HEXWORTH PRIME · MATRIX HOUSE · OPERATOR FIELD MANUAL

Week 3 Operator Field Manual

DNS & Automation | DNS Fundamentals | BIND9 | Bash Scripting | Scheduling

COURSE	**CTS4321C Advanced Linux Administration**
WEEK	**3**
SUBTITLE	**DNS & Automation | DNS Fundamentals | BIND9 | Bash Scripting | Scheduling**

OPERATOR QUICK REFERENCE — CTS4321C Advanced Linux Administration. Contains terse command tables, zone file skeletons, and incident playbooks for L07 (Name Authority), L08 (The Night Shift), and L09 (Poisoned Records). YOUR CELL IS YOUR RESPONSIBILITY.

WEEK 3 OPERATOR QUICK REFERENCE

DNS & Automation | Bash Scripting | Scheduling — YOUR CELL IS YOUR RESPONSIBILITY

The rule: one command answers one question. Use these workflows to keep names resolving, jobs running, and zones intact.

DNS FAILURE FLOW

Layer	Question	Tool
Local cache	Is the OS cache stale?	`resolvectl flush-caches`
Local resolver	Does `@127.0.0.1` answer?	`dig A name @127.0.0.1`
External resolver	Does `@8.8.8.8` answer?	`dig A name @8.8.8.8`
Authority	Who owns the zone?	`dig NS domain +short`
Full chain	Where does it break?	`dig +trace name`
Service	Is named running?	`systemctl status named`

MATRIX HOUSE OPERATOR RULES

- Validate before reload: `named-checkconf -z` then `rndc reload`
- Never edit zone files without incrementing the serial
- Cron scripts need absolute paths — always set PATH at crontab top
- Set `set -euo pipefail` on every bash script, no exceptions
- `allow-update { none; }` on every static zone

First Commands at Lab Start

Command	Use
`systemctl status named`	BIND service state
`dig A name @127.0.0.1`	Test local authoritative answer
`named-checkconf -z`	Validate all zones before reload
`crontab -l`	Review scheduled jobs
`systemctl list-timers --all`	Show all systemd timer next-fires

Operator Quick-Reference: Problem to Command

Problem	First Question	Command
Name does not resolve	Resolver or zone?	`dig +trace name`
BIND not answering	Service up? Port 53?	`systemctl status named` / `ss -tulnp | grep :53`
Zone not loading	Syntax error?	`named-checkconf -z`
Secondary stale	Serial higher? Transfer OK?	`dig SOA zone @ns2` / `rndc retransfer zone`
Cron job not firing	PATH or permission?	`env -i PATH=/usr/bin:/bin script.sh`
systemd timer missed	Persistent set?	`systemctl status timer` / `journalctl -u service`
Cache poisoned	Zone matches backup?	`diff db.zone db.zone.bak`
Dynamic update blocked	allow-update set?	`grep allow-update /etc/bind/named.conf.local`

> *Week 3 Goal: names resolve authoritatively, jobs run reliably, zones stay clean.*

DNS FUNDAMENTALS

Hierarchy | Records | Resolution — YOUR CELL IS YOUR RESPONSIBILITY

A name is just a pointer. Everything on the network runs on IP. DNS is the translator.

DNS Record Type Quick Reference

Type	Purpose	Example
A	Name → IPv4 address	`ops IN A 10.0.1.50`
AAAA	Name → IPv6 address	`ops IN AAAA 2001:db8::50`
CNAME	Alias → canonical name	`www IN CNAME ops.matrix.internal.`
MX	Domain → mail server hostname	`@ IN MX 10 mail.matrix.internal.`
NS	Zone → authoritative nameservers	`@ IN NS ns1.matrix.internal.`
SOA	Zone metadata (serial, timing)	`@ IN SOA ns1 admin (2026040901 ...)`
PTR	IP → name (reverse DNS)	`50 IN PTR ops.matrix.internal.`
TXT	Arbitrary text (SPF, DKIM, ownership)	`@ IN TXT "v=spf1 mx -all"`

Resolution Path (recursive resolver, cache-miss walk)

```
1  Client calls getaddrinfo()
2  OS checks /etc/hosts                --> match = done
3  OS sends recursive query to /etc/resolv.conf nameserver
4  Resolver cache hit                  --> returns with remaining TTL
5  Cache miss: resolver queries root   --> referral to .com TLD servers
6  Resolver queries TLD                --> referral to authoritative NS
7  Resolver queries authoritative NS   --> answer (aa flag set)
8  Resolver caches answer for TTL; client receives it
```

