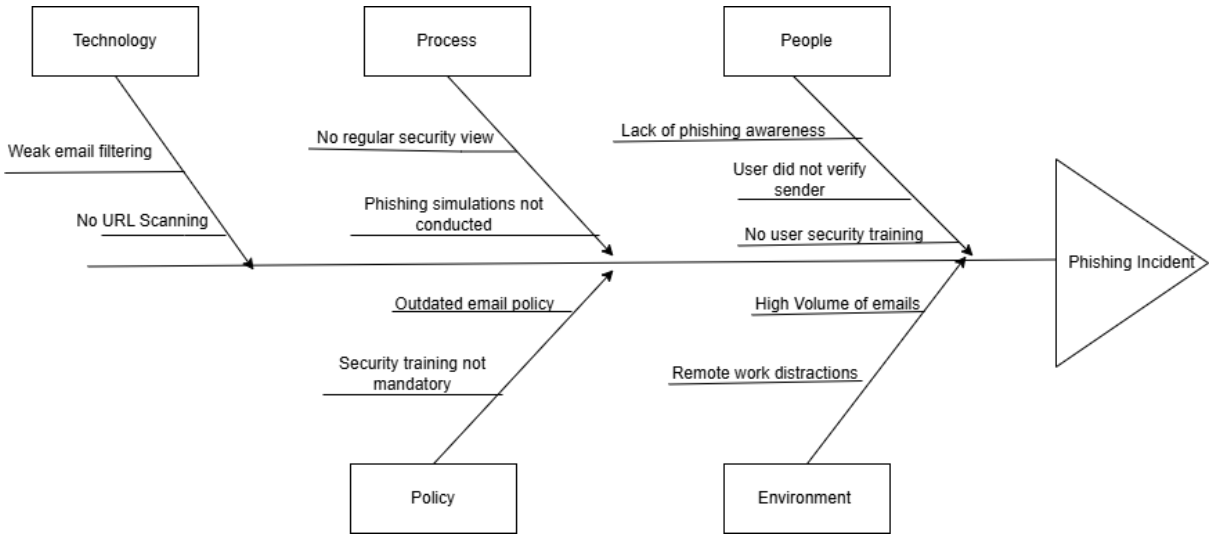
  Below mentioned steps are to be followed by doing the task :

  a) The first step is to login into Splunk phantom UI.
  b) Confirm all the integrations like Hive, Crowdsec and wazuh.
  c) The next step is to create a SOAR playbook in splunk phantom
  d) There navigate to playbooks there click on create new.
  e) The next step is to name the playbook and select the type.
  f) Below is the logic flow :
  Wazuh phishing alert -> Extract Source IP -> Check IP reputation -> Block IP via crowdsec -> Create a case in Hive
  g) The next step is to check the IP reputation in playbook.
  h) There we can see that the IP is malicious and we need to taken preventive measures.
  i) For preventive measures we block the IP via crowdsec.
  j) Below is the output of IP being blocked



  k) After blocking the IP the next step is to create ticket in Hive there give the name, and also select the severity in Hive for ticket creation.
  l) After that save the playbook and check for active mode enabled.
  m) The next step is to generate phishing alert in Wazuh and check the alert observed in wazuh.
  n) Now we have checked the Ip reputation and it was found to be malicious we have blocked it using crowdsec and also created a ticket in Hive.

- **Post Incident Analysis** :
  The fishbone diagram is used to illustrate the phishing incident by categorizing various factors like people, process, technology, policy, environment which help us study the root cause analysis and find the areas of improvement.



➢ **Metrics Calculation** :

Event	Time Taken
Incident Occurred	00:00
Incident Detected	02:00
Incident Resolved	06:00

Mean Time to Detect : (MTTD)
MTTD = Time of Detection – Time of Occurrence
= 2 Hours

Mean Time to Respond : (MTTR)
MTTR = Time of Resolution – Time of Detection
= 4 Hours

Summary :
The phishing incident was detected in 2 hours indicating monitoring effectiveness and the complete response and containment took 4 hours which demonstrates timely action. By improving the security controls detection techniques we can reduce the response time in future which will increase the efficiency.

- **Alert Triage with Automation** : We have used following tools for this task
    - Wazuh
    - VirusTotal
    - Hive
    - Ubuntu VM

Following steps are to be followed to do the task.

a) The first step is to update the system packages.
b) If wazuh is already installed check the status it should be active.
c) The next step is to access the wazuh dashboard.
d) After accessing it the next step is to simulate a suspicious file download
e) Then go to security events tab in wazuh dashboard and filter group by malware and event category by file_download
f) It shows the output of alert detected with source IP, severity and rule.

