

Why I Use CSI Linux as My Digital Forensics Workstation

Tue, Jun 9, 2:05 PM EDT · <https://scottslab.io/posts/csi-linux-digital-forensics-workstation>



TL;DR

CSI Linux as the DFIR workstation, mostly because 50-odd forensic tools are already installed and an investigation should not start with dependency hunting. Four independent ways back onto the box (browser desktop over SSO, SSH, hypervisor guest agent, VPN), with evidence on a central SMB share that systemd automount keeps alive across reboots. Snapshot before analysis and roll back after, so one case never contaminates the next.

Published: June 2026 **Category:** Digital Forensics **Reading Time:** 14 minutes

Executive Summary

- Deployed CSI Linux as the primary DFIR workstation - 50+ forensic tools pre-installed across 9 categories with zero dependency hunting
- Integrated with centralized evidence management: SMB share on desktop, web upload portal with zero-trust access for external investigators
- Configured 4 access methods: web-based remote desktop (SSO), SSH, hypervisor guest agent, and VPN backup

- Solved persistent network mount issues using systemd automount for reliable evidence share access across reboots
 - Established snapshot-based investigation workflow - snapshot before analysis, rollback after to prevent workstation contamination
-

Goal

Problem: When an incident happens, you need to start investigating immediately - not spend three hours installing Autopsy, hunting down Volatility dependencies, and figuring out why your disk imaging tool won't compile. I needed a forensic workstation that was ready to go the moment I needed it, with tools for disk forensics, memory analysis, network capture, OSINT, and malware triage all in one place.

Why it mattered: Incident response is time-sensitive. Every hour spent setting up tools is an hour the attacker has to cover tracks, exfiltrate data, or move laterally. A pre-configured forensic workstation means I can start analyzing evidence within minutes of receiving it, not days. Plus, running forensics on the same machine as production services risks contaminating both the investigation and the infrastructure.

Scope and Constraints

In Scope

- DFIR workstation deployment and configuration
- Evidence management infrastructure (centralized share, upload portal)
- Remote access methods (web desktop, SSH, guest agent)
- Tool inventory and verification
- Investigation workflow design

Out of Scope

- Mobile forensics (P1 improvement - MVT, Andriller)
- Cloud forensics (P2 improvement - AWS IR, Azure tools)
- Malware sandboxing (P2 improvement - Cuckoo/CAPE)
- Custom tool development
- Forensic report automation

Key Constraints

- **VM resources** - 4 CPU cores, 8GB RAM, 50GB SSD (home lab budget)

- **Evidence integrity** - Must prevent accidental modification of evidence
 - **Investigation isolation** - Forensic work must be isolated from production infrastructure
 - **Accessibility** - Investigators need GUI access for tools like Autopsy and Wireshark
-

Tools and References

CSI Linux Tool Categories (50+ Tools)

Category	Tools	Purpose
Disk & File Forensics	Autopsy, Sleuth Kit, Foremost, Scalpel, TestDisk, PhotoRec, Binwalk	Disk imaging, file carving, partition recovery, firmware extraction
Memory Forensics	Volatility 3	RAM analysis, process enumeration, malware artifact detection
Network Forensics	Wireshark, Nmap, Kismet, Bettercap	Packet capture, network discovery, wireless analysis
OSINT	Maltego, Recon-ng, Sherlock, theHarvester, Amass, SpiderFoot	Username lookup, subdomain enumeration, link analysis
Password Analysis	Hashcat, John the Ripper, Hydra, Medusa	Hash cracking, credential testing
Malware Analysis	ClamAV, Radare2, YARA	AV scanning, reverse engineering, pattern matching
Steganography	Steghide, ExifTool	Hidden data extraction, metadata analysis
Web Testing	GoBuster, Nikto, SQLMap	Directory enumeration, vulnerability scanning
Wireless Security	Aircrack-ng, Kismet	WiFi security testing, multi-protocol sniffing

Infrastructure Tools

Tool	Role
CSI Linux	Ubuntu 22.04 LTS-based DFIR distribution

Tool	Role
Proxmox	VM hypervisor hosting the forensic workstation
Apache Guacamole	Web-based remote desktop access
Samba/CIFS	Evidence share file system
Cloudflare Access	Zero-trust authentication for upload portal

References:

- [CSI Linux Official Documentation](#)
 - [Autopsy Digital Forensics](#)
 - [Volatility 3 Documentation](#)
 - [SANS DFIR Poster](#)
-

Approach

Phase 1: Distribution Selection

What I did: Evaluated forensic Linux distributions (SIFT, REMnux, Kali, CSI Linux) against requirements: pre-installed DFIR tools, OSINT capability, VM-friendly design, active maintenance.

