

AI-Driven Incident Response

Using AI as your co-pilot to detect, contain, eradicate, and recover from security incidents faster and more effectively.

What You Will Learn

- IR frameworks (NIST & SANS)
- AI-assisted triage and containment
- Evidence collection with AI documentation
- Post-incident analysis and runbooks

CTF Challenge

Respond to an active security incident — identify, contain, and eradicate the threat using AI-assisted IR techniques.

⚠ Educational Use Only

Incident response procedures are for authorized security operations and CTF competitions. In real incidents, follow your organization's established IR procedures and legal requirements.

Why AI in Incident Response?

Modern SOC's face alert volumes that overwhelm human analysts. AI transforms IR from a reactive scramble into a structured, accelerated workflow.

- **Speed** — AI can correlate thousands of log entries in seconds, reducing mean time to detect (MTTD)
- **Consistency** — AI follows playbooks without fatigue, ensuring no steps are skipped at 3 AM
- **Context synthesis** — LLMs summarize complex attack chains across multiple data sources simultaneously
- **Documentation** — AI generates real-time incident timelines, reducing post-incident paperwork by hours
- **Knowledge amplification** — Junior analysts gain senior-level reasoning through AI-assisted decision support

Notes: AI augments human decision-making — it does not replace the incident commander. Final containment and eradication decisions must have human approval.

NIST SP 800-61: The Foundation

NIST Special Publication 800-61 Rev. 2 defines the standard incident response lifecycle used across government and industry.

Phase	Purpose	AI Role
Preparation	Build IR capability before incidents occur	Generate playbooks, train scenarios
Detection & Analysis	Identify and validate incidents	Alert correlation, log analysis, severity scoring
Containment, Eradication & Recovery	Stop the threat, remove it, restore services	Recommend containment actions, verify eradication
Post-Incident Activity	Learn and improve	Root cause analysis, lessons-learned reports

Key NIST Principle

IR is a **cycle**, not a one-time event. Lessons from each incident feed back into preparation for the next.

SANS 6-Step IR Process

The SANS Institute breaks IR into six discrete phases — more granular than NIST, making it ideal for operational playbooks.

1. **Preparation** — Policies, tools, team training, communication plans
2. **Identification** — Detect anomalies, validate alerts, determine scope
3. **Containment** — Short-term and long-term isolation of affected systems
4. **Eradication** — Remove root cause: malware, backdoors, compromised credentials
5. **Recovery** — Restore systems, verify integrity, monitor for re-infection
6. **Lessons Learned** — Post-mortem, documentation, process improvements

NIST vs SANS

NIST combines Containment, Eradication, and Recovery into one phase. SANS separates them for clearer operational focus.

AI Advantage

AI can track which phase you are in and provide phase-specific recommendations, preventing analysts from jumping ahead.

Phase 1: Preparation

Effective incident response starts long before the first alert fires. AI accelerates preparation by generating and maintaining IR documentation.

AI-Assisted Preparation Tasks

- **Playbook generation** — Prompt AI with your infrastructure details to draft IR playbooks for common attack types
- **Asset inventory analysis** — AI parses CMDB exports to identify critical assets and dependencies
- **Tabletop exercise design** — AI generates realistic incident scenarios tailored to your environment
- **Contact list maintenance** — AI cross-references org charts and on-call schedules

```
# AI-assisted playbook generation prompt
prompt = """Generate an IR playbook for ransomware incidents targeting
our environment:
- 200 Windows endpoints on 10.0.0.0/16
- Active Directory domain: corp.example.com
- Critical assets: SQL Server (10.0.5.10), File Server (10.0.5.20)
```

- Backup system: Veeam with air-gapped copies

Include: detection signatures, containment steps, eradication checklist, recovery procedures, and communication templates."""

```
response = client.messages.create(  
    model="claude-sonnet-4-20250514",  
    max_tokens=4096,
```

Phase 2: Identification — AI-Powered Triage

The identification phase is where AI delivers the most immediate value: transforming a flood of alerts into actionable intelligence.

