

Safe Lab Telemetry Cleanup

Alerts to Answers Analyst Field Guide 06 - v20

Introduction

This instructor-focused field guide provides an evidence-preserving workflow for removing student-generated telemetry from the dedicated Alerts to Answers lab between exercises. It separates generated data from protected configuration and requires a dry-run, verified backups, integrity comparisons, and post-cleanup telemetry validation before the environment is returned to students.

Purpose and Scope

Use this guide only on the validated disposable classroom environment: Wazuh 4.14.5, an Ubuntu 26.04 Suricata sensor, and a Windows 11 endpoint running Sysmon and the Wazuh agent. The procedure is not a production retention policy and must never be used on systems holding evidence required for grading, incident response, legal preservation, or audit retention.

Critical safety boundary. Cleanup is destructive. Dry-run is mandatory, execution requires instructor authorization, and Wazuh index deletion is permitted only after a successful snapshot. The source package has passed a static safety audit but still requires separate dry-runs on the actual Wazuh, Ubuntu, and Windows lab virtual machines before destructive use.

Chapter and Lab Alignment

Alignment	Placement
Primary chapter alignment	Chapter 9 - Operating the Platform
Supporting chapter alignment	Chapters 3 and 5
Lab alignment	Instructor reset procedure supporting Lab 0, Lab 0A, and Labs 1-9
Recommended point of use	Between graded exercises, after evidence retention requirements are satisfied, and before the next student cohort begins

Required Companion Scripts

The package includes three instructor-vetted scripts: `cleanup-wazuh-lab_v20.sh`, `cleanup-suricata-lab_v20.sh`, and `Clear-Windows-Sysmon-LabLogs_v20.ps1`. Keep the package together, verify its published checksums before use, and do not substitute similarly named files.

1. Safety Model

Each script uses a closed allowlist. It can act only on explicitly named generated-data locations or event channels. A path supplied for Suricata must resolve beneath `/var/log/suricata`. The Wazuh index option can address only `wazuh-alerts-4.x-*` and `wazuh-archives-4.x-*`.

Controls that must all pass

Control	Effect
Dry-run default	Inventories targets and creates integrity baselines; performs no deletion.
Root/Administrator requirement	Prevents partial execution under insufficient privileges.
Typed confirmation	<code>-confirm CLEAN-WAZUH-LAB</code> , <code>-confirm CLEAN-SURICATA-LAB</code> , or <code>-Confirmation CLEAN-WINDOWS-LAB</code> .
Dated .bak backup and protected-state hash	Creates and verifies dated <code>.bak</code> backups before deletion, then hashes configuration, rules, enrollment keys, and relevant service/registry state before and after.
Service-aware execution	Stops only the data-producing Linux service, performs cleanup, then restarts it if it was previously active.
Post-checks	Runs service checks; Suricata also runs <code>suricata -T</code> ; Windows confirms protected state is byte-for-byte/logically identical.

Never-clean list

- `/var/ossec/etc/client.keys` and every agent-side `client.keys` file
- `/var/ossec/etc`, `/var/ossec/ruleset`, custom rules and decoders, CDB lists, certificates, and agent-group configuration
- `/var/ossec/queue`, Wazuh databases, registration and enrollment APIs, and `manage_agents`
- `/etc/suricata`, `/var/lib/suricata/rules`, `/etc/suricata/rules`, and `suricata.yaml`
- Sysmon binaries, service, driver, registry configuration, or installed XML policy

Evidence rule. Never clean a system holding evidence needed for grading, an incident, an investigation, legal preservation, or audit retention.

2. Pre-Lab Preparation

1. Copy the package to each applicable VM. Keep the original read-only.

2. Open an administrative shell: sudo-capable Bash on Linux; PowerShell as Administrator on WIN11.
3. Confirm system identity and time before cleanup.
4. Run the appropriate dry-run. Review every displayed target and the report directory.
5. If the target list contains anything unexpected, stop. Do not use `--execute`.

```
# Wazuh server
hostnamectl --static
date -u
sudo ./cleanup-wazuh-lab_v20.sh --dry-run

# UB2604
hostnamectl --static
date -u
sudo ./cleanup-suricata-lab_v20.sh --dry-run

# WIN11 - PowerShell as Administrator
hostname
Get-Date -AsUTC
.\Clear-Windows-Sysmon-LabLogs_v20.ps1
```

Why do I care? Dry-run exposes the exact deletion boundary before any service is stopped. This is the most important human verification step.

Expected dry-run result

- A timestamped report directory is created.
- Candidate files/channels and sizes or record counts are listed.
- Protected configuration hashes/state are recorded.
- The output ends with `DRY-RUN COMPLETE`.
- No monitored service is restarted and no telemetry is deleted.

3. Wazuh Server Cleanup

Local generated logs only

WAZUH-SRV • TERMINAL

```
sudo ./cleanup-wazuh-lab_v20.sh --execute --confirm CLEAN-WAZUH-LAB
```

This clears generated files only inside `/var/ossec/logs/alerts` and `/var/ossec/logs/archives`. It does not delete Wazuh operational logs, `/var/ossec/queue`, databases, configuration, rules, certificates, or `client.keys`.

