Linux Endpoint Investigation

FOR402

Respond decisively when Linux systems are under attack. Learn a proven forensic methodology to investigate, contain, and eradicate threats in Linux, cloud, and container environments.

Modules	18
Lessons	150
Phases	5
Free preview	Course Orientation (11 lessons)

The method

Linux investigation is done against a system that logs less than Windows and tells you more if you know where to look. This course runs the same loop for every compromise it works.

1. Establish what this distribution actually records. Log locations, retention and audit configuration vary by distribution and by whoever built the image. Assume nothing that has not been read.

2. Take the volatile evidence first. Processes, connections, open files and loaded modules. On a server nobody wants to reboot, this is the material with an expiry on it.

3. Read the filesystem for what the attacker had to touch. Timestamps, permissions, and the directories a technique cannot avoid. This is where the intrusion leaves marks it did not choose to leave.

4. Follow the account, not the process. Escalation and lateral movement are identity events on Linux as much as anywhere else, and the trail runs through sudo, SSH and the key files.

5. Say what the evidence supports and where it stops. A finding with a gap named is stronger than a finding with the gap hidden, and Linux estates have gaps.

What this course assumes

No minimum experience and no prerequisite course. The filesystem layout, the log stack and the process model are explained where they first matter, and the course does not assume you administer Linux.

What makes it go faster: a Linux machine you can run commands on, and comfort at a shell. Neither is required, and every command in the course is shown with its output.

What this course does not cover: Windows and macOS forensics, network investigation, and Linux system administration. Those are their own courses, and this one stays on the endpoint.

Full syllabus

Phase 1: Foundations

Course Orientation FREE PREVIEW

What Linux Endpoint Investigation teaches: a proven forensic method to investigate, contain, and eradicate threats across Linux servers, cloud VMs, and containers.

1. LX0.1 Why Linux IR Breaks Windows-Trained Analysts
2. LX0.2 How a Linux Compromise Actually Unfolds
3. LX0.3 The Linux Evidence Landscape and Order of Volatility
4. LX0.4 Everything Is a File: The Forensic Consequence
5. LX0.5 Reading the Authentication and System Logs
6. LX0.6 Where Attackers Hide: A First Look at Persistence
7. LX0.7 The Linux IR Lifecycle
8. LX0.8 Measuring What Matters in Linux IR
9. LX0.9 The Environment This Course References
10. LX0.10 Course Toolkit and Lab Setup
11. LX0.11 Interactive: Triage a Compromised Host

Linux Forensic Foundations

The architectural model a Linux investigator must hold before any applied technique makes sense: the filesystem and inode model, the four timestamps, the user and permission model, the process model, how the kernel exposes the running system, the distinction between memory and disk, and the...

1. LX1.1 The Linux Filesystem Model
2. LX1.2 Inodes and the Four Timestamps
3. LX1.3 Users, Groups, and the Permission Model
4. LX1.4 The Process Model
5. LX1.5 How the Kernel Exposes the Running System
6. LX1.6 Memory and the Running System
7. LX1.7 Live Host versus Disk Image
8. LX1.8 The Boot and Service Model
9. LX1.9 Interactive: Mapping the Foundations to an Investigation
10. Module Summary
11. Check My Knowledge

Phase 2: Evidence and Analysis

Linux Filesystem Forensics

Applied filesystem investigation on the two filesystems that dominate Linux estates: ext4 and xfs.

1. LX2.1 The Filesystem on Disk: ext4 and xfs for Investigators
2. LX2.2 Timestamp Forensics at Depth: Beneath stat and Into the Inode
3. LX2.3 Deleted File Recovery: What Comes Back, What Does Not
4. LX2.4 Timeline Generation: From Bodyfile to Super-Timeline
5. LX2.5 Attacker Staging Areas: Where Intruders Work
6. LX2.6 SUID, SGID, and Capability Indicators on Disk
7. LX2.7 User Artifact Forensics: History, Keys, and the Home Directory
8. LX2.8 Anti-Forensics Detection: Reading the Cleanup
9. LX2.9 The Filesystem Investigation Workflow
10. LX2.10 Acquiring the Disk Image
11. LX2.11 Complex Disk Geometries
12. Module Summary
13. Check My Knowledge

Linux Log Analysis

The Linux logging architecture from an investigator's seat: syslog, journald, and auditd, what each captures and how to query it, authentication-log analysis for SSH and sudo, journald and auditd as forensic sources, web server log forensics, system and service logs, cross-log correlation into...

