NAME : PATHI ASHOK
TEAM : SOC
EMAIL : ashokpathi697@gmail.com

1. Alert Priority Levels

1.1 Core Concepts

- **Priority Definitions**:
  - **Critical** → Maximum impact, urgent action needed.
    Example: Ransomware actively encrypting production data.
  - **High** → Severe risk but not yet catastrophic.
    Example: Unauthorized admin access on a server.
  - **Medium** → Moderate impact, may escalate if ignored.
    Example: Malware on a workstation, not yet spreading.
  - **Low** → Low risk or informational alert.
    Example: Failed login attempts on a test VM.
- **Assignment Criteria**:
  Prioritization should consider:
  - **Asset Criticality**: Is it a production server or a lab machine?
  - **Exploit Likelihood**: Is there a known CVE with a public exploit available?
  - **Business Impact**: Will it cause financial or reputational loss?

Example: *Log4Shell (CVE-2021-44228)* had a CVSS score of 9.8 → immediately Critical.

- **Scoring Systems**:
  - **CVSS (Common Vulnerability Scoring System):** Widely used risk quantification (Base, Temporal, Environmental metrics).
  - **SOC Tools Risk Scoring:** Splunk, QRadar, and Elastic SIEM assign "risk scores" to alerts for triage.

Key Objectives

- Learn to **quickly assess alerts** based on severity, impact, and urgency.
- Develop a **standardized method** to prioritize work in SOC operations.

How to Learn

1. Study CVSS metrics → FIRST CVSS Guide.
2. Review **NIST SP 800-61** for severity classification and workflows.
3. Analyze case studies: CISA's Log4Shell alerts show how CVSS maps to real-world Critical/High responses.

2. Incident Classification

Core Concepts

- **Incident Categories**:
  - **Malware** → Viruses, worms, ransomware.
  - **Phishing** → Credential harvesting via fake emails/sites.

  - **DDoS** → Service unavailability due to traffic flooding.
  - **Insider Threat** → Employee stealing or leaking data.
  - **Data Exfiltration** → Sensitive data leaving the organization.
- **Taxonomy & Frameworks**:
  - **MITRE ATT&CK**: Maps incidents to adversary techniques. Example: T1566 → *Phishing*.
  - **ENISA Incident Taxonomy**: Standardized classification used in EU.
  - **VERIS Framework**: Vocabulary for structured incident reporting.
- **Contextual Metadata**:
  Each incident should be enriched with:
  - **Affected Systems** (server, endpoint, cloud)
  - **Timestamps** (first seen, last seen)
  - **IOCs (Indicators of Compromise)** → Hashes, IPs, domains, URLs

Example: Phishing incident → Email from attacker, malicious attachment hash, compromised user account, and IP address of C2 server.

Key Objectives

- Develop skill to **categorize and label incidents** using frameworks.
- Learn to **enrich incidents with metadata** to support faster investigations.

How to Learn

1. Study **MITRE ATT&CK Navigator** (hands-on mapping).
2. Read **ENISA & VERIS taxonomies**.
3. Practice classifying real-world incidents (e.g., SANS Phishing case studies).

3. Basic Incident Response

Core Concepts

- **Incident Lifecycle (NIST SP 800-61)**:
  1. **Preparation** → Build playbooks, train SOC analysts, define escalation.
  2. **Identification** → Triage alerts, confirm incidents via logs & tools.
  3. **Containment** → Stop spread (isolate systems, block IPs).
  4. **Eradication** → Remove malware, close vulnerabilities.
  5. **Recovery** → Restore services, monitor for reinfection.
  6. **Lessons Learned** → Conduct post-incident review, update playbooks.
- **Procedures**:
  - **System Isolation** → Take compromised host offline.
  - **Evidence Preservation** → Memory dumps, disk images, file hashing.
  - **Communication Protocols** → Notify stakeholders, follow escalation chain.
  - **SOAR Tools (Security Orchestration, Automation, and Response)** → Automate repetitive steps. Example: Splunk Phantom, Palo Alto Cortex XSOAR.

Example: Ransomware outbreak →

- Detect unusual file encryption → isolate infected endpoints → preserve logs → remove malware → restore backups → update detection rules.

Key Objectives

- Be able to **respond systematically** to incidents.
- Gain confidence in **technical + communication skills** during response.

