contact@namiq.io · namiq.io

NamiQ

SECURITY ADVISORY

Linux kernel CVE-2026-46300 (Fragnesia)

CVE-2026-46300 · Linux kernel · chained with PHP2Shell

Assessment & Action Plan

A local privilege escalation from May, scored 7.8, was added to CISA's exploited list on 28 October with a three-day deadline. CISA says it can be chained with the two PHP flaws to plant a webshell and run commands as root. What to patch, in what order, and what the chain changes.

PREPARED BY	Hung Le, NamiQ	DATE	2 November 2026
FACTS AS OF	28 October 2026	VERSION	1.1
SEVERITY	**High: CVSS 3.1 7.8; in CISA KEV**	STATUS	Fixed in stable kernels; CISA due date 31 October

NamiQ © 2026 NamiQ. All rights reserved. This document is the property of NamiQ. It may be shared unaltered, with attribution.
Companion article: namiq.io/insights/linux-kernel-fragnesia-cve-2026-46300

Update, 2 November 2026: this note now ends with a section on checking your own servers and projects with lattice-scan, for IT admins, AI agent skills and CI/CD, and the analysis report is reissued with it. Everything else is as of 28 October 2026; check CISA's catalog for any change since.

Security teams rank bugs one at a time. Attackers do not. On 28 October CISA added a Linux kernel flaw from May to its list of vulnerabilities that are being exploited, gave it a three-day deadline, and said in the entry that it can be chained with the two PHP flaws we wrote about on 22 October to implant PHP webshells and run commands as root. A local bug that most teams had slotted as "patch later" is now the second half of a remote chain.

This note is for you if you run Linux servers, especially ones that run PHP or that let untrusted users or containers run code. Everything is as of 28 October 2026 and every claim links to its source at the end. One caution on sourcing: the statement about the PHP chain is in CISA's own catalog entry. We found no independent write-up of the chain, so we state it as CISA's claim and no more.

The short version

- CVE-2026-46300, nicknamed Fragnesia, is a flaw in the Linux kernel's network stack. CVSS 3.1 7.8: local, low privileges. A third-party write-up says it was disclosed publicly on 13 May 2026, with a proof of concept, by William Bowling of V12 Security.
- CISA added it to the Known Exploited Vulnerabilities catalog on 28 October, due 31 October. Ransomware use is not recorded.
- CISA's entry says it can be chained with CVE-2026-109618 and CVE-2026-109619, the PHP flaws, to implant PHP webshells and run commands as root.
- Fixed in stable kernels, among them 5.10.257, 5.15.208, 6.1.174, 6.6.141, 6.12.91 and 6.18.33. Distributions backport, so use your distribution's advisory and the running kernel, not a version number alone.
- A stop-gap exists, with a cost: block the esp4, esp6 and rxrpc modules. It breaks IPsec on hosts that use it.
- If a host was exposed to the PHP flaws and has not had this kernel fix, assume the chain is possible and look for evidence it ran.

What the bug is

The kernel moves network data around in buffers called socket buffers, and sometimes merges one into another to save work. Some of those buffers point at memory that belongs to someone else, such as the file cache, and carry a marker that says so. The marker tells later code not to write into that memory in place.

The CVE record describes the fault precisely. When the kernel merges fragments from one buffer into another, the function that does it did not carry the marker across. The merged buffer then holds the same externally owned pages, but without the flag. Later, the code that handles incoming encrypted traffic (ESP, the part of IPsec that wraps packets) checks the flag to decide whether it can

decrypt in place. Seeing no flag, it decrypts in place, over memory it should have copied first. The upstream fix is two lines that preserve the marker.

The practical result, as a third-party write-up describes it, is that a local user can alter the in-memory copy of a file that the system caches, such as a setuid program, without touching the file on disk. The next time that program runs it runs the altered version, which gives root. The same write-up says the proof of concept targets the ESP-in-TCP path and depends on the espintcp module being present. We have not run it.

Why the score undersells it

CVSS 7.8 describes the flaw as it is: it needs a foothold on the machine. A score is a statement about one bug. It cannot say what else is on the machine, and what is on the machine is the point. A web server that runs PHP is a machine where an attacker who finds a remote flaw in PHP lands as the web server's account. That account is exactly the local, low-privilege foothold this bug needs.

That is the chain CISA describes. The PHP flaws provide the way in and the kernel flaw provides the way up. Neither needs the other to be dangerous, and together they remove the thing that usually limits the damage of a PHP compromise: that the attacker is stuck as an unprivileged user.