Alert ID	Description	Source IP	Priority	Status
005	File Downloaded	192.168.1.102	High	Open

g) The next step is to install Hive if already installed start and enable the hive.
h) The next step is to check the status it should be active and running.
i) The next step is to access the hive UI.
j) After accessing the UI navigate to virustotal website and create the account.
k) Then configure virustotal in hive by going to admin-> analyzers there enable Virustotal analyzer and then save the configuration.
l) Then create a new case in Hive then enter the title, severity, and then add file hash.
m) Then click on add observable and type = hash paste the hash and click on save.'
n) Then run the virustotal analyzer, click on run analyzer and get the output.

Summary :

The suspicious hash file was validated automatically using tools like VirusTotal alongwith Hive integration. The analysis concluded that detection through various tools confirming that file was malicious. It was done with automated generation of alerts, reduced manual efforts and faster escalation and response for high priority alerts.

- **Evidence Analysis** : For doing this task we have tools like velociraptor, FTK Imager and Ubuntu VM. Following steps are to be followed for this task :
    a) Download and install velociraptor if not installed and check the status or version after download is complete so as to verify.
    b) The next step is to create a directory and change the directory in vm so it points towards the directory created.
    c) The next step is to collect the network connection evidence so as to verify if any external or suspicious connection.
    d) For connection evidence or to check suspicious connection check it through SELECT * FROM netstat.

e) It show all connections with port numbers and collect that data and store it in netstat_output.csv

f) Then analyze the network connection if any suspicious connection is there.

g) The output of netstat or network connections is attached below.

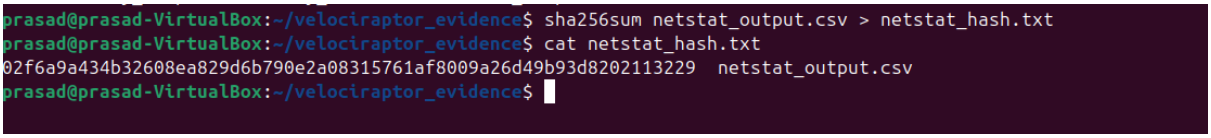
```
Fd,Family,Type,Laddr,Raddr,Status,Pid,FamilyString,TypeString
0,2,1,"{
 ""IP"": ""127.0.0.53"",
 ""Port"": 53
}","{
 ""IP"": ""0.0.0.0"",
 ""Port"": 0
}",LISTEN,-1,IPv4,TCP
0,2,1,"{
 ""IP"": ""127.0.0.1"",
 ""Port"": 631
}","{
 ""IP"": ""0.0.0.0"",
 ""Port"": 0
}",LISTEN,-1,IPv4,TCP
0,2,1,"{
 ""IP"": ""127.0.0.54"",
 ""Port"": 53
}","{
 ""IP"": ""0.0.0.0"",
 ""Port"": 0
}",LISTEN,-1,IPv4,TCP
0,2,1,"{
 ""IP"": ""0.0.0.0"",
 ""Port"": 22
}","{
 ""IP"": ""0.0.0.0""
 ""Port"": 68
}","{
 ""IP"": ""10.0.2.2"",
 ""Port"": 67
}",ESTAB,-1,IPv4,UDP
0,2,2,"{
 ""IP"": ""192.168.56.102"",
 ""Port"": 68
}","{
 ""IP"": ""192.168.56.100"",
 ""Port"": 67
}",ESTAB,-1,IPv4,UDP
0,10,2,"{
 ""IP"": ""::"",
 ""Port"": 5353
}","{
 ""IP"": ""::"",
 ""Port"": 0
}",CLOSE,-1,IPv6,UDP
0,10,2,"{
 ""IP"": ""::"",
 ""Port"": 40232
}","{
 ""IP"": ""::"",
 ""Port"": 0
}",CLOSE,-1,IPv6,UDP
(END)
```

h) The next step after checking the connection is to hash the network log.

i) We will generate the sha256sum to hash the log.

j) And view the log with cat netstat_hash.txt

k) The output of hashed log is attached below.



Item	Description	Collected By	Date	Hash Value
Network log	Server Z log	SOC Analyst	2025-08-18	02f6a9a434b32608ea829b790

Network Connections evidence was collected using velociraptor netstat. No malicious or external IP or port connection was identified in the logs and the logs were hashed with help of sha256sum and stored in csv file for further reference.

