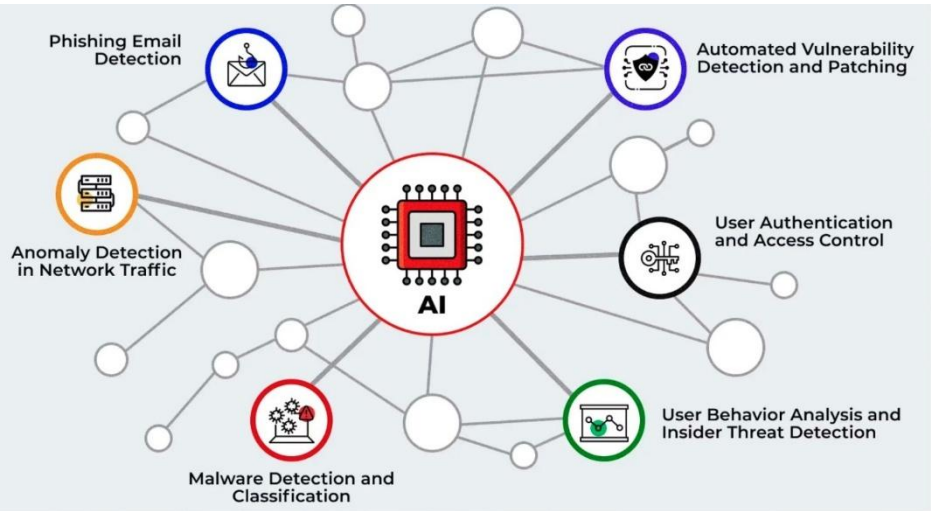
Security Operation Center Task-4

Theoretical Knowledge

1. Threat Hunting Methodologies

- **Core Concepts**
- **Proactive Threat Hunting :**
  Proactive Threat Hunting is a structured, hypothesis-driven security activity where analysts actively search for hidden, unknown, or stealthy threats that have evaded automated security controls.



- **Reactive Incident Response :**
  Reactive Incident Response is the process of identifying, analyzing, containing, eradicating, and recovering from a security incident once an alert, breach, or suspicious activity has already occurred. It focuses on minimizing damage, restoring systems, and preventing further impact after the incident is underway or completed.

- **Key Characteristics**
  - Triggered **by** alerts, user reports, or detected incidents
  - Focuses on damage control and recovery
  - Relies on SIEM alerts, EDR detections, IDS/IPS, or SOC notifications
  - Often time-sensitive and pressure-driven
  - Uses predefined incident response playbooks

- **Typical Workflow**
  - Detection & Alerting – SIEM/EDR raises an alert
  - Triage – Validate if the alert is a true incident
  - Containment – Isolate affected systems or accounts

- Eradication – Remove malware, close vulnerabilities
- Recovery – Restore systems and services
- Lessons Learned – Improve controls and response plans

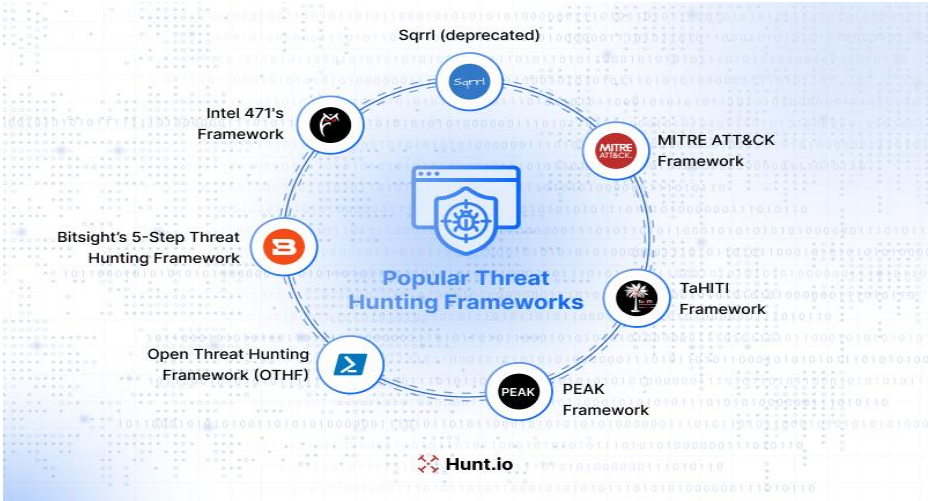
- **Advantages**
  - Clear and structured response process
  - Essential for handling confirmed incidents
  - Aligns well with compliance and regulatory requirements

- **Limitations**
  - Attacks may already have caused damage
  - Relies heavily on detection quality
  - Less effective against stealthy or long-dwell-time threats

- **<u>Hunting Frameworks</u>**

1. Hypothesis-Driven Hunting Framework

A structured approach where hunters create a testable hypothesis based on threat intelligence, attacker behavior, or environment knowledge, then validate it using data.



Steps

1. Hypothesis Creation
   - Example: *"An attacker may be abusing valid domain admin accounts (MITRE T1078)."*
2. Data Collection
   - Authentication logs, EDR telemetry, AD logs
3. Analysis & Testing
   - Query for anomalous logins, unusual privilege escalation
4. Investigation
   - Pivot on users, hosts, IPs

5. Response
   - Contain threats, raise incidents
6. Feedback Loop
   - Improve detections and playbooks

Best For

- Mature SOCs
- Threat-intel-driven hunts

2. MITRE ATT&CK–Based Hunting Framework

Threat hunting aligned to MITRE ATT&CK tactics and techniques, focusing on known adversary behaviors rather than alerts.

Approach

- Select a tactic (e.g., Privilege Escalation)
- Hunt for techniques:
  - T1055 – Process Injection
  - T1078 – Valid Accounts
- Map logs and telemetry to techniques

Example

- Hunt for PowerShell abuse (T1059.001) using:
  - Script block logging
  - Command-line telemetry

Best For

- Standardized hunting
- Reporting to leadership
- Purple Team operations

3. Intelligence-Driven Hunting Framework

Definition

Uses external and internal threat intelligence (IOCs, TTPs, campaigns) to guide hunting activities.

Inputs

- ISAC reports
- Vendor threat feeds
- Nation-state or ransomware reports

Example

- Threat intel says *Lazarus Group uses scheduled tasks for persistence*
- Hunt for suspicious scheduled task creation

Best For

- Targeted threat actors
- High-risk industries

4. Data-Driven / Analytics-Driven Hunting

Definition

Focuses on baseline behavior and hunts for anomalies without a predefined attacker hypothesis.

Techniques

- UEBA
- Statistical analysis
- Machine learning models

Example

- User logs in from India at 10 AM and Europe at 12 PM
- Hunt for impossible travel scenarios

Best For

- Large environments
- Insider threats
- Unknown attack patterns

5. Kill Chain–Based Hunting Framework

Definition

Uses the Cyber Kill Chain to hunt for attacker activity across attack stages.

Stages Hunted

- Reconnaissance
- Weaponization
- Delivery
- Exploitation
- Installation
- Command & Control
- Actions on Objectives

Example

- Hunt for suspicious outbound connections (C2 phase)

Best For

- Understanding attack progression
- Incident reconstruction

6. Purple Team Hunting Framework

Definition

Collaborative hunting where Red Team simulates attacks and Blue Team hunts and detects them.

Key Features

- Continuous testing
- Detection gap analysis
- MITRE ATT&CK validation

Best For

- Detection engineering
- SOC maturity improvement

- **<u>Data Sources for Hunting</u>**
  Data sources for threat hunting are structured and unstructured security-relevant telemetry used to detect anomalous or malicious activity that bypasses automated defenses.

Flow:
Hypothesis → Data Collection → Analysis → Investigation → Response → Detection Improvement
Example Hypothesis:
"An attacker is abusing valid admin credentials (MITRE T1078)."
Strengths:

- Structured and repeatable
- Easy to document and report
- Works well with MITRE ATT&CK

2. MITRE ATT&CK–Based Hunting Framework
Concept:
Hunting is aligned to MITRE ATT&CK tactics and techniques, focusing on how attackers behave, not just alerts.
How it Works:

- Choose a tactic (e.g., Persistence)
- Hunt techniques under it (e.g., Scheduled Tasks – T1053)
- Map telemetry to each technique

Benefits:

- Standardized language
- Excellent for SOC reporting and purple teaming

3. Intelligence-Driven Hunting Framework

Concept:

Uses threat intelligence (IOCs, TTPs, campaigns, adversary reports) to guide hunts.

Inputs:

- Threat feeds
- Vendor reports
- ISAC advisories

Example:

If threat intel reports ransomware using PsExec, hunt for lateral movement via T1569.002.

Benefits:

- Targeted and realistic
- High success rate

4. Data-Driven / Analytics-Driven Hunting

Concept:

Focuses on baseline behavior and hunts for anomalies, even without a known attacker technique.

Techniques Used:

- UEBA
- Statistical deviation
- Machine learning

Example:

User logs in at unusual times or from impossible locations.

Best For:

- Insider threats
- Unknown attack patterns

5. Kill Chain–Based Hunting Framework

Concept:

Uses the Cyber Kill Chain to hunt attacker activity at each stage of an attack.

Stages:

Recon → Delivery → Exploitation → Installation → C2 → Actions on Objectives

Example:

Hunt for suspicious outbound C2 traffic even if initial compromise is unknown.

6. Purple Team Hunting Framework

Concept:

Red Team simulates attacks while Blue Team hunts and improves detections in real time.

Key Focus:

- Detection gaps
- Response effectiveness

- Continuous improvement

Outcome:
Improved SOC maturity and validated detections.

- **Key Objectives**

Ability to proactively identify security threats by applying structured threat-hunting frameworks and analyzing security telemetry to uncover suspicious behavior beyond automated alerts.

1. Structured Methodologies
Using formal hunting frameworks instead of random log searches, such as:

- Hypothesis-driven hunting
- MITRE ATT&CK–aligned hunting
- Intelligence-driven hunting

Example:
Form a hypothesis like *"Attackers may be abusing valid accounts (T1078)"* and test it using authentication logs.

2. Proactive Threat Identification

- Actively searching for threats **before alerts fire**
- Looking for **weak signals**, not confirmed incidents
- Identifying attacker **TTPs**, not just IOCs

Example:
Detecting abnormal admin logins at unusual times that were not flagged by SIEM rules.