Alert Classification Pipeline

```
def ai_triage_alert(alert_data):  
    """Use AI to classify and prioritize a security alert."""  
    prompt = f"""Analyze this security alert and provide:  
1. Classification (True Positive / False Positive / Needs Investigation)  
2. Severity (Critical / High / Medium / Low)  
3. Attack category (MITRE ATT&CK technique if applicable)  
4. Recommended immediate actions  
5. Related IOCs to search for  
  
Alert Data:  
{json.dumps(alert_data, indent=2)}  
  
Context: Enterprise network, 500 employees, Windows/Linux mixed env.  
Normal business hours: 08:00-18:00 EST."""
```

Notes: Always validate AI triage decisions against your SIEM data. AI may lack context about known maintenance windows or authorized penetration tests.

Severity Assessment Matrix

AI can apply a consistent severity assessment framework to every alert, reducing the variability between analyst shift handoffs.

Factor	Critical	High	Medium	Low
Data at risk	PII / financial / classified	Internal sensitive	Internal general	Public data
Systems affected	Domain controller / auth	Production servers	Dev/staging	Individual workstation
Lateral movement	Confirmed spread	Indicators present	Possible	Isolated
Active exfiltration	Confirmed outbound	Suspicious DNS/HTTP	Anomalous traffic	None detected
Business impact	Revenue-generating down	Major function impaired	Minor disruption	No impact

AI Scoring Prompt Pattern

Feed the matrix to the AI as system context, then provide raw alert data. Ask the AI to score each factor and compute an overall severity with justification.

Real-Time Analysis: Incident Timeline

During an active incident, AI can continuously build and maintain a timeline from multiple log sources. Here is an example of an AI-generated incident timeline:

Time (UTC)	Source	Event	Severity
14:23:01	EDR	Suspicious PowerShell execution on WS-042	Medium
14:23:18	AD Logs	Service account svc_backup authenticates from WS-042	High
14:24:05	Firewall	WS-042 connects to 185.220.101.x:443 (known C2)	Critical
14:25:33	EDR	Mimikatz signature detected in memory on WS-042	Critical
14:26:10	AD Logs	svc_backup queries all domain admin group members	Critical
14:28:44	SIEM	Lateral movement: svc_backup authenticates to SRV-DB-01	Critical
14:30:02	DLP	Large data transfer initiated from SRV-DB-01 to external IP	Critical

AI Insight

Attack pattern identified: Initial access via compromised workstation → credential harvesting (Mimikatz) → privilege escalation (service account abuse) → lateral movement → data exfiltration. Matches MITRE ATT&CK chain: T1059.001 → T1003.001 → T1078.002 → T1021.002 → T1041.

AI as an IR Co-Pilot: Live Analysis

During an active incident, feed real-time data to AI for continuous analysis and recommendations.

```
class IRCoPilot:
    """AI-driven incident response co-pilot."""

    def __init__(self, client):
        self.client = client
        self.timeline = []
        self.context = []

    def analyze_event(self, event):
        """Analyze a new event in the context of the ongoing incident."""
        self.timeline.append(event)
        prompt = f"""You are an incident response co-pilot assisting a
SOC analyst during an active security incident.

INCIDENT TIMELINE SO FAR:
```

Phase 3: Containment Strategies

Containment is about stopping the bleeding. AI helps choose the right containment strategy based on the attack type, affected systems, and business impact.

Short-Term Containment

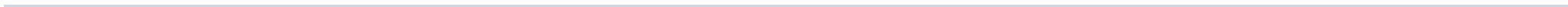
- Network isolation (VLAN change)
- Firewall block rules
- Account disable/lockout
- Process termination
- DNS sinkhole for C2 domains

Long-Term Containment

- Rebuild from clean image
- Network segmentation changes
- Credential rotation (bulk)
- Certificate revocation
- Firewall policy hardening

Critical Decision Point

AI can recommend containment actions, but **a human must authorize execution**. Isolating a production database server has business consequences that AI cannot fully evaluate.