Key Files

File	Purpose
`/etc/resolv.conf`	List of resolver IPs and search domains (managed by systemd-resolved or netplan)
`/etc/hosts`	Static local overrides — checked before DNS
`/etc/nsswitch.conf`	Order of name resolution methods (`hosts: files dns`)

dig Command Reference

Need	Command
Quick answer	`dig +short A ops.matrix.internal`
Full response	`dig A ops.matrix.internal`
Trace full chain	`dig +trace ops.matrix.internal`
Ask specific resolver	`dig @8.8.8.8 A ops.matrix.internal`
Reverse lookup	`dig -x 10.0.1.50`
SOA / serial check	`dig SOA matrix.internal +multiline`
Zone NS delegation	`dig NS matrix.internal`
MX records	`dig MX matrix.internal +short`
TXT records	`dig TXT matrix.internal`
Authority flag check	look for `aa` in flags line

DNS Operator Rules

- `aa` flag in `dig` output = authoritative answer; no `aa` = cached or recursive
- Missing trailing dot in zone file = relative name, BIND appends zone origin = silent misdelegation
- CNAME cannot coexist with any other record type at the same name
- CNAME cannot be placed at the zone apex (`@`)
- MX record target must be a hostname (A record), not an IP address or CNAME

BIND9 DEPLOYMENT

Install | Configure | Validate | Reload — YOUR CELL IS YOUR RESPONSIBILITY

BIND9 is the reference DNS implementation. Every configuration decision has a reason.

Install and First Checks

```
sudo apt update && sudo apt install -y bind9 bind9utils bind9-doc

# Disable systemd-resolved stub listener (conflicts on port 53)
sudo sed -i 's/#DNSStubListener=yes/DNSStubListener=no/' /etc/systemd/resolved.conf
sudo systemctl restart systemd-resolved

sudo systemctl enable --now named
sudo ss -tulnp | grep :53          # named must own port 53, not stub
```

File Layout

Path	Purpose
`/etc/bind/named.conf`	Root config — includes the three below
`/etc/bind/named.conf.options`	Global options: recursion, forwarders, ACLs
`/etc/bind/named.conf.local`	Zone declarations
`/etc/bind/named.conf.default-zones`	Root hints + localhost zones (do not edit)
`/etc/bind/db.*`	Zone files (forward and reverse)
`/var/cache/bind/`	Journal files, secondary zone data
`/var/run/named/`	PID file, control socket

named.conf.options — Authoritative-Only Server

```
// /etc/bind/named.conf.options
acl "trusted" { 127.0.0.1; 10.0.0.0/8; };

options {
    directory "/var/cache/bind";
    recursion no;                          // authoritative-only
    allow-query { trusted; };
    allow-transfer { none; };              // default deny; override per zone
    dnssec-validation auto;
    listen-on { any; };
};
```

named.conf.local — Zone Declarations

```
// Forward zone
zone "matrix.internal" {
    type master;
    file "/etc/bind/db.matrix.internal";
    allow-transfer { 10.0.1.11; };        // allow ns2 to transfer
    allow-update { none; };               // block dynamic updates
    notify yes;
};

// Reverse zone for 10.0.1.0/24
zone "1.0.10.in-addr.arpa" {
    type master;
    file "/etc/bind/db.10.0.1";
    allow-transfer { 10.0.1.11; };
```

```
    allow-update { none; };
};

// Secondary (on ns2)
zone "matrix.internal" {
    type slave;
    masters { 10.0.1.10; };
    file "/var/cache/bind/db.matrix.internal";
};
```

Forward Zone File Skeleton

```
; /etc/bind/db.matrix.internal
$TTL    3600
@   IN  SOA  ns1.matrix.internal.  admin.matrix.internal. (
               2026040901   ; serial -- YYYYMMDDNN; increment on every change
               3600         ; refresh
               900          ; retry
               604800       ; expire
               300          ; minimum TTL (negative cache)
           )

@           IN  NS      ns1.matrix.internal.
@           IN  NS      ns2.matrix.internal.
ns1         IN  A       10.0.1.10
ns2         IN  A       10.0.1.11
ops         IN  A       10.0.1.50
www         IN  CNAME   ops.matrix.internal.    ; trailing dot = absolute FQDN
@           IN  MX  10  mail.matrix.internal.
mail        IN  A       10.0.1.70
```