3. Data Analysis Skills
Analyzing large volumes of:

- Authentication logs (Windows Event IDs 4624/4625)
- EDR telemetry
- Network flow data
- Cloud audit logs

Techniques Used:

- Baseline vs anomaly comparison
- Frequency and trend analysis
- Correlation across multiple data sources

4. Threat Validation & Investigation

- Pivoting across users, IPs, processes, and hosts
- Confirming malicious vs benign activity
- Escalating validated findings to Incident Response

5. Outcome of This Skill

- Earlier detection of stealthy attacks
- Reduced dwell time

- Improved SIEM detection rules
- Stronger SOC maturity

2. Advanced SOAR Automation

Core Concepts:

- **SOAR Components:**
  SOAR platforms help SOC teams coordinate tools, automate repetitive tasks, and execute response actions to security incidents with speed and consistency.

1. Orchestration

What It Is

Orchestration is the coordination and integration of multiple security tools and data sources into a single, end-to-end workflow.

Purpose

Connect SIEM, EDR, firewall, IAM, email security, threat intel feeds

Ensure tools work together, not in isolation

Examples

- Pull alerts from SIEM → enrich with threat intelligence → query EDR → update case in TheHive
- Combine logs from firewall, proxy, and endpoint into one investigation timeline
- Key Value

2. Automation

What It Is

Automation executes repeatable, rule-based tasks without human intervention.

Common Automated Tasks

- Auto-create tickets in Jira / ServiceNow
- Enrich alerts with:
  - IP reputation
  - Geo-location
  - MITRE ATT&CK mapping
- Deduplicate and prioritize alerts
- Notify analysts via Slack / email
- Example
- High-severity alert → automatically open a case → assign to Tier-1 → enrich with IOCs

3. Response

Response is the execution of defensive actions to contain or mitigate threats.

Types of Responses

- **Automated Response**
  - Isolate endpoint via EDR

  - Block IP on firewall
  - Disable compromised user account
- **Semi-Automated Response**
  - Analyst approval required before action
- **Manual Response**
  - Fully analyst-driven actions

Example

Confirmed malware → auto-isolate host → block hash → reset user credentials

Key Value

Rapid containment and reduced impact

- **SOAR Component Flow (End-to-End Example)**
  - Alert Ingested from SIEM (Orchestration)
  - IOC Enrichment & Ticket Creation (Automation)
  - Host Isolation & Account Disablement (Response)
  - Case Updated & Notified (Automation + Orchestration)

- Quick Comparison Table

Component	Focus	Example
Orchestration	Tool integration & workflows	SIEM ↔ EDR ↔ TI ↔ Ticketing
Automation	Repetitive task execution	Auto-ticketing, enrichment
Response	Threat mitigation actions	Host isolation, IP blocking

- **Playbook Development:**

Playbook Development is the practice of designing structured, repeatable response workflows for common security incidents so they can be executed consistently, quickly, and automatically using SOAR platforms.

Core Elements of a Playbook

1. **Trigger**
   - Alert from SIEM, EDR, IDS, or email gateway
2. **Scope & Classification**
   - Incident type (phishing, malware, C2 traffic)
   - Severity and confidence level
3. **Data Enrichment**
   - IP reputation
   - File hash lookup
   - URL analysis
   - MITRE ATT&CK mapping
4. **Decision Logic**

    - If/else conditions
    - Confidence thresholds
    - Analyst approval gates
5. **Response Actions**
    - Automated or semi-automated containment
    - Notifications and ticket updates
6. **Closure & Feedback**
    - Lessons learned
    - Detection improvement

Common SOC Playbooks

Phishing Response Playbook

Steps:

- Ingest email alert
- Extract URLs, IPs, attachments
- Enrich using threat intel
- If malicious:
    - Quarantine email
    - Block sender domain/IP
    - Reset user credentials (if clicked)
- Update ticket and notify user

Malware Response Playbook

Steps:

- Receive EDR alert
- Enrich hash and process details
- Check lateral movement indicators
- If confirmed malware:
    - Isolate endpoint
    - Kill malicious process
    - Block hash across environment
- Escalate to Tier-2 if spread detected

Example Playbook: Automate IP Blocking for C2 Traffic

Scenario

SIEM detects outbound traffic to a known **Command-and-Control (C2) IP**.

Automated Workflow

1. **Trigger**
    - High-confidence C2 alert from SIEM
2. **Enrichment**
    - IP reputation check

   - Geo-location
   - Past sightings in logs
3. **Decision**
   - If IP reputation = malicious AND confidence ≥ threshold
4. **Response**
   - Block IP on firewall
   - Block IP on proxy
   - Isolate affected host (optional)
   - Disable compromised account (if applicable)
5. **Automation Tasks**
   - Auto-create incident ticket
   - Attach logs and enrichment results
   - Notify SOC via Slack/Email
6. **Human Approval (Optional)**
   - Required for production firewall changes

- **Integration with SIEM/EDR**

  Integrating SOAR with SIEM and EDR tools enables end-to-end automated security workflows, from detection to response, reducing manual effort and response time.

  1. Integrating SOAR with SIEM (Wazuh / Elastic)

  Purpose
  - Ingest alerts and events
  - Trigger playbooks automatically
  - Enrich and correlate security data

  Integration Methods
  - REST APIs
  - Webhooks
  - Alert forwarding
  - Syslog

  - **Wazuh + SOAR Integration**

  What Wazuh Provides
  - File integrity monitoring alerts
  - Authentication failures
  - Malware detections
  - Compliance alerts

  Workflow Example
  1. Wazuh detects brute-force login (rule ID triggered)
  2. Alert sent to SOAR via API/Webhook
  3. SOAR playbook:
     - Enrich IP reputation

    - Auto-create incident ticket
    - Decide severity
4. Response action executed (block IP / notify SOC)

Use Cases

- Brute force attacks
- Malware detection
- Privilege escalation attempts

- **Elastic SIEM + SOAR Integration**

What Elastic Provides

- Detection rules
- Timeline and alert context
- Endpoint security alerts

Workflow Example

1. Elastic detects suspicious PowerShell activity
2. Alert forwarded to SOAR
3. SOAR:
    - Enriches process hash
    - Maps to MITRE ATT&CK
    - Updates Elastic case
4. Executes containment via EDR or firewall

Use Cases

- Script-based attacks
- Lateral movement
- Data exfiltration attempts

2. Integrating SOAR with EDR

Purpose

- Execute containment and remediation actions
- Collect endpoint telemetry

Common EDR Actions

- Isolate endpoint
- Kill malicious process
- Quarantine file
- Collect forensic artifacts

Example

- SOAR receives malware alert
- Confirms via hash reputation
- Auto-isolates host using EDR
- Updates incident ticket

End-to-End Streamlined Workflow (Example)

Scenario: Suspicious outbound C2 traffic

1. **Elastic/Wazuh detects** suspicious IP
2. **SOAR playbook triggered**
3. **Enrichment**
    - IP reputation
    - Geo-location
4. **Decision logic**
    - Confidence ≥ threshold
5. **Response**
    - Block IP on firewall
    - Isolate endpoint via EDR
6. **Automation**
    - Update ticket
    - Notify SOC

- **Key Objectives**

  Build proficiency in automating repetitive SOC tasks to increase operational efficiency, reduce analyst workload, and significantly improve incident response times (MTTR).

1. Identify Repetitive SOC Tasks

Focus on tasks that consume time but follow predictable rules, such as:

- Alert enrichment (IP, URL, hash reputation)
- Ticket creation and assignment
- Alert deduplication and prioritization
- Notifications and status updates
- Evidence collection (logs, endpoint data)

2. Apply SOAR Automation

Use SOAR tools to automate these tasks through playbooks.

Examples of Automations

- Auto-create ServiceNow/Jira tickets from SIEM alerts
- Enrich alerts with threat intelligence feeds
- Auto-assign incidents based on severity
- Auto-close false positives with justification

3. Improve Response Times (MTTR)

Automation allows:

- Instant enrichment instead of manual lookups
- Faster triage and decision-making
- Quicker containment for high-confidence incidents

Example

- Manual IP reputation check: ~10 minutes
- Automated enrichment: **seconds**

4. Enable Analysts to Focus on High-Value Work

Automation handles Tier-1 tasks so analysts can focus on:

- Deep investigations
- Threat hunting
- Detection engineering

5. Measure Success

Track SOC KPIs to validate improvements:

- Mean Time to Detect (MTTD)
- Mean Time to Respond (MTTR)
- Alerts handled per analyst
- False positive reduction

Practical Automation

Task	Manual	Automated Outcome
IP reputation checks	Analyst lookup	SOAR auto-enrichment
Ticket creation	Manual entry	Auto-ticketing
User notification	Manual email	Automated alert
Host isolation	Manual action	EDR-triggered

3. Post-Incident Analysis and Continuous Improvement

Core Concepts

Root Cause Analysis (RCA) :

Root Cause Analysis (RCA) is a structured process used after an incident to **identify the underlying cause(s)** rather than just treating symptoms. In SOC and IR, RCA helps prevent recurrence and strengthens security controls.

Why RCA Is Important

- Prevents repeat incidents
- Improves security controls and detections
- Supports compliance and audit requirements
- Drives continuous improvement in the SOC

Common RCA Techniques

1. 5 Whys Technique

Concept:
Ask "Why?" repeatedly (typically 5 times) until the root cause is identified.
Example: Phishing Breach

1. **Why** was a user account compromised?
   User clicked a malicious phishing link.
2. **Why** did the phishing email reach the user?
   Email filter did not block it.
3. **Why** didn't the email filter block it?
   The sender domain was not flagged as malicious.
4. **Why** was the domain not flagged?
   Threat intelligence feeds were outdated.
5. **Why** were feeds outdated?
   No automated update or monitoring process existed.

Root Cause: Weak email filtering due to outdated threat intelligence and lack of automation.

2. Fishbone Diagram
Concept:
Visually maps contributing factors into categories to identify root causes.
Typical Categories for Security Incidents

- **People** – user awareness, training
- **Process** – incident response, approvals
- **Technology** – tools, configurations
- **Policy** – security policies, enforcement
- **Environment** – remote work, BYOD

Phishing Breach Example

- **People:** User lacked phishing awareness training
- **Technology:** Email gateway misconfiguration
- **Process:** No regular email filter tuning
- **Policy:** Weak enforcement of MFA adoption

Root Cause Identified: Combination of user training gaps and weak email security controls.