AI-Assisted Containment Commands

AI generates specific, copy-paste-ready containment commands tailored to the incident context.

Network Isolation

```
# Isolate compromised host via firewall (iptables)
sudo iptables -I FORWARD -s 10.0.1.42 -j DROP
sudo iptables -I FORWARD -d 10.0.1.42 -j DROP
sudo iptables -I INPUT -s 10.0.1.42 -j DROP

# Allow only IR team access for forensics
sudo iptables -I INPUT -s 10.0.1.42 -d 10.99.0.0/24 -j ACCEPT

# Block C2 domain at DNS level
echo "address=/malicious-c2.example.com/127.0.0.1" >> /etc/dnsmasq.d/ir-blocks.conf
sudo systemctl restart dnsmasq
```

Account Lockout

```
# Disable compromised service account in Active Directory
Disable-ADAccount -Identity "svc_backup"
```

```
# Force password reset for potentially compromised accounts
Get-ADGroupMember "Domain Admins" | Set-ADUser -ChangePasswordAtLogon $true

# Revoke all active sessions for the compromised user
Revoke-AzureADUserAllRefreshToken -ObjectId (Get-AzureADUser `
    -SearchString "svc_backup").ObjectId
```

AI Containment Decision Engine

Structure AI recommendations as scored options so the incident commander can make informed decisions quickly.

```
def recommend_containment(incident_data):  
    prompt = f"""Based on this active incident, recommend containment  
strategies ranked by effectiveness and business impact.  
  
INCIDENT SUMMARY:  
- Type: {incident_data['type']}  
- Affected systems: {incident_data['systems']}  
- Current spread: {incident_data['spread_status']}  
- Business-critical systems at risk: {incident_data['critical_assets']}  
- Current business hours: {incident_data['is_business_hours']}  
  
For each option provide:  
| Option | Action | Effectiveness | Business Impact | Risk |  
  
Also provide the exact commands/steps to execute each option."""
```

Phase 4: Eradication

Eradication removes the root cause of the incident. AI helps ensure nothing is missed by systematically checking for persistence mechanisms.

AI-Driven Eradication Checklist

```
eradication_prompt = """Given the following incident details, generate  
a comprehensive eradication checklist:
```

```
Attack type: Credential theft + lateral movement
```

```
Compromised accounts: svc_backup, jsmith
```

```
Malware found: Mimikatz (memory-resident), Cobalt Strike beacon
```

```
Affected systems: WS-042, SRV-DB-01
```

```
C2 infrastructure: 185.220.101.0/24
```

```
For each item, provide:
```

- Specific check to perform
- Command or tool to use
- How to verify successful eradication
- Common places attackers hide persistence"""

Common Persistence Locations to Check

- **Scheduled Tasks** — `schtasks /query /fo LIST /v`
 - **Registry Run keys** — `reg query HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run`
 - **WMI subscriptions** — `Get-WMIObject -Namespace root\subscription -Class __EventFilter`
 - **Services** — `Get-Service | Where-Object {$_.StartType -eq 'Automatic'}`
 - **SSH authorized_keys** — `find / -name authorized_keys -type f 2>/dev/null`
-

Phase 5: Recovery

Recovery restores affected systems to normal operation. AI assists with verification procedures and monitoring for re-infection.

AI-Assisted Recovery Workflow

1. **System restoration** — Rebuild from known-good images or restore from verified backups
2. **Credential rotation** — AI generates a comprehensive list of credentials to reset based on blast radius analysis
3. **Integrity verification** — Compare file hashes against known-good baselines
4. **Monitoring enhancement** — AI generates custom detection rules for re-infection indicators
5. **Staged reconnection** — Bring systems back online incrementally with enhanced monitoring

```
# AI-generated YARA rule for post-recovery monitoring
recovery_prompt = """Based on the IOCs from this incident, generate:
1. YARA rules to detect the malware variants observed
2. Sigma rules for the behavioral patterns identified
3. Network signatures (Snort/Suricata) for C2 communication
4. A 30-day enhanced monitoring plan
```

IOCs:

- File hash: a1b2c3d4e5f6... (Cobalt Strike beacon)
 - C2 domains: update.legit-software[.]com, cdn.trusted-host[.]net
 - User-Agent: Mozilla/5.0 (compatible; MSIE 10.0; Windows NT)
 - Named pipe: \\.\pipe\msse-1234-server""
-

Evidence Collection & Chain of Custody

Proper evidence handling can make or break a legal case. AI helps document everything meticulously while analysts focus on technical collection.