Reverse Zone File Skeleton

```
; /etc/bind/db.10.0.1  (zone: 1.0.10.in-addr.arpa)
$TTL    3600
@   IN  SOA  ns1.matrix.internal.  admin.matrix.internal. (
               2026040901  3600  900  604800  300 )
@   IN  NS   ns1.matrix.internal.
@   IN  NS   ns2.matrix.internal.
; Only the last octet as the owner name
10  IN  PTR  ns1.matrix.internal.
11  IN  PTR  ns2.matrix.internal.
50  IN  PTR  ops.matrix.internal.
70  IN  PTR  mail.matrix.internal.
```

Validate → Reload Workflow (mandatory every time)

```
sudo named-checkconf                              # validate named.conf
sudo named-checkconf -z                           # validate all zone files too
sudo named-checkzone matrix.internal /etc/bind/db.matrix.internal
# Expected: zone matrix.internal/IN: loaded serial 2026040901 / OK

sudo rndc reload                                  # reload all zones
sudo rndc reload matrix.internal                  # reload specific zone only
```

rndc Command Reference

Command	Purpose
`rndc status`	Uptime, version, zone count

Command	Purpose
`rndc reload`	Reload all zones and config
`rndc reload ZONE`	Reload one zone
`rndc reconfig`	Re-read named.conf (no zone reload)
`rndc flush`	Clear entire resolver cache
`rndc flushname NAME`	Remove one name from cache
`rndc retransfer ZONE`	Force full AXFR on secondary
`rndc freeze ZONE`	Suspend dynamic updates for manual edit
`rndc thaw ZONE`	Resume dynamic updates
`rndc querylog on/off`	Enable/disable per-query logging
`rndc zonestatus ZONE`	Detailed zone health

Post-Change Verification Queries

```
dig A ops.matrix.internal @127.0.0.1         # forward -- expect aa flag, correct IP
dig -x 10.0.1.50 @127.0.0.1                  # reverse -- expect ops.matrix.internal.
dig SOA matrix.internal @127.0.0.1 +multiline # serial must match zone file
dig A ghost.matrix.internal @127.0.0.1        # NXDOMAIN with aa flag = correct behavior
```

BIND9 Hardening Checklist

- `recursion no` on authoritative-only servers
- `allow-query { trusted; }` — not `any`
- `allow-transfer { none; }` globally, override per zone with specific IPs
- `allow-update { none; }` on all static zones
- Zone files owned `bind:bind`, mode `640`
- TSIG key for authenticated zone transfers: `tsig-keygen zone-xfer`

BASH SCRIPTING

Variables | Control Flow | Functions | Exit Codes — YOUR CELL IS YOUR RESPONSIBILITY

Every repeatable action gets scripted. Every script is safe under failure.

Script Structure

```
#!/usr/bin/env bash
# script-name.sh  --  one-line description
# Usage: ./script-name.sh [OPTIONS] <argument>
set -euo pipefail              # exit on error | unset var = error | pipe fail
IFS=$'\n\t'                    # safe word splitting

readonly SCRIPT_DIR="$(cd "$(dirname "${BASH_SOURCE[0]}")" && pwd)"
LOG="/var/log/sector/$(basename "$0" .sh).log"

log() { echo "[$(date +%T)] $*" | tee -a "$LOG"; }

# -- main logic below --
```

Variable Cheat Sheet

Syntax	Meaning
`NAME="value"`	Local variable
`export NAME="value"`	Available to child processes

Syntax	Meaning
`readonly NAME=value`	Cannot be reassigned
`declare -i COUNT=0`	Integer type
`"${VAR:-default}"`	Use default if VAR is unset or empty
`"${VAR:?'message'}"`	Abort with message if VAR is unset
`"$1"` `"$2"`	Positional arguments
`"$@"`	All arguments as separate words (safe)
`$#`	Argument count
`$?`	Exit code of last command
`$$`	PID of current shell (unique temp file suffix)
`$!`	PID of last background process

Conditionals

```
# Prefer [[ ]] over [ ] in bash -- no word-splitting traps
if [[ "$USER" == "root" ]]; then echo "root"; fi
if [[ -z "$VAR" ]]; then echo "empty"; fi      # -z = zero-length
if [[ -n "$VAR" ]]; then echo "set"; fi        # -n = non-zero-length
if [[ -f "$FILE" ]]; then echo "file exists"; fi
if [[ -d "$DIR" ]]; then echo "dir exists"; fi
if [[ $COUNT -gt 4 ]]; then echo "high"; fi    # numeric: -eq -ne -lt -le -gt -ge
if [[ "$STR" =~ ^[0-9]+$ ]]; then echo "numeric"; fi  # regex

# case: cleaner than if/elif for discrete values
case "$action" in
    start)   systemctl start nginx ;;
    stop)    systemctl stop nginx ;;
    *)       echo "unknown: $action" >&2; exit 1 ;;
esac
```

