SECURITY INCIDENT RESPONSE · 30 APRIL 2026

# Root compromise of a shared-hosting control-panel server

ROOT COMPROMISE | CVE EXPLOITATION | PRIVILEGE ESCALATION | BOTNET IMPLANT | CONTROL-PANEL ABUSE | REBUILD

## 01 Executive summary

On 30 April 2026 the UPNIX team responded to the root compromise of a client's shared-hosting control-panel server. An attacker chained a vulnerable web application with a freshly disclosed privilege-escalation vulnerability to gain root, then established several persistence layers: a rogue control-panel reseller with full privileges, SSH backdoor keys across multiple accounts, dozens of disguised API tokens, and a botnet (Mirai-variant) binary. The malicious objective — enlisting the server into a botnet — was not completed; the intrusion was caught early. The UPNIX team isolated the host, enumerated every persistence layer, and planned and supported a full rebuild with complete credential rotation.

## 02 Environment

A single shared-hosting control-panel server (cPanel/WHM on CloudLinux) hosting a client's corporate and billing sites alongside customer hosting accounts.

## 03 Entry and escalation

A PHP web shell was planted through a vulnerable content-management site and initially ran at user level. After a privilege-escalation vulnerability (CVE-2026-31431) was publicly disclosed on 29 April, the attacker used it to escalate to root the following day. Notably, SSH was never used by the attacker: remote logins came only from the owner's own addresses, and the original administrative password kept working. The attacker operated entirely through the web shell and the hosting control panel.

## 04 Timeline (UTC)

| | |
|---|---|
| **29 Apr** | The privilege-escalation vulnerability (CVE-2026-31431) is publicly disclosed. |
| **30 Apr 05:34** | A valid control-panel root session is abused to create a rogue reseller account with full “grant-root” privileges. |
| **30 Apr** | Further rogue accounts created; SSH backdoor keys injected into several hosting-account home directories; dozens of disguised control-panel API tokens created; a botnet binary deployed. |
| **1 May** | The rogue accounts are suspended; forensic capture and containment by the UPNIX team. |

## 05 Persistence found

- A rogue, full-privilege reseller account (re-applied three times).
- SSH backdoor keys planted in six hosting-account home directories.
- More than thirty API tokens named to resemble legitimate system and control-panel jobs.
- A Mirai-variant botnet binary — deleted from disk but still running in memory, listening on a high port with a hard-coded command-and-control address.
- Empty loader-stub files. No kernel rootkit was present (no `LD_PRELOAD` preload, no rogue system services).

## 06 Containment and recovery

With an approximately 36-hour root window and multiple persistence layers, an in-place clean-up was judged high-risk. The UPNIX team's plan: isolate the host at the hypervisor firewall to the owner's addresses and block known command-and-control; back up the legitimate accounts after verifying none carried web shells; rebuild on a fresh server; restore from pre-incident backups; rotate every credential (control-panel root, all account passwords, SSH keys); delete all API tokens; and enable control-panel two-factor authentication.

## 07 Lessons learned

- **Patch the moment a relevant vulnerability is disclosed.** The window between public disclosure (29 April) and exploitation (30 April) was under a day.
- **Keep hosted content-management sites patched.** A vulnerable web application was the initial foothold.
- **Treat control-panel sessions and API tokens as high-value.** Enable two-factor authentication, review resellers and tokens regularly, and alert on new reseller or grant-root events.
- **A binary deleted from disk can still be running in memory.** Inspect process memory, not just the filesystem.
- **Multiple persistence layers plus unverifiable integrity mean rebuild, not clean-up.**

*This report has been fully anonymized. All client identifiers — organization names, hostnames, domains, IP addresses, account names and storage locations — have been removed. Technical findings, methods and recommendations are retained unchanged so the case remains useful to other operators.*

**About UPNIX Technologies.** The UPNIX team builds, monitors, protects and keeps running the infrastructure businesses depend on — security, 24/7 monitoring and operations for clients worldwide, alongside software and AI engineering.