AI-Documented Evidence Log

Evidence ID	Type	Source	Collected By	Hash (SHA-256)	Time (UTC)
EV-001	Memory dump	WS-042	J. Analyst	a3f8c2...e91b	15:02:33
EV-002	Disk image	WS-042	J. Analyst	7b2d41...f03a	15:45:10
EV-003	Network PCAP	FW-Core	K. Engineer	d9e1a7...c44f	15:12:08
EV-004	AD event logs	DC-01	K. Engineer	1f5e83...b27d	15:20:55
EV-005	Malware sample	WS-042	J. Analyst	c2a9f1...88e3	16:01:22

AI Chain of Custody Documentation

AI automatically logs every evidence interaction: who accessed it, when, what tools were used, and hash verification at each step. This creates a defensible audit trail.

Evidence Collection Commands

AI provides the exact forensic collection commands and automatically documents each step with timestamps and hashes.

Volatile Evidence (Collect First!)

```
# Memory acquisition (Linux)
sudo dd if=/dev/mem of=/evidence/mem_dump_$(hostname)_$(date +%s).raw
sha256sum /evidence/mem_dump_*.raw >> /evidence/hashes.txt

# Network connections snapshot
ss -tunapeo > /evidence/netstat_$(date +%s).txt
ip route show > /evidence/routes_$(date +%s).txt

# Running processes with full details
ps auxef > /evidence/processes_$(date +%s).txt
ls -la /proc/*/exe 2>/dev/null > /evidence/proc_links_$(date +%s).txt
```

Disk Evidence

```
# Create forensic disk image with hash verification
sudo dc3dd if=/dev/sda of=/evidence/disk_image.dd \
```

```
hash=sha256 log=/evidence/dc3dd_log.txt  
  
# Verify image integrity  
sha256sum /evidence/disk_image.dd  
# Compare against dc3dd output hash
```

Notes: Order of volatility matters: memory first, then network state, then running processes, then disk. AI should enforce this ordering in its recommendations.

Communication Templates: Stakeholder Notifications

AI drafts communication at each stage of the incident, ensuring consistent, professional messaging tailored to each audience.

Initial Notification (Technical Team)

Subject: [IR-2024-042] Active Security Incident - SEVERITY: CRITICAL

Status: Active - Containment in progress

Incident Commander: Sarah Chen

Bridge Line: +1-555-0142 / PIN: 8834

Summary: At 14:23 UTC, EDR detected credential theft activity on WS-042. Attacker has obtained svc_backup credentials and moved laterally to SRV-DB-01. Data exfiltration to external IP confirmed.

Immediate Actions Taken: C2 IP blocked at perimeter, svc_backup account disabled, WS-042 network-isolated.

Required: All domain admin accounts — verify your recent logon activity. Report any anomalies to IR bridge immediately.

Communication Templates: Executive & Legal

Executive Briefing (AI-Drafted)

Subject: Security Incident Briefing - Action Required

A security incident involving unauthorized access to an internal database server was detected today at 2:23 PM. The security team has contained the threat and is working to determine the full scope of impact. Customer-facing services remain operational.

Business impact: Internal database server temporarily offline for forensic analysis. Estimated restoration: 4-6 hours.

Customer impact: Under investigation. Preliminary analysis suggests customer PII may have been accessed.

Next update: 6:00 PM today or sooner if status changes.

Legal / Compliance Notification

Regulatory consideration: If PII exposure is confirmed, GDPR Article 33 requires notification to supervisory authority within **72 hours** of becoming aware. AI should flag applicable notification deadlines based on data types and jurisdictions involved.