Why: Building a forensic workstation from scratch on bare Ubuntu takes days and creates dependency nightmares. A purpose-built distro provides a tested toolchain where everything works together.

Phase 2: VM Deployment

What I did: Created a VM on Proxmox with 4 CPU cores, 8GB RAM, and 50GB SSD. Installed CSI Linux from ISO, enabled QEMU guest agent for remote management.

Why: Running as a VM provides isolation from production, enables snapshots before investigations, and allows resource scaling if needed. Guest agent enables shutdown/restart from hypervisor without console access.

Phase 3: Access Configuration

What I did: Configured four access methods: web-based remote desktop via Guacamole (behind SSO), SSH with password auth enabled via drop-in config, hypervisor guest agent for remote management, VPN mesh as backup path.

Why: Forensic work often requires GUI tools (Autopsy, Wireshark, Maltego). Web desktop provides full GUI access from any browser without VPN. Multiple access methods ensure I can

always reach the workstation.

Phase 4: Evidence Share Integration

What I did: Mounted centralized evidence share to the forensic workstation desktop using CIFS with systemd automount. Configured credentials file, automount options, and network wait service.

Why: Evidence needs to be accessible directly from the forensic tools. Dragging files between the desktop evidence folder and Autopsy is more intuitive than transferring via SCP. Automount ensures the share survives reboots.

Phase 5: Investigation Workflow Design

What I did: Established case management workflow using CSI Linux's built-in case creation, integrated with evidence share directory structure. Documented snapshot-before-investigation procedure.

Why: Consistent workflow prevents mistakes under pressure. Snapshots before investigation allow rollback if a tool corrupts the workstation environment.

Implementation Notes

VM Configuration (Sanitized)

```
VM Name: <FORENSIC_VM>
Hypervisor: Proxmox
Resources:
  CPU: 4 cores
  RAM: 8GB
  Disk: 50GB SSD (VirtIO)
Network: Bridge to lab VLAN
Guest Agent: QEMU (enabled for remote management)
```

Evidence Share Mount (Sanitized)

```
# /etc/fstab entry with systemd automount
//<EVIDENCE_IP>/evidence /home/csi/Desktop/Evidence cifs
credentials=/root/.smbcreds,uid=1000,gid=1000,vers=3.0,x-
systemd.automount,x-systemd.mount-timeout=30,_netdev,nofail 0 0
```

Mount options explained:

- `x-systemd.automount` - Creates automount point, actual mount on first access
- `x-systemd.mount-timeout=30` - Don't hang boot if share unavailable
- `_netdev` - Wait for network before attempting mount
- `nofail` - Don't fail boot if mount fails
- `vers=3.0` - Explicit SMB protocol version

Credentials File (Sanitized)

```
# /root/.smbcreds (chmod 600)
username=<SMB_USER>
password=<SMB_PASSWORD>
domain=<SMB_DOMAIN>
```

Enable Network Wait Service

```
# Required for boot-time network dependencies
systemctl enable NetworkManager-wait-online.service
```

SSH Drop-in Configuration (Sanitized)

```
# /etc/ssh/sshd_config.d/99-local.conf
PasswordAuthentication yes
PermitRootLogin prohibit-password
```

Snapshot Before Investigation

```
# From Proxmox CLI or GUI
qm snapshot <VMID> pre-investigation-$(date +%Y%m%d)

# After investigation, rollback if needed
qm rollback <VMID> pre-investigation-<DATE>
```

Validation and Evidence

Signals That Proved It Worked

Check	Expected	Actual
Forensic tools installed	50+	50+ verified
Evidence share mount	Auto on boot	Working with automount
Web desktop access	GUI via browser	Functional via Guacamole
SSH access	Remote CLI	Enabled and working
Guest agent	Remote management	Responding to Proxmox
Snapshot/rollback	Clean state recovery	Tested successfully

Tool Verification Commands (Sanitized)

```
# Verify key forensic tools
which autopsy && autopsy --version
which volatility3 && vol -h | head -1
which wireshark && wireshark --version
which maltego && echo "Maltego installed"
which hashcat && hashcat --version

# Verify evidence share mount
ls /home/csi/Desktop/Evidence/
mount | grep Evidence

# Verify access methods
systemctl status sshd
systemctl status qemu-guest-agent
```

Case Management Workflow Test

1. Created new case via CSI Linux case manager
 2. Copied sample evidence from share to case directory
 3. Opened evidence in Autopsy
 4. Verified file carving results
 5. Exported findings to case notes
-