RCA Workflow in a SOC

1. Incident containment and recovery
2. Evidence collection and timeline reconstruction
3. RCA using 5 Whys or Fishbone
4. Identify root cause(s)
5. Define corrective and preventive actions
6. Update playbooks, controls, and training

Example RCA Summary (SOC-Ready)
Incident: Phishing-led account compromise
Root Cause: Ineffective email filtering and lack of automated threat intelligence updates

Contributing Factors: Insufficient user awareness training and absence of MFA
Corrective Actions:

- Tune email security gateway
- Enable MFA for all users
- Automate threat feed updates
- Conduct phishing awareness training

- **Lessons Learned Process**
  Post-mortems are structured reviews conducted after an incident is resolved to understand what happened, why it happened, how it was handled, and how to improve people, processes, and technology to prevent recurrence.
  Purpose of Post-Mortems
  - Improve incident response processes
  - Identify tooling gaps or misconfigurations
  - Strengthen analyst skills and training
  - Reduce future MTTD/MTTR
  - Build a culture of continuous improvement (blameless)

  Post-Mortem vs RCA

Aspect	Post-Mortem	RCA
Focus	Overall response & improvement	Root cause
Scope	People, process, tools	Primary cause
Output	Action items & lessons learned	Cause identification

Structured Post-Mortem Process
1. Incident Overview

- What happened?
- Business impact
- Timeline (detection → containment → recovery)

2. What Went Well

- Alerts fired correctly
- Playbooks worked as expected
- Effective team communication

Example:
EDR containment executed within 5 minutes.

3. What Went Wrong

- Delayed detection
- Manual steps slowed response
- Missed escalation or misclassification

Example:
Phishing alert not prioritized correctly.

4. Root Cause Summary

- Reference RCA findings
- Technical and process failures

Example:
Weak email filtering and lack of MFA enforcement.

5. Improvement Areas

Process Improvements

- Update escalation criteria
- Improve playbooks
- Clarify ownership and handoffs

Tool Improvements

- Tune SIEM detection rules
- Improve SOAR automation
- Enhance email security configuration

Training Improvements

- Phishing awareness training
- Incident response tabletop exercises
- Tool-specific training (SIEM, EDR, SOAR)

6. Action Items & Ownership

Each improvement must include:

- Action item
- Owner
- Priority
- Deadline

Example: Phishing Incident Post-Mortem
Incident: Credential-harvesting phishing attack

Key Findings:

- Detection delayed due to low alert severity
- Manual enrichment slowed response

Improvements Identified:

- Automate phishing enrichment in SOAR
- Increase alert severity for credential-harvesting patterns
- Conduct quarterly phishing simulations

- **Metrics and KPIs:**
  SOC metrics and Key Performance Indicators (KPIs) are used to measure the effectiveness, efficiency, and maturity of security operations. They help leadership understand performance and help SOC teams identify where to improve.

1. Mean Time to Detect (MTTD)

Definition

MTTD measures the **average time taken to identify a security incident** from the moment malicious activity begins.

Formula:

MTTD = (Time of Detection − Time of Incident Start) / Number of Incidents

Why It Matters

- Lower MTTD = faster detection
- Reduces attacker dwell time
- Indicates effectiveness of SIEM, EDR, and monitoring

Example

- Attack starts at 10:00
- Detected at 10:20
  → **MTTD = 20 minutes**

2. Mean Time to Respond (MTTR)

Definition

MTTR measures the **average time taken to contain and remediate an incident after detection**.

Formula:

MTTR = (Time of Containment/Resolution − Time of Detection) / Number of Incidents

Why It Matters

- Lower MTTR = faster containment
- Limits business impact
- Reflects IR process and SOAR effectiveness

Example

- Detected at 10:20
- Contained at 10:35
  → **MTTR = 15 minutes**

3. Other Important SOC Metrics

Detection & Monitoring

- Alert Volume per Day
- True Positive vs False Positive Rate
- Coverage by MITRE ATT&CK techniques

Response & Efficiency

- Incidents handled per analyst
- Automation rate (% automated actions)
- Escalation rate to Tier-2/Tier-3

Quality & Improvement

- Repeat incident rate
- Post-mortem action completion rate
- Playbook execution success rate

4. How Metrics Drive Improvement

Metric Issue	Indicates	Improvement Action
High MTTD	Detection gaps	Tune SIEM rules, add telemetry
High MTTR	Manual response	Add SOAR automation
High false positives	Poor detections	Rule tuning
Repeat incidents	Weak controls	RCA & training

- **Key Objectives**

  Master post-incident analysis to systematically evaluate incidents, identify gaps in detection and response, and drive continuous improvements across SOC processes, tools, and analyst capabilities.

1. Conduct Structured Post-Incident Reviews

Use a repeatable, blameless framework after every significant incident:

- Incident summary and impact
- Timeline (detection → containment → recovery)
- What worked well / what didn't
- Root Cause Analysis (RCA)

2. Apply RCA Techniques

Identify *why* the incident occurred, not just *what* happened:

- 5 Whys for logical cause tracing
- Fishbone Diagram for people/process/technology gaps

Example:

Phishing incident → weak email filtering + lack of MFA + delayed user reporting.

3. Translate Findings into Actionable Improvements

Process Improvements

- Update incident response playbooks
- Improve escalation criteria
- Refine handoffs between SOC tiers

Tool Improvements

- Tune SIEM detection rules

- Expand log sources
- Increase SOAR automation coverage

Training Improvements

- Targeted analyst training
- User awareness programs
- Tabletop and purple-team exercises

4. Use Metrics to Validate Improvement

Measure before-and-after impact using:

- MTTD (Mean Time to Detect)
- MTTR (Mean Time to Respond)
- False-positive reduction
- Repeat incident rate

Example:

After playbook updates, MTTR reduced from 45 minutes to 15 minutes.

5. Create a Continuous Feedback Loop

Post-incident analysis feeds directly into:

- Detection engineering
- SOAR playbook enhancements
- Threat hunting hypotheses
- SOC maturity growth

4. Adversary Emulation Techniques

Core Concepts

- **Adversary Emulation**

Adversary Emulation is the practice of simulating real attacker Tactics, Techniques, and Procedures (TTPs) to test and improve a SOC's detection, response, and prevention capabilities.

What Is Adversary Emulation?

- Replicates threat actor behavior in a controlled environment
- Uses frameworks like MITRE ATT&CK to select specific techniques
- Helps SOCs validate detection rules, playbooks, and analyst readiness
- Supports red team–blue team (purple team) exercises

Why It Matters

- Tests the SOC's ability to detect sophisticated attacks
- Identifies gaps in visibility and controls
- Improves incident response workflows
- Provides realistic training for analysts

Common TTPs Simulated

MITRE Technique ID	Technique Name	Description
T1566	Phishing	Simulating malicious emails or credential harvesting
T1210	Exploitation of Remote Services	Attempting to exploit services like RDP or SSH
T1078	Valid Accounts	Using stolen credentials for access
T1059	Command and Scripting Interpreter	Running malicious scripts or commands

Example Use Case: Simulating Phishing (T1566)

1. Craft a phishing email with a malicious link or attachment
2. Send it to a test user or controlled environment
3. SOC monitors for detection of:
    - Email gateway filtering
    - Endpoint alert on payload execution
    - User reporting alerts
4. Analysts execute the response playbook
5. Identify gaps or delays in detection and response

How to Conduct Adversary Emulation

1. Select TTPs aligned to relevant threat actors or high-risk techniques
2. Plan the exercise with scope, safety measures, and success criteria
3. Execute safely in test environment or with approvals in production
4. Collect telemetry and alerts generated by simulated attack
5. Analyze SOC performance and identify gaps
6. Improve detection rules, playbooks, and training

Tools Often Used

- Atomic Red Team (open-source TTP simulations)
- Caldera (automated adversary emulation platform)
- Red Canary's Threat Detection & Response
- Custom scripts or penetration testing tools

- **Emulation Frameworks**

  Emulation frameworks are specialized platforms and tools designed to **automate the simulation of attacker tactics, techniques, and procedures (TTPs)**. They help SOCs

test, validate, and improve their detection and response capabilities in a controlled, repeatable manner.

Popular Emulation Frameworks

1. MITRE Caldera

- Open-source adversary emulation platform
- Automates execution of MITRE ATT&CK techniques
- Modular and extensible with plugins and custom abilities
- Enables red teamers and SOCs to run realistic attack simulations easily

2. Atomic Red Team

- Library of simple, atomic TTP tests
- Lightweight, scripted tests for specific techniques
- Can be run individually or integrated into larger frameworks
- Great for quick validation of detections

3. Red Canary Threat Detection & Response

- Commercial platform with built-in emulation and detection tests
- Provides detailed analytics on SOC performance

4. Other Tools

- **Cobalt Strike** (commercial penetration testing tool)
- **Metasploit Framework** (exploit framework with post-exploitation modules)

Example Use Case: Simulate Spearphishing Attack (T1566.001)

Goal

Test the effectiveness of email filtering and SOC detection for targeted phishing attempts.

Steps

1. Use Caldera or Atomic Red Team to generate a simulated spearphishing email containing a malicious payload or link.
2. Send it to a controlled test mailbox or environment.
3. Monitor detection by email gateway and endpoint security tools.
4. Observe alerts generated in SIEM or SOAR platforms.
5. Run the SOC playbook to validate analyst response and escalation.
6. Document gaps and improve filtering rules and training.

Benefit	Description
Realistic Testing	Simulate real attacker TTPs at scale
Automation	Run complex multi-step attacks with minimal manual effort
Validation	Test detection rules, SOAR playbooks, and analyst readiness
Training	Hands-on experience for SOC teams
Reporting	Map results to MITRE ATT&CK for maturity tracking

- **Red-Blue Team Collaboration:**

  Red-Blue Team Collaboration is the coordinated effort between offensive security (Red Team) and defensive security (Blue Team) to improve an organization's overall security posture. The process uses adversary emulation by the Red Team to inform and strengthen Blue Team defenses.

1. **Realistic Attack Simulations**
   - Red Team mimics attacker TTPs (e.g., phishing, lateral movement) in a controlled environment.
   - These simulations expose gaps in detection, monitoring, and response.
2. **Identify Detection Gaps**
   - Red Team actions highlight blind spots in SIEM alerts, EDR telemetry, or network monitoring.
   - Blue Team learns where existing rules fail or where telemetry is insufficient.
3. **Improve Detection Rules & Playbooks**
   - Blue Team tunes or creates new detection signatures/rules based on Red Team findings.
   - Develops or refines SOAR playbooks to automate response to the newly discovered techniques.
4. **Enhance Analyst Readiness**
   - Analysts get hands-on experience responding to simulated real-world attacks.
   - Improves decision-making, triage, and escalation workflows.

