

Remote access — validation (Linux CLI + mobile parity)

Linux CLI validation — device sim + E2E

Table of contents

[What the CLI suite simulates](#)

[Scripts](#)

[Quick start \(local orchestration\)](#)

[Production / staging BE](#)

[Optional: bring up WireGuard on Linux](#)

[Mobile vs CLI checklist](#)

[Troubleshooting](#)

Source: REMOTE_ACCESS_VALIDATION.md — section links work in PDF viewers that support internal anchors.

Companion to REMOTE_ACCESS_IMPL.md. Use this when verifying WireGuard v1 **before** or **after** deploying the BE, and when debugging without an Android device.

What the CLI suite simulates

The Android app (RemoteAccessService) does the following over HTTP only (no RSSH in v1):

App behaviour	CLI equivalent
Persist stable device_id	RA_DEVICE_ID or --device-id
Persist stable random in prefs	file RA_RANDOM_FILE (default /tmp/androidcast-ra-<id>.random)
Mode Disabled → status:disable	ra_device_sim.sh disable
Mode WireGuard → jittered status:enable poll	ra_device_sim.sh poll --until connect (fixed interval via RA_POLL_INTERVAL_S, default 3s for tests)
Capabilities + tunnel_mode: wireguard	same JSON in ra_lib.sh / ra_ra_post
Apply WG config on connect	ra_wg_quick_from_connect + optional --apply-wg (wg-quick up)
Operator whitelist + open session	ra_e2e_cli.sh or manual remote_access.php curls

The **control plane** is validated by HTTP alone. **Data plane** (UDP to wg_endpoint) requires DNAT, wg0 on the BE, and optionally wg-quick on the machine running the device sim.

Scripts

All under examples/crash_reporter/backend/scripts/:

Script	Role
ra_lib.sh	Shared heartbeat + admin helpers (sourced, not run alone)
ra_device_sim.sh	Device-only: disable / enable / poll / optional WG
ra_e2e_cli.sh	Full flow: disable → wait → whitelist → open → connect → close → disable
test_remote_access_api.sh	Minimal API smoke (grep, no jq)
verify_remote_access_prod.sh	BE preflight: config, wg tools, DB tables, optional smoke

```
cd examples/crash_reporter/backend
chmod +x scripts/ra_*.sh scripts/test_remote_access_api.sh scripts/verify_remote_access_prod.sh
```

Dependencies: curl, jq (device sim and E2E), openssl (random). Optional: wireguard-tools for --apply-wg.

Quick start (local orchestration)

```
With the crashes stack listening on the default orchestration URL:
export CRASHES_BASE=http://127.0.0.1:8080/app/androidcast_project/crashes
export ADMIN_USER=admin ADMIN_PASS=admin

./scripts/verify_remote_access_prod.sh
./scripts/test_remote_access_api.sh
./scripts/ra_e2e_cli.sh

Single device poll after you have opened a session in the UI:
export RA_DEVICE_ID=my-lab-phone
./scripts/ra_device_sim.sh poll --until connect
```

Production / staging BE

```
Verified 2026-06-04 on https://apps.f0xx.org/app/androidcast_project/crashes: ra_e2e_cli.sh passed (connect returned real
peer_public_key, endpoint ra.apps.f0xx.org:51820). Poll step may take up to ~45s per min_poll_interval_s. UDP handshake still needs
a device or ra_device_sim.sh --apply-wg from a reachable host.
ssh alpine-be # or your BE host
cd /var/www/localhost/htdocs/apps/app/androidcast_project/android_cast/examples/crash_reporter/backend

export CRASHES_BASE=https://apps.f0xx.org/app/androidcast_project/crashes
export ADMIN_USER=admin ADMIN_PASS='...'

./scripts/verify_remote_access_prod.sh
BASE="$CRASHES_BASE" ./scripts/test_remote_access_api.sh
BASE="$CRASHES_BASE" ./scripts/ra_e2e_cli.sh

Or from your workstation (same cookies/session as browser login):
export CRASHES_BASE=https://apps.f0xx.org/app/androidcast_project/crashes
./scripts/ra_e2e_cli.sh --device-id ra-cli-$(hostname -s)

Full E2E from verify script:
RA_E2E=1 BASE="$CRASHES_BASE" ./scripts/verify_remote_access_prod.sh
```

Optional: bring up WireGuard on Linux

After poll --until connect returns credentials:

```
./scripts/ra_device_sim.sh poll --until connect --apply-wg
```

Manual:

```
RESP="$(curl -sf ...)" # connect response
source scripts/ra_lib.sh
ra_wg_quick_from_connect "$RESP" | sudo tee /tmp/ra-test.conf
sudo wg-quick up /tmp/ra-test.conf
ping -c1 10.66.66.1 # BE wg0 address if routed
sudo wg-quick down /tmp/ra-test.conf
```

This step fails if UDP wg_endpoint is not reachable (FE DNAT, firewall). HTTP E2E can still pass.

Mobile vs CLI checklist

Step	Android	Pass criteria
1	Developer settings → Disabled	action: disabled
2	WireGuard mode on, not whitelisted	action: wait
3	Dashboard whitelist device	API ok: true
4	Operator open session	action: wait then connect on next poll
5	VPN comes up	wg show / ping hub via tunnel
6	Close session / disable app	tunnel down; peer removed if provision_peers

CLI E2E covers steps 1–4 and 6 over HTTP; step 5 needs --apply-wg or a real device.

Troubleshooting

Symptom	Likely cause
action: deny + wg_tools_required	require_wg_tools true but wg genkey fails for PHP-FPM user — fix permissions / install wireguard-tools
connect but no handshake	wg_server_public_key mismatch, UDP not DNAT'd, or wrong wg_endpoint
wait after open session	Poll rate limit (min_poll_interval_s); wait and retry
jq required	Install jq on the host running device sim
API smoke fails on verify	Set BASE / CRASHES_BASE to live crashes URL; ensure login user has remote_access_admin

See also REMOTE_ACCESS_IMPL.md § Production deploy.