How to Learn

1. **NIST SP 800-61 Guide** → Must-read for IR lifecycle.
2. **SANS Incident Handler's Handbook** → Templates & best practices.
3. Use **Let's Defend** → Hands-on labs for simulated IR scenarios.
4. Explore SOAR use cases → Automating repetitive IR workflows.

Summary Table for Quick Reference

Topic	Core Skill	Frameworks/Tools	Learning Resource
Alert Priority Levels	Assess severity & urgency	CVSS, NIST, SIEM scoring	FIRST CVSS, NIST 800-61
Incident Classification	Categorize & enrich incidents	MITRE ATT&CK, ENISA, VERIS	ATT&CK Navigator, SANS case studies
Basic Incident Response	Lifecycle response actions	NIST IR, SOAR tools	NIST 800-61, SANS IR Handbook, Let's Defend

Practical Application

Step-by-step on Kali Linux

 # Update & install Docker

1. sudo apt update
2. sudo apt install -y docker.io docker-compose-plugin
3. sudo usermod -aG docker $USER
4. newgrp docker

1.Bring up Wazuh (all-in-one via Docker)

 # Get Wazuh Docker templates

git clone https://github.com/wazuh/wazuh-docker.git
cd wazuh-docker/single-node

 # Generate certs and start stack (manager + indexer + dashboard)

docker compose -f generate-indexer-certs.yml run --rm generator
docker compose up -d
(After a minute or two, open the **Wazuh Dashboard** in your browser (localhost URL shown by docker compose ps).
(Default user/password are shown in the repo README (change them after first login).

(Fastest way to simulate alerts)
1.Enter the manager container:

docker exec -it wazuh.manager bash

2.Add local rules to map your mock alerts and include MITRE tags:

```
cat >/var/ossec/etc/rules/local_rules.xml << 'EOF'
<group name="local,">
 <!-- Phishing -->
 <rule id="100100" level="10">
  <description>Phishing Email: Suspicious Link</description>
  <match>Phishing Email: Suspicious Link</match>
  <mitre>
   <id>T1566</id>
   <tactic>Initial Access</tactic>
   <technique>Phishing</technique>
  </mitre>
  <group>phishing,</group>
 </rule>

 <!-- Log4Shell (Critical) -->
 <rule id="100101" level="13">
  <description>Log4Shell Exploit Detected</description>
  <match>Log4Shell Exploit Detected</match>
  <mitre>
   <id>T1190</id>
   <tactic>Initial Access</tactic>
   <technique>Exploit Public-Facing Application</technique>
  </mitre>
  <group>vuln, exploit,</group>
 </rule>

 <!-- Port scan (Low) -->
 <rule id="100102" level="3">
  <description>Port Scan Observed</description>
  <match>Port Scan</match>
  <group>recon,</group>
 </rule>
</group>
EOF

# Restart Wazuh manager inside the container
/var/ossec/bin/wazuh-control restart
```

Wazuh rule **level** ≈ priority proxy
Critical: ≥12, High: 8–11, Medium: 5–7, Low: ≤4

2) Create a priority pie chart in Wazuh Dashboard

In **Discover**, confirm your alerts appear in index wazuh-alerts-*.

Create a **runtime/scripted field** to map rule.level → Priority:

- Go to **Stack Management → Index Patterns → wazuh-alerts-*** → **Add field** (runtime).
- Name: priority_bucket (type: keyword)
- Script (Painless):

```
if (doc.containsKey('rule.level') && !doc['rule.level'].empty) {
  int l = doc['rule.level'].value;
  if (l >= 12) emit('Critical');
  else if (l >= 8) emit('High');
  else if (l >= 5) emit('Medium');
  else emit('Low');
} else { emit('Unknown'); }
```

3.Visualize → Create → Pie

- Open **Wazuh Dashboard → Dashboards → Create dashboard**.
- **Create visualization → Pie**.
- **Data view**: wazuh-alerts-*.
- **Buckets**: Split Slices by rule.level or a derived priority field; size by count.
- Save to your dashboard (Wazuh docs show custom dashboards/visuals).