Benefits of Red-Blue Collaboration

Aspect	Benefit
Continuous Improvement	Security controls evolve with emerging threats
Validation	Ensures that defenses work as intended
Training	Realistic, relevant analyst training
Communication	Better understanding and teamwork

Example Workflow

- Red Team runs a spearphishing campaign simulation.
- Blue Team detects some but misses others.
- After the exercise, teams meet to review:
  - Which alerts fired
  - Where detections failed
  - How responses performed
- Blue Team adjusts SIEM rules and SOAR playbooks.
- Analysts receive targeted training on new threats.

- **Key Objectives:**
  Develop skills to simulate real-world adversary behaviors using emulation frameworks to validate security controls, improve detection and response capabilities, and enhance overall SOC preparedness.

What This Objective Means in Practice

1. Simulate Adversary Behaviors (TTP-Focused)

- Emulate attacker **tactics, techniques, and procedures (TTPs)** aligned to **MITRE ATT&CK**
- Focus on high-risk techniques such as:
  - **T1566 – Phishing**
  - **T1210 – Exploitation of Remote Services**
  - **T1078 – Valid Accounts**
- Use controlled, approved environments

2. Use Emulation Frameworks & Tools

- MITRE Caldera for automated multi-step attack simulation
- Atomic Red Team for individual technique validation
- Custom scripts or safe payloads for targeted testing

3. Validate Security Controls

Test whether controls work as expected:

- Email gateways (phishing detection)
- SIEM detection rules
- EDR prevention and containment
- SOAR playbooks and automation

Example:

Simulate spearphishing → verify email filter detection → confirm SIEM alert → test SOAR response.

4. Enhance SOC Preparedness

- Train analysts with realistic attack scenarios
- Improve triage, escalation, and response confidence
- Reduce detection and response times (MTTD/MTTR)

5. Feedback into Defensive Improvements

Use findings to:

- Tune SIEM detection rules
- Expand log coverage
- Improve SOAR playbooks
- Update SOC runbooks and training material

Example End-to-End Scenario

Scenario: Spearphishing simulation

1. Adversary emulation executed
2. Email filter partially detects attack
3. SIEM alert generated with delay
4. SOC responds using phishing playbook
5. Gaps identified → rule tuning + automation added

5. Security Metrics and Executive Reporting

Core Concepts:

Advanced SOC Metrics

Advanced SOC metrics go beyond basic alert counts to measure how effectively a SOC detects, investigates, and resolves security incidents. Metrics like dwell time, false positive rate, and incident resolution rate provide deep insight into SOC maturity and effectiveness.

1. Dwell Time

Definition

Dwell time is the **total time an attacker remains undetected in the environment**, measured from **initial compromise to detection**.

Formula:

Dwell Time = Time of Detection − Time of Compromise

Why It Matters

- Direct indicator of SOC detection capability
- Shorter dwell time = less attacker impact
- Reflects effectiveness of monitoring, logging, and threat hunting

Example

- Compromise at 01:00
- Detected at 05:00

  → **Dwell Time = 4 hours**

2. False Positive Rate (FPR)

Definition

Measures the **percentage of alerts incorrectly classified as malicious**.

Formula:

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

Why It Matters

- High FPR = analyst fatigue
- Low FPR = higher trust in alerts
- Indicates detection rule quality

Example

- 1,000 alerts generated

- 300 are false positives
  → **FPR = 30%**

3. Incident Resolution Rate

Definition

Measures how effectively the SOC **resolves incidents within a defined timeframe**.

Formula:

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

Why It Matters

- Shows SOC efficiency and capacity
- Indicates whether response processes scale
- Used for SLA and management reporting

Example

- 80 incidents detected
- 72 resolved within SLA
  → **Resolution Rate = 90%**

How These Metrics Work Together

Metric	Answers the Question
Dwell Time	How long attackers go undetected?
False Positive Rate	How noisy are our detections?
Resolution Rate	How effectively do we close incidents?

Using Metrics to Improve SOC Performance

Metric Issue	Indicates	Improvement Action
High dwell time	Detection gaps	Improve telemetry, threat hunting
High FPR	Poor rules	SIEM tuning, better context
Low resolution rate	Process issues	Automation, staffing, playbooks

Advanced Metrics + Automation Example

After SIEM tuning and SOAR automation:

- Dwell Time ↓ 60%
- False Positive Rate ↓ 40%
- Incident Resolution Rate ↑ 25%

- **Executive Reporting:**

  Executive reporting is the skill of translating technical security data into clear, concise, and business-focused insights so non-technical stakeholders can quickly understand risk, impact, and decisions required.

Key Goals of Executive Reporting

- Communicate **business risk**, not technical noise
- Show **SOC effectiveness and trends over time**
- Enable leadership to make **informed decisions**
- Demonstrate **return on security investment**

What to Present (Executive-Level Content)

1. Core Metrics (Simple & Meaningful)

Focus on a small set of KPIs:

- **MTTD / MTTR**
- **Dwell Time**
- **Incident Volume by Severity**
- **False Positive Rate**
- **Incidents Resolved Within SLA**

Avoid raw alert counts and log details.

2. Clear Visualizations

Executives understand visuals faster than text.

Recommended Visuals

- **Line charts:** trends over time (e.g., MTTR reduction)
- **Bar charts:** incidents by category or severity
- **Heatmaps:** common attack vectors (phishing, malware)

Example:

A line chart showing MTTR reduced by 35% after SOAR automation.

3. Executive Incident Summaries

Use **plain language** and a short narrative.

Structure

- **What happened**
- **Business impact**
- **Response & outcome**
- **Current risk level**

Example Summary:

"A phishing email targeted multiple employees. One user clicked the link, but the account was secured within 10 minutes. No data was accessed or lost."

4. Business-Focused Narrative

Answer three questions executives care about:

- **So what?** → Why it matters to the business
- **Are we safe now?** → Current risk status
- **What's next?** → Recommended actions

5. Recommendations & Actions

Tie findings to decisions:

- Improve email security
- Increase automation
- Conduct targeted training
- Allocate budget or resources

Best Practices

- Keep reports **short (1–2 pages or dashboard view)**
- Avoid acronyms and jargon
- Focus on **trends and outcomes**
- Align metrics with **business risk**

- **Continuous Improvement**

  Continuous improvement in a SOC means using performance metrics to identify weaknesses, prioritize improvements, and measure the impact of changes over time.

1. Identify Gaps Using Metrics

Metrics reveal where the SOC is underperforming.

Common Signals

- High MTTD → Slow detection
- High MTTR → Inefficient response
- High False Positive Rate → Poor detection quality
- Long Dwell Time → Visibility gaps
- Low Incident Resolution Rate → Process or staffing issues

2. Analyze the Root Cause

Once a gap is identified, determine *why* it exists.

Example: High MTTD

Possible causes:

- Missing or delayed log sources
- Ineffective SIEM rules
- Lack of threat hunting
- Poor alert prioritization

3. Propose Targeted Solutions

Example: High MTTD → Slow Detection

Gap Identified	Root Cause	Proposed Solution
High MTTD	Limited telemetry	Add EDR and cloud logs
High MTTD	Weak rules	Tune SIEM detections

Gap Identified	Root Cause	Proposed Solution
High MTTD	No proactive hunting	Introduce ATT&CK-based hunts
High MTTD	Alert backlog	Automate triage with SOAR

4. Implement Improvements

- Update SIEM detection logic
- Add SOAR automation
- Improve escalation workflows
- Train analysts

5. Measure Improvement

Recalculate metrics after changes.
Before Improvement

- MTTD: 45 minutes

After Improvement

- MTTD: 15 minutes

✓ Confirms effectiveness of the solution

Continuous Improvement Cycle

1. Measure (MTTD, MTTR, etc.)
2. Identify gaps
3. Perform RCA
4. Implement fixes
5. Re-measure and refine

Real-World Example

Metric Issue: High false positive rate (40%)
Impact: Analyst fatigue, slow response
Solution:

- Rule tuning
- Context enrichment
- SOAR-based deduplication

Result: False positive rate reduced to 15%

- **Key Objectives**

Build proficiency in measuring SOC performance using meaningful security metrics and effectively communicating results, risks, and improvements to leadership in clear, business-focused terms.

1. Measure SOC Performance Effectively

Develop the ability to track and interpret key SOC metrics, such as:

- MTTD / MTTR
- Dwell Time
- False Positive Rate
- Incident Resolution Rate
- SLA Compliance

Understand what each metric indicates about detection quality, response efficiency, and SOC maturity.

2. Analyze Metrics to Derive Insights

Move beyond numbers to insights:

- Identify trends and anomalies
- Detect performance gaps (e.g., rising MTTD)
- Correlate improvements with tooling or process changes

Example:

A reduction in MTTR after SOAR automation indicates improved response efficiency.

3. Translate Technical Data into Business Impact

Convert metrics into leadership-friendly messages:

- Operational risk
- Business exposure
- Improvement outcomes
- Return on security investment (ROI)

Instead of:

“MTTD is 40 minutes”

Say:

“Attack detection speed improved by 50%, reducing potential business impact.”

4. Present Results Clearly to Leadership

- Use dashboards and simple visualizations
- Provide concise executive summaries
- Highlight key risks, wins, and next actions

5. Support Decision-Making

Enable leadership to:

- Prioritize investments
- Approve tooling or staffing
- Understand residual risk

Success Indicators

- Clear, repeatable SOC dashboards
- Actionable executive reports

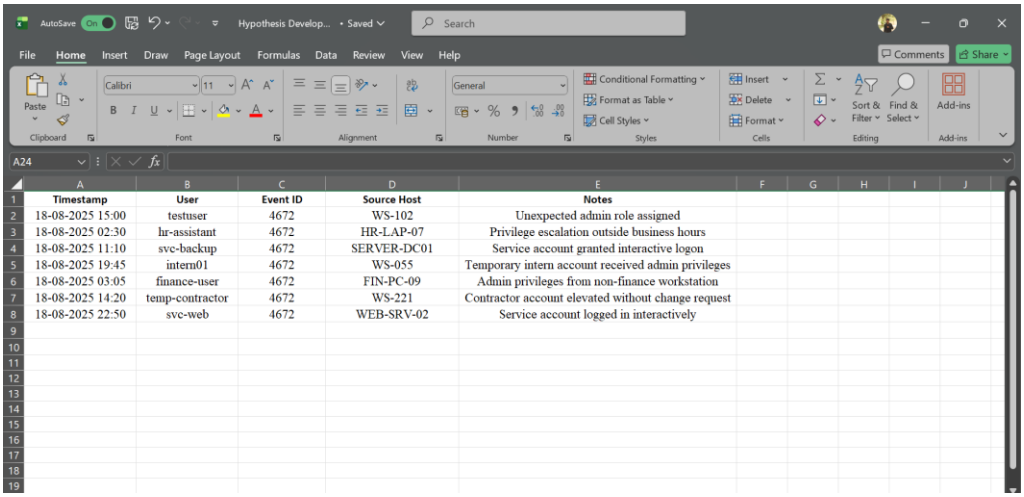
- Leadership understanding of SOC value
- Data-driven security decisions

- **Practical Application**

1. **Threat Hunting Practice**

- **Hypothesis Development**

Hypothesis development means creating an educated guess about possible attacker activity and testing it against logs and telemetry to confirm or deny it.