1. LX3.1 The Linux Logging Architecture
2. LX3.2 Authentication Log Analysis
3. LX3.3 Journald Investigation
4. LX3.4 Auditd Forensics
5. LX3.5 Web Server Log Analysis
6. LX3.6 System, Service, and Application Logs
7. LX3.7 Cross-Log Correlation
8. LX3.8 Log Integrity and Tampering Detection
9. LX3.9 The Log Analysis Workflow
10. Module Summary
11. Check My Knowledge

Phase 3: Compromise Scenarios

Initial Access and Server Compromise

How a Linux server gets breached, and how you prove which way it happened.

1. LX4.1 The Two Entry Vectors
2. LX4.2 Scoping the SSH Compromise
3. LX4.3 Finding the SSH Breach Point
4. LX4.4 The Web Application Vector
5. LX4.5 Web Shells and Reverse Shells
6. LX4.6 Application and Error Logs as Exploit Evidence
7. LX4.7 Comparing the Two Vectors
8. LX4.8 The Initial-Access Investigation Workflow
9. Module Summary
10. Check My Knowledge

Escalation, Persistence and Lateral Movement

What an attacker does after the foothold, and how you reconstruct all of it.

1. LX5.1 The Post-Foothold Arc
2. LX5.2 Escalation Vectors
3. LX5.3 Reconstructing the Escalation Chain
4. LX5.4 Persistence: Accounts, Keys, and Scheduled Jobs
5. LX5.5 Persistence: Services and Shell Initialization
6. LX5.6 Persistence: Library, PAM, and Kernel
7. LX5.7 Lateral Movement: Trust, Pivots, and Config Management
8. LX5.8 Lateral Movement: Credential Reuse and Internal Services
9. LX5.9 The Multi-Host Campaign
10. Module Summary
11. Check My Knowledge

Container and Cloud Compromise

Incident response when the host is no longer the whole story.

1. LX6.1 The Cloud and Container Investigation Surface
2. LX6.2 Cloud-Native Evidence and Its Volatility
3. LX6.3 The Instance Metadata Entry Vector
4. LX6.4 IAM Credential Abuse and Cloud Enumeration
5. LX6.5 Cloud Data Exfiltration
6. LX6.6 Container Runtime and the Image Supply Chain
7. LX6.7 Kubernetes Audit Logs and Service-Account/RBAC Abuse
8. LX6.8 Container Escape to the Node
9. LX6.9 Cloud Persistence and Cryptojacking
10. LX6.10 The Cross-Environment Campaign
11. LX6.11 ESXi Host Forensics
12. LX6.12 vCenter and the Blast Radius
13. Module Summary
14. Check My Knowledge

Phase 4: Live Response and Deep Analysis

Memory and Malware Analysis

Investigating the evidence that exists only in RAM, and reverse-engineering what you recover.

1. LX7.1 Why Memory Is Ground Truth
2. LX7.2 Acquiring Memory Without Destroying It
3. LX7.3 Building the Picture: Process and Profile Analysis
4. LX7.4 Recovering What the Rootkit Hid
5. LX7.5 Pulling Evidence From RAM
6. LX7.6 Memory Timeline and the Disk Cross-Check
7. LX7.7 Static Malware Triage for IR
8. LX7.8 Dynamic Analysis in a Sandbox
9. LX7.9 IOCs and YARA
10. LX7.10 Memory and Malware Campaign
11. LX7.11 Module Summary
12. LX7.12 Check My Knowledge

Live Response and Volatile-Data Triage

The first hour on a running, compromised Linux host.

