PatchMon PATCHMON BLOG

cPanel CVE-2026-41940: The Authentication Bypass That Was Already in the Wild Before the Patch

A practical breakdown of the cPanel & WHM authentication bypass disclosed on 28 April 2026. How it works, which versions are patched, what to do right now, and how to confirm your hosting fleet is no longer exposed.

AUTHOR

PatchMon Team

PUBLISHED

2 May 2026

CATEGORIES

Security, Patch Management

READ ONLINE

https://patchmon.net/blog/cpanel-whm-cve-2026-41940-authentication-bypass-patch

Contents

There is a particular shape of bad week you have when you operate a hosting estate.

It usually starts with a tweet. Then a tweet thread. Then a watchTowr blog post. Then your Slack lights up with screenshots from your peers at other providers, all asking the same question: *"are you blocking 2083 and 2087 yet?"* By the time the upstream advisory drops, the largest hosts in the industry have already taken cPanel and WHM offline at the network edge, and you are trying to work out whether your customers will be more upset about a few hours of access loss or about the alternative.

That has been the last week of April 2026 for cPanel operators. The CVE is CVE-2026-41940 (https://www.rapid7.com/blog/post/etr-cve-2026-41940-cpanel-whm-authentication-bypass/), disclosed by WebPros on 28 April 2026 (https://support.cpanel.net/hc/en-us/articles/40073787579671-cPanel-WHM-Security-Update-04-28-2026), CVSS 9.8 critical, unauthenticated, remote, root. It is now in the CISA Known Exploited Vulnerabilities catalogue (https://cyberscoop.com/cpanel-authentication-bypass-vulnerability-cve-2026-41940-exploited/), with confirmed in-the-wild exploitation going back to at least 23 February 2026, two months before the patch.

This post is the practical version of the conversation. What the bug actually is, which versions contain the fix, what to do today, and how to verify your fleet is not still sat on a vulnerable build.

The Short Version

If you only have two minutes:

> **What it is:** An unauthenticated authentication bypass in cPanel and WHM. CRLF injection in the session handling code lets an attacker promote themselves to root with a single crafted HTTP request.
>
> **Severity:** CVSS 9.8 Critical (https://thehackernews.com/2026/04/critical-cpanel-authentication.html), no authentication required, remote, low complexity.
>
> **Disclosed:** 28 April 2026. Exploited in the wild since at least 23 February 2026. Two-month head start for the attackers.
>
> **Who is affected:** Every cPanel and WHM version after 11.40 prior to the 28 April fixes. WP Squared 136.1.6 and earlier. Roughly 1.5 million instances are exposed to the public internet (https://hadrian.io/blog/cpanel-critical-authentication-bypass-actively-exploited).
>
> **What an attacker gets:** Root on the cPanel host. Full control of every site, mailbox, and database hosted on it.
>
> **What you do about it:** Update to a patched build, restart `cpsrvd`, and assume any unpatched server has been touched.

The rest of this post is the why and the how.

What CVE-2026-41940 Actually Is

cPanel runs a single privileged daemon called `cpsrvd` that handles WHM, cPanel, and webmail traffic on ports 2083, 2087, 2095, and 2096. The login flow does what every login flow does: parse credentials, validate them, mint a session, write the session to disk, hand the user a cookie.

The bug, per the watchTowr Labs technical writeup (https://labs.watchtowr.com/the-internet-is-falling-down-falling-down-falling-down-cpanel-whm-authentication-bypass-cve-2026-41940/), lives in two unfortunate decisions in that flow.

The first is that the password sanitiser, `set_pass()`, strips null bytes but happily leaves `\r\n` alone. The second is that `saveSession()` writes the session payload to disk without invoking the usual `filter_sessiondata()` cleanup.

If you submit HTTP Basic credentials whose decoded form looks like:

```
root:x\r\nhasroot=1\r\ntfa_verified=1\r\nuser=root\r\nsuccessful_internal_auth_wit
h_timestamp=...
```

then those `\r\n` sequences become real line breaks inside the raw session file. cPanel's session format is a flat key=value list, so what was meant to be a password field is now a sequence of new top-level session keys.

From there it is a matter of getting cPanel to re-parse the file and promote the injected records to authenticated session state. The watchTowr researchers found that an unauthenticated request to a path like `/scripts2/listaccts` triggers `check_security_token() → do_token_denied() → Cpanel::Session::Modify→save()`, which reads the raw file, parses it as if every line is legitimate, and rewrites the JSON session cache with the injected keys at the top level.

