SNAPSMACK — SECURITY FINDING

smack-2fa-verify.php: Insecure Cookie Flag + Recovery Code Policy Gap

Date: 2026-05-19

File: smack-2fa-verify.php

Severity: Low

Status: Fixed in 0.7.159

Summary

Two issues were identified in smack-2fa-verify.php, the TOTP interstitial page that sits between password verification and full session grant. Neither issue permits bypassing 2FA entirely, but both represent policy gaps relative to the rest of the authentication flow.

Finding 1 — Session Cookie Missing secure Flag on Reverse-Proxy Installs

snap-in.php sets the session cookie secure flag by checking both $_SERVER['HTTPS'] and HTTP_X_FORWARDED_PROTO. This correctly handles installs behind a reverse proxy (Nginx, Cloudflare, load balancer) where PHP itself runs on plain HTTP internally while the browser connection is HTTPS.

smack-2fa-verify.php used only $_SERVER['HTTPS']. On a reverse-proxy install, $_SERVER['HTTPS'] is never 'on' inside PHP, so the secure flag was absent from the cookie emitted at the TOTP step. This means the session cookie could be transmitted over HTTP during the 2FA verification step on proxy-terminated setups.

Affected Code — Finding 1

smack-2fa-verify.php (pre-0.7.159):

```
'secure' => isset($_SERVER['HTTPS']) && $_SERVER['HTTPS'] === 'on',
```

snap-in.php (correct):

```
'secure' => (isset($_SERVER['HTTPS']) && $_SERVER['HTTPS'] === 'on')
         || (isset($_SERVER['HTTP_X_FORWARDED_PROTO']) && $_SERVER['HTTP_X_FORWARDED_PROTO'] === 'https'),
```

Finding 2 — Recovery Code at TOTP Stage Does Not Force Password Change

When a recovery code is used at the snap-in.php stage (i.e. the user has forgotten their password entirely), a forced password change is triggered on the next page. This signals to the admin that something was wrong and prompts a security review.

When a recovery code is used at the smack-2fa-verify.php stage (i.e. the user passed password verification but cannot produce a TOTP code — lost authenticator, device failure, etc.), no forced password change was triggered. The user was logged directly into the admin panel.

This is a policy inconsistency. Burning a recovery code at any stage indicates that normal authentication could not complete. The forced password change ensures the admin is aware something went wrong and prompts re-enrollment of 2FA.

Fix

Finding 1: smack-2fa-verify.php updated to match snap-in.php's secure flag logic, including HTTP_X_FORWARDED_PROTO check.

Finding 2: smack-2fa-verify.php now sets $_SESSION['force_password_change'] = true and redirects to smack-change-password.php when a recovery code is consumed at the TOTP stage, consistent with the snap-in.php recovery code path.

Notes

Neither finding permits bypassing 2FA. The TOTP gate in smack-2fa-verify.php is structurally sound: totp_pending_user_id must be present in session (planted only by snap-in.php after successful password verification), user_login is never set before TOTP or recovery code is confirmed, and 5 failed TOTP attempts wipe the pending session entirely.

The cookie flag issue is only exploitable if an attacker can perform an active network interception between the user and the server — unlikely on HTTPS installs but a configuration flaw nonetheless.

A stale comment referencing core/auth.php (renamed to core/auth-smack.php in 0.7.155) was also corrected in this fix.