4. Create a TheHive incident ticket (UI or API)

1. **UI**: *New Case* → Title, Severity, TLP/PAP, Description, Tags, add Artifacts (IP/hash/filename). See TheHive docs for case creation. MITRE ATT&CK

2. **API**: POST /api/alert with JSON like the provided **thehive_alert_template.json**. (Official API docs list fields.)

5) Escalation email (≈100 words)

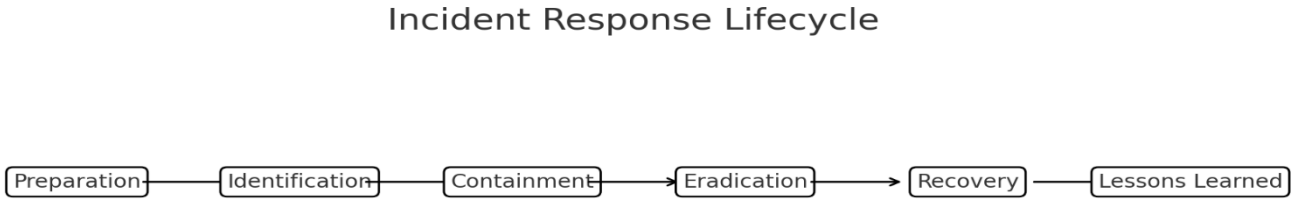
At 12:14 IST, 18-Aug-2025, Wazuh raised a **Critical** alert (rule.level=13) for suspected ransomware on **Server-X**. Indicators include process creation crypto_locker.exe, accelerated file renames in the user profile, and outbound traffic to **192.168.1.50:4444**. Asset is a **production file server**. Immediate actions taken: network isolation, privileged account lock, and volatile evidence preservation (RAM capture, session logs). Please triage encryption artifacts, verify

shadow copies, hunt lateral movement, and prepare recovery steps. Linked case: **TheHive TH-2025-001**. Observables and logs attached.

2.Response Documentation

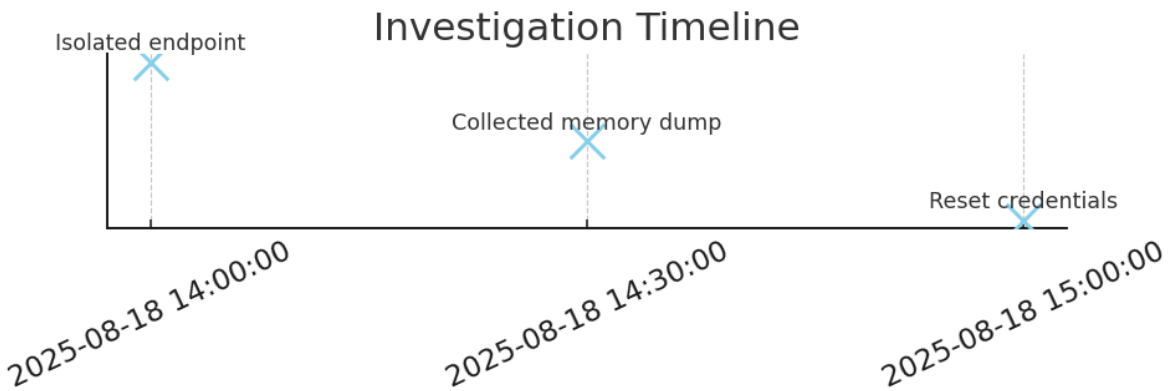
2.1 Incident Response Template (SANS-style)

- Executive Summary – what happened, when, impact, and initial response.
- Timeline – chronological order of detection and response actions.
- Impact Analysis – affected users, systems, data, or services.
- Remediation Steps – containment, eradication, and fixes applied.
- Lessons Learned – what to improve for the future.



2.2 Investigation Steps

Timestamp	Action
2025-08-18 14:00:00	**Isolated endpoint**
2025-08-18 14:30:00	**Collected memory dump**
2025-08-18 15:00:00	**Reset credentials**



2.3 Phishing Checklist

- Confirm email headers (SPF/DKIM/DMARC)
- Check link/file reputation (VirusTotal)
- Identify affected users
- Block sender/domain
- Reset credentials

Phishing Investigation Checklist

Task	Done
Confirm email headers	[ ]
Check link reputation (VirusTotal)	[ ]
Identify affected users	[ ]
Block sender/domain	[ ]
Reset credentials	[ ]

2.4 Post-Mortem (50-word summary)

Email filtering rules were insufficient to block targeted phishing attempts. User awareness training gaps contributed to delayed reporting. Stronger DMARC enforcement, updated phishing playbooks, and quarterly awareness campaigns are recommended. While containment was effective, response speed can be further improved through automation and predefined escalation paths.