Loops

```
# for: list of words
for host in web-01 web-02 db-01; do
    ping -c1 -W1 "$host" >/dev/null 2>&1 && echo "$host UP" || echo "$host DOWN"
done

# while: read file line-by-line (safe pattern)
while IFS= read -r line; do
    echo "Host: $line"
done < /etc/sector/hosts.txt

# while: retry with limit
RETRIES=0; MAX=5
while ! pg_isready -h db-01 >/dev/null 2>&1; do
    (( RETRIES++ )); [[ $RETRIES -ge $MAX ]] && { echo "DB unreachable" >&2; exit 1; }
    sleep 3
done

# C-style for: numbered iterations
for (( i=1; i<=5; i++ )); do
    curl -sf http://api.sector.local/health && break
    sleep 2
done
```

Functions

```
log_info()  { echo "[INFO]  $(date +%T) $*"; }
log_error() { echo "[ERROR] $(date +%T) $*" >&2; }

deploy_config() {
    local src="$1" dest="$2"
    local backup="${dest}.bak.$(date +%s)"
    [[ -f "$dest" ]] && cp "$dest" "$backup"
    cp "$src" "$dest"
    log_info "Deployed $src -> $dest"
}

get_primary_ip() {
    local iface="${1:-eth0}"
    ip -4 addr show "$iface" | awk '/inet /{gsub(/\/.*/, "", $2); print $2}'
}
```

Exit Codes and Error Handling

```
# Explicit exit codes (0 = success, 1-255 = failure)
[[ $(id -u) -eq 0 ]] || { echo "Must run as root" >&2; exit 1; }

# Cleanup on exit (runs even when set -e exits early)
TMPFILE="/tmp/work-$$"
trap 'rm -f "$TMPFILE"' EXIT

# Capture exit code explicitly without set -e aborting
ping -c1 "$HOST" >/dev/null 2>&1
status=$?
[[ $status -eq 0 ]] && echo "$HOST UP" || echo "$HOST DOWN"
```

Common One-Liners

```
# Remote SSH command loop over a host list
while IFS=' ' read -r name ip; do
    ssh "$name" "systemctl is-active sshd && df -h /var" 2>/dev/null
done < /home/operator/cell-list.txt

# Timed backup with datestamp
rsync -avz --delete /opt/data/ backup-node:/backup/$(date +%F)/

# Parse last-column field
awk '{print $NF}' file.txt

# Count occurrences and rank
sort file | uniq -c | sort -rn | head -10
```

AUTOMATION & SCHEDULING

Cron | systemd Timers | logrotate | rsync — YOUR CELL IS YOUR RESPONSIBILITY

Manual intervention is a liability. Automate it once, observe it forever.

Cron Syntax

```
# .---------- minute      (0-59)
# |  .-------- hour        (0-23)
# |  |  .----- day of month (1-31)
# |  |  |  .-- month       (1-12)
```

```
# |  |  |  |  . day of week (0-7, 0 and 7 = Sunday)
# |  |  |  |  |
# *  *  *  *  *  command
```

Schedule	Expression
Every day at 02:30	`30 2 * * *`
Every Monday at 04:00	`0 4 * * 1`
Every 15 minutes	`*/15 * * * *`
1st of month midnight	`0 0 1 * *`
Weekdays at 08:00	`0 8 * * 1-5`
06:00, 12:00, 18:00	`0 6,12,18 * * *`

Crontab Management

```
crontab -e                    # edit current user's crontab
crontab -l                    # list current user's crontab
crontab -u deploy -l          # list another user's crontab
crontab -r                    # remove all (destructive -- confirm first)
```

Crontab locations:

Location	Format	Use
`crontab -e` / `/var/spool/cron/crontabs/USER`	no username field	Per-user jobs
`/etc/cron.d/`	includes username field	System drop-in jobs
`/etc/cron.daily|weekly|monthly/`	executable scripts, no time syntax	Simple frequency tasks