Why it is on the list now

CISA's Known Exploited Vulnerabilities catalog is for vulnerabilities it has evidence are being exploited. A listing is CISA's statement that exploitation has been seen; it does not say who, how widely or since when. The disclosure was in May. The listing is in October, two days after the PHP listings. We do not know what changed, and the entry does not say. Two reasonable readings fit what we can see: exploitation was observed recently, or the chain with PHP made it newly worth listing. Neither is stated. What is stated is the deadline: 31 October, three days from the entry.

Who is exposed

- Any Linux host with a kernel from 3.9 up to the fixed builds, in the sense of the CVE record. Distribution kernels differ; check the vendor.
- Hosts where an attacker can get code running as an unprivileged user: web servers (the PHP chain), shared development and build servers, CI runners, shared hosting, container hosts.
- On container hosts, the third-party write-up argues a single compromised container can reach the host.
- The write-up says Amazon Linux was not affected at the time of writing because it does not ship the loadable espintcp module. Treat that as a May statement about one distribution and check your own.

Things people will get wrong

- "It is only local." So is the second half of every chain. Ask what remote flaws put an attacker on the machine.
- "The file is unchanged on disk, so we are clean." The write-up says the on-disk file and its hash stay the same. File-integrity tools that read the disk will not see this.
- "A reboot cleared it." Rebooting or dropping caches clears altered cache pages. It does not undo anything the attacker did once they had root.
- "We patched PHP." Good, but the PHP flaws are only half of the chain, and an attacker with any other foothold still needs the kernel fixed.
- "The kernel is installed, so we are patched." The running kernel changes only after a reboot or a live patch.

How we would triage it

This is our judgement. The facts are from the sources.

- Rank by foothold. Hosts that run PHP, or let untrusted users or containers run code, come first, because for them the chain's first step is available.
- Treat the CISA deadline as the schedule. Three days is short. If you cannot reboot everything, reboot the exposed hosts and use the module stop-gap on the rest.
- Assume the PHP flaws may already have been used on any unpatched server and look for evidence, because this entry says the second step exists.

Step 1: what kernel is running, and is it fixed

```
# The running kernel (not the newest installed one)
uname -r

# Debian and Ubuntu: does the package changelog list the fix?
apt changelog linux-image-$(uname -r) | grep -i CVE-2026-46300

# RHEL family
rpm -q --changelog kernel-$(uname -r) | grep -i CVE-2026-46300

# Are the ESP modules loaded?
lsmod | grep -E '^(esp4|esp6|espintcp|rxrpc)'
```

Compare with your distribution's security advisory for CVE-2026-46300. If the changelog does not mention it, do not assume the build is fixed.

Step 2: patch, in order

- Hosts that run PHP or let untrusted code run: update the kernel and reboot, or apply a vendor live patch if you have one.
- Then the rest of the fleet in your normal cycle, with a short deadline.
- Update PHP everywhere it runs, to the fixed releases in our PHP2Shell note.

Step 3: the stop-gap, if you cannot patch yet

```
# Stop the modules loading, and unload them if nothing is using them
echo 'install esp4 /bin/false'  | sudo tee    /etc/modprobe.d/block-esp.conf
echo 'install esp6 /bin/false'  | sudo tee -a /etc/modprobe.d/block-esp.conf
echo 'install rxrpc /bin/false' | sudo tee -a /etc/modprobe.d/block-esp.conf
sudo modprobe -r esp4 esp6 rxrpc 2>/dev/null || true
```

The cost is real. Hosts that terminate IPsec VPNs, for example with strongSwan or Libreswan, will stop working. Do not apply this to a VPN gateway without a plan. It is a stop-gap and should have an end date.

Step 4: assume the chain may have run

- On every PHP server that was unpatched for the PHP flaws, look in web roots and upload directories for files you did not put there, particularly PHP files with recent modification times.
- Review the web server and PHP-FPM accounts' activity: new child processes, shells, outbound connections.
- Look for root activity that does not belong to an administrator: new user accounts, new SSH keys, new scheduled jobs, new services.
- Remember that disk-based integrity checks will not see an altered cache. Rely on process and authentication logs, and on a reboot to clear any lingering cache changes.
- If you find a webshell, treat the host as compromised and rebuild it. Rotate the secrets it could read.

