SNAPSMACK — SECURITY FINDING

install.php Step 4 Creates Admin Account on Existing Install (2FA Bypass)

Date: 2026-05-19

File: install.php

Severity: High

Status: Fixed in 0.7.157

Summary

Step 4 of the installer creates an admin user account. It used INSERT IGNORE, which silently skips the insert if the submitted username already exists — but would succeed if the attacker chose a different username. On a site where install.php was not deleted after initial setup, an attacker could create a second admin account with arbitrary credentials and no 2FA, gaining full administrative access to the CMS.

Affected Code

install.php step 4 handler (pre-0.7.157):

```
$hash = password_hash($admin_pass, PASSWORD_DEFAULT);
$stmt = $pdo->prepare("INSERT IGNORE INTO `{$prefix}users`
    (username, email, password_hash, user_role, preferred_skin)
    VALUES (?, ?, ?, 'admin', 'midnight-lime')");
$stmt->execute([$admin_user, $admin_email, $hash]);
```

No check was made for existing admin users before executing the insert.

Attack Scenario

1. Target site has a functioning SnapSmack install with an admin user who has TOTP 2FA enabled.

2. install.php was not deleted after install (or was re-uploaded).

3. Attacker navigates to install.php, passes step 1 (environment check), step 2 (enters any valid DB credentials), step 3 (schema install — IF NOT EXISTS means no damage to existing tables).

4. In step 4, attacker submits a new username (e.g. 'support'), any email, and a known password.

5. INSERT IGNORE skips if 'support' is unique-constrained to an existing row — but since 'support' is a new username, the insert succeeds.

6. New admin account created with totp_enabled = 0. Attacker logs in at /snap-in with support / chosen password — no 2FA prompt.

7. Full admin access to content, settings, credentials, and file paths.

Fix

Before executing the insert, check for any existing admin user:

```
$existing_admin = $pdo->query(
    "SELECT COUNT(*) FROM `{$prefix}users` WHERE user_role = 'admin' LIMIT 1"
```

```
)->fetchColumn();
if ($existing_admin > 0) {
    $errors[] = 'An admin account already exists on this database. '
              . 'The installer cannot create another. '
              . 'Delete install.php and log in normally.';
    $step = 4;
}
```

If any admin exists, step 4 is hard-blocked with an error message directing the user to delete install.php and log in through the normal login page.

Notes

install.php self-deletes on successful completion of step 5. The attack window exists only if the installer was not completed (failed partway through) or was manually re-uploaded.

The probe-ban list in .htaccess blocks install.php on established installs — but .htaccess itself is generated by the installer and may not exist on a partial install.

The fix does not affect fresh installs: on a truly empty database the snap_users table either doesn't exist (caught by the PDOException) or is empty, and step 4 proceeds normally.

The new account would have totp_enabled = 0 regardless of other users' 2FA status, since 2FA enrollment is per-user and not enforced at account creation time.