Cron Environment Rules (most common source of failures)

```
# Top of every /etc/cron.d/ file or crontab:
SHELL=/bin/bash
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
MAILTO=ops@sector.local         # empty string to silence

# cron entry with output redirection
30 2 * * *  root  /usr/local/bin/backup.sh >> /var/log/backup.log 2>&1

# % in crontab is a newline -- escape it
0 5 * * *   /path/script.sh >> /var/log/out-$(date +\%F).txt 2>&1

# Debug: simulate cron environment manually
env -i HOME=/root PATH=/usr/bin:/bin SHELL=/bin/bash /usr/local/bin/script.sh
```

systemd Timer Unit Pair

```
# /etc/systemd/system/sector-backup.service
[Unit]
Description=Sector Node Backup
After=network-online.target

[Service]
Type=oneshot
User=backup
ExecStart=/usr/local/bin/sector-backup.sh
StandardOutput=journal
StandardError=journal
```

```
# /etc/systemd/system/sector-backup.timer
[Unit]
Description=Daily sector backup at 02:30

[Timer]
OnCalendar=*-*-* 02:30:00
Persistent=true              # catch missed runs after downtime
RandomizedDelaySec=15min     # stagger across nodes

[Install]
WantedBy=timers.target
```

```
systemctl daemon-reload
systemctl enable --now sector-backup.timer
systemctl list-timers sector-backup.timer    # confirm next trigger
journalctl -u sector-backup.service -n 50    # check last run output
```

systemd Calendar vs Cron

cron	systemd OnCalendar
`0 0 * * *`	`daily`
`0 0 * * 1`	`Mon *-*-* 00:00:00`
`*/15 * * * *`	`*:0/15`
`0 8 * * 1-5`	`Mon..Fri *-*-* 08:00`

```
# Validate calendar expression before deploying
systemd-analyze calendar '*-*-* 02:30:00'
systemd-analyze calendar 'Mon..Fri *-*-* 08:00'
```

at — One-Shot Scheduled Command

```
echo "/usr/local/bin/patch-rollout.sh" | at 03:00 tomorrow
echo "/usr/local/bin/reboot-node.sh"   | at 02:00 2026-04-15
atq                                     # list pending at jobs
atrm 3                                  # remove job number 3
```

rsync Backup Patterns

```
# Basic sync (push local to remote)
rsync -avz --delete /opt/sector/ backup-node:/mnt/backup/sector/

# Dry run first -- always before --delete
rsync -avn --delete /opt/sector/ backup-node:/mnt/backup/sector/

# Incremental snapshot with --link-dest (hardlink unchanged files)
TODAY=$(date +%F)
SNAPSHOT="/mnt/backup/snapshots/${TODAY}"
rsync -av --delete --link-dest="/mnt/backup/latest" \
    /opt/sector/ "${SNAPSHOT}/"
ln -sfn "${SNAPSHOT}" /mnt/backup/latest    # update symlink for next run

# Remote pull (backup from remote cell to local)
rsync -avz cell-14:/etc/ /var/backups/cell-14/$(date +%F)/etc/
```

logrotate Configuration

```
# /etc/logrotate.d/sector-app
/var/log/sector/app.log {
    daily
    rotate 14
    compress
    delaycompress
    missingok
    notifempty
    create 0640 sectorapp adm
    dateext
    sharedscripts
    postrotate
        systemctl reload sector-app 2>/dev/null || true
    endscript
}
```

```
logrotate --debug /etc/logrotate.d/sector-app    # dry-run test
logrotate --force /etc/logrotate.d/sector-app    # force rotation now
```

INCIDENT PLAYBOOKS

Terse step-by-step playbooks for the three Week 3 labs — YOUR CELL IS YOUR RESPONSIBILITY

Playbooks keep operators from guessing under pressure.

L07 — Name Authority: Stand Up BIND9 from Scratch

Goal: Forward zone + reverse zone + zone transfer on `sector7.matrix.net`

```
1   apt install -y bind9 bind9utils bind9-doc
2   Edit named.conf.options    -- listen-on, allow-query, recursion no
3   Edit named.conf.local      -- two zones (forward + 1.0.10.in-addr.arpa)
4   Write db.sector7.matrix.net  -- SOA NS A CNAME MX (trailing dots on FQDNs)
5   Write db.10.0.1            -- SOA NS PTR (last-octet owner names only)
6   named-checkconf && named-checkzone sector7.matrix.net /etc/bind/zones/db.sector7.matrix.net
7   named-checkzone 1.0.10.in-addr.arpa /etc/bind/zones/db.10.0.1
8   systemctl start named
9   dig A cell-071.sector7.matrix.net @10.0.1.1   -- expect 10.0.1.71, aa flag
10  dig -x 10.0.1.71 @10.0.1.1                    -- expect cell-071.sector7.matrix.net.
11  /opt/verify/check-forward.sh  (FLAG 1)
12  /opt/verify/check-reverse.sh  (FLAG 2)
13  /opt/verify/check-transfer.sh -- requires allow-transfer { 10.0.1.2; }; (FLAG 3)
```