When the attacker now hits any authenticated endpoint, cpsrvd loads that cache, sees `successful_internal_auth_with_timestamp` set, and short-circuits straight past the password check:

```
if ($SESSION_ref→{'successful_internal_auth_with_timestamp'}) {
    return AUTH_OK
}
```

You are root. The whole exploit is "log in unsuccessfully, then send one request with a CRLF in the password and a stripped session cookie." The Register's coverage (https://www.theregister.com/2026/04/30/cpanel_whn_cves/) summarises it neatly: "create a session by failing to log in, then promote yourself."

Why This One Hurts More Than Most cPanel CVEs

cPanel patches plenty of bugs. Most do not look like this.

It is unauthenticated. No credentials, no foothold, no chain of bugs to combine. A single attacker on the public internet can go from "nothing" to "root on your hosting node" in one request.

The exploit is small and reliable. Once you know the steps, the request fits in a curl one-liner. There are no offsets, no races, no version-specific tuning. Bleeping Computer reports (https://www.bleepingcomputer.com/news/security/critical-cpanel-and-whm-bug-exploited-as-a-zero-day-poc-now-available/) that working PoC exploit code became publicly available within hours of disclosure.

It was already being exploited. Per Help Net Security (https://www.helpnetsecurity.com/2026/04/30/cpanel-zero-day-vulnerability-cve-2026-41940-exploited/) and SecurityWeek (https://www.securityweek.com/critical-cpanel-whm-vulnerability-exploited-as-zero-day-for-months/), exploitation traces back to at least 23 February 2026, with KnownHost and other hosts confirming targeted activity well before 28 April. If your servers were exposed during that window, the right operating assumption is that you may already have been compromised, not that you might be a target going forward.

The blast radius is enormous. Shadowserver Foundation (https://www.shadowserver.org/) sees roughly 650,000 IPs running publicly accessible cPanel/WHM instances, and 44,000 unique IPs scanning or actively exploiting their honeypots. cPanel sits in front of an estimated 70 million domains globally. The total population of vulnerable assets is several orders of magnitude larger than most CVEs you will respond to this year.

This is not a "schedule the patch for next maintenance window" CVE. It is a "verify every box was patched today, then go look for indicators of compromise" CVE.

Which Versions Are Patched

Per the Rapid7 advisory (https://www.rapid7.com/blog/post/etr-cve-2026-41940-cpanel-whm-authentication-bypass/) and the official cPanel security bulletin (https://support.cpanel.net/hc/en-us/articles/40073787579671-cPanel-WHM-Security-Update-04-28-2026), the patched builds are:

Branch	Patched build
11.86	**11.86.0.41**
11.110	**11.110.0.97**
11.118	**11.118.0.63**
11.126	**11.126.0.54**
11.130	**11.130.0.19**
11.132	**11.132.0.29**
11.134	**11.134.0.20**
11.136	**11.136.0.5**
WP Squared	**136.1.7**

Anything older than the build listed for your branch is vulnerable. The affected range starts at any version "after 11.40", which is to say practically every cPanel install in the field.

Confirm your build with:

```
/usr/local/cpanel/cpanel -V
# or
whmapi1 version
```

If the answer is not at or above the patched build for your branch, you have not yet applied the fix.

The Update, Step by Step

The cPanel security bulletin (https://support.cpanel.net/hc/en-us/articles/40073787579671-cPanel-WHM-Security-Update-04-28-2026) prescribes three things: update, verify, restart. The full sequence on a single host:

```
# 1. Force an immediate update from your current tier
/scripts/upcp --force

# 2. Confirm the patched build is now installed
/usr/local/cpanel/cpanel -V

# 3. Restart cpsrvd so the running daemon picks up the patched code
/scripts/restartsrv_cpsrvd
```

`upcp` will pick the latest patched build on whatever release tier you have configured (`STABLE`, `RELEASE`, `CURRENT`, `EDGE`, or an LTS tier). If you have automatic updates disabled or pinned to an older tier, you will need to either bump the tier or apply the build manually.

Restart of `cpsrvd` is the bit that catches people. Updating the package on disk does not change the running daemon; the in-memory copy is still vulnerable until you bounce it. Treat the restart as a non-negotiable step rather than something you defer to the next reboot.