- **Adversary Emulation Practice** : We have used following tools for doing this task :
  - **a)** MITRE Caldera
  - **b)** Wazuh
  - **c)** Ubuntu VM

Following steps are to be followed to do the task.

a) The first step is to install MITRE Caldera and requirements related to it.
b) The next step is to start the Caldera
c) After starting caldera the next step is to start the wazuh if already installed or install the wazuh
d) After installing start the wazuh-agent and check the status it should be active (running)
e) The next step is to enable email detection in wazuh and for that edit the rules in the local_rules
f) After editing the rules restart the wazuh-manager
g) The next step is to configure the Caldera for spearphishing.
h) For that login into caldera and then go to settings and select plugins in that enable stockpile.
i) The next step is to create adversary profile for that select adversaries and click on create and fill in required details
j) The next step is to run emulation that is to attack for that go to operations and select new operations there select adversary, agent and click on run.
k) It will show the output of delivered successfully.
l) The next step is to validate the detection in wazuh for that go to wazuh dashboards.
m) There navigate to security and filter it by T1566. It will show the output of phishing activity along with level.

Summary :

The exercise of emulation successfully emulated a spearphishing attack using MITRE Caldera aligned with MITRE ATTACK technique T1566. The phishing execution on endpoint generated indicators that were collected by wazuh agent and the wazuh manager. The SOC detection successfully identified the activity and generated high severity alert which was mapped to T1566. There was also a gap in email gateway logging. Enhancing email header and attachment sandboxing would improve detection for future spearphishing attacks on the system.

- **Security Metrics and Executive Reporting** :

To do this task we will need tools like Elastic security, Google sheets and google docs.
Following steps are to be followed to do the task

a) The first step is to install elastic security and if installed enable it and start the services.
b) The next step is to prepare the data in the elastic security
c) For that purpose navigate to elastic -> security -> alerts.
d) The next step is to filter alerts like for the last 7 days.
e) The next step is to select 5-10 mock alerts.
f) The next step is to calculate metrics logs.

- **Mean time to detect (MTTD)**
  Detection Time – Event Occurred time
  Eg :10:00 – 08:00 = 2 Hours

- **Mean time to Respond (MTTR)**
  Incident Close time – Incident Detection Time
  Eg :14:00 -10:00 = 4 hours

- **False Positive Rate**
  False Alerts / Total Alerts x 100
  Eg : 2/10 x 100 = 20%

g) The next step is to create elastic security metrics dashboard.
h) For that navigate to Elastic -> Dashboard -> Create new dashboard
i) Then click on add visualisation for Average Detection time, Average response time and false positive.
j) Then you can see the values shown for MTTD and MTTR.

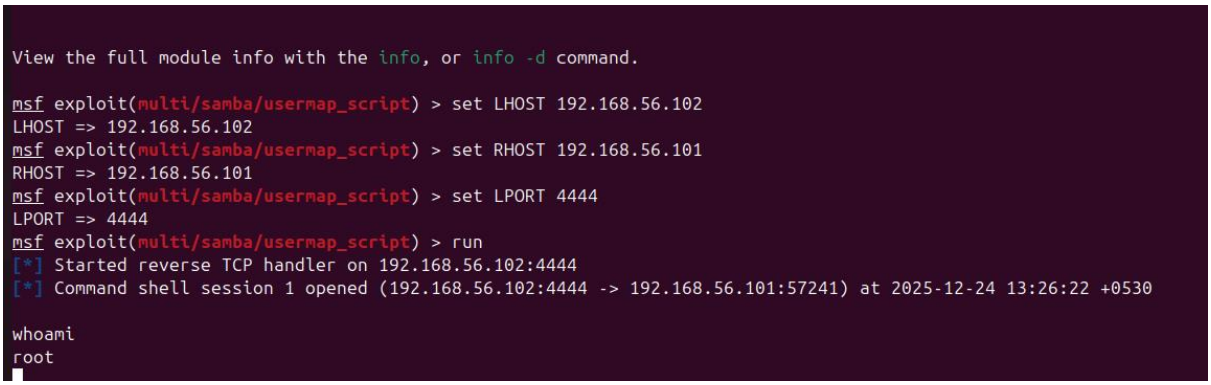
Dwell Time Analysis :

Dwell time analysis summarized that delay in detection would significantly impact the incident. Incident with high positive rates or alerts would experience longer dwell times. Improving the alert accuracy and managing the processed faster ca effectively reduce the dwell time and would also limit the potential damage that would be cost to the network or the server.

