HEXWORTH PRIME · INSTRUCTOR SOLUTION GUIDE

ALA Scavenger Hunt #3: Lost Authority

Solution Guide

COURSE	**CTS4321C Advanced Linux Administration**
WEEK	**3**
FORMAT	**In-class team race + printed worksheet**
DURATION	**20 minutes (target)**
FLAGS	**7 (one per worksheet row)**
DIFFICULTY	**Beginner**
POINTS	**350 (50 per flag, base 0)**
WORKSHEET	**`_app/houses/matrix/handouts/scavengerHunt-lost-authority.docx`**

Objective

A DNS incident response exercise on cell-071. An attacker gained root via a stolen SSH key (203.0.113.99, 02:10 UTC), broke the BIND zone for sector7.matrix.net, poisoned the grid-api A record, and installed a rogue cron job that re-poisons the zone every 5 minutes. Students diagnose the failure, repair BIND, expose the poison, find and neutralise the automation, and confirm authoritative resolution is restored.

Each worksheet row maps to one lab flag. The engine awards the flag automatically when the student runs the matching W3 command.

Team-race scoring

Team rank	Score
First team to capture all 7 flags AND fill the worksheet correctly	100/100
Other teams who finish correctly	95/100
Each wrong sheet answer	-2 pts

Max box score: 350 points (7 flags x 50 pts; base score = 0). Wrong-flag penalty: -25 pts. Hint penalty: -50 pts per hint used.

Starting State (sourced from config.js lore + filesystem)

- **named:** failed (systemd) -- zone parse error at 02:15 UTC, PID 1847 exited
- **Zone file:** `/etc/bind/zones/db.sector7.matrix.net` -- two attacker edits:
- Line 18: `grid-api IN A 203.0.113.99` (correct value: 10.0.1.50)
- Line 20: `www IN CNAME ops.sector7.matrix.net` (missing trailing dot -> out-of-zone; line 19 is a comment)
- **State bits:** `_syntaxFixed=false`, `_recordFixed=false`, `_cronNeutralized=false`, `_namedRunning=false`
- **Rogue cron:** `/etc/cron.d/grid-resync` (installed 02:13 UTC, fires every 5 min)
- **Rogue script:** `/usr/local/bin/zone-resync.sh` (re-poisons A record + increments serial)
- **Backup:** `/etc/bind/zones/db.sector7.matrix.net.bak` (clean pre-attack copy, serial 2026050901)
- **Attacker IP:** 203.0.113.99 (RFC 5737 TEST-NET-3 documentation range)
- **Correct grid-api IP:** 10.0.1.50

- **Incident date:** 2026-05-09

Flag-by-Flag Walkthrough

Flag 1 (cmd01) -- DNS Diagnose

Worksheet row: "Confirm that DNS resolution is broken on cell-071. Query the local nameserver for grid-api. What status do you get?"

Command:

```
dig A grid-api.sector7.matrix.net @127.0.0.1
```

Expected output (from config.js dig handler, _namedRunning=false branch):

```
; <<>> DiG 9.18.1 <<>> A grid-api.sector7.matrix.net @127.0.0.1
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 17382
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;grid-api.sector7.matrix.net.   IN A

;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Sat May 09 02:14:00 UTC 2026
;; MSG SIZE  rcvd: 45

Status: SERVFAIL -- named is not running or zone failed to load.
Hint: systemctl status named  then  named-checkconf -z
```

`status: SERVFAIL` with no answer section confirms the authoritative nameserver is not responding. The `aa` (authoritative answer) flag is absent, and the ANSWER count is 0. Follow-up: `systemctl status named` to confirm the service has exited.

Flag value: `FLAG{ala-hunt3_cmd01_dns_diagnose}`

Flag 2 (cmd02) -- BIND Checkconf

Worksheet row: "Run the BIND configuration and zone-file validator. Which file has an error, and what is the error on line 20?"