AI-Assisted Communication Generator

```
def generate_ir_communication(incident, audience, phase):  
    """Generate audience-appropriate incident communications."""  
    prompt = f"""Draft an incident response communication for:  
  
Audience: {audience} # "technical", "executive", "legal", "customers"  
IR Phase: {phase}      # "detection", "containment", "eradication", "recovery"  
  
Incident Details:  
- ID: {incident['id']}  
- Type: {incident['type']}  
- Severity: {incident['severity']}  
- Affected systems: {incident['systems']}  
- Data at risk: {incident['data_types']}  
- Current status: {incident['status']}  
- Business impact: {incident['business impact']}
```

Phase 6: Lessons Learned

The most frequently skipped IR phase is also the most valuable. AI ensures thorough post-incident analysis by structuring the review process.

AI-Driven Root Cause Analysis

```
rca_prompt = ""Perform a root cause analysis for this security incident.  
Use the "5 Whys" technique and provide a fishbone (Ishikawa) diagram  
structure.
```

INCIDENT TIMELINE:

```
{full_timeline}
```

ATTACK CHAIN:

1. Phishing email delivered to jsmith@corp.example.com
 2. User clicked link, downloaded malicious macro document
 3. Macro executed PowerShell downloader
 4. Cobalt Strike beacon established C2 channel
 5. Mimikatz extracted svc_backup credentials from memory
 6. Lateral movement to SRV-DB-01 via SMB
 7. Database export and exfiltration via HTTPS to external IP
-

Root Cause Analysis: 5 Whys Example

AI applies structured analytical frameworks to produce actionable findings from incident data.

Level	Question	Answer
Why 1	Why was data exfiltrated?	Attacker had database access via svc_backup credentials
Why 2	Why did the attacker have those credentials?	Mimikatz extracted them from WS-042 memory
Why 3	Why were privileged credentials cached on a workstation?	svc_backup had interactive logon rights and was used for manual backup tasks
Why 4	Why was a service account used interactively?	No PAM solution; team used shared credentials for convenience
Why 5	Why was there no PAM solution?	PAM project was deprioritized in Q2 budget cycle

Root Cause

Technical: Privileged service account with interactive logon rights cached on standard workstation. **Process:** Lack of privileged access management controls. **Recommendation:** Deploy PAM, enforce credential tiering, remove interactive logon for service accounts.

Lessons Learned Documentation

AI generates comprehensive post-incident reports that capture institutional knowledge and drive security improvements.

```
lessons_prompt = """Generate a Lessons Learned report for incident  
IR-2024-042 using this structure:
```

- ```
1. EXECUTIVE SUMMARY (3-4 sentences)
2. INCIDENT METRICS
 - Time to detect (TTD)
 - Time to contain (TTC)
 - Time to eradicate (TTE)
 - Time to recover (TTR)
 - Total incident duration
3. WHAT WENT WELL
4. WHAT NEEDS IMPROVEMENT
5. ACTION ITEMS (owner, priority, deadline)
6. DETECTION GAP ANALYSIS
7. UPDATED RISK ASSESSMENT
```

## Key Metrics from Our Example Incident

TTD: 7 minutes | TTC: 12 minutes | TTE: 4.5 hours | TTR: 18 hours | Total: 23 hours

Industry average for similar incidents without AI: TTD: 197 days. AI-assisted IR can compress timelines by orders of magnitude.

---

# Building IR Runbooks with AI

IR runbooks are step-by-step procedures for specific incident types. AI can generate, maintain, and improve runbooks based on lessons learned from actual incidents.