1. LX8.1 The Live-Response Workflow
2. LX8.2 Capturing Volatile State in Order
3. LX8.3 The /proc Live Investigation
4. LX8.4 Detecting Concealment on a Live Host
5. LX8.5 Triage Collection and Custody
6. LX8.6 Handoff and Containment
7. LX8.7 Live-Response Capstone
8. LX8.8 Module Summary
9. LX8.9 Check My Knowledge

Phase 5: Readiness and Reporting

IR Readiness and Reporting

The two things that bracket every Linux investigation: being ready before one and reporting after it.

1. LX9.1 Why Readiness and Reporting Decide the Outcome
2. LX9.2 The Linux IR Playbook
3. LX9.3 Readiness: Tooling and Collection Pre-Staged
4. LX9.4 Tabletop Exercises
5. LX9.5 IR Metrics and Maturity
6. LX9.6 The Technical Findings Report
7. LX9.7 The Executive Summary
8. LX9.8 IOCs, Remediation, and Lessons Learned
9. LX9.9 Legal, Regulatory, and Cross-Platform Reporting
10. Module Summary
11. Check My Knowledge

Phase 0: Course Resources

Cheatsheets

Ten sheets of Linux investigation commands, each entry showing what the output proves, what it does not, and where the course teaches the reasoning behind it.

1. Orienting on an Unfamiliar Host
2. Filesystem Structure and Timestamps
3. Recovery and Anti-Forensics Detection
4. Reading the Logs
5. Establishing the Entry Point
6. Escalation, Persistence, and Lateral Movement
7. Containers, Cloud, and Virtualization
8. Memory and Malware Triage
9. Live Response and the Handoff
10. Paths, Logs, and Distribution Deltas

Cookbooks

Seven end-to-end Linux investigation procedures, each with the commands, the branch points, and what the procedure does not cover.

1. The Sixty-Minute Linux Triage
2. Acquiring Memory From a Production Host
3. Sweeping Every Persistence Mechanism
4. Investigating a Container Compromise
5. Investigating a Cloud Workload Compromise
6. Containing and Eradicating Without Losing the Case
7. Recovering What Was Deleted

Lab Setup

Building a Linux investigation lab on hardware you already own, with the logging Linux ships disabled turned on, and a written ground truth to check your analysis against.

1. What To Build On
2. Building the Hosts
3. Turning On the Logging Linux Ships Disabled
4. Generating Evidence With a Ground Truth
5. Verify the Lab, and Know Its Limits

Walkthroughs

Seven worked Linux investigations, each carried from the referral to a written finding, including the ones that end in a defensible no.

1. Ten Minutes to a Defensible No
2. From an Upload Form to Root in Four Minutes
3. The File That Claimed to Be a Year Old
4. The Process Nobody Could See
5. The Container That Did Not Stay In Its Container
6. The Credential That Was Already Gone
7. The Sixty Minutes

Playbooks

Seven Linux threats, each with the examination sequence that establishes what happened, the check that could end it early, and what the handoff must carry.

1. Suspected SSH Compromise
2. Web Application Compromise
3. Cryptomining on a Linux Host
4. Suspected Concealment or Rootkit
5. Container or Cloud Workload Compromise
6. Data Staging on a Server
7. Ransomware Precursor on Linux

Playground

Where to practice Linux investigation and use it at work: guided scenarios against Linux evidence, the toolkit, the detection library, and your own lab.

References & Further Reading

Linux forensic tool documentation, distribution and kernel references, container and cloud provider resources, and incident response frameworks used throughout the Linux Endpoint Investigation course.

Course Completion

Course Exam

Linux Endpoint Investigation end-of-course exam: a simulation-based assessment testing your ability to triage, investigate, and report on a Linux server compromise you have not seen before, using the artifacts and the method built across the course.

1. Course Completion. Linux Endpoint Investigation

Generated 2026-09-01 from the published course. The current syllabus is always at ridgelinecyber.com/training/courses/linux-endpoint-investigation/