HEXWORTH PRIME · INSTRUCTOR SOLUTION GUIDE

ALA Scavenger Hunt #2: The Perimeter Is Open

Solution Guide

COURSE	**CTS4321C Advanced Linux Administration**
WEEK	**2**
FORMAT	**In-class team race + printed worksheet**
DURATION	**20 minutes (target)**
FLAGS	**8 (one per worksheet row)**
DIFFICULTY	**Beginner-Intermediate**
POINTS	**400 (50 per flag)**
WORKSHEET	**`_app/houses/matrix/handouts/scavengerHunt-perimeter-open.docx`**

Objective

A post-breach hardening exercise on cell-071. Each worksheet row maps to one lab flag. The student runs the correct Linux command in the simulated terminal, the engine detects the command pattern, and `awardFlag('cmdN')` fires. The student writes the command on the printed worksheet for the matching row.

The activity is an **in-class team race**:

Team rank	Score
First team to capture all 8 flags AND fill the worksheet correctly	100/100
Other teams who finish correctly	95/100
Each wrong sheet answer	−2 pts

The box does not enforce ordering — students may capture flags in any sequence. The worksheet rows are numbered in hardening-sequence order to guide the natural flow.

Starting State

Cell-071 after a partial intrusion:

- Firewall INPUT chain policy: `ACCEPT` (no rules — wide open)
- Rogue process: `python3 /tmp/cell-implant.py` listening on port 8888 (PID 3142)
- SSH daemon config: `PermitRootLogin yes` (default/insecure)
- Brute-forced account: `rogue-ops` — 5 failures from 203.0.113.44, currently locked
- Malware: `/tmp/cell-implant.py` (Trojan.ShellScript-7) and `/home/rogue-ops/.local/share/.beacon` (PUA.Tool.NetCat-3)
- Package tamper: `dpkg -V openssh-server` shows MD5 mismatch on `/usr/sbin/sshd`

Login banner asserts BREACHED state and instructs students to open `~/notes.txt`.

Flag-by-Flag Walkthrough

Flag 1 (cmd01) — Firewall status

Worksheet row: "Inspect the current firewall rules on cell-071. What is the default policy on the INPUT chain?"

Command:

```
sudo iptables -L -n -v
```

Expected output:

```
Chain INPUT (policy ACCEPT)
target   prot opt source   destination
(no rules listed)
```

`policy ACCEPT` means all inbound traffic is permitted by default — the firewall is wide open. This is the condition the exercise is hardening against.

Flag value: `FLAG{ala-hunt2_cmd01_firewall_status}`

Flag 2 (cmd02) — Default-deny policy

Worksheet row: "Apply a default-deny policy so all NEW inbound connections are blocked. Add the established-sessions rule first."

Command (accept either form):

```
sudo iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
sudo iptables -P INPUT DROP
```

```
# UFW alternative:
sudo ufw default deny incoming
```

Expected output (iptables -L -n -v after applying):

```
Chain INPUT (policy DROP)
target     prot opt source    destination
ACCEPT     all  --  anywhere  anywhere  state ESTABLISHED,RELATED
```

Critical teaching point: The ESTABLISHED,RELATED rule must be added *before* setting `-P INPUT DROP`. Reversing the order kills the operator's own SSH session.

Flag value: `FLAG{ala-hunt2_cmd02_default_deny}`

Flag 3 (cmd03) — Allow SSH

Worksheet row: "Explicitly allow new SSH connections on port 22 so future logins are possible."

Command (accept either form):

```
sudo iptables -A INPUT -p tcp --dport 22 -m state --state NEW -j ACCEPT
```

```
# UFW alternative:
sudo ufw allow 22/tcp
```

Expected output (iptables -L -n -v after applying):

```
Chain INPUT (policy DROP)
target  prot opt  source    destination
ACCEPT  all  --   anywhere  anywhere  state ESTABLISHED,RELATED
ACCEPT  tcp  --   anywhere  anywhere  tcp dpt:ssh state NEW
```

Flag value: `FLAG{ala-hunt2_cmd03_allow_ssh}`

Flag 4 (cmd04) — Rogue port detection

Worksheet row: "Identify every service listening on a TCP port. There is a rogue process running on an unexpected port — which port is it?"