Results

Metric	Outcome
Forensic Tools	50+ tools across 9 categories, ready to use
Evidence Access	Centralized share mounted on desktop + web upload portal
Access Methods	4 paths (web desktop, SSH, guest agent, VPN)
Time to Ready	Minutes (vs. days for manual build)
Investigation Isolation	VM separation from production
State Recovery	Snapshot-based rollback available

What I Learned

1. **CSI Linux dramatically accelerates forensic workstation setup.** What would take days of manual installation and dependency resolution is ready in under an hour. The toolchain is pre-tested and integrated.
2. **systemd automount (`x-systemd.automount`) is essential for network mounts.** Standard fstab mounts fail when the network isn't ready at boot. Automount defers the actual mount until first directory access, completely avoiding race conditions.
3. **`_netdev` and `nofail` flags prevent boot hangs.** Without these, an unavailable network share can stall your entire boot process. Forensic workstations need to boot reliably even when infrastructure is down.
4. **The `NetworkManager-wait-online.service` must be enabled.** If you have any boot-time network dependencies, this service ensures the network is actually up before systemd continues. It's often disabled by default.
5. **CIFS mounts may require explicit protocol version.** Some SMB servers reject version negotiation. Adding `vers=3.0` explicitly can fix mysterious mount failures.
6. **Evidence on the desktop creates intuitive workflow.** Drag-and-drop between the evidence folder and tools like Autopsy feels natural. SCP transfers for every file would be tedious.
7. **VM snapshots before investigations are critical.** One wrong tool execution - accidentally writing to a disk image, corrupting a memory dump - can contaminate your workstation. Snapshots provide a clean rollback path.

8. **Web-based remote desktop works better than expected.** I was skeptical about running GUI forensic tools through a browser, but Guacamole handles Autopsy, Wireshark, and even Maltego smoothly.
 9. **Running forensics on a separate VM prevents cross-contamination.** The forensic workstation never touches production services directly. Evidence comes in through the share; findings go out through reports.
 10. **Zero-trust access controls on the evidence portal prevent unauthorized submissions.** External investigators can upload evidence without VPN access, but only after email-based identity verification.
-

What I Would Improve Next

P0 (Do This Week)

- **Automated evidence hashing** - SHA-256 hash every file on upload for chain of custody verification
- **Write-once evidence storage** - Implement append-only storage for original evidence to prevent tampering

P1 (Do This Month)

- **Additional memory forensics tools** - Add MemProcFS and standardize on Volatility 3 alongside the base Volatility (skip Rekall, it's archived and unmaintained)
- **Automated triage scripts** - Build scripts that run common tools (strings, file, exiftool, clamscan) on new evidence automatically
- **Mobile forensics capability** - Add MVT (Mobile Verification Toolkit) and Andriller for mobile device analysis

P2 (Do This Quarter)

- **SIEM integration** - Log forensic workstation activity to Wazuh for audit trail
 - **Malware sandbox** - Add Cuckoo or CAPE for dynamic malware analysis
 - **Report templates** - Build forensic report templates that auto-populate from tool outputs
 - **Cloud forensics tools** - Add AWS IR tools and Azure forensics capabilities
-

Common Failure Modes

1. **"Evidence folder is empty after reboot"** - Network mount failed. Check that `x-systemd.automount` is in `fstab`, `NetworkManager-wait-online.service` is enabled, and credentials file exists with correct permissions (600).
 2. **"SSH connection refused"** - CSI Linux ships with SSH disabled by default. Create drop-in config at `/etc/ssh/sshd_config.d/` to enable password auth, then restart `sshd`.
 3. **"Wrong IP address / can't reach workstation"** - IP conflict with another device. Check DHCP leases and static configurations on other hosts. Create DHCP reservation with MAC binding to prevent future conflicts.
 4. **"QEMU guest agent not responding"** - Agent may not be running inside the VM. Must enable from within the desktop GUI first - can't bootstrap remotely without it.
 5. **"Mount fails with protocol error"** - SMB version mismatch. Add `vers=3.0` (or `vers=2.1`) explicitly to the `fstab` mount options to force a specific protocol version.
-

Security Considerations

Investigation Isolation

- Forensic workstation runs on separate VM from production services
- No direct network path between forensic tools and production infrastructure
- Evidence enters through controlled share, not direct disk access

Evidence Integrity

- Centralized evidence share provides single source of truth
- Automount ensures consistent access without manual intervention
- Future: write-once storage and automated hashing for chain of custody

Access Control

- Web desktop behind SSO authentication
- SSH requires explicit enablement (disabled by default)
- Evidence upload portal uses zero-trust access control (email verification)
- Multiple access methods provide redundancy without sacrificing security

State Recovery

- Snapshot before investigation preserves clean workstation state
 - Rollback capability ensures contaminated environments can be restored
 - VM isolation means workstation corruption doesn't affect production
-