Check your own servers and projects

The facts above are as of 28 October 2026. To compare your own estate with this CVE, the open-source lattice-scan command line checks the parts it can see. Naming the CVEs needs no key, only the CVE ids leave your machine, and a match is a candidate for a person to confirm, not a finding.

On a server: run it on the server itself, not on the laptop or CI runner that deploys to it. It reads installed packages (dpkg, rpm or apk), snaps, the names of running container images and the kernel version, compares them on the machine, and keeps the inventory there. A distribution can fix a CVE

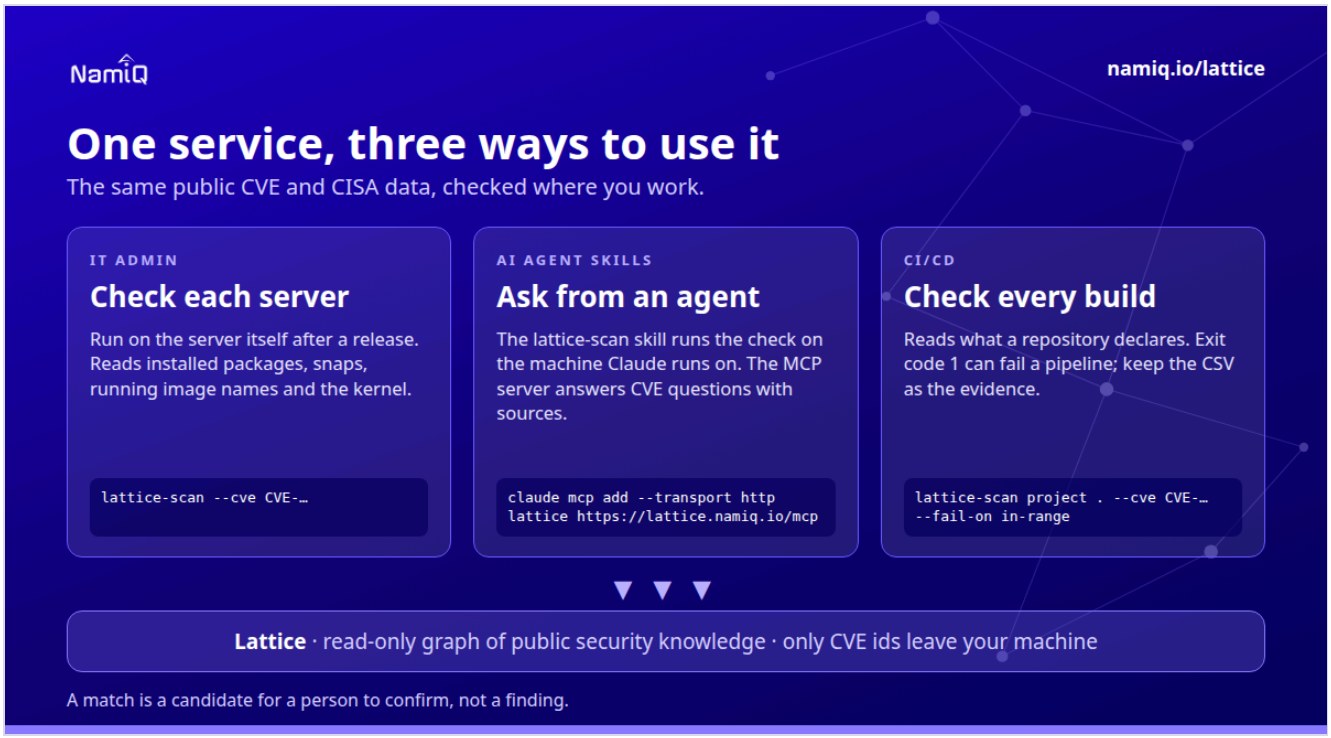
under an unchanged version number, so on Debian-family hosts read the changelog line the report shows.

```
lattice-scan --cve CVE-2026-46300
```

The host check compares the running kernel's upstream version with the ranges, and a distribution can backport the fix under an unchanged version number, so read the kernel package's changelog line too (Step 1). It does not read a repository, so there is no project check for a kernel flaw.

Who can use it, and how

For	What to do	Command
IT admin	Run the host check on each server after a release, and again when a new entry lands.	lattice-scan --cve CVE-2026-46300
AI agent skills	The lattice-scan skill runs the check on the machine Claude runs on. The Lattice MCP server answers questions about the CVE with its sources.	claude mcp add --transport http lattice https://lattice.namiq.io/mcp



Where the same check is used. The three ways read the same public data; only CVE ids leave your machine.

