MAMP Server Vulnerability Remediation Plan (Tickets #157, #160, #161)

Last Updated: 2026-02-17

Live Validation Snapshot (from terminal audit)

- Active Apache: `2.4.33 (Win64)`
- Active PHP (via response headers): `8.3.1`
- Scanner-visible header currently includes full versions:
  - `Server: Apache/2.4.33 (Win64) OpenSSL/1.0.2u mod_fcgid/2.3.9 PHP/8.3.1`
  - `X-Powered-By: PHP/8.3.1`

This confirms why unauthenticated version/banner-based detections remain open until Apache/PHP are upgraded and banner exposure is reduced.

Database Impact Statement

- These tickets are web-stack vulnerabilities (Apache/PHP), not MySQL schema/data vulnerabilities.
- No database object/data changes are required to remediate.
- Detailed per-database audit and no-DB-change plan:
  `docs/MAMP_Database_Audit_and_WebStack_Remediation.md` .

Summary (What is impacted)

The WIP server `v-mtmfg-5` is reporting medium-severity vulnerabilities tied to:

- PHP versions below patched releases (Ticket #157)
- Apache HTTP Server `2.4.53` and earlier (Ticket #160)
- Apache `mod_proxy` X-Forwarded-For handling issue in `2.4.53` and earlier (Ticket #161)

Required Target Versions

Use one of the following patched PHP versions (or newer):

- PHP `8.4.5+`
- PHP `8.3.19+`
- PHP `8.2.28+`
- PHP `8.1.32+`

Use Apache HTTP Server version newer than `2.4.53` (latest stable 2.4.x is recommended).

User-Friendly Fix Plan

1) Prepare and schedule (30-60 min)

- Schedule a maintenance window for `v-mtmfg-5`.
- Confirm app owner approval for restart/downtime.
- Take backups before touching anything:
  - MAMP Apache config (`httpd.conf`, vhosts, SSL config)
  - PHP config (`php.ini` and custom extensions)
  - Web root/application files
  - Database backup/snapshot if this server hosts DB components

2) Confirm current versions (baseline)

Record current versions so you can prove what changed:

- `php -v`
- `httpd -v`

Also capture externally visible headers (what scanners often detect):

- `curl -I http://<server-or-site>`
- `curl -I https://<server-or-site>`

3) Upgrade MAMP components

Choose the safest option for your environment:

- Preferred: Upgrade to the latest MAMP package that includes patched Apache and PHP.
- Alternate: Upgrade Apache and PHP binaries in place to patched versions supported by your MAMP installation.

After upgrade:

- Set active PHP runtime to a patched version (`8.4.5+` preferred).
- Ensure Apache is running a version newer than `2.4.53` .
- Re-apply only required custom config from backup (avoid copying old insecure defaults blindly).

4) Restart services and smoke test

- Restart Apache (and PHP service manager if applicable).
- Validate:
  - Site loads over HTTP/HTTPS
  - Login/critical workflows still function
  - No startup errors in Apache/PHP logs

5) Security hardening (recommended)

These do not replace patching, but reduce exposure:

- In Apache config:
  - `ServerTokens Prod`
  - `ServerSignature Off`
- In `php.ini` :
  - `expose_php = Off`
- Disable unused modules, especially:
  - `mod_lua` (if not required)
  - `mod_isapi` (if not required)

6) Validate closure evidence

Collect evidence for ticket closure:

- New version outputs:
  - `php -v` showing patched version
  - `httpd -v` showing version newer than 2.4.53

- Header checks:
  - `curl -I` results after patch
- Security scan rerun result:
  - No findings for tickets #157, #160, #161

Why this fixes all 3 tickets

- Ticket #157 is resolved by upgrading PHP to the patched release line.
- Ticket #160 is resolved by upgrading Apache above `2.4.53`.
- Ticket #161 is tied to the same vulnerable Apache range and is resolved by the Apache upgrade.

Suggested Customer Response (copy/paste)

```
We reviewed vulnerabilities on server v-mtmfg-5 (MAMP stack) and completed remediation planning

Actions taken / planned:
1. Backed up Apache/PHP configs and application data.
2. Upgraded Apache HTTP Server to a version newer than 2.4.53.
3. Upgraded PHP to a patched release (8.4.5+ / 8.3.19+ / 8.2.28+ / 8.1.32+).
4. Restarted services and validated application functionality.
5. Applied hardening settings (ServerTokens Prod, ServerSignature Off, expose_php Off).
6. Re-ran vulnerability scan to confirm findings are cleared.

Validation evidence:
- php -v output attached
- httpd -v output attached
- Post-remediation scan results attached

Please review and confirm if any additional evidence is required to close these tickets.
```

Official References

- PHP downloads: https://www.php.net/downloads
- PHP changelogs:
  - https://www.php.net/ChangeLog-8.php#8.4.5
  - https://www.php.net/ChangeLog-8.php#8.3.19
  - https://www.php.net/ChangeLog-8.php#8.2.28
  - https://www.php.net/ChangeLog-8.php#8.1.32

- Apache vulnerabilities: https://httpd.apache.org/security/vulnerabilities_24.html