Runbook

If Evidence Share Is Empty

1. **Check mount status** - `mount | grep Evidence`
2. **Try manual access** - `ls /home/csi/Desktop/Evidence/` (triggers automount)
3. **Verify automount unit** - `systemctl status home-csi-Desktop-Evidence.automount`
4. **Test network connectivity** - `ping <EVIDENCE_IP>`
5. **Check credentials file** - Verify `/root/.smbcreds` exists and has 600 permissions

If Web Desktop Won't Connect

1. **Verify Guacamole service** - Check tunnel/proxy status
2. **Test direct SSH** - If SSH works, tunnel is the issue
3. **Check SSO authentication** - Verify identity provider is responding
4. **Review browser console** - WebSocket errors indicate connection issues
5. **Try alternate access** - Use VPN + direct RDP as fallback

How to Snapshot Before Investigation

```
# From Proxmox host
qm snapshot <VMID> pre-case-$(date +%Y%m%d-%H%M)

# Verify snapshot exists
qm listsnapshot <VMID>

# After investigation, rollback if needed
qm rollback <VMID> <SNAPSHOT_NAME>
```

How to Upload Evidence via Web Portal

1. Navigate to `<UPLOAD_PORTAL>` in browser
2. Authenticate via email verification (Cloudflare Access)

- 3. Select files for upload (drag-and-drop supported)
- 4. Verify upload completion
- 5. Files appear in evidence share within seconds

How to Start a New Case

- 1. Open CSI Linux case manager from desktop
- 2. Enter case name and investigator information
- 3. Case directory created with standard structure
- 4. Copy relevant evidence from share to case directory
- 5. Begin analysis with appropriate tools
- 6. Document findings in case notes

Appendix

Glossary

Term	Definition
CSI Linux	Ubuntu-based distribution purpose-built for digital forensics and OSINT
DFIR	Digital Forensics and Incident Response
OSINT	Open Source Intelligence - gathering information from public sources
Volatility	Memory forensics framework for analyzing RAM dumps
Autopsy	GUI forensic platform built on Sleuth Kit for disk analysis
Sleuth Kit	CLI tools for disk forensics (file system analysis, timeline creation)
Guacamole	Clientless remote desktop gateway (access via web browser)
Steganography	Hiding data within other files (images, audio, video)
YARA	Pattern matching language for malware identification
File Carving	Recovering files from disk images based on file signatures

MITRE ATT&CK Mapping (Forensic Tool Coverage)

Technique ID	Name	Tools That Detect/Analyze
T1005	Data from Local System	Autopsy, Sleuth Kit, Foremost, Scalpel

Technique ID	Name	Tools That Detect/Analyze
T1039	Data from Network Shared Drive	Wireshark, network log analysis
T1083	File and Directory Discovery	Autopsy timeline, Sleuth Kit fls
T1057	Process Discovery	Volatility pslist, pstree, psxview
T1049	System Network Connections Discovery	Volatility netscan, Wireshark
T1003	OS Credential Dumping	Volatility hashdump, mimikatz plugin
T1027	Obfuscated Files or Information	YARA, Binwalk, strings analysis
T1071	Application Layer Protocol	Wireshark, Zeek log analysis
T1059	Command and Scripting Interpreter	Volatility cmdline, consoles

Tool-to-Technique Coverage Matrix

Tool	Primary Techniques Covered
Autopsy	T1005, T1083, T1027 (disk-level data, files, hidden content)
Volatility	T1057, T1049, T1003, T1059 (process, network, credentials, commands)
Wireshark	T1039, T1071, T1049 (network traffic, protocols, connections)
YARA	T1027 (obfuscated/packed malware identification)
Hashcat/John	T1003 (credential hash analysis)
Binwalk	T1027 (embedded files, firmware extraction)
ExifTool	T1005 (metadata extraction from files)

Artifacts Produced

- **Forensic Workstation VM** - CSI Linux deployment with 50+ tools configured
- **Evidence Share Infrastructure** - Centralized SMB share mounted to workstation desktop
- **Web Upload Portal** - Zero-trust evidence submission for external investigators
- **Systemd Automount Configuration** - Reliable network mount across reboots
- **SSH Access Configuration** - Drop-in config enabling remote CLI access
- **Case Management Workflow** - Documented procedure for investigation lifecycle

Building your own forensic lab? CSI Linux won't give you everything, but it gives you a running start. The network mount gotchas will still bite you though - hope the automount notes help.

Written by Scott S. — Contact: scott@scottschlangen.com — scottslab.io — <https://scottslab.io/posts/csi-linux-digital-forensics-workstation>