If You Cannot Patch Right Now

Sometimes you cannot patch a fleet of cPanel nodes within minutes of disclosure. Customer SLAs, change windows, multi-tenant impact, all the usual reasons.

For those situations, the same workaround the major hosts implemented on 28 April still applies: **block inbound traffic on the cPanel/WHM service ports at the network edge.**

```
# CSF (typical on cPanel servers)
# Edit /etc/csf/csf.conf and remove 2083, 2087, 2095, 2096 from TCP_IN
csf -r

# Or with iptables directly
iptables -I INPUT -p tcp -m multiport --dports 2083,2087,2095,2096 -j DROP
```

Allowlisting your office or VPN ranges instead of an outright block keeps your own access while shutting the door on the public internet:

```
iptables -I INPUT -p tcp -m multiport --dports 2083,2087,2095,2096 -s
YOUR.IP.RANGE/24 -j ACCEPT
iptables -I INPUT -p tcp -m multiport --dports 2083,2087,2095,2096 -j DROP
```

This is a stopgap, not a fix. As the official cPanel guidance (https://support.cpanel.net/hc/en-us/articles/40073787579671-cPanel-WHM-Security-Update-04-28-2026) puts it, defenders are strongly advised to patch rather than rely on workarounds. The port block buys you a maintenance window, not the rest of the year.

Look for Indicators of Compromise

Because this CVE was actively exploited for two months before the patch, the right next step after patching is to assume any internet-exposed cPanel host might already have been touched, and check.

The watchTowr writeup (https://labs.watchtowr.com/the-internet-is-falling-down-falling-down-falling-down-cpanel-whm-authentication-bypass-cve-2026-41940/) and several SOC vendors flag the following:

Session files containing injected records. Search `/var/cpanel/sessions/raw/` for any session file containing `hasroot=1`, `tfa_verified=1`, or `successful_internal_auth_with_timestamp` written outside of the expected legitimate fields:

```
grep -rE 'hasroot=1|tfa_verified=1|successful_internal_auth_with_timestamp'
/var/cpanel/sessions/raw/
```

Failed-login then token-missing patterns from the same IP. In `/usr/local/cpanel/logs/access_log` and `error_log`, look for a failed login (HTTP 401 on `/login/`) immediately followed by a request to a non-token path (e.g., `/scripts2/listaccts`) from the same source IP within seconds.

Unexpected root-owned WHM sessions. Sessions created without a corresponding successful login event in the logs are a strong signal.

New WHM API tokens, FTP accounts, or cPanel users created during the suspected exploitation window. The attacker's typical follow-on is to mint a persistent backdoor through the legitimate API once they have root.

`/etc/wwwacct.conf` **and** `/var/cpanel/users/` **modifications** that do not match a known change ticket.

If any of those turn up, treat the host as compromised. Patching is necessary but not sufficient; you also need to rotate credentials, audit hosted accounts, and consider rebuilding the box if you cannot fully scope the activity.

How to Tell Your Hosting Fleet Is Actually Patched

If you only run a handful of cPanel nodes, you can SSH into each one, run `/usr/local/cpanel/cpanel -V`, eyeball the output, and call it done. If you run a hundred, that approach falls apart fast. And that is before you factor in the WP Squared, CloudLinux, and AlmaLinux base systems sitting underneath cPanel, all of which need their own patch verification.

This is exactly the workflow PatchMon (https://patchmon.net) was built for. The agent reports the running OS, the kernel, the installed RPM/DEB package set including the cPanel and `cpanel-*` packages, and the available update queue from every node in your fleet to a single dashboard. Filtering to *"every host whose `cpanel` package version is below the patched build for its branch"* takes a few seconds rather than an afternoon of `pssh` loops. When the next CVE drops, the workflow is the same: filter, patch, watch the count fall to zero, export the report for your customers or your auditors.

For hosting providers and MSPs specifically, this is also the evidence trail you will want when a customer asks "were my sites on a patched node during the exposure window?". A timestamped, per-host record of installed package versions answers that question without you having to reconstruct it from log scraps.

The fastest way to evaluate is the PatchMon Cloud trial (https://patchmon.net/pricing). It runs on real hosts, billing tracks the hosts that actually check in rather than a fleet tier, and the trial is 14 days with no charge if you cancel before day 14. If you would rather self-host, the Community Edition (https://patchmon.net/open-source) is AGPLv3 and free; you can have it monitoring your cPanel estate in under half an hour.