## Runbook Structure

runbook:

name: "Ransomware Response"

version: 2.3

last\_updated: "2024-11-15"

author: "IR Team + AI Assistant"

severity\_default: "Critical"

triggers:

- "EDR alert: ransomware behavior detected"
- "Multiple file extension changes in short period"
- "Ransom note file detected (README.txt, DECRYPT\_FILES.html)"

immediate\_actions:

- step: "Isolate affected system from network"  
command: "netsh advfirewall set allprofiles state on"
-

# AI Runbook Generator

```
def generate_runbook(incident_type, environment):
 """Generate a tailored IR runbook for a specific incident type."""
 prompt = f"""Create a detailed IR runbook for: {incident_type}
```

Environment context:

```
{json.dumps(environment, indent=2)}
```

The runbook must include:

1. TRIGGERS - What alerts/events activate this runbook
2. SEVERITY CLASSIFICATION - How to determine initial severity
3. IMMEDIATE ACTIONS (first 15 minutes)
4. INVESTIGATION STEPS (with specific commands for our OS mix)
5. CONTAINMENT OPTIONS (ranked by impact)
6. ERADICATION CHECKLIST
7. RECOVERY PROCEDURES

**Notes:** AI-generated runbooks should be reviewed by senior IR staff and tested in tabletop exercises before being approved for production use.

# Runbook Library: Common Incident Types

Build a library of AI-generated runbooks covering your most common and most critical incident types.

| Incident Type               | Typical Severity | Key Containment Action             | AI Value-Add                                 |
|-----------------------------|------------------|------------------------------------|----------------------------------------------|
| Ransomware                  | Critical         | Network isolation                  | Variant ID, decryptor search                 |
| Phishing (credential theft) | High             | Password reset, session revoke     | Analyze phishing kit, find other victims     |
| Insider threat              | High             | Account suspension                 | Behavioral analysis, data access audit       |
| DDoS attack                 | High             | Traffic filtering, CDN scaling     | Traffic pattern analysis, source attribution |
| Supply chain compromise     | Critical         | Vendor isolation, dependency audit | Impact blast radius analysis                 |

| Incident Type     | Typical Severity | Key Containment Action | AI Value-Add                                |
|-------------------|------------------|------------------------|---------------------------------------------|
| Data exfiltration | Critical         | Egress blocking        | Data classification, scope assessment       |
| Cryptomining      | Medium           | Process termination    | Identify entry point, check for persistence |

# Integrating AI into Your IR Workflow

Practical architecture for embedding AI into your existing incident response tools and processes.

## Integration Points

- **SIEM → AI** — Forward high-fidelity alerts for AI triage and enrichment
- **SOAR → AI** — AI generates containment playbook steps, SOAR orchestrates execution
- **Ticketing → AI** — AI updates incident tickets with analysis, timeline entries, and status changes
- **Chat/Bridge → AI** — AI bot on the IR bridge provides real-time analysis and answers questions
- **Evidence Platform → AI** — AI maintains chain of custody logs and generates forensic reports

### Architecture Principle

AI should be **in the loop** (providing recommendations) not **on the loop** (making autonomous decisions). All containment and eradication actions require human approval before execution.

**Notes:** Start small: integrate AI triage first, measure improvement in MTTD/MTTC, then expand to other IR phases.

---

# IR Playbook Snippet: Phishing Response

A condensed example of an AI-generated IR playbook for a phishing incident with credential compromise.

```
playbook: phishing_credential_compromise
phase: identification
steps:
 - action: "Extract IOCs from reported phishing email"
 ai_prompt: |
 Analyze this email header and body. Extract:
 - Sender IP and domain (check SPF/DKIM/DMARC)
 - All URLs (defang them)
 - Attachment hashes
 - Social engineering technique used
 output: ioc_list

 - action: "Search for other recipients"
 command: |
 # Exchange Online
```

# Handling AI Limitations in IR

AI is a powerful tool but has critical limitations that must be understood during incident response.

