# Runbook: Fixing PHP CVEs for RLSA-2026:87080 (php:7.4)

| Item | Value |
|---|---|
| Advisory | 87080( https://errata.rockylinux.org/RLSA-2026:87080): Important: php:7.4 security update |
| Affected platform | Rocky Linux 8 (x86_64): ClusterStor management nodes |
| Fixed version | `7.4.33-8.module+el8.10.0+40393+c071e357` |
| Nodes to patch | Both management nodes (for example `<mgmt-node-0>` and `<mgmt-node-1>`) |
| Service impact | Brief restart of the web/monitoring services on the ***active*** management node |
| Estimated time | About 30 minutes |

Modification history

| Version | Date | Description |
|---|---|---|
| 1.0 | 2026-10-23 | Initial version of the runbook |
| 1.1 | 2026-10-26 | Added a note with the mitigation for CVE-2026-115560 (RHSB-2026-013( https://access.redhat.com/security/vulnerabilities/RHSB-2026-013 )) |
| 1.2 | 2026-10-26 | Added a note on running the procedure as the `admin` user with `sudo` |

## 1. CVEs addressed

| CVE | CVSS v3 | Summary |
|---|---|---|
| CVE-2026-109618 | 8.1 | OOM bailout during request parsing causes memory corruption and possible RCE |
| CVE-2026-109619 | 8.1 | Fatal error handling keeps stale request state, causing a use-after-free |
| CVE-2026-91765 | 7.5 | DoS through unbounded recursion in the SOAP parser |
| CVE-2025-14181 | 6.5 | DoS through a heap buffer overflow in the SOAP client |
| CVE-2026-91767 | 6.5 | Information disclosure through a crafted TLS server certificate |
| CVE-2026-91768 | 6.5 | Access control bypass through a partial IPv6 address comparison |
| CVE-2026-91766 | 5.9 | Credential disclosure through cross-origin HTTP redirects |
| CVE-2026-92842 | 5.9 | Information disclosure through an out-of-bounds read in stream filters |
| CVE-2026-93682 | 5.8 | Out-of-bounds read through an empty HTTP redirect `Location` header |
| CVE-2026-6103 | 4.3 | Archive entry injection through an integer overflow in the TAR parser |
| CVE-2026-91769 | 4.3 | Server impersonation through Common Name fallback in TLS verification |
| CVE-2025-1218 | 3.7 | DoS through an out-of-bounds read in the mysqlnd wire protocol parser |

> ***Note:*** *CVE-2026-115560 (RHSB-2026-013) is a third issue in PHP's session upload progress feature. This PHP update doesn't fix it. Apply the mitigation described in the note in Step 6 until a later PHP release fixes it.*

## 2. Overview

The management nodes have no direct internet access. To install the fix:

1. **Download host** (any Linux machine with internet access that can also reach the management nodes): Download the fixed RPMs from the Rocky Linux 8 AppStream repository, then copy them and the repository file to both management nodes.
2. **Each management node**: Create a local repository from the RPMs, then upgrade PHP from that repository.
3. **Active management node**: Restart the services that depend on PHP.

## 3. Prerequisites

> ***Note:*** *If root access isn't available, you can run this procedure as the `admin` user with `sudo`. `admin` can't write to `/root` or `/etc/yum.repos.d/`, so `scp` the RPM directory and `php-fixes.repo` to the `admin` home directory, then use `sudo` to move them to `/root/` and `/etc/yum.repos.d/` on each management node. Run the remaining commands that change the system with `sudo`.*

- You have root access to both management nodes.
- The download host has `curl` and outbound HTTPS access to `dl.rockylinux.org`.
- The download host can reach both management nodes over SSH (`scp`). If it can't, use the customer-approved transfer method (for example, USB media) to move the files.
- `createrepo` (`createrepo_c`) is installed on the management nodes. Check with `which createrepo`.
- Agreed brief downtime of the web UI and monitoring (httpd, Zabbix, Icinga, Plex) as the services restart.

## 4. Pre-checks (run on each management node)

Record the current PHP package versions so you can verify the upgrade and roll back if needed:

```
rpm -qa 'php*' | sort | tee /root/php-pre-RLSA-2026-87080.txt
php -v
```

Expected: The `php*` packages are an older `7.4.33-*` build that is not `7.4.33-8.module+el8.10.0+40393+c071e357`.