Command:

```
sudo named-checkconf -z
```

Expected output (from config.js named-checkconf handler, _syntaxFixed=false branch):

```
/etc/bind/named.conf: OK
/etc/bind/named.conf.options: OK
/etc/bind/named.conf.local: OK
zone sector7.matrix.net/IN: loading from master file /etc/bind/zones/
db.sector7.matrix.net failed: dns_rdata_fromtext: /etc/bind/zones/db.sector7.matrix.net:
20: out of zone data
zone sector7.matrix.net/IN: not loaded due to errors.
zone 1.0.10.in-addr.arpa/IN: loaded serial 2026050901

Error on line 20 of /etc/bind/zones/db.sector7.matrix.net:
  www  IN  CNAME  ops.sector7.matrix.net    <-- missing trailing dot!

Fix: change to ops.sector7.matrix.net.  (trailing dot = absolute FQDN)
Without the dot BIND appends the zone origin, producing out-of-zone data.
```

The `-z` flag instructs `named-checkconf` to also validate zone files, not just `named.conf`. Without `-z` the conf files pass silently. The diagnostic message `dns_rdata_fromtext: ...:20: out of zone data` pinpoints the exact line. Line 19 is the attacker's comment (`; ATTACKER SYNTAX ERROR: missing trailing dot on CNAME target`); the actual defective record is on line 20. The root cause: BIND CNAME targets require a fully-qualified domain name ending in `.`; without the dot, BIND appends the zone origin (`sector7.matrix.net`) producing the unresolvable name `ops.sector7.matrix.net.sector7.matrix.net` -- out-of-zone data.

Flag value: `FLAG{ala-hunt3_cmd02_checkconf}`

Flag 3 (cmd03) -- BIND Reload

Worksheet row: "After fixing the zone file, reload only the sector7.matrix.net zone. What does rndc report on success?"

Fix the CNAME syntax error first (the parse error that prevents named from starting). Note: you do NOT need to fix the A-record poison before reloading -- named will start with the poisoned record. The poison is a separate step (cmd04/cmd07).

```
# Option A: editor (nano or vim) -- fix line 20 (add trailing dot to CNAME target)
sudo nano /etc/bind/zones/db.sector7.matrix.net

# Option B: restore from backup (fixes BOTH the syntax error AND the A-record poison)
sudo cp /etc/bind/zones/db.sector7.matrix.net.bak /etc/bind/zones/db.sector7.matrix.net

# Option C: sed -- fix only the syntax error (A record stays poisoned)
sudo sed -i 's/ops.sector7.matrix.net$/ops.sector7.matrix.net./' /etc/bind/zones/
db.sector7.matrix.net
```

Then reload (the command that captures cmd03):

```
sudo rndc reload sector7.matrix.net
```

Expected output (from config.js rndc handler, _syntaxFixed=true, zone arg present):

```
zone reload queued
(named reloaded zone sector7.matrix.net -- serial 2026050902)
Verify: dig A grid-api.sector7.matrix.net @127.0.0.1 +short
```

`rndc reload <zone>` reloads a single zone without restarting the full daemon -- the correct production procedure. The engine gates cmd3 on `_syntaxFixed` only: once the CNAME parse error is fixed, named can load the zone and start answering queries -- even though the A record is still poisoned. Serial shown is `2026050902`, the value in the zone file at that point (the attacker's edit set it to `2026050902`; the `.bak` file has `2026050901`).

Flag value: `FLAG{ala-hunt3_cmd03_reload_verify}`

Flag 4 (cmd04) -- DNS Integrity

Worksheet row: "With BIND running again, query grid-api with +short. What IP address does the zone currently return?"

Command:

```
dig +short A grid-api.sector7.matrix.net @127.0.0.1
```

Expected output (from config.js dig handler, _namedRunning=true, !(_recordFixed && _cronNeutralized), +short branch):

```
203.0.113.99
```

The `+short` flag suppresses all header and section lines, returning only the answer. Named is now running and the syntax error is resolved (cmd03 fired), but the A record is still poisoned -- the attacker's IP 203.0.113.99 is being returned.