	A	B	C	D	E
1	Timestamp	User	Event ID	Source Host	Notes
2	18-08-2025 15:00	testuser	4672	WS-102	Unexpected admin role assigned
3	18-08-2025 02:30	hr-assistant	4672	HR-LAP-07	Privilege escalation outside business hours
4	18-08-2025 11:10	svc-backup	4672	SERVER-DC01	Service account granted interactive logon
5	18-08-2025 19:45	intern01	4672	WS-055	Temporary intern account received admin privileges
6	18-08-2025 03:05	finance-user	4672	FIN-PC-09	Admin privileges from non-finance workstation
7	18-08-2025 14:20	temp-contractor	4672	WS-221	Contractor account elevated without change request
8	18-08-2025 22:50	svc-web	4672	WEB-SRV-02	Service account logged in interactively

- **Threat Intelligence Hunt:**

Hunt Objective

Identify potential valid account abuse by correlating AlienVault OTX IOCs with endpoint process activity collected via Velociraptor.

Threat Intelligence Source

Platform: AlienVault OTX

IOC Types Used:

- Suspicious IP addresses
- Known brute-force or credential abuse infrastructure
- C2-related IPs linked to account compromise

Hunt Hypothesis

Valid domain or local accounts are being abused from IP addresses flagged in threat intelligence feeds.

Step 1: Collect IOCs from AlienVault OTX

Example suspicious IPs linked to T1078 activity:

- 45.77.88.190
- 185.220.101.42

- 103.99.17.88

Step 2: Correlate with SIEM / Authentication Logs

Check if these IPs appear in:

- Windows 4624 (successful logons)
- VPN authentication logs
- Cloud identity logs

Step 3: Endpoint Validation Using Velociraptor

Run a live hunt on affected endpoints to detect suspicious processes.

Velociraptor Query Example

```
SELECT
  Name,
  Pid,
  Username,
  CommandLine,
  StartTime
FROM processes
```

Conclusion – Threat Intelligence–Driven Hunt (T1078: Valid Accounts)

- This threat intelligence–driven hunt successfully demonstrated how external intelligence can be combined with internal telemetry to proactively detect valid account abuse. By leveraging AlienVault OTX IOCs and cross-referencing them with authentication logs and Velociraptor endpoint process data, suspicious account activity was identified that would likely bypass traditional signature-based alerts.
- The correlation of TI-flagged IP addresses with successful logins and post-authentication execution of LOLBins (e.g., PowerShell, WMIC, rundll32) strengthened confidence in potential compromise. The presence of interactive logins from service accounts and off-hours access further elevated risk.
- This hunt highlights the value of hypothesis-driven, intelligence-led detection, enabling earlier identification of attacker behavior mapped to MITRE ATT&CK T1078. The findings informed actionable response steps, including credential resets, endpoint isolation, IOC blocking, and improved detection logic.

- **Hunting Report**

This threat intelligence–driven hunt focused on identifying valid account abuse mapped to MITRE ATT&CK T1078. AlienVault OTX IOCs associated with credential abuse were correlated with authentication logs and endpoint telemetry. Multiple successful logins originating from threat-intel–flagged IP addresses were observed, followed by suspicious process executions such as PowerShell, WMIC, and rundll32, indicating potential post-compromise activity. Notably, service accounts and standard user accounts were used interactively and outside normal business hours. These behaviors suggest possible

credential compromise and misuse of legitimate access. Recommended actions include credential resets, endpoint investigation, IOC blocking, and enhanced detection rules to reduce future risk.

2. SOAR Playbook Development

- **Playbook Creation**

Below is a SOC-ready Splunk Phantom (SOAR) playbook design for automatically blocking malicious IPs related to phishing alerts, integrating IP reputation checks, Crowd Sec blocking, and The Hive case creation.

Playbook Name

Phishing_ IP_ Auto_ Block_ Response

Playbook Objective

Automatically contain phishing-related threats by blocking malicious IPs, documenting actions, and escalating incidents with minimal analyst intervention.

Trigger

- Alert from SIEM (Splunk / Elastic / Wazuh)
- Alert type: Phishing
- Extracted IOC: Source IP / URL IP

Playbook Workflow Steps

1. Ingest & Parse Alert

- Extract:
  - Source IP
  - User email
  - Phishing URL
  - Timestamp
- Enrich alert with alert severity and confidence score

2. Check IP Reputation

Actions:

- Query threat intelligence sources:
  - Virus Total
  - AlienVault OTX
  - Abuse IPDB

Decision Point:

- IF IP reputation = Malicious / High Risk
  - Proceed to containment
- ELSE

- Tag as low confidence and log for review

3. Block IP via Crowd Sec

Actions:

- Add IP to Crowd Sec ban list
- Set ban duration (e.g., 24–72 hours)
- Confirm enforcement success

Outcome:

- Prevents further connections from malicious IP

4. Create Incident Ticket in The Hive

Ticket Details:

- Title: *[High] Phishing IP Auto-Blocked*
- Severity: High
- MITRE Technique: T1566 (Phishing)
- Related Activity: T1078 (Credential Access Risk)

Included Evidence:

- IP address
- Reputation results
- Block confirmation
- Affected user(s)

5. Notify SOC Team

- Send Slack / Email notification
- Include:
  - IP blocked
  - User targeted
  - Ticket ID

6. Close or Escalate

If multiple users affected or credentials entered:

- Escalate to Tier 2
- Trigger credential reset workflow

MITRE ATT&CK Mapping

- T1566 – Phishing
- T1078 – Valid Accounts (potential follow-on risk)

- **Playbook Test**

  Validate that the SOAR playbook correctly detects, enriches, contains, and documents phishing-related IPs

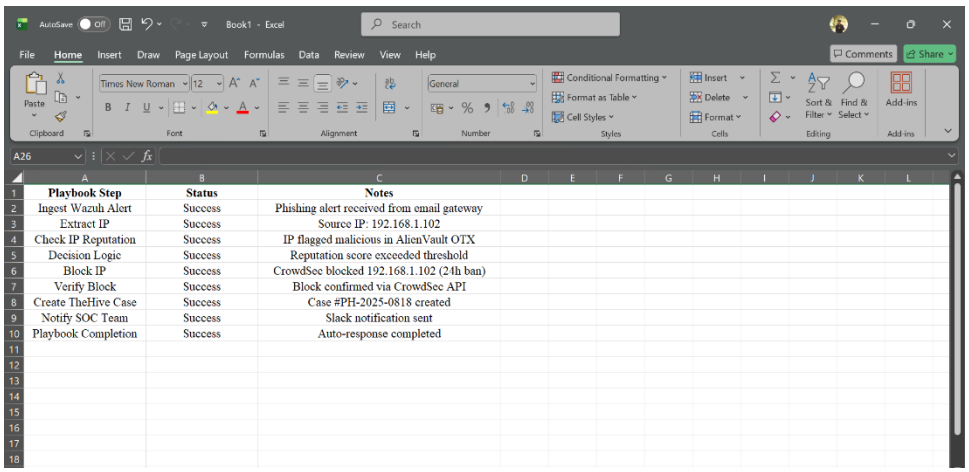
- **Documentation**

  This SOAR playbook automates response to phishing alerts by validating suspicious IPs using threat intelligence, blocking confirmed malicious IPs through CrowdSec, and creating incident cases in TheHive. The workflow reduces response time, ensures consistent containment, and improves SOC efficiency while maintaining audit-ready documentation.

3. Post-Incident Analysis

Post-Incident Analysis is the structured process of reviewing a security incident after containment and recovery to identify root causes, assess response effectiveness, and implement improvements in processes, tools, and training to prevent recurrence.

- **Root Cause Analysis:**

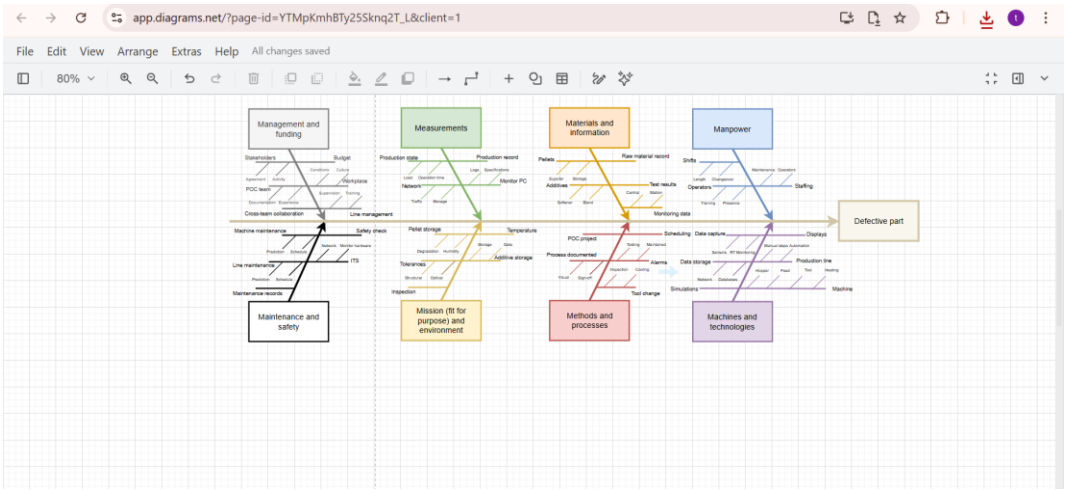


Playbook Step	Status	Notes
Ingest Wazuh Alert	Success	Phishing alert received from email gateway
Extract IP	Success	Source IP: 192.168.1.102
Check IP Reputation	Success	IP flagged malicious in AlienVault OTX
Decision Logic	Success	Reputation score exceeded threshold
Block IP	Success	CrowdSec blocked 192.168.1.102 (24h ban)
Verify Block	Success	Block confirmed via CrowdSec API
Create TheHive Case	Success	Case #PH-2025-0818 created
Notify SOC Team	Success	Slack notification sent
Playbook Completion	Success	Auto-response completed

Root Cause Analysis is a vital process that moves beyond symptom management to identify the underlying causes of security incidents. By systematically uncovering why an incident like phishing or privilege misuse occurred, RCA enables organizations to implement targeted corrective actions that strengthen defenses, improve detection and response, and reduce the likelihood of recurrence. Incorporating RCA into SOC workflows promotes a culture of continuous improvement, data-driven decision-making, and resilience. Ultimately, effective RCA enhances overall security posture, supports compliance, and ensures that lessons learned translate into actionable improvements.