## AI Limitations

- **Hallucination risk** — AI may generate plausible but incorrect IOCs or commands
- **Context window limits** — Large log volumes may exceed token limits
- **No real-time access** — AI cannot query your SIEM directly (without tool use)
- **Stale threat intel** — Training data has a knowledge cutoff
- **Confidentiality** — Sending sensitive incident data to cloud AI has privacy implications

## Mitigations

- **Verify all commands** before execution
- **Chunk large logs** and summarize iteratively
- **Use tool/function calling** to give AI safe access to systems
- **Supplement with live threat feeds**
- **Use local/private AI models** for sensitive incidents

**Notes:** For classified or highly sensitive incidents, consider running local LLMs (Llama, Mistral) that keep all data on-premises. The trade-off is reduced capability vs. full data sovereignty.

---

# CTF Challenge: Active Incident Response

You are the on-call SOC analyst. An active security incident has been detected. Use AI-assisted IR techniques to respond.

## Scenario

At 02:17 UTC, your SIEM fires multiple alerts. A workstation (WS-117, user: mwilliams) is exhibiting the following behaviors:

- PowerShell executing base64-encoded commands
- DNS queries to `x8f2.update-service[.]xyz` every 60 seconds (beaconing)
- SMB connections to three file servers: FS-01, FS-02, FS-03
- Windows Event Log cleared on WS-117
- New scheduled task created: "SystemHealthCheck" running every 5 minutes

- Large ZIP file (2.3 GB) copied to WS-117 from FS-02
-

# CTF Challenge: Your Mission

## Tasks

1. **Identification** — Use AI to analyze the alert data. Classify the incident type, determine severity, and map to MITRE ATT&CK techniques
2. **Timeline** — Construct a detailed incident timeline from the provided log data using AI assistance
3. **Containment** — Propose and justify at least 3 containment actions with exact commands. Rank them by priority
4. **Evidence** — List all evidence to collect, in order of volatility. Provide collection commands and chain-of-custody entries
5. **Communication** — Draft notifications for: (a) the IR team, (b) the CISO, and (c) the affected user's manager
6. **Eradication Plan** — Detail how you would eradicate the threat, including persistence mechanism checks
7. **Lessons Learned** — Write a preliminary post-incident report with root cause hypothesis and 3 improvement recommendations

## Scoring Criteria

**Completeness** (all IR phases addressed) | **Technical accuracy** (valid commands and techniques) | **AI utilization** (effective prompting) | **Communication quality** (audience-appropriate messaging) | **Documentation** (evidence handling and chain of custody)

---

# CTF Hint: MITRE ATT&CK Mapping

Map each observed behavior to MITRE ATT&CK techniques to understand the full attack chain.

| Observed Behavior                  | MITRE Technique | Tactic            |
|------------------------------------|-----------------|-------------------|
| Base64-encoded PowerShell          | T1059.001       | Execution         |
| DNS beaconing to suspicious domain | T1071.004       | Command & Control |
| SMB connections to file servers    | T1021.002       | Lateral Movement  |
| Event log cleared                  | T1070.001       | Defense Evasion   |
| Scheduled task persistence         | T1053.005       | Persistence       |
| Large file staging                 | T1074.001       | Collection        |

## Attack Stage Assessment

The attacker is in the **Collection / Exfiltration preparation** stage. The 2.3 GB ZIP file indicates data staging before exfiltration. **Containment is urgent** — you may still prevent data leaving the network.

---

# Key Takeaways

- **IR frameworks provide structure** — NIST SP 800-61 and SANS 6-step processes ensure no phase is skipped, even under pressure
- **AI excels at triage speed** — Automated alert classification and severity scoring reduce MTTD from hours to minutes
- **Containment requires human judgment** — AI recommends actions; humans authorize them based on business context
- **Evidence integrity is non-negotiable** — AI documentation ensures chain of custody is maintained even during chaotic incidents
- **Communication is an IR skill** — AI drafts audience-appropriate notifications so analysts can focus on technical response
- **Runbooks capture knowledge** — AI-generated runbooks encode lessons learned into repeatable, testable procedures
- **Post-incident analysis drives improvement** — AI-powered root cause analysis and lessons learned create a continuous improvement cycle

## Next Lesson

Lesson 09 will cover **AI-Powered Threat Hunting** — proactively searching for threats that evade automated detection using AI-driven hypothesis generation and investigation.

---