Persistence mechanic: Even if a student fixes the A record at this point (changing 203.0.113.99 back to 10.0.1.50 in the zone file), `dig +short` will continue to return `203.0.113.99` because the rogue cron job (`/etc/cron.d/grid-resync`) is still running and will re-overwrite the A record within 5 minutes. The engine models this: `dig` returns the attacker IP whenever `_namedRunning=true` AND `NOT (_recordFixed AND _cronNeutralized)` . cmd07 (the correct IP) only fires once BOTH the record is fixed AND the cron is removed.

Flag value: `FLAG{ala-hunt3_cmd04_poisoned_record}`

Flag 5 (cmd05) -- Automation

Worksheet row: "List the system cron drop-in directory. Which file should not be there?"

Command:

```
ls /etc/cron.d
```

Expected output (from config.js ls handler for /etc/cron.d, _cronNeutralized=false):

```
grid-backup  grid-resync
```

`grid-backup` is the legitimate daily backup job (pre-existing, installed at 02:30 on 2026-05-08). `grid-resync` was installed by the attacker at 02:13 UTC and is the rogue drop-in. The alternate command `cat /etc/cron.d/grid-resync` also awards this flag and reveals the full cron entry:

```
# /etc/cron.d/grid-resync -- ROGUE (installed by attacker 2026-05-09 02:13 UTC)
SHELL=/bin/bash
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
MAILTO=""

*/5 * * * *   root   /usr/local/bin/zone-resync.sh >> /var/log/grid-resync.log 2>&1
```

`*/5 * * * *` fires every 5 minutes. The job runs as `root`, giving the script full write access to the BIND zone directory.

Flag value: `FLAG{ala-hunt3_cmd05_rogue_cron}`

Flag 6 (cmd06) -- Bash/Automation

Worksheet row: "Read the malicious script that the rogue cron job runs. What IP address does the script re-poison the A record to?"

Command:

```
cat /usr/local/bin/zone-resync.sh
```

Expected output (from config.js cat handler -> filesystem zone-resync.sh content):

```
#!/usr/bin/env bash
# zone-resync.sh -- GRID ZONE SYNC UTILITY
# Dropped by attacker at 2026-05-09 02:13 UTC -- C2: 203.0.113.99
# Runs every 5 minutes via /etc/cron.d/grid-resync.
# Rewrites grid-api A record to attacker IP, increments serial, reloads zone.
set -euo pipefail
ZONEFILE="/etc/bind/zones/db.sector7.matrix.net"
ATTACKER_IP="203.0.113.99"
sed -i "s/^grid-api[[:space:]]*IN[[:space:]]*A.*/grid-api    IN  A       ${ATTACKER_IP}/"
"${ZONEFILE}"
SERIAL=$(awk '/serial/{print $1}' "${ZONEFILE}")
NEWSERIAL=$((SERIAL + 1))
sed -i "s/${SERIAL}[[:space:]]*; serial/${NEWSERIAL}   ; serial/" "${ZONEFILE}"
rndc reload sector7.matrix.net 2>/dev/null || true
```

The script performs three operations on every execution: (1) overwrites the `grid-api` A record with the attacker IP `203.0.113.99` using `sed -i`; (2) increments the zone serial number so resolvers treat the poisoned zone as authoritative; (3) calls `rndc reload` to load the re-poisoned zone. The serial increment is the critical detail -- without it, secondary nameservers and caching resolvers would not fetch the updated (poisoned) zone.