Bottom Line

CVE-2026-41940 is the worst kind of CVE. Pre-authentication, single request, root, and weeks of in-the-wild exploitation before the rest of us knew about it.

The week of action looks like this:

- **Update every cPanel and WHM node to the patched build for its branch.** Confirm the version with `/usr/local/cpanel/cpanel -V`, not just by package manager output.
- **Restart** `cpsrvd` on every node. The patch on disk does not protect you until the daemon is restarted.
- **Block ports 2083, 2087, 2095, 2096 from the public internet** until you have confirmed every node is patched, and ideally keep them allowlisted afterwards.
- **Hunt for indicators of compromise** on every node that was internet-exposed during the 23 February to 28 April window. Treat any positive finding as a compromise, not a maybe.
- **Verify across the entire fleet, with timestamps**, so you can answer the question your customers are about to ask.

Hosting providers who handled this calmly were not the ones with the most expensive tooling. They were the ones who could confidently answer *"is every node patched, with evidence?"* on the same day the question came in. Whatever tool you use to do that, this is the week to make sure it actually works.

If you want to see how PatchMon handles a CVE response across a hosting fleet, the 14-day Cloud trial (https://patchmon.net/pricing) is the quickest way in, and the Community Edition (https://patchmon.net/open-source) is free to deploy on your own stack. Either way, get every node patched and `cpsrvd` restarted first. The rest is housekeeping.

Sources and Further Reading

- cPanel Security Bulletin: cPanel & WHM Security Update 04-28-2026 (https://support.cpanel.net/hc/en-us/articles/40073787579671-cPanel-WHM-Security-Update-04-28-2026)

Rapid7: CVE-2026-41940 cPanel & WHM Authentication Bypass (https://www.rapid7.com/blog/post/etr-cve-2026-41940-cpanel-whm-authentication-bypass/)

watchTowr Labs: The Internet Is Falling Down (CVE-2026-41940) (https://labs.watchtowr.com/the-internet-is-falling-down-falling-down-falling-down-cpanel-whm-authentication-bypass-cve-2026-41940/)

Help Net Security: cPanel zero-day exploited for months before patch release (https://www.helpnetsecurity.com/2026/04/30/cpanel-zero-day-vulnerability-cve-2026-41940-exploited/)

SecurityWeek: Critical cPanel & WHM Vulnerability Exploited as Zero-Day for Months (https://www.securityweek.com/critical-cpanel-whm-vulnerability-exploited-as-zero-day-for-months/)

The Hacker News: Critical cPanel Authentication Vulnerability Identified (https://thehackernews.com/2026/04/critical-cpanel-authentication.html)

Bleeping Computer: cPanel & WHM emergency update fixes critical auth bypass bug (https://www.bleepingcomputer.com/news/security/cpanel-whm-emergency-update-fixes-critical-auth-bypass-bug/)

Bleeping Computer: Critical cPanel and WHM bug exploited as a zero-day, PoC now available (https://www.bleepingcomputer.com/news/security/critical-cpanel-and-whm-bug-exploited-as-a-zero-day-poc-now-available/)

The Register: Critical cPanel, WHM flaw probably exploited as 0-day (https://www.theregister.com/2026/04/30/cpanel_whn_cves/)

CyberScoop: cPanel's authentication bypass bug is being exploited in the wild, CISA warns (https://cyberscoop.com/cpanel-authentication-bypass-vulnerability-cve-2026-41940-exploited/)

Hadrian: cPanel Critical Authentication Bypass Actively Exploited (https://hadrian.io/blog/cpanel-critical-authentication-bypass-actively-exploited)

Malwarebytes: Actively exploited cPanel bug exposes millions of websites to takeover (https://www.malwarebytes.com/blog/news/2026/05/actively-exploited-cpanel-bug-exposes-millions-of-websites-to-takeover)

Australian Cyber Security Centre: Active exploitation of cPanel/WHM critical vulnerability (https://www.cyber.gov.au/about-us/view-all-content/alerts-and-advisories/active-exploitation-of-cpanel-whm-critical-vulnerability)

PatchMon

ABOUT PATCHMON

The open source Linux patch management platform

PatchMon gives sysadmins one dashboard for patching across Linux, FreeBSD, and Windows fleets. Run it as a managed SaaS on PatchMon Cloud (per-host billing, 14-day trial, no fleet minimum) or self-host the AGPLv3 Community Edition on your own infrastructure.

Start a trial: patchmon.net/pricing