Command:

```
ss -tlnp
```

Expected output:

```
State   Recv-Q  Send-Q  Local Address:Port  Peer Address:Port  Process
LISTEN  0       128     0.0.0.0:22          0.0.0.0:*     users:(("sshd",pid=847,fd=3))
LISTEN  0       5       0.0.0.0:8888        0.0.0.0:*     users:
(("python3",pid=3142,fd=4))
```

Rogue port: **8888**. Process: `python3 /tmp/cell-implant.py` (PID 3142).

Flag value: `FLAG{ala-hunt2_cmd04_rogue_port}`

Flag 5 (cmd05) — Disable root SSH login

Worksheet row: "Disable root SSH login in the SSH daemon configuration, syntax-check the config, then apply the change without dropping active connections."

Command:

```
# Edit /etc/ssh/sshd_config — set:
#   PermitRootLogin no
sudo sed -i 's/^#*PermitRootLogin.*/PermitRootLogin no/' /etc/ssh/sshd_config
sshd -t && sudo systemctl reload sshd
```

Expected output:

```
(no output from sshd -t means the config is syntactically valid)

grep PermitRootLogin /etc/ssh/sshd_config
PermitRootLogin no
```

`sshd -t` (test mode) must return exit code 0 before `systemctl reload` runs. `systemctl reload sshd` performs a graceful reload — active sessions survive; new connections immediately use the hardened config.

Flag value: `FLAG{ala-hunt2_cmd05_harden_ssh}`

Flag 6 (cmd06) — Brute-force audit

Worksheet row: "Identify the account that was targeted by a brute-force attack and check how many login failures it has recorded."

Command:

```
faillock --user rogue-ops
```

```
# Also useful:
lastb | head -20
```

Expected output:

```
rogue-ops:
When                Type    Source          Valid
2026-04-10 23:44:01 RHOST     203.0.113.44    V
2026-04-10 23:44:14 RHOST     203.0.113.44    V
2026-04-10 23:44:28 RHOST     203.0.113.44    V
2026-04-10 23:44:41 RHOST     203.0.113.44    V
2026-04-10 23:44:51 RHOST     203.0.113.44    V
```

Five failures from `203.0.113.44` — account is currently locked per `deny=5` in `/etc/security/faillock.conf`. The source IP is in the RFC 5737 documentation range (TEST-NET-3) used throughout the ALA CTF engine.

Flag value: `FLAG{ala-hunt2_cmd06_lock_account}`

Flag 7 (cmd07) — AV sweep and quarantine

Worksheet row: "Scan /tmp and /home recursively for malware and move any infected files to quarantine."

Command:

```
sudo clamscan -r -i --move=/var/quarantine /tmp /home
```

Expected output:

```
/tmp/cell-implant.py: Trojan.ShellScript-7 FOUND
/home/rogue-ops/.local/share/.beacon: PUA.Tool.NetCat-3 FOUND

----------- SCAN SUMMARY -----------
Infected files: 2
Time: 4.321 sec (2 file(s) moved to /var/quarantine/)
```

`-r` = recursive, `-i` = infected files only, `--move` = quarantine (do not delete — preserve evidence for forensic analysis).

Flag value: `FLAG{ala-hunt2_cmd07_av_scan}`

Flag 8 (cmd08) — Package integrity verification

Worksheet row: "Verify the integrity of the openssh-server package against its stored checksums. What does the output tell you?"

Command:

```
dpkg -V openssh-server
```

Expected output:

```
??5??????   /usr/sbin/sshd
```

What the output means:

Column	Character	Meaning
5	`5`	MD5 checksum mismatch — the file's actual hash does not match the hash recorded at install time

`/usr/sbin/sshd` is an executable binary, not a configuration file, so `dpkg -V` does not print a `c` type flag. The nine-character field shows `??5??????` followed by spaces (no type letter), then the path. A `5` in position 3 means the installed `/usr/sbin/sshd` binary does not match the checksum recorded at package install time — the binary was modified or replaced by the attacker. Remediation:

```
sudo apt install --reinstall openssh-server
dpkg -V openssh-server   # verify clean — should produce no output
```

Flag value: `FLAG{ala-hunt2_cmd08_dpkg_verify}`

Order Dependency

All 8 flags are independent with one practical constraint:

- **cmd02 before cmd03** is the *correct operational sequence* (add ESTABLISHED rule, set DROP, then whitelist SSH). The engine awards both flags regardless of order, but the worksheet description explicitly states the safe sequence. A team that sets DROP before the ESTABLISHED rule will lose their SSH session in the real world — use this as a teachable moment if it happens.