Optional: remove dashboard alert/archive history

Local files and dashboard history are separate storage layers. Use the index option only when the class needs a visually clean Dashboard and the environment is dedicated to this lab.

```
export WAZUH_INDEXER_URL='https://127.0.0.1:9200'
export WAZUH_INDEXER_USER='admin'

read -rsp 'Indexer password: ' WAZUH_INDEXER_PASSWORD; export
WAZUH_INDEXER_PASSWORD; echo

export WAZUH_INDEXER_CA='/etc/wazuh-indexer/certs/root-ca.pem'

export WAZUH_SNAPSHOT_REPOSITORY='wazuh-lab-backups'

sudo
--preserve-
env=WAZUH_INDEXER_URL,WAZUH_INDEXER_USER,WAZUH_INDEXER_PASSWORD,WAZUH_INDEXER_
CA,WAZUH_SNAPSHOT_REPOSITORY
\

./cleanup-wazuh-lab_v20.sh --dry-run --delete-indexes

sudo
--preserve-
env=WAZUH_INDEXER_URL,WAZUH_INDEXER_USER,WAZUH_INDEXER_PASSWORD,WAZUH_INDEXER_
CA,WAZUH_SNAPSHOT_REPOSITORY
\

./cleanup-wazuh-lab_v20.sh --execute --delete-indexes \

--confirm CLEAN-WAZUH-LAB

unset WAZUH_INDEXER_PASSWORD
```

Backup gate. Local cleanup verifies dated Wazuh configuration and log .bak archives. Index deletion requires a configured snapshot repository and a SUCCESS snapshot result. Monitoring, state, security, and saved-object indices remain excluded.

4. Ubuntu Suricata Cleanup

```
sudo ./cleanup-suricata-lab_v20.sh --dry-run
sudo ./cleanup-suricata-lab_v20.sh --execute \
--confirm CLEAN-SURICATA-LAB
```

Before cleanup, the script creates and verifies dated backups of Suricata configuration, ET rules, Ubuntu Wazuh agent configuration/enrollment, and Suricata logs. It deletes only regular files beneath /var/log/suricata, then validates hashes and service health.

Why do I care? Stopping the writer avoids deleting an open log while Suricata is appending to it. Restarting lets Suricata create fresh log files with its configured ownership and format.

Validation

```
sudo systemctl is-active suricata
sudo `suricata -T` -c /etc/suricata/suricata.yaml
sudo find /var/log/suricata -maxdepth 1 -type f -printf '%TY-%Tm-%Td
%TH:%TM %s %p\n' | sort
sudo suricata-update list-sources 2>/dev/null | head
```

- Service result should be active.
- The configuration test should complete successfully.
- New log files may appear immediately after restart; their timestamps should be current and their sizes small.
- ET rule content and configuration hashes must match the pre-cleanup manifest.

Common mistake. Do not point `--log-dir` at /var/log, /etc/suricata, or a symlink outside /var/log/suricata. The script rejects those targets.

5. Windows 11 and Sysmon Cleanup

```
# Dry-run

.\Clear-Windows-Sysmon-LabLogs_v20.ps1

# Execute

.\Clear-Windows-Sysmon-LabLogs_v20.ps1 -Execute `
-Confirmation CLEAN-WINDOWS-LAB -Confirm:$false
```

Default channels are Sysmon Operational, Security, System, and Application. Before clearing, each is exported as an .evtx.<UTC date>.bak file; Wazuh configuration/enrollment files and Sysmon registry keys also receive dated .bak copies.

```
.\Clear-Windows-Sysmon-LabLogs_v20.ps1 -Execute `
-Confirmation CLEAN-WINDOWS-LAB -Confirm:$false `

-Logs
'Microsoft-Windows-Sysmon/Operational','Security','System','Application',`
'Microsoft-Windows-PowerShell/Operational','Windows PowerShell'
```

Validation

```
Get-Service Sysmon64, WazuhSvc -ErrorAction SilentlyContinue

Get-WinEvent -ListLog 'Microsoft-Windows-Sysmon/Operational' |

Select-Object LogName,RecordCount,IsEnabled,FileSize

Get-WinEvent -FilterHashtable @{LogName='Security'; Id=1102} -MaxEvents
1 -ErrorAction SilentlyContinue
```

Expected artifact. Clearing the Security log can generate Event ID 1102. A small nonzero post-clear record count is therefore normal and proves that logging resumed.

Preserved. The script never invokes Sysmon with -i, -u, or -c; never edits its registry configuration; and never stops or reconfigures the Wazuh agent.

6. End-to-End Validation Checklist

Check	Pass condition
Wazuh manager	systemctl is-active wazuh-manager returns active.

Check	Pass condition
Agent enrollment	/var/ossec/etc/client.keys remains nonempty and hash-identical; u2604 and win11 remain enrolled/connected.
Wazuh configuration	config-before.sha256 and config-after.sha256 match; config-diff.txt is empty.
Fresh Wazuh data	Generate one benign event and confirm only current timestamps appear in the Dashboard.
Suricata	Service is active; <code>suricata -T</code> succeeds; ET rules hashes match; eve.json begins receiving new events.
Sysmon	Sysmon64 service remains present/running and new Event ID 1 or 22 records appear after test activity.
Wazuh agent on WIN11	WazuhSvc remains running and client.keys/ossec.conf hashes match.
Reports	Retain the timestamped run folders with target inventory, pre/post state, and results.