Guide: namiq.io/lattice/guides/choose-your-channel. MCP server (io.namiq/lattice in the MCP registry): namiq.io/lattice. Package: pypi.org/project/lattice-scan, or namiq.io/apt.

What we do not know yet

- How the chain works in detail, and whether it has been used. CISA's entry states it and we found no independent account.
- Why the listing came in October when the disclosure was in May.
- Which distributions have shipped a fix for which of their kernels. Vendors differ and the picture was changing.
- Whether the espintcp requirement in the write-up applies to every route to the bug.

If you take one thing from this: stop ranking bugs alone. For every local flaw, ask which remote flaws on the same machine put an attacker where it needs them. This week, PHP is the answer.

Appendix 1. Severity decoded

CVE-2026-46300: score 7.8, `CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H` From the CVE record. Local, low privileges: the score describes the kernel flaw on its own, not the chain.

Metric	Value	What it means here
Attack vector	Local	Needs an account or code already running on the machine.
Attack complexity	Low	No special conditions beyond the flaw itself.
Privileges required	Low	An ordinary account is enough.
User interaction	None	No one has to do anything.
Scope	Unchanged	Damage stays inside the vulnerable component.
Confidentiality	High	Total loss of confidentiality.
Integrity	High	Total loss of integrity.
Availability	High	Total loss of availability.

Appendix 2. Timeline

Date	Event
13 May 2026	Public disclosure of Fragnesia by William Bowling of V12 Security, with a proof of concept (as reported by a third-party write-up).
23 May 2026	CVE record for CVE-2026-46300 published.
21 October 2026	PHP releases fixes for CVE-2026-109618 and CVE-2026-109619 (PHP2Shell).
26 October 2026	CISA adds both PHP CVEs to its Known Exploited Vulnerabilities catalog, due 29 October.
28 October 2026	CISA adds CVE-2026-46300, due 31 October, and says it can be chained with the two PHP CVEs to implant PHP webshells and run commands as root.

Appendix 3. Exposure by situation

These priorities are our judgement from the attacker prerequisites in the sources. They are not vendor ratings.

Your situation	Priority	Timeframe

Linux host running PHP that is not yet patched for PHP2Shell, and not patched for this kernel flaw	Critical	Patch both now; the CISA deadline is 31 October
Multi-tenant host, CI runner or container host where untrusted users run code	High	Kernel update this week
Host with espintcp or the ESP modules loaded and untrusted local users	High	Patch or apply the module stop-gap now
Single-purpose host with no untrusted local users and no PHP	Lower	Next kernel update
Kernel already updated to a fixed build	None	Confirm the running kernel, not just the installed one

Appendix 4. Definition of done

- Every Linux host's running kernel is known, and the distribution's fixed build for CVE-2026-46300 is identified.
- Hosts exposed to untrusted users or to PHP are updated and rebooted first.
- PHP is updated to a fixed release on every host that serves PHP.
- Where a patch cannot be applied, the ESP modules are blocked and the effect on IPsec has been accepted in writing.
- Web server and PHP accounts have been checked for webshells and unexpected privileged activity, on the assumption that the chain may already have run.
- Disk-based integrity checks are not the only evidence relied on.

Sources

- CISA Known Exploited Vulnerabilities catalog entry for CVE-2026-46300 (added 28 October 2026): https://www.cisa.gov/known-exploited-vulnerabilities-catalog
- CVE-2026-46300 record (Linux CNA): https://www.cve.org/CVERecord?id=CVE-2026-46300
- Upstream fix, stable tree commit 3599e6b3cc1ada96883d496a50a210d3afbb6987: https://git.kernel.org/stable/c/3599e6b3cc1ada96883d496a50a210d3afbb6987
- Fragnesia (CVE-2026-46300): Linux page cache LPE survives Dirty Frag mitigations (third-party write-up): https://lilting.ch/en/articles/linux-fragnesia-page-cache-root
- CISA alert: critical vulnerabilities in PHP require immediate patching (21 October 2026): https://www.cisa.gov/news-events/alerts/2026/10/21/critical-vulnerabilities-php-require-immediate-patching
- NamiQ: PHP2Shell, CVE-2026-109618 and 109619: https://namiq.io/insights/php2shell-cve-2026-109618-109619