3. Alert Triage Practice

3.1. Activities

- **Tools Used**:
  - **Wazuh** → Detects and displays alerts from endpoints and servers.
  - **VirusTotal (VT)** → Reputation check for IPs, domains, hashes.
  - **AlienVault OTX** → Community threat intel platform to cross-check IOCs.

3.2. Triage Simulation in Wazuh

Wazuh Steps (UI in OpenSearch Dashboards):

1. Go to Wazuh Dashboard → Security Events.
2. Filter by keyword:

rule.groups: authentication AND message:"Failed password"

OR by MITRE technique: rule.mitre.id: T1110*

3.2.3 Check frequency, affected host, and login attempts.

Document in triage log (Google Sheet or CSV):

Alert ID	Description	Source IP	Priority	Status
002	Brute-force SSH	192.168.1.100	Medium	Open

3.3. Threat Intelligence Validation

AlienVault OTX

1. Log in to OTX.
2. Enter the IP in search.
3. Check if it appears in pulses (collections of known IOCs).
4. Review related indicators (domains, hashes, campaigns).

Virus Total

1. Go to VT.
2. Paste the IP or file hash (if present).
3. Look at *Detection* → how many security vendors flagged it.
4. Check *Relations* (domains/URLs linked).

4. Analysis Outcome

- If IOC is in OTX pulses & flagged by VT → Confirmed malicious (raise severity, escalate).
- If IOC not seen anywhere & context is weak → Could be false positive (e.g., pentest IP, misconfigured scanner).
- If inconclusive → Leave Medium, continue monitoring & correlation.

5. 50-Word IOC Validation

The IP 192.168.1.100 triggered a brute-force SSH alert in Wazuh. Cross-check with AlienVault OTX and VirusTotal showed no reputation hits or associated campaigns. Activity appears consistent with generic scanning. Incident remains *Medium* severity, marked for monitoring, with firewall rules recommended to reduce repeated login attempts.

4. Evidence Preservation

Volatile data (memory, network connections, clipboard, process lists) can disappear quickly. Capture volatile artifacts **first**, then non-volatile data (disk), and always preserve integrity by hashing and documenting a chain of custody. NIST's guidance on integrating forensic techniques into incident response covers this approach and legal/admissibility concerns.

1.Order of volatility
Capture in roughly this order: CPU/registers → Memory (RAM) → Network connections/sockets → Running processes → Disk → Logs → Backups. Collect memory and network connections early to avoid losing crucial transient evidence. (See NIST SP 800-86 for rationale.)

2.Collecting network connections with Velociraptor
Why use Velociraptor? It's an endpoint collection/forensics tool with built-in artifacts for Windows and Linux; it includes Windows. Network. Netstat to capture open sockets.
GUI steps (recommended)

1. Open Velociraptor Web UI → **Collections** (or **Hunts** for multi-host).
2. Create **New Collection** → Search artifacts → choose **Windows.Network.Netstat**.
3. Configure target client(s) and start collection.
4. After the run completes, go to **Collected** tab → download results (CSV/JSON). Many users save long outputs to CSV for offline review.

3. Acquiring memory (Velociraptor & FTK Imager)
Velociraptor built-in artifact: Windows.Memory. Acquisition uses the WinPmem driver to capture a full memory image.
Note: memory images are large — increase collection timeout and bandwidth allowances. After collection, artifacts may be compressed; Velociraptor docs show using go-winpmem.exe to expand compressed images
Velociraptor memory capture (GUI)

1. Collections → New → select **Windows.Memory.Acquisition**.
2. Set a longer timeout (default may be too short for large RAM).
3. Run; once finished, download the artifact ZIP from the client's collected results. It typically contains a compressed memory image. Use go-winpmem or similar to expand if needed

FTK Imager (alternative GUI tool)

1. Open FTK Imager → File → Capture Memory.
2. Choose destination filename & options, then start capture. FTK writes a .mem or .dd style file. FTK Imager is commonly used in labs and validated for volatile memory capture.