Top traps: missing trailing dot on CNAME target | `listen-on` missing server IP | PTR uses last octet only | serial must match in both zones

L08 — The Night Shift: Bash Automation + Cron

Goal: Three scripts on `cell-night-ops` scheduled at 05:00 daily

```
1   cat ~/cell-list.txt       -- cell-14 10.0.1.14 / cell-27 10.0.1.27 / cell-33 10.0.1.33
2   Write /opt/cell-services/scripts/rotate-logs.sh
      while IFS=' ' read -r name ip; do
          ssh "$name" bash <<'REMOTE'
              cp /var/log/cell-ops/ops.log /var/log/archive/ops-$(date +%F).log
              gzip /var/log/archive/ops-$(date +%F).log
              > /var/log/cell-ops/ops.log
          REMOTE
      done < /home/operator/cell-list.txt
3   chmod +x rotate-logs.sh && ./rotate-logs.sh
```

```
4   /opt/verify/check-rotation.sh  (FLAG 1)
5   Write backup.sh
      rsync -avz --link-dest="${LATEST}" cell-14:/etc/ /var/backups/cell-14/$(date +%F)/etc/
      ln -sfn "${TODAY_DIR}" "${LATEST_LINK}"
6   chmod +x backup.sh && ./backup.sh
7   /opt/verify/check-backup.sh  (FLAG 2)
8   Write health-check.sh  -- ssh per cell: sshd status, df /var, ping gateway
      exit code = 0 if all PASS, 1 if any FAIL
9   chmod +x health-check.sh && ./health-check.sh
10  addcron 0 5 * * * /opt/cell-services/scripts/rotate-logs.sh
    addcron 0 5 * * * /opt/cell-services/scripts/backup.sh
    addcron 0 5 * * * /opt/cell-services/scripts/health-check.sh
11  /opt/verify/check-health.sh  (FLAG 3)
```

Top traps: `%` in crontab must be escaped as `\%` | heredoc delimiter must be quoted `<<'REMOTE'` | `--link-dest` symlink needs `ln -sfn` update after each run | `chmod +x` required

L09 — Poisoned Records: DNS Incident Response

Goal: Identify tampered A records, restore zone, harden BIND

```
INVESTIGATION
1   cat ~/incident-report.txt && cat ~/MISSION.txt
2   diff /etc/bind/zones/db.sector7.matrix.net /etc/bind/zones/db.sector7.matrix.net.bak
    -- three lines with 203.0.113.x are poisoned
3   sudo ausearch -f /etc/bind/zones/db.sector7.matrix.net
    -- uid=0 via vi at 02:13:44
4   grep '02:13' /var/log/named/named.log
    -- rndc reload 11 seconds after file edit

RESTORATION
5   sudo cp /etc/bind/zones/db.sector7.matrix.net.bak /etc/bind/zones/db.sector7.matrix.net
6   sudo nano /etc/bind/zones/db.sector7.matrix.net
    -- increment serial: 2026041000 --> 2026041001
7   sudo named-checkzone sector7.matrix.net /etc/bind/zones/db.sector7.matrix.net
8   sudo rndc reload sector7.matrix.net
9   dig grid-api.sector7.matrix.net @10.0.1.1 +short   -- must NOT be 203.0.113.99
10  /opt/verify/check-restoration.sh  (FLAG 1)

HARDENING
11  Edit named.conf.local: add allow-update { none; }; to zone declaration
12  sudo chown -R bind:bind /etc/bind/zones
    sudo chmod 640 /etc/bind/zones/*
13  tsig-keygen sector7-xfer | sudo tee /etc/bind/tsig-sector7.key
14  /opt/verify/check-hardening.sh  (FLAG 2)
```

Top traps: must increment serial after restore or secondaries ignore update | `named-checkzone` before `rndc reload` | `chown` and `chmod` as separate commands | `diff` direction: `diff current backup` shows current on `<` lines (those are poisoned)

Matrix House | CTS4321C Advanced Linux Administration | Week 3 Operator Field Manual — YOUR CELL IS YOUR RESPONSIBILITY