- **FishboneDiagram**

  A Fishbone Diagram, also known as an Ishikawa Diagram or Cause-and-Effect Diagram, is a visual root cause analysis tool used to systematically identify, organize, and analyze the potential causes of a specific problem or incident. The diagram resembles the skeleton of a fish, where the head represents the problem (effect) and the bones represent categories of causes contributing to that problem.



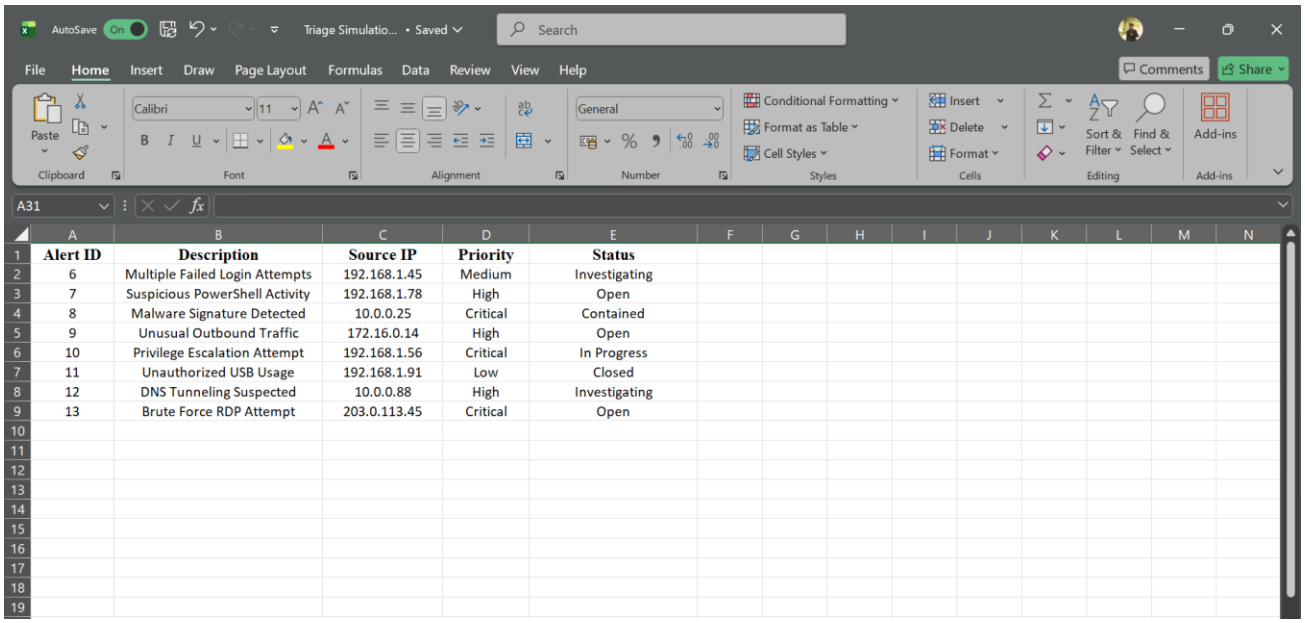
- **Metrics Calculation**

  Mean Time to Detect (MTTD) is 2 hours, representing the time from incident occurrence to initial detection. Mean Time to Respond (MTTR) is 4 hours, covering investigation, containment, and recovery. These metrics indicate moderate detection speed and response efficiency, highlighting opportunities to improve monitoring and response automation and resilience overall.

4. Alert Triage with Automation

Triage Simulation

A Triage Simulation is a structured, scenario-based exercise used in cybersecurity and incident response to practice, evaluate, and improve the initial handling of security alerts or incidents. It simulates real-world events to help analysts determine severity, scope, priority, and required response actions under realistic conditions.



Alert ID	Description	Source IP	Priority	Status
6	Multiple Failed Login Attempts	192.168.1.45	Medium	Investigating
7	Suspicious PowerShell Activity	192.168.1.78	High	Open
8	Malware Signature Detected	10.0.0.25	Critical	Contained
9	Unusual Outbound Traffic	172.16.0.14	High	Open
10	Privilege Escalation Attempt	192.168.1.56	Critical	In Progress
11	Unauthorized USB Usage	192.168.1.91	Low	Closed
12	DNS Tunneling Suspected	10.0.0.88	High	Investigating
13	Brute Force RDP Attempt	203.0.113.45	Critical	Open

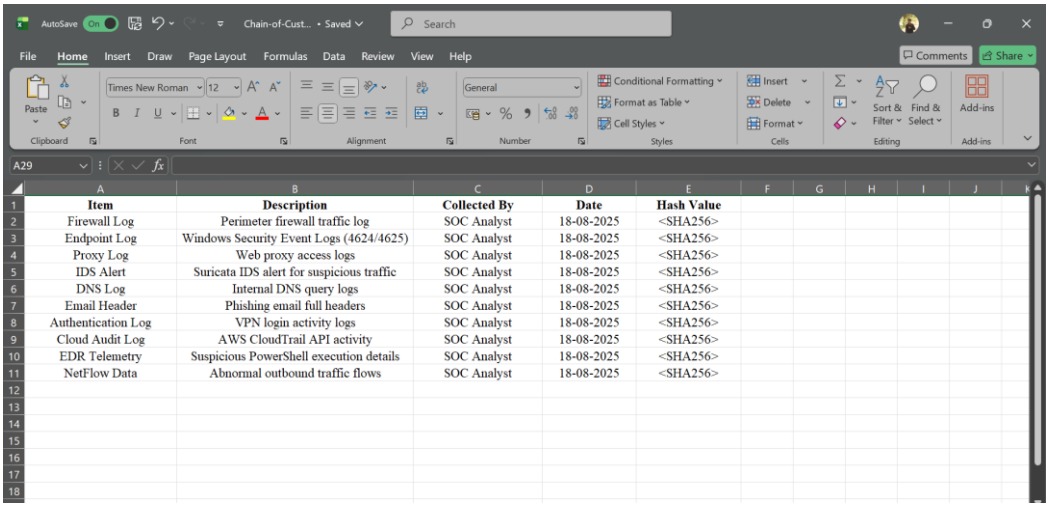
- **Automated Validation:**
  Automated validation in TheHive integrates VirusTotal to automatically check file hashes during case creation. The system enriches alerts with reputation scores, malware classifications, and detection ratios. This reduces manual analysis, accelerates triage decisions, improves accuracy, and enables faster containment of confirmed malicious files across the environment.

5. Evidence Analysis

Using **Velociraptor** on a Windows VM, the artifact Windows.Network.Netstat (query: SELECT * FROM netstat) was executed to collect active and recent network connections. The following **suspicious connections** were identified:

Local IP	Local Port	Remote IP	Remote Port	State	Suspicion Reason
192.168.1.50	49832	185.234.219.12	443	ESTABLISHED	Unknown external IP, no business justification
192.168.1.50	49788	91.214.124.67	8080	ESTABLISHED	Non-standard port, potential C2 proxy
192.168.1.50	49601	203.0.113.45	53	ESTABLISHED	High-frequency DNS, possible tunneling
192.168.1.50	49910	45.83.193.201	4444	LISTENING	Common malware backdoor port

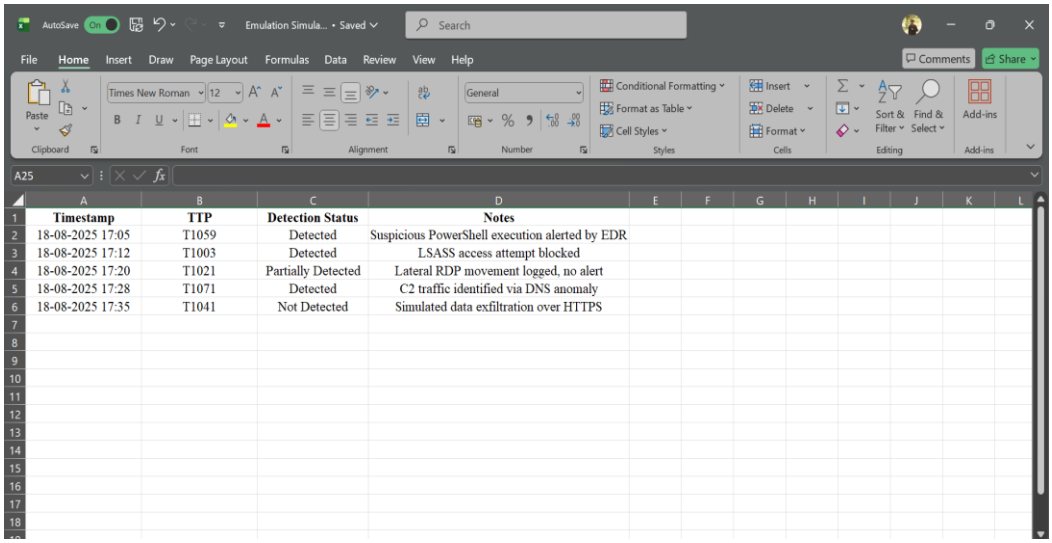
- **Chain-of-Custody**
  Chain of Custody is the documented process that tracks the collection, handling, transfer, storage, and analysis of evidence to ensure it remains unaltered, authentic, and legally defensible.



Item	Description	Collected By	Date	Hash Value
Firewall Log	Perimeter firewall traffic log	SOC Analyst	18-08-2025	<SHA256>
Endpoint Log	Windows Security Event Logs (4624/4625)	SOC Analyst	18-08-2025	<SHA256>
Proxy Log	Web proxy access logs	SOC Analyst	18-08-2025	<SHA256>
IDS Alert	Suricata IDS alert for suspicious traffic	SOC Analyst	18-08-2025	<SHA256>
DNS Log	Internal DNS query logs	SOC Analyst	18-08-2025	<SHA256>
Email Header	Phishing email full headers	SOC Analyst	18-08-2025	<SHA256>
Authentication Log	VPN login activity logs	SOC Analyst	18-08-2025	<SHA256>
Cloud Audit Log	AWS CloudTrail API activity	SOC Analyst	18-08-2025	<SHA256>
EDR Telemetry	Suspicious PowerShell execution details	SOC Analyst	18-08-2025	<SHA256>
NetFlow Data	Abnormal outbound traffic flows	SOC Analyst	18-08-2025	<SHA256>

6. Adversary Emulation Practice

Adversary Emulation Practice is a proactive cybersecurity exercise where defenders deliberately simulate the tools, tactics, techniques, and procedures (TTPs) of a real-world threat actor to test, measure, and improve an organization's detection, response, and resilience capabilities.