4. Hashing & verification (integrity)
Linux/macOS : sha256sum memdump.raw
output: <SHA256> memdump.raw
Windows PowerShell: Get-FileHash -Path C:\evidence\memdump.raw -Algorithm SHA256

5) Chain-of-custody (CoC) — what to record

- Item description (file name, artifact type),
- Who collected it (name & role),
- Date/time (ISO format),
- Collection method & tool (Velociraptor vX.Y, FTK Imager vA.B),
- Source host & location (hostnames, IPs),
- Destination storage path (evidence locker),
- Hash (SHA256),
- Notes & signatures for transfers.

6) Transfer & storage best practices

- Use secure transfer (SCP over SSH, or a dedicated evidence transfer method). After transfer, recompute hash and compare.
- Keep original evidence read-only. For disks, use a hardware write blocker. For memory images, preserve the raw image file.
- Log every access/transfer (who, why, when). Keep copies in a secure evidence store with restricted access.

7) Example documentation & proof

- sample_netstat.csv — simulated netstat output (rows show repeated SSH connections from an attacker IP).
- memdump.raw — dummy memory file (512 KB) used to demonstrate hashing (do not treat as real evidence).
- chain_of_custody_example.csv — CoC CSV with a completed example row including a SHA256 hash.
- chain_of_custody_example.png — image of the CoC entry for quick inclusion in reports.
- evidence_flow.png — illustrated evidence-handling flow (Detect → Isolate → Collect → Acquire → Hash → Store → Document).
- evidence_preservation_readme.txt — short guide and the computed SHA256 for the dummy file.

5.Capstone Project: Full Alert-to-Response Cycle

Step 1: Attack Simulation (Red Team)
Target: Metasploitable2 (vulnerable Linux VM)
Exploit: VSFTPD v2.3.4 Backdoor (Metasploit module)

msfconsole
use exploit/unix/ftp/vsftpd_234_backdoor
set RHOSTS 192.168.1.105
set RPORT 21
run

Step 2: Detection with Wazuh (Blue Team)

- Wazuh collects logs from Metasploitable2 via an installed agent.
- An exploit attempt triggers Wazuh rules for suspicious FTP activity.

Timestamp	Source IP	Alert Description	MITRE Technique
2025-08-18 11:00:00	**192.168.1.100**	**VSFTPD exploit attempt**	**T1190 (Exploit Public-Facing Application)**

Step 3: Triage

- Source IP → Attacker machine 192.168.1.100
- Technique → MITRE ATT&CK T1190
- Severity → Critical (remote code execution risk)

Step 4: Response (Containment with CrowdSec)

CrowdSec firewall bouncer is used to block the attacker IP.

Command : cscli decisions add --ip 192.168.1.100 --duration 24h

ping test: ping 192.168.1.100

Step 5: Reporting (SANS Incident Template)

On 18th August 2025, a Metasploitable2 server was targeted with a known FTP backdoor exploit (VSFTPD v2.3.4) from IP 192.168.1.100. Wazuh detected the attempt, categorizing it as a Critical alert mapped to MITRE ATT&CK T1190. Immediate action was taken to block the attacker using CrowdSec.

Timeline

- 11:00 – Wazuh flagged exploit attempt.
- 11:05 – SOC analyst verified logs.
- 11:10 – CrowdSec rule applied to block source IP.
- 11:15 – Verified isolation via ping test.

Impact Analysis

No data exfiltration was detected. The attacker gained no persistence as the exploit was contained at detection.

Remediation Steps

- Blocked IP via CrowdSec.
- Recommended patching FTP service.
- Reviewed firewall logs for lateral movement.

Recommendations

- Disable unused FTP services.
- Regular vulnerability scans.
- Automated response playbooks to shorten detection-to-containment time.

Step 6: Stakeholder Briefing (100 Words)

On August 18, 2025, our monitoring system detected an attempted exploitation of a known FTP vulnerability on the Metasploitable2 server. The attack originated from IP 192.168.1.100. Security controls successfully detected and blocked the attempt using CrowdSec within minutes, preventing potential compromise. No business data or services were impacted. The vulnerability has been logged for patching, and monitoring continues to ensure no follow-up attacks occur. Overall, this incident demonstrates our SOC's capability to quickly identify and contain threats before they affect operations.