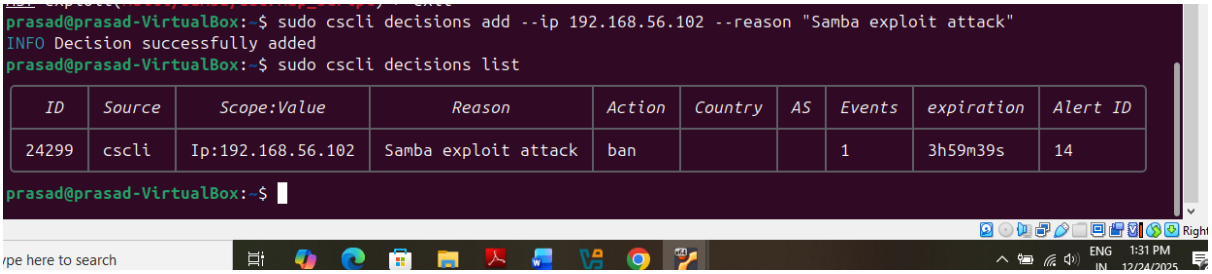
- **Capstone Project : Comprehensive SOC Incident Response**
  To do the task we will use tools like Metasploit, Wazuh, Crowdsec, The Hive, MITRE Caldera, Elastic security, Ubuntu VM
  Following steps are to be followed to do the task:
  a) The first step is to install Metasploit alongwith framework and verify by checking the version.
  b) The next step is to verify the ip address of ubuntu vm and also the Metasploit.
  c) The next step is to install wazuh if already installed enable and start the services.
  d) After that install wazuh-agent on Metasploitable2 and after installing start the services.
  e) The next step is to install Hive if already installed start the services and check the status.

f) After that install crowdsec and if installed check the services.
g) The next step is to initiate a attack on the vulnerable metasploitable2 through our VM.
h) For that purpose login into msf console that is Metasploit on VM.
i) The snapshot of msf login is attached below.

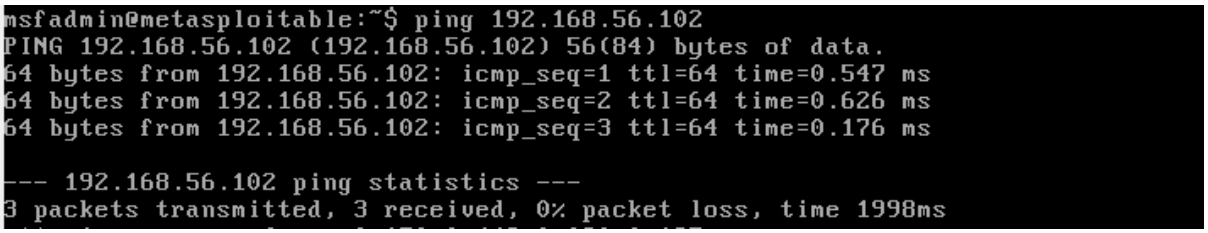
j) The next step is to use usermap samba script for the attack.
k) For that set target set the LHOST and the RHOST and run
l) You will see the output of command shell access with root privileges when checked through whoami this tells us attack was successful.
m) The output of the script run is attached below.



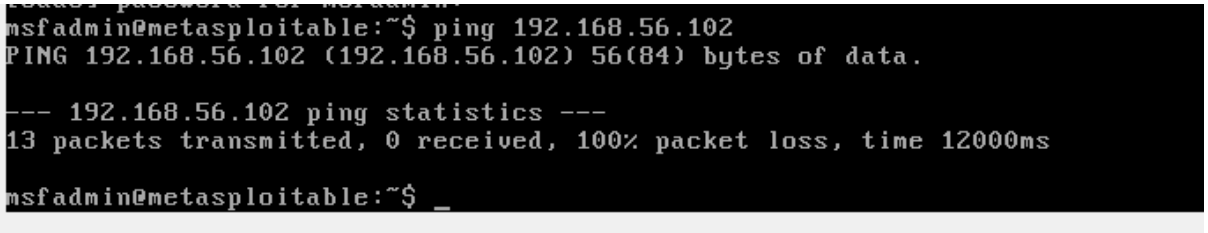
n) The next step is attack detection in Wazuh for that login into Wzuh dashboard and navigate to Security events -> alerts.
o) Then find the samba exploit logs and unauthorized shell access.
p) The next step is to create a case in the Hive and for that select on create new case and fill in the required details like Case Title, Severity.
q) The next step is to isolate the VM by using crowdesc.
r) The first step is to isolate the victims VM and block the IP
s) For blocking the IP add the IP to decisions list and write the reason Samba exploit attack and check or verify the IP is blocked with a ping test
t) Below is snapshot of attackers IP 192.168.56.102 added to decisions list for block.



u) Below is snapshot of ping test of IP before blocking



v) Below is snapshot of ping test of IP after blocking we can see 100% packet loss.



w) The next step is to automate the IP blocking via playbook for that purpose create a case in the Hive, extract the attacker IP, Block the IP using crowdsec and log action for the audit.

x) The SOAR automation reduces the response time by automatically creating an incident, blocking the attackers IP and improving the efficiency.