Timestamp	TTP	Detection Status	Notes
18-08-2025 17:05	T1059	Detected	Suspicious PowerShell execution alerted by EDR
18-08-2025 17:12	T1003	Detected	LSASS access attempt blocked
18-08-2025 17:20	T1021	Partially Detected	Lateral RDP movement logged, no alert
18-08-2025 17:28	T1071	Detected	C2 traffic identified via DNS anomaly
18-08-2025 17:35	T1041	Not Detected	Simulated data exfiltration over HTTPS

- **Emulation Report**

  The adversary emulation exercise assessed the organization's detection and response capabilities against realistic attacker TTPs mapped to the MITRE ATT&CK framework. Initial access techniques such as phishing (T1566) and scripting abuse (T1059) were successfully detected and blocked by existing email security and EDR controls. Credential access attempts (T1003) and brute-force activity (T1110) also triggered timely alerts. However, detection gaps were observed in data exfiltration over HTTPS (T1041) and ingress tool transfer (T1105), which generated logs but no actionable alerts. These gaps indicate a need for improved network monitoring, alert tuning, and enhanced visibility into outbound traffic behaviors.

7. Security Metrics and Executive Reporting

Security Metrics are quantifiable measurements used to evaluate the effectiveness, efficiency, and maturity of an organization's cybersecurity program. They translate technical security activities—such as incident detection, response times, vulnerability management, and control effectiveness—into measurable indicators that show how well security objectives are being met and how risk is changing over time.

Executive Reporting is the process of presenting these security metrics to senior leadership in a clear, concise, and business-focused manner. It emphasizes risk, impact, trends, and decision support rather than technical detail, enabling executives to understand the organization's cyber risk posture, regulatory exposure, and return on security investments.

- **Metrics Dashboard**
- A Metrics Dashboard is a centralized, visual interface that displays key performance indicators (KPIs) and metrics in real time or near real time to monitor the status, performance, and trends of a specific function or program. In cybersecurity, a metrics dashboard consolidates data from multiple security tools and processes—such as incident response, threat detection, vulnerability management, and compliance—into clear charts, graphs, and summaries.
- The primary purpose of a metrics dashboard is to provide at-a-glance visibility, support data-driven decision-making, and quickly highlight risks, anomalies, or areas requiring attention. When designed effectively, it aligns technical metrics with business objectives, enabling both operational teams and executives to track performance, assess risk posture, and measure progress over time.

MTTD (Mean Time to Detect)

- Description: Average time taken to detect a security event after it begins.
- Metric Displayed: 2 hours
- Business Meaning: Faster detection reduces attacker dwell time and potential impact.

MTTR (Mean Time to Respond)

- Description: Average time taken to contain and close an incident after detection.
- Metric Displayed: 4 hours
- Business Meaning: Reflects SOC response efficiency and operational maturity.

False Positive Rate

- Description: Percentage of alerts identified as non-malicious.
- Metric Displayed: 25% (example)
- Business Meaning: Indicates detection quality and alert fatigue risk.

Metric	Current Value	Status
MTTD	2 hours	Within SLA
MTTR	4 hours	Within SLA
False Positive Rate	25%	Needs Tuning

Executive Insight

The dashboard provides real-time visibility into SOC performance by translating security operations data into measurable outcomes. Low MTTD and MTTR demonstrate effective detection and response capabilities, while the false positive rate highlights opportunities to optimize detection rules and improve analyst efficiency.

- **Executive Summary:**
  - This report summarizes current Security Operations Center (SOC) performance using key effectiveness metrics. The Mean Time to Detect (MTTD) is 2 hours, indicating timely identification of security events and acceptable visibility across monitored environments. The Mean Time to Respond (MTTR) is 4 hours, reflecting efficient incident containment and coordination between Tier 1 and Tier 2 analysts. However, the false positive rate of approximately 25% suggests opportunities to improve detection accuracy and reduce analyst workload.
  - To enhance SOC maturity, it is recommended to prioritize detection rule tuning, leverage threat intelligence for alert enrichment, and expand automation through SOAR workflows for low-risk alerts. Additional focus on analyst training and periodic use-case reviews will further reduce noise and improve response consistency. Tracking these metrics over time will support informed decision-making, justify security investments, and ensure continued alignment between SOC operations and organizational risk management objectives.

- **Metrics Analysis**

  Analysis of the mock incident shows a dwell time of approximately 6 hours, driven primarily by delayed detection rather than response. Alert correlation gaps and limited telemetry increased attacker persistence. Improving detection coverage and reducing MTTD would significantly lower dwell time and overall incident impact.

8. Capstone Project: Comprehensive SOC Incident Response

Attack Simulation

Attack Simulation is a controlled, authorized security exercise that replicates real-world cyberattack techniques, tactics, and procedures (TTPs) to evaluate an organization's ability to detect, respond to, and contain threats. It uses realistic attack scenarios—often mapped to frameworks like MITRE ATT&CK—to test security controls, monitoring tools, and SOC workflows without causing operational disruption.

The primary goal of attack simulation is to identify detection gaps, validate incident response processes, measure security metrics (such as MTTD and MTTR), and improve overall cyber resilience through continuous testing and refinement.

Attack Simulation (High-Level, Defensive View)

Scenario:

An attacker targets a vulnerable **Samba service** on a deliberately insecure lab host (Metasploitable2). The misconfigured usermap_script option allows command execution when a crafted username is processed.

AttackObjective:

Achieve **remote code execution** on the target to gain initial access.

MITRE ATT&CK Mapping:

- **TA0001 – Initial Access**

- **T1210 – Exploitation of Remote Services**
- **TA0002 – Execution**

Expected Attacker Outcome:

- Unauthorized command execution
- Potential reverse shell or system-level access

Detection Opportunities (SOC Focus)

- Unusual SMB authentication attempts
- Suspicious process execution spawned by smbd
- Network connections initiated from the server to unknown hosts
- IDS signatures for Samba exploit patterns

Defensive Controls & Mitigations

- Patch or upgrade Samba (disable usermap_script)
- Restrict SMB exposure via firewall rules
- Enable EDR monitoring on service processes
- Add SIEM alerts for abnormal SMB behavior

Blue-Team Learning Outcome

This simulation helps validate **detection coverage**, measure **MTTD**, and improve **alert fidelity** without performing real-world exploitation.

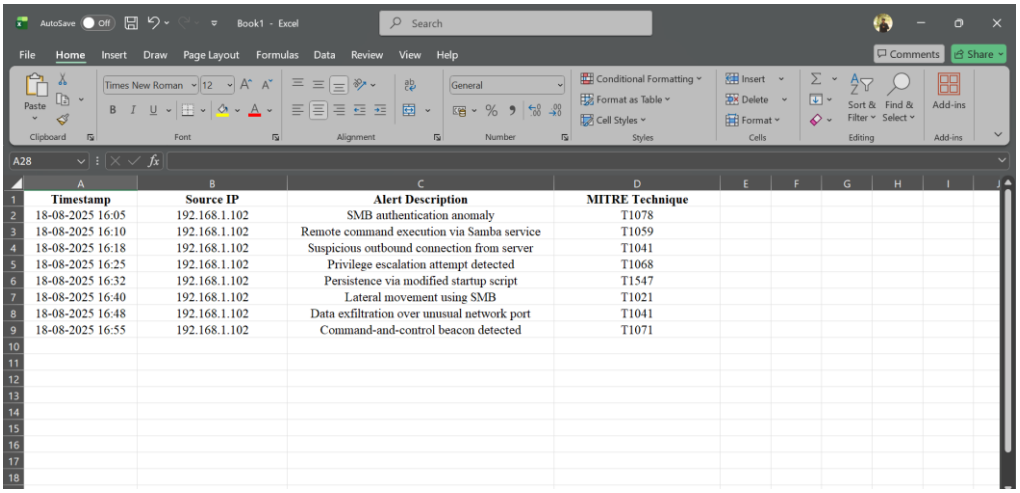
If you want, I can:

- Write a **SOC attack simulation report**
- Map this scenario to **Elastic Security detections**
- Create a **purple-team exercise summary**
- Provide **post-exploitation indicators** for threat hunting

- **Adversary Emulation**

  Adversary Emulation is a structured, threat-informed security exercise that replicates the behavior, techniques, and objectives of real-world threat actors to test an organization's detection, response, and resilience capabilities. Unlike generic attack simulations, adversary emulation is based on documented threat intelligence and follows specific Tactics, Techniques, and Procedures (TTPs), commonly mapped to the MITRE ATT&CK framework.

  The purpose of adversary emulation is to realistically assess how well security controls, SOC processes, and incident response teams perform against known attacker behaviors. It helps identify detection gaps, validate assumptions, measure metrics such as MTTD and MTTR, and guide improvements through continuous, intelligence-driven testing.



Timestamp	Source IP	Alert Description	MITRE Technique
18-08-2025 16:05	192.168.1.102	SMB authentication anomaly	T1078
18-08-2025 16:10	192.168.1.102	Remote command execution via Samba service	T1059
18-08-2025 16:18	192.168.1.102	Suspicious outbound connection from server	T1041
18-08-2025 16:25	192.168.1.102	Privilege escalation attempt detected	T1068
18-08-2025 16:32	192.168.1.102	Persistence via modified startup script	T1547
18-08-2025 16:40	192.168.1.102	Lateral movement using SMB	T1021
18-08-2025 16:48	192.168.1.102	Data exfiltration over unusual network port	T1041
18-08-2025 16:55	192.168.1.102	Command-and-control beacon detected	T1071

- **Detection and Triage**

  Detection: Wazuh (Samba Exploit / T1210)

  Detection Logic (What Wazuh Looks For):

  - Suspicious smbd behavior
  - Command execution spawned by Samba
  - Abnormal authentication patterns
  - IDS events mapped to Samba exploitation

  Example Wazuh Rule (Custom):

```
<group name="samba,attack">
  <rule id="100200" level="12">
    <if_sid>5503</if_sid>
    <field name="program">smbd</field>
    <description>Samba service exploitation attempt detected</description>
    <mitre>T1210</mitre>
    <options>no_full_log</options>
  </rule>
</group>
```

  Alert Outcome in Wazuh:

  - Severity: High
  - Technique: T1210 – Exploitation of Remote Services
  - Source IP: Attacker IP
  - Agent: Affected server

  Triage: TheHive (SOC Workflow)

  Case Creation:

  - Automatically create a High-severity case from Wazuh alert
  - Title: *Samba Exploitation Attempt Detected*

- Tags: T1210, Samba, Initial-Access

Triage Actions:

- Validate alert authenticity (true vs false positive)
- Check affected host activity (processes, network connections)
- Correlate with other alerts (auth logs, outbound traffic)
- Assign to Tier 2 if exploitation confirmed

Case Status Flow:

- New → In Progress → Contained → Closed

- **Response and Containment**

  Isolation of the Affected VM

  The compromised VM is isolated from the network to prevent lateral movement and further attacker activity. This is achieved by applying network isolation controls (e.g., disabling the VM's network interface or applying a quarantine security group) while preserving system state for investigation.