Find the **active** management node, which is the node currently running the web and monitoring services:

```
systemctl is-active httpd zabbix-server icinga2 plex
```

> ***Note:*** *You can ignore the warning `Failed loading plugin "changelog": No module named 'dateutil'` in the `dnf` output.*

## 5. Procedure

### Step 1: Download the fixed RPMs (download host)

Create `get-php-rpms.sh`:

```
#!/usr/bin/env bash
set -euo pipefail

BASE="https://dl.rockylinux.org/pub/rocky/8/AppStream/x86_64/os/Packages/p"
VR="7.4.33-8.module+el8.10.0+40393+c071e357"
DEST="${1:-./php-7.4.33-8-rpms}"
mkdir -p "$DEST"

PKGS=(php php-bcmath php-cli php-common php-dba php-dbg php-devel php-embedded
      php-enchant php-ffi php-fpm php-gd php-gmp php-intl php-json php-ldap
      php-mbstring php-mysqlnd php-odbc php-opcache php-pdo php-pgsql
      php-process php-snmp php-soap php-xml php-xmlrpc)

for p in "${PKGS[@]}"; do
  f="${p}-${VR}.x86_64.rpm"
  echo "-> $f"
  curl -fSL --retry 3 -o "$DEST/$f" "$BASE/$f"
done

echo "Downloaded $(ls "$DEST"/*.rpm | wc -l) RPMs to $DEST"
```

Run the script:

```
bash get-php-rpms.sh
```

Expected: `Downloaded 27 RPMs to ./php-7.4.33-8-rpms`

> ***Note:*** *If a download returns HTTP 404, a newer build has replaced this one in the repository. Get the package from `https://dl.rockylinux.org/vault/rocky/8.10/AppStream/x86_64/os/Packages/p/` instead.*

### (Optional) Verify the RPM signatures

Each RPM carries a GPG signature from Rocky Linux. Checking it confirms two things:

- **Authenticity:** Rocky Linux built and signed the package, not a third party.
- **Integrity:** The package has not been changed since it was signed, for example by a corrupted download or tampering.

The local repository uses `gpgcheck=0`, so dnf does not check signatures during the upgrade. Verify them on the download host before you transfer the files. This check uses a temporary RPM database and leaves the host's own database unchanged:

```
curl -fsSL -o /tmp/RPM-GPG-KEY-Rocky-8 https://dl.rockylinux.org/pub/rocky/RPM-GPG-KEY-Rocky-8
mkdir -p /tmp/rocky8-keydb
rpmkeys --dbpath /tmp/rocky8-keydb --import /tmp/RPM-GPG-KEY-Rocky-8
rpmkeys --dbpath /tmp/rocky8-keydb -K php-7.4.33-8-rpms/*.rpm
```

Expected: Every file reports `digests signatures OK`. If any file reports `NOT OK`, do not use it; download it again. If any file reports `NOKEY`, the key import failed; re-run the import.

## Step 2: Copy the RPMs to both management nodes (download host)

```
scp -r php-7.4.33-8-rpms/ root@<mgmt-node-0>:~
scp -r php-7.4.33-8-rpms/ root@<mgmt-node-1>:~
```

The RPMs must end up in `/root/php-7.4.33-8-rpms/` on each management node, because the repository file in Step 3 points to that path.

## Step 3: Create and deploy the repository file (download host)

Create `php-fixes.repo` on the download host:

```
[Php-CVE-fixes]
name=Php CVE fixes Repository
baseurl=file:///root/php-7.4.33-8-rpms
enabled=1
gpgcheck=0
module_hotfixes=true
```

> ***Important:*** *You need `module_hotfixes=true`. PHP 7.4 is delivered as a modular stream (`php:7.4`). Without this option, dnf filters out the packages in this repository, and the upgrade reports "Nothing to do".*

Copy the repository file to both management nodes:

```
scp php-fixes.repo root@<mgmt-node-0>:/etc/yum.repos.d/
scp php-fixes.repo root@<mgmt-node-1>:/etc/yum.repos.d/
```

## Step 4: Create the local repository metadata (each management node)

```
[root@<mgmt-node-0> ~]# createrepo /root/php-7.4.33-8-rpms/
[root@<mgmt-node-1> ~]# createrepo /root/php-7.4.33-8-rpms/
```

Expected: `Directory walk done - 27 packages`

Confirm that each management node lists the repository:

```
dnf repolist
```

Expected: `Php-CVE-fixes` appears alongside the existing `cs` repository.

> ***Note:*** *Run `createrepo` before any other `dnf` command on the management nodes. Until the repository metadata exists, dnf reports an error for the `Php-CVE-fixes` repository.*

## Step 5: Upgrade PHP (each management node)

```
dnf update php -y
```

Upgrading `php` also upgrades every installed PHP extension, because each extension requires the same version of `php-common`.

In the command output, check that the transaction:

- upgrades only `php*` packages, all to `7.4.33-8.module+el8.10.0+40393+c071e357` from the `Php-CVE-fixes` repository,
- may also install `php-opcache` as a weak dependency (this is expected), and
- removes no packages.