- **Post Incident Analysis** : The main objective is to identify why did the incident happened.

  5 WHY Analysis

  - Why was the system compromised ?
    The attacker exploited a vulnerable samba service.

  - Why was the Samba Vulnerable ?
    The samba version was outdated and not properly patched.

  - Why was it unpatched ?
    There was no regular patch management process.

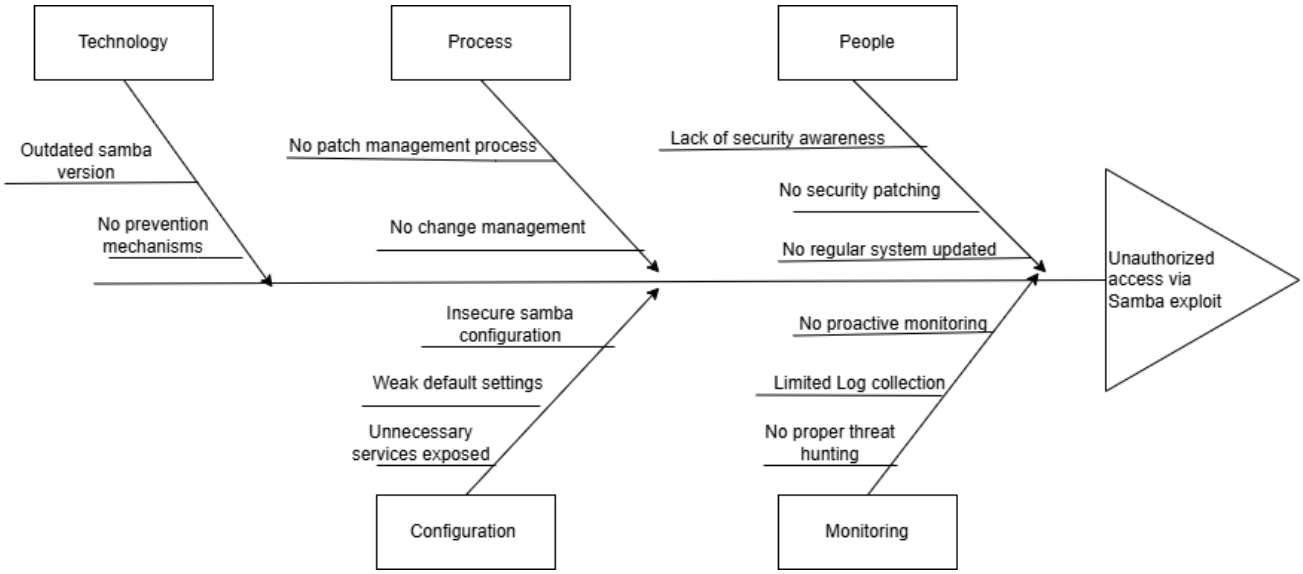
  - Why was the patch management missing ?

The system lacked in regular maintenance and security.

- Why was no regular maintenance and security ?
  Because the process was not properly implemented and maintained.

The root cause analysis say that the incident might have occurred due to inadequate patch management process and absence of security controls which led to samba exploit.

- Fishbone Diagram :



- **Metrics Reporting** :
  The objective is to measure the SOC performance using standard metrics. In elastic dashboard it should contain components like alert timeline, response duration, IP blocked.

  - **Mean Time to Detect** :
    Attack start Time : 14:00
    Alert generated : 16:00
    MTTD – 2 Hours

  - **Mean Time to Respond** :
    Alert Time : 16:00
    IP blocked : 20:00
    MTTR – 4 Hours

  - **Dwell Time** :
    Attack Start :14:00
    Containment : 20:00
    Dwell Time – 6 Hours

Stakeholder Briefing :

On 18th August 2025 a security incident took place which involved unauthorized system access and it was timely identified and contained within the organization. The activity was detected by the security monitoring and was investigated by security operations team. The affected system was isolated and the attackers IP was blocked as containment.

The incident was detected within 2 hours and fully contained within 4 hours of detection which resulted in total presence of 6 hours. No evidence of loss of data or further system compromise was found.

Following the incident improvements were needed in continuous monitoring, patch management and increase in automation so as to reduce manual work and less time of incident handling.

Overall the incident demonstrated effective response system alongwith opportunities to improve in particular areas.