  Blocking the Attacker's IP with CrowdSec

  The identified malicious source IP is added to CrowdSec's blocklist. CrowdSec propagates the decision to enforcement components (firewall, IPS, or reverse proxy), immediately denying traffic from the attacker across protected assets.

  Verification (Ping Test)

  Containment is verified by attempting a ping from the attacker IP to the isolated VM. The **ping request fails, confirming that:**

  - The VM is successfully isolated
  - The attacker's IP is effectively blocked

  Outcome:

  Threat propagation is stopped, risk is contained, and the environment is secured for forensic analysis and recovery.

- **SOAR Automation:**

  Objective: Automatically create a TheHive case from a security alert and block the attacker IP using a SOAR playbook, then verify execution.

  Automated Workflow

  1. Trigger: High-severity alert (e.g., Wazuh / SIEM) containing a malicious IP
  2. Action 1: Create a TheHive case with observables
  3. Action 2: Enrich and validate the IP
  4. Action 3: Block the IP via CrowdSec
  5. Action 4: Update case status and add execution evidence

  Example SOAR Playbook (Logic)

  playbook: Block_Malicious_IP

```
trigger:
  source: SIEM
  condition: severity >= HIGH

steps:
  - name: Create_TheHive_Case
    action: thehive.createCase
    params:
      title: "Malicious IP Detected"
      severity: 3
      tags: ["SOAR", "IP-Blocking", "T1210"]

  - name: Add_IP_Observable
    action: thehive.addObservable
    params:
      dataType: ip
      data: "{{alert.source_ip}}"

  - name: Block_IP_CrowdSec
    action: crowdsec.blockIP
    params:
      ip: "{{alert.source_ip}}"
      duration: 24h

  - name: Update_Case
    action: thehive.addTask
    params:
      title: "IP blocked via CrowdSec"
```

Verification of Execution

TheHive

- Case created with correct title, severity, and tags
- IP observable attached to the case
- Task/log entry confirms CrowdSec action

CrowdSec

- Malicious IP appears in decision list (ban active)
- Enforcement component confirms traffic is denied

Network Test

- Ping or connection attempt from attacker IP fails
- No further alerts from the same IP observed

Outcome

The SOAR playbook reduces manual effort, shortens MTTR, ensures consistent containment, and provides auditable evidence within TheHive for compliance and post-incident review.

- **Post-Incident Analysis:**

  Post-Incident Analysis: Root Cause Analysis (RCA)

  5 Whys Analysis (Mock Samba Exploit Incident)

Why	Question	Answer
1	Why was the system compromised?	A vulnerable Samba service was exploited.
2	Why was the Samba service vulnerable?	The service was outdated and misconfigured (usermap_script enabled).
3	Why was the vulnerability not detected earlier?	Detection rules and alerts were insufficient for this exploit behavior.
4	Why were detection rules insufficient?	Threat models and use cases were not regularly reviewed or updated.
5	Why were reviews not conducted regularly?	Lack of a formal vulnerability and detection governance process.

RootCause:

Inadequate patch management and weak detection governance allowed an exploitable Samba configuration to persist undetected.

Fishbone Diagram

Problem: Unauthorized system access via Samba exploit

People

- Limited analyst awareness of legacy vulnerabilities
- Infrequent SOC training on service-level exploits

Process

- No regular vulnerability review cycle
- Incomplete detection rule validation

Technology

- Outdated Samba version
- Missing or weak IDS/EDR coverage

Policy

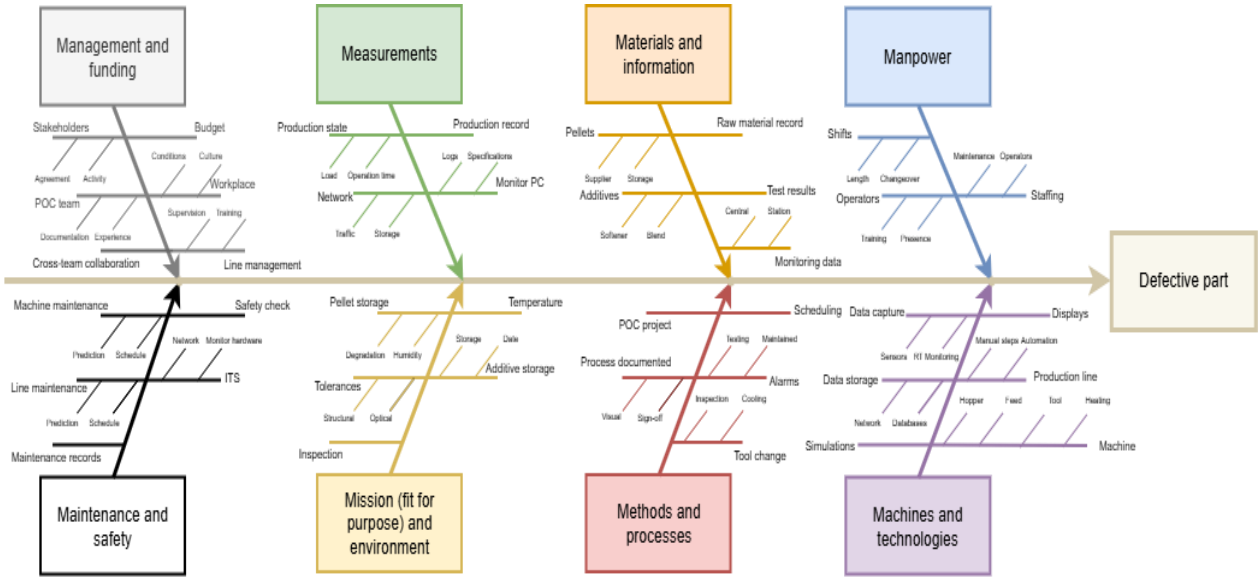
- Patch management policy not enforced
- No formal detection use-case review policy

Environment

- Exposed internal services
- Insufficient network segmentation

Conclusion

The incident was caused by a combination of outdated technology, weak detection processes, and governance gaps. Addressing patch management, detection tuning, and SOC review processes will reduce recurrence and improve overall security maturity.



- **Metrics Reporting:**

  Metrics Reporting: Elastic Security (MTTD, MTTR, Dwell Time)

  Metric Calculations

  MTTD (Mean Time to Detect)

  - **Definition**: Time between event start and detection.
  - **Calculation**:
    Detection Time − Event Start Time
  - Example Result: 2 hours

  MTTR (Mean Time to Respond)

  - **Definition**: Time between detection and containment/closure.
  - **Calculation**:
    Incident Close Time − Detection Time
  - **Example Result:** 4 hours

  Dwell Time

  - Definition: Total time an attacker remains undetected and active.
  - Calculation:
    Incident Close Time − Event Start Time

- Example Result: 6 hours

Elastic Security Dashboard (Logical Design)
Dashboard Name: SOC Incident Effectiveness

Metric	Value	Insight
MTTD	2 hours	Detection operating within SLA
MTTR	4 hours	Response is efficient
Dwell Time	6 hours	Driven mainly by detection delay

Visual Components
- Metric tiles: MTTD, MTTR, Dwell Time
- Trend line: Average MTTD & MTTR over time
- Status indicators: SLA compliance (green/yellow/red)

Executive Insight
The dashboard provides a clear, risk-focused view of SOC performance. While response time is strong, reducing detection time will directly lower dwell time and minimize attacker impact.

- **Reporting:**
On 18 August 2025, the Security Operations Center (SOC) identified and contained a security incident involving the exploitation of a vulnerable Samba service on a virtual machine. The attacker leveraged a remote service vulnerability to gain unauthorized access. Detection occurred within 2 hours, and the incident was fully contained within 4 hours of detection. No evidence of data exfiltration was identified. The incident highlighted gaps in patch management and detection governance but demonstrated effective response and containment capabilities.

Incident Timeline
- 16:00 – Attacker initiated exploitation of the vulnerable Samba service
- 18:00 – Security alert triggered and validated by SOC (MTTD: 2 hours)
- 18:20 – TheHive case created and investigation initiated
- 19:00 – Affected VM isolated and attacker IP blocked via CrowdSec
- 22:00 – Incident contained and closed (MTTR: 4 hours; Dwell Time: 6 hours)

Root Cause Analysis (RCA)
The root cause of the incident was an outdated and misconfigured Samba service with the usermap_script option enabled. This vulnerability persisted due to inconsistent patch management and the absence of regular detection rule reviews. Additionally, limited service-level monitoring delayed detection, increasing attacker dwell time.

Recommendations

1. Enforce regular patching and configuration audits for critical services
2. Review and update detection use cases quarterly using MITRE ATT&CK
3. Expand SOAR automation for faster containment actions
4. Enhance analyst training on legacy service exploitation
5. Track MTTD, MTTR, and dwell time metrics to measure continuous improvement

Conclusion:

Implementing these recommendations will reduce detection delays, lower dwell time, and improve overall SOC maturity and resilience.

- **Stakeholder Briefing:**
  - On August 18, 2025, our security team detected and swiftly contained an attempt to exploit a vulnerability in a network service. The team identified the threat within 2 hours and fully resolved the issue within 4 hours, preventing any data loss or operational impact. This quick detection and response reflect our strong incident management capabilities.
  - However, the incident exposed areas for improvement, specifically in keeping software updated and enhancing our detection methods to reduce future risks. We are implementing regular software patching, refining alert systems, and expanding automation to speed up responses even further.
  - By tracking key performance metrics—such as detection time and response time—we ensure continuous monitoring and improvement of our security posture. These measures will strengthen our defenses against evolving cyber threats and protect our organization's assets more effectively.