Remediation sequence: remove the cron job FIRST (stops re-poisoning), then fix the zone A record, then reload. The engine requires `_cronNeutralized=true` before `dig` returns the correct IP -- so removing the cron is mandatory, not optional.

```
sudo rm /etc/cron.d/grid-resync
# Fix zone file (correct the A record back to 10.0.1.50 if not already done)
sudo nano /etc/bind/zones/db.sector7.matrix.net
sudo rndc reload sector7.matrix.net
```

Flag value: `FLAG{ala-hunt3_cmd06_malicious_script}`

Flag 7 (cmd07) -- Verify

Worksheet row: "After removing the rogue cron job, correcting the poisoned A record, and reloading the zone, confirm resolution is restored. What IP does grid-api now return?"

Command:

```
dig +short A grid-api.sector7.matrix.net @127.0.0.1
```

Expected output (from config.js dig handler, _namedRunning=true, _recordFixed=true, _cronNeutralized=true, +short branch):

```
10.0.1.50
```

`10.0.1.50` is the correct IP for `grid-api.sector7.matrix.net` (per the pre-attack backup at `/etc/bind/zones/db.sector7.matrix.net.bak`). The full (non-short) dig output confirms the `aa` (authoritative answer) flag is set, verifying that cell-071 is again the authoritative nameserver for the zone:

```
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 2

;; ANSWER SECTION:
grid-api.sector7.matrix.net. 3600 IN A 10.0.1.50
```

Note: the same `dig +short` command was used for Flag 4 (cmd04). The engine distinguishes the two calls using three independent state bits: cmd04 fires when `_namedRunning=true` and `NOT (_recordFixed AND _cronNeutralized)`; cmd07 fires only when all three conditions are met: `_namedRunning=true`, `_recordFixed=true`, AND `_cronNeutralized=true`. Removing the cron is required -- the engine will not return `10.0.1.50` if the rogue job is still in place, even if the A record has been manually corrected.

Flag value: `FLAG{ala-hunt3_cmd07_restored}`

Order Dependency

The seven flags follow the natural DNS incident-response sequence:

Step	Dependency
cmd01	None -- diagnose first
cmd02	None -- can run before or after cmd01
cmd03	Requires `_syntaxFixed=true` (CNAME trailing-dot error corrected) -- A-record fix NOT required
cmd04	Requires `_namedRunning=true` (cmd03 fired or `systemctl restart named` ran)
cmd05	None -- can run at any time
cmd06	None -- can run at any time
cmd07	Requires `_namedRunning=true` AND `_recordFixed=true` AND `_cronNeutralized=true`

Critical sequence for cmd07: cmd04 (sees 203.0.113.99) must come BEFORE cmd07. The persistence mechanic ensures cmd04 stays reachable: even after the A record is manually corrected, `dig` still returns `203.0.113.99` while the rogue cron is present (`_cronNeutralized=false`). Removing the cron is what unlocks cmd07.

The box awards each flag as soon as the triggering command runs, regardless of worksheet order. The numbered sequence on the worksheet reflects operational best practice, not an enforced engine constraint.

Worksheet Integration

The printed worksheet has 7 numbered rows + 7 flag-value spaces:

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

Auto-captured flags appear in the CTF box flag panel (cmd01 through cmd07, turning from grey to green as each is captured).

Scoring (instructor side)

Item	Points
Each correct worksheet command (7 rows)	50 pts
Max box score (7 flags x 50 pts)	350 pts
Wrong flag entry	-25 pts
Hint used	-50 pts per hint
First team to finish correctly	100/100 sheet grade
Other correct teams	95/100 sheet grade
Each wrong sheet answer	-2 pts