Worksheet Integration

The printed worksheet has 8 numbered rows + 8 flag-value spaces:

1. **Row N** — student writes the command they ran
2. **Flag table** — student copies the captured `FLAG{...}` value

Auto-captured flags are visible in the CTF box flag panel (cmd01.txt through cmd08.txt, turning grey to green as each is captured).

Scoring (instructor side):

Item	Points
Each correct worksheet command (8 rows)	12.5 pts
Each wrong answer	−2 pts
All 8 lab flags captured	Required for correct completion
First team to finish correctly	100/100
Other correct teams	95/100

Common Student Mistakes

- **Running `iptables -P INPUT DROP` first.** Drops the SSH session immediately. Engine warns ("WARNING: add ESTABLISHED,RELATED ACCEPT rule before applying DROP"), but a student who ignores it must reconnect.
- **Running `ss -tn` or `ss -tnp` (no -l).** Shows established connections, not listening sockets. `-l` is required to surface the rogue port 8888.

- **Skipping `sshd -t` before reload.** A syntax error in sshd_config combined with `systemctl reload` may kill SSH acceptance without showing an obvious error.
- **Running `clamscan` without `--move`.** `clamscan -r -i` reports infections but leaves files in place; the engine requires the `--move=/var/quarantine` flag.
- **Reading `dpkg -V` output as an error.** Students often assume no output = clean. Clarify: `dpkg -V` is silent when everything is correct; output = problem found.

Teaching Notes

- **Primary concept:** The W2 hardening workflow — firewall, auth hardening, AV, and package integrity — compressed into a single incident scenario.
- **Why a race:** Forces students to *actually run* the W2 commands rather than identifying them on a multiple-choice quiz.
- **The ordering puzzle (cmd02/cmd03):** The ESTABLISHED,RELATED-before-DROP rule is the single highest-stakes sequencing error a new admin can make. Having students discover it in a consequence-free lab environment is the point.
- **Rogue port 8888:** Chosen deliberately (not 443 as in W1) so students practice reading `ss -tlnp` output critically rather than pattern-matching a well-known port.
- **dpkg -V:** Most students have never used this command. The tampered-sshd scenario makes the result visceral — they see why package integrity matters.
- **Box engine:** Forked from the W2 Lockdown Protocol lab. The 8 CTF flag triggers are new; the firewall/sshd/clamav/dpkg simulation machinery is inherited intact.

Validator Checklist

- [x] **BOX-002a** Walkthrough exists at `~/hexworth-shared/Solutions/Advanced Linux Administration/ALA-Hunt2-Perimeter-Open-SOLUTION.md`
- [x] **BOX-002b** Walkthrough cites all 8 flag values verbatim
- [x] **BOX-002c** Flag values use `ala-hunt2_` prefix throughout (no collision with Hunt 1 `ala-hunt1_` prefix)
- [x] **BOX-024** No duplicate flag values across boxes

Related Artifacts

- **Lab config:** `_app/houses/matrix/adv-linux/labs/ala-hunt2-perimeter-open/config.js`
- **Lab index:** `_app/houses/matrix/adv-linux/labs/ala-hunt2-perimeter-open/index.html`
- **Hub wiring:** `_app/houses/matrix/adv-linux/index.html` (Week 2 card -> In-class Activity subsection)
- **Worksheet (printable):** `_app/houses/matrix/handouts/scavengerHunt-perimeter-open.docx`
- **Lecture deck reference:** W2 topic decks (Firewalls, Auth Hardening, AV, Package Management)
- **Operator Field Manual:** `~/hexworth-shared/Raw sources/ALA/Matrix_House_ALA_Week2_Operator_Field_Manual.md`
- **W1 solution (parallel reference):** `~/hexworth-shared/Solutions/Advanced Linux Administration/ALA-Hunt1-Website-Down-SOLUTION.md`

Last updated: 2026-06-15