In validation, the transaction upgraded 15 packages and installed 1:

```
Upgrading: php php-bcmath php-cli php-common php-fpm php-gd php-intl php-json php-ldap
           php-mbstring php-mysqlnd php-pdo php-pgsql php-soap php-xml
Installing weak dependencies: php-opcache
Transaction Summary: Install 1 Package, Upgrade 15 Packages
```

Repeat this step on the second management node.

### Step 6: Restart the dependent services (active management node only)

> ***Note: Additional mitigation for CVE-2026-115560 (RHSB-2026-013)***
>
> *This PHP update doesn't fix CVE-2026-115560. Disabling PHP session upload progress fully protects against it and also blocks the confirmed session paths for CVE-2026-109619. The PHP project expects to fix CVE-2026-115560 in a later routine release. Keep this setting until that update is installed.*
>
> *The issue affects only applications that set `session.serialize_handler` to `php_binary` or `php_serialize` in configuration and then switch it to `php` at runtime with `ini_set()` before calling `session_start()`. Red Hat recommends the mitigation regardless.*
>
> ***Warning:** Applications that use PHP's built-in session upload progress tracking will no longer receive progress data. File uploads still work.*
>
> *Apply this on **each** management node. A drop-in file in `/etc/php.d/` is used instead of editing `/etc/php.ini`, so the change survives PHP package updates:*
>
> ```
> ## on each management node
> cat > /etc/php.d/99-security.ini <<'EOF'
> session.upload_progress.enabled=0
> EOF
> ```
>
> *Then restart php-fpm on all management nodes so that the change takes effect. Run this from either node:*
>
> ```
> pdsh -g mgmt "systemctl restart php-fpm"
> ```
>
> *Check the setting on each management node. Expected: `session.upload_progress.enabled => Off => Off`.*

Run these commands on the active management node that you identified in the pre-checks. Restart `php-fpm` and `httpd` first, because they serve the PHP web UIs (Zabbix and Icinga Web). Then restart the backend daemons:

```
systemctl restart php-fpm
systemctl restart httpd
systemctl restart zabbix-server
systemctl restart icinga2
systemctl restart plex
```

Check that all the services are running:

```
systemctl is-active php-fpm httpd zabbix-server icinga2 plex
```

Expected: Each service reports `active`. On the passive node, the new packages take effect when it next becomes active, because the services start with the patched binaries.

## 6. Post-checks (each management node)

```
dnf list installed 'php*'
php -v
rpm -q --changelog php-common | head -20
php -i | grep 'session.upload_progress.enabled'
```

Expected:

- Every PHP package from the PHP module (for example `php`, `php-cli`, `php-common`, `php-fpm`, `php-xml`) reports version `7.4.33-8.module+el8.10.0+40393+c071e357` from `@Php-CVE-fixes`.
- Packages that ClusterStor provides (`php-Icinga`, `icinga-php-*`, `icingaweb2-vendor-lessphp`) are unchanged and still come from `@cs`. This is expected.
- The `php-common` changelog lists the CVEs in section 1.
- `session.upload_progress.enabled => Off => Off`.

Functional checks:

- Log in to the ClusterStor web UI and confirm that the pages load.
- Confirm that the Zabbix and Icinga dashboards load and show current data.
- Check `/var/log/httpd/error_log` and `/var/log/php-fpm/error.log` for new errors.

## 7. Cleanup (each management node)

Remove the temporary repository so that it does not interfere with future ClusterStor updates:

```
rm -f /etc/yum.repos.d/php-fixes.repo
dnf clean all
dnf repolist          # Php-CVE-fixes should no longer be listed
```

Optionally, remove the RPM directory with `rm -rf /root/php-7.4.33-8-rpms`. If you need to roll back, keep it until you have confirmed that the system is stable.

## 8. Rollback (only if issues occur)

Find the upgrade transaction, then undo it:

```
dnf history list php
dnf history undo <transaction-id>
```

For the undo to work, the previous PHP versions must be available in an enabled repository, such as `cs`. Compare the result with `/root/php-pre-RLSA-2026-87080.txt`, then restart the services as in Step 6 of section 5.

## 9. Summary checklist

- [ ] Pre-check package versions recorded on both management nodes
- [ ] 27 RPMs downloaded and signatures verified
- [ ] RPMs copied from the download host to both management nodes
- [ ] `createrepo` run on both management nodes
- [ ] `php-fixes.repo` deployed and `Php-CVE-fixes` shown in `dnf repolist`
- [ ] PHP upgraded to `7.4.33-8.module+el8.10.0+40393+c071e357` on both management nodes
- [ ] php-fpm, httpd, zabbix-server, icinga2, plex restarted on the active node
- [ ] Post-checks passed (versions, web UI, monitoring)
- [ ] Temporary repository removed