Common Student Mistakes

- **Running `rndc reload` before fixing the CNAME syntax error.** Returns `rndc: 'reload' failed: out of zone data`. The engine blocks cmd03 until `_syntaxFixed=true`. Fix: add the trailing dot to the `www CNAME` target on line 20.
- **Fixing only the CNAME error and expecting `dig` to return the correct IP.** `named-checkconf -z` passes and named starts (cmd03 fires), but `dig +short` returns `203.0.113.99` (cmd04). The A-record poison is a separate, second attacker edit -- a semantic change with no parse error.
- **Fixing the A record but not removing the cron job.** `dig +short` still returns `203.0.113.99` because the engine keeps cmd4 active while `_cronNeutralized=false`, modeling the real-world behavior: the cron job would re-overwrite the record within 5 minutes. **`sudo rm /etc/cron.d/grid-resync` is required before cmd07 fires.**
- **Using `crontab -l` instead of `ls /etc/cron.d`.** `crontab -l` lists the operator's per-user crontab (empty). System-level drop-in files live in `/etc/cron.d/` and are invisible to `crontab -l`. The box engine includes a correct `crontab -l` response ("no scheduled jobs for this user") to reinforce this.
- **Forgetting `sudo` on `named-checkconf`, `rndc`, and `rm`.** The sudoers policy at `/etc/sudoers.d/operator` grants passwordless sudo for these specific commands. Students who omit `sudo` receive `permission denied`.

Teaching Notes

- **Primary concept:** BIND zone integrity + incident response. The W3 DNS labs (L07 Name Authority, L09 Poisoned Records) are the direct predecessors of this hunt.
- **Why the two-bug design:** A single bug (just the parse error, or just the A-record poison) would be solvable with one command. The combination models a real attacker who deliberately introduces a crash to hide the semantic poison: a distracted operator fixes the crash, sees names resolving, and misses that traffic is going to the wrong IP.
- **The rogue cron persistence:** Forces students to think beyond the immediate fix. Removing the cron job is a separate step from fixing the zone. This mirrors real incident response where multiple persistence mechanisms must be identified and removed.
- **The serial increment:** The attacker script increments the SOA serial. Students who read the script closely see that even a fixed zone will appear "stale" to secondaries until they do a fresh `rndc reload` or allow the refresh cycle to run.
- **LPIC-1 alignment:** cmd01/cmd04/cmd07 map to objective 108.4 (Manage DNS client); cmd02/cmd03 map to 108.1 (Maintain DNS zones); cmd05 maps to 107.2 (Automate system administration tasks); cmd06 maps to 105.1 (Customize and use the shell environment).

Validator Checklist

- [x] **BOX-002a** Walkthrough exists at `~/hexworth-shared/Solutions/Advanced Linux Administration/ALA-Hunt3-Lost-Authority-SOLUTION.md`
- [x] **BOX-002b** Walkthrough cites all 7 flag values verbatim (sourced from config.js flags[])
- [x] **BOX-002c** Flag values use `ala-hunt3_` prefix throughout (no collision with Hunt 1 `ala-hunt1_` or Hunt 2 `ala-hunt2_` prefix)
- [x] **BOX-002d** Points stated as 350 (7 x 50, base 0) matching config.js scoring block
- [x] **BOX-024** No duplicate flag values across boxes

Related Artifacts

- **Lab config:** `_app/houses/matrix/adv-linux/labs/ala-hunt3-lost-authority/config.js`
- **Lab index:** `_app/houses/matrix/adv-linux/labs/ala-hunt3-lost-authority/index.html`
- **Worksheet (printable):** `_app/houses/matrix/handouts/scavengerHunt-lost-authority.docx`
- **Worksheet (PDF):** `_app/houses/matrix/handouts/scavengerHunt-lost-authority.pdf`

- **Operator Field Manual:** `~/hexworth-shared/Raw sources/ALA/Matrix_House_ALA_Week3_Operator_Field_Manual.md`
- **W3 Lab 07 solution:** `~/hexworth-shared/Solutions/Advanced Linux Administration/ALA-L07-Name-Authority-SOLUTION.md`
- **W3 Lab 09 solution:** `~/hexworth-shared/Solutions/Advanced Linux Administration/ALA-L09-Poisoned-Records-SOLUTION.md`
- **W2 hunt solution (parallel reference):** `~/hexworth-shared/Solutions/Advanced Linux Administration/ALA-Hunt2-Perimeter-Open-SOLUTION.md`

Last updated: 2026-06-15