

Lab Guide

Four exercises across four log sources: web access logs, Linux syslog, Windows security events, and cloud audit trails — learn to triage at SOC speed with AI assistance.

Authorized Use Only

All log samples are **synthetic**. Never paste real production logs with customer data into a public AI — scrub PII and use redaction or a local LLM for real incidents.

Prerequisites

- Completed the [Theory](#) reading
- Access to an AI assistant (Claude, ChatGPT, or local LLM)
- Downloaded all four files into a folder named `log-lab/`
- Familiarity with: common log formats, MITRE ATT&CK basics, Sigma rule syntax

Lab Scenario

You are on SOC tier-2 at AcmeCorp. Overnight, four log feeds dropped into the case queue: web server access logs, a mixed Linux syslog stream, Windows security events from a domain controller, and a cloud audit trail. Your job: identify what happened, rank severity, and produce detection rules that would have caught it earlier.

Download Lab Files

Save all 4 files into a folder named `log-lab/` before starting the exercises.

↓ [cloud-audit.txt](#) ~15 KB

↓ [syslog-mixed.txt](#) ~10 KB

↓ [web-access.log.txt](#) ~12 KB

↓ [windows-security.txt](#) ~11 KB

Exercise 1 — Web Access Log Triage

Open [web-access.log.txt](#). Web logs are noisy — the attack hides in the volume.

Task 1.1 — Find the attacker in the noise

You are a SOC analyst reviewing Apache/nginx access logs. Analyze:

1. Top source IPs by request count, and flag any that look like scans (high rate, 404-heavy, user-agent anomalies).
2. Any path-traversal attempts (../, %2e%2e, encoded variants).
3. SQL injection indicators (UNION SELECT, OR 1=1, sleep()).
4. Command injection (;, |, backticks, \$()).
5. File upload abuse (.php, .jsp, .aspx uploads to wrong paths).
6. Credential stuffing (many POSTs to /login from one IP, or one user from many IPs).
7. Successful responses (2xx) after a long 4xx run — this is the attacker finding the valid path.
8. Any user-agent that is obviously a tool (sqlmap, nikto, gobuster).

For each finding:

- Source IP, timestamp, URL/method, status code
- MITRE ATT&CK initial-access or discovery technique
- Severity (low / medium / high / critical)
- Recommended immediate action (block / monitor / alert)

Then: write ONE Sigma rule that would catch the primary attack.

[paste web-access.log.txt]

✓ Self-check: You should identify at least one attacker IP with a concrete MITRE technique and have a working Sigma rule draft.

Exercise 2 — Linux Syslog Analysis

Open `syslog-mixed.txt`. Linux syslog mixes SSH, sudo, cron, and kernel messages — the signal is sparse.

Task 2.1 — Reconstruct what happened on the box

Below is a mixed Linux syslog stream. As a defender, identify:

1. SSH authentication failures (brute force? password spray?).
2. Successful logins that followed repeated failures (the compromised account).
3. sudo / su escalations — was the account authorized?
4. New cron jobs, systemd units, or scheduled tasks (persistence).
5. Unexpected kernel / auditd events (module loads, capability changes, apparmor denials).
6. Commands that look like post-exploitation (wget/curl to pastebin, base64-encoded shells, nc listeners).
7. User creation / group membership changes.

Output a chronological incident narrative + a list of the 3 most dangerous events with MITRE technique IDs.

[paste syslog-mixed.txt]

✓ Self-check: Your narrative should read like a story — "at T+0 attacker tried brute force, at T+5 got in, at T+10 escalated, at T+15 persisted."

Exercise 3 — Windows Security Events

Open `windows-security.txt`. Windows event IDs are the alphabet of Windows defense.

Task 3.1 — Decode the Event IDs

Below are Windows security events (4624, 4625, 4672, 4688, 4720, 4728, 5140, etc.). For each event:

1. Translate the Event ID to plain English.
2. Note the account, logon type, source workstation/IP.
3. Flag suspicious logon types (type 3 network, type 10 RDP, type 9 runas with explicit creds).
4. Identify:
 - Pass-the-hash / pass-the-ticket indicators
 - Kerberoasting (4769 with RC4_HMAC)
 - Golden/silver ticket signs (4624 with mismatched domain)
 - Lateral movement (admin logon to many hosts from one)
 - Privilege escalation (4672 special privileges assigned)
5. Map to MITRE ATT&CK and rank severity.

Output as a detection report with a "Top 3 Findings" executive summary.

[paste windows-security.txt]

✓ **Self-check: You should catch at least one lateral movement or privesc pattern and map it to a MITRE technique.**

Exercise 4 — Cloud Audit Log Review

Open `cloud-audit.txt`. AWS CloudTrail / Azure activity / GCP audit logs record every API call in your cloud.

Task 4.1 — Spot the cloud compromise

Below is a cloud audit log. As a cloud defender, analyze:

1. IAM anomalies:
 - New access keys created on service accounts
 - Policy modifications granting broader permissions
 - AssumeRole from unusual principals
 - MFA disabled / deleted

2. Data plane anomalies:

- S3 bucket ACLs opened to public
- Large-scale GetObject from a single principal
- Snapshot / AMI shared with external account

3. Compute anomalies:

- RunInstances in unusual regions
- Security group modifications opening 0.0.0.0/0
- Instance metadata service accessed from inside

4. Source-IP analysis: known corp ranges vs. Tor / VPN / foreign ASN.

For each finding:

- Principal, action, resource, source IP
- MITRE ATT&CK cloud technique
- Containment step (revoke key? disable user? close bucket?)

Finally: identify the root-cause event (where the attacker first got in).

[paste cloud-audit.txt]

✓ **Self-check: You should trace the attack from first suspicious API call to current state in a chain.**

Final Exercise — Cross-Log Correlation

Task 5.1 — Unify the four streams

Here are four analyses of four log sources from the same AcmeCorp environment over the same 24-hour window.

Correlate:

1. Which source shows the earliest sign of compromise?
2. Does the same attacker IP / account appear in multiple logs?
3. Build a single timeline across all four sources.
4. Which source had the weakest detection (most missed)?
5. If you could deploy ONE new detection rule tonight, what would it be?

Output as a 1-page incident summary + 3 tactical SOC improvements.

✓ Self-check: A good correlation will connect at least 2 of the 4 sources by a shared indicator (IP, account, time window).

What's Next

Lesson 02 moves from endpoint logs to the **network layer** — NIDS alerts, pcaps, and traffic baselines.