Minimal telemetry generation

```
# UB2604

curl -I http://example.com

sudo tail -n 5 /var/log/suricata/eve.json

# WIN11

Resolve-DnsName example.com

Get-WinEvent -FilterHashtable
@{LogName='Microsoft-Windows-Sysmon/Operational'; Id=22} -MaxEvents
3
```

Validation gate. Do not hand the environment to students until every protected-state comparison passes and all three telemetry pipelines produce a new current-time event.

7. Troubleshooting and Recovery

Symptom	Action
Protected-state mismatch	Stop. Preserve the run report. Compare the before/after JSON or checksum manifests. Do not repeat cleanup.
Wazuh manager inactive	Run <code>systemctl status wazuh-manager</code> and

Symptom	Action
Dashboard still shows old alerts	journalctl -u wazuh-manager -n 100 --no-pager. Configuration was not intentionally edited.
Suricata test fails	Confirm --delete-indexes was intentionally used and inspect index-targets.tsv. Also confirm Dashboard time range and refresh. Read suricata-test.log. Do not change rules during cleanup troubleshooting; restore only from a known preexisting backup if required.
Windows access denied	Reopen Windows PowerShell as Administrator. Security log clearing requires elevation.
Post-clear logs are nonzero	Usually normal: active services immediately write startup/status events. Check timestamps rather than expecting permanent zero.
Script interrupted	The Linux EXIT trap attempts to restart any service that was active. Verify service state and integrity reports before rerunning.

Report locations

- /var/tmp/wazuh-lab-cleanup/<UTC timestamp>/
- /var/tmp/suricata-lab-cleanup/<UTC timestamp>/
- C:\UTC timestamp>\

Recovery boundary. Dated .bak files support recovery of cleaned logs and protected state. Wazuh index recovery uses the required snapshot. Test restoration separately before treating these as production-grade backups.

8. Instructor Operating Procedure

- Take a VM snapshot or platform checkpoint before the first cleanup rehearsal.
- Run all dry-runs and inspect the exact target inventories.
- Execute one component at a time: Windows, Suricata, then Wazuh local logs; delete Wazuh indices last.
- Complete the end-to-end validation checklist.
- Record the UTC cleanup timestamp and retain the reports for the class session.

- Generate a small baseline event from each sensor so students can confirm collection health.

Recommended classroom policy

- Instructor executes cleanup; students may run dry-run and validation only.
- Use only on dedicated disposable lab systems between exercises.
- Never schedule these scripts automatically.
- Never broaden wildcard patterns or protected directories for convenience.
- Never clear evidence before grades, incident review, or required retention is complete.

Design decision. The package intentionally omits agent database resets, Filebeat registry resets, Wazuh queue deletion, and Sysmon reinstall commands. Those operations can damage enrollment, duplicate ingestion, or change endpoint configuration.

Version applicability

Validated for the stated classroom design: Wazuh 4.14.5, Ubuntu 26.04 sensor with Suricata 8.0.x and ET rules, and Windows 11 with Sysmon and the Wazuh agent. Re-run dry-run and integrity checks after any platform upgrade.

References

Microsoft. (2024). wevtutil. Microsoft Learn. <https://learn.microsoft.com/en-us/windows-server/administration/windows-commands/wevtutil>

Open Information Security Foundation. (n.d.). Log rotation. Suricata user guide. <https://docs.suricata.io/en/suricata-8.0.6/output/log-rotation.html>

Wazuh, Inc. (n.d.). Alert management. Wazuh documentation. <https://documentation.wazuh.com/current/user-manual/manager/alert-management.html>

Wazuh, Inc. (n.d.). Client keys file. Wazuh documentation. <https://documentation.wazuh.com/current/development/client-keys.html>

Wazuh, Inc. (n.d.). Index lifecycle management. Wazuh documentation. <https://documentation.wazuh.com/current/user-manual/wazuh-indexer-cluster/index-lifecycle-management.html>

Wazuh, Inc. (n.d.). Wazuh indexer. Wazuh documentation. <https://documentation.wazuh.com/current/user-manual/wazuh-indexer/index.html>

Technical basis

Wazuh stores generated alerts locally and forwards indexed data for Dashboard queries (Wazuh, Inc., n.d.). The Wazuh `client.keys` file authenticates and identifies enrolled agents, so it is explicitly protected (Wazuh, Inc., n.d.). Suricata supports reopening log files around rotation; this guide instead uses a controlled stop-delete-restart sequence for a disposable lab reset (Open Information Security Foundation, n.d.). Microsoft documents `wevtutil clear-log` as the supported command for clearing an event channel (Microsoft, 2024).

Final rule. A cleanup is successful only when old telemetry is gone, configuration hashes are unchanged, agents remain enrolled, and fresh telemetry reaches Wazuh.