SECAUDIT 035 - Client-address spoofing across the IP ban subsystem (IP SMACKER and the login shield)

SECURITY AUDIT / PUBLIC REMEDIATION RECORD

Field	Value
Audit ID	2026-07-27-035
Date	2026-07-27
Severity	**CRITICAL** - login brute-force protection can be evaded entirely, and an arbitrary address (including the owner's) can be banned by an unauthenticated request. Availability and authentication impact; no direct confidentiality or integrity impact.
Component	`snap_ip_bans` and every producer/consumer of it: `probe-ban.php` (IP SMACKER), `snap-in.php` (`snap_client_ip()`, login ban gate, brute-force counter), `core/smackverse.php` (inbox limiter), `core/flkrfckr-api.php` (auth limiter), `smack-fingerprints.php` (admin view)
Status	**CLOSED in 0.7.454** - trusted client resolution, guarded fixed-lifetime writers, recoverable historical cleanup, bounded pruning, and owner lockout alerting are complete
Reporter	Sean (persistent 403 on squared.pixhellated.ca; questioned what the ban table was storing) + Claude (traced address resolution, ran the fleet sweep, traced the shared-table consumers)
Related	**005 (login hardening / IP shield - this report supersedes its threat model)**, 021/021A (hub-spoke attack surface), 017 (SMACKBACK), 034 (closure discipline - one route removed is not proof the capability is gone)
Disclosure	No targeted exploitation is known. Forged addresses are present in 16 of 16 site databases from ordinary scanner traffic (section 6), so the weakness is being exercised in the wild. Reachable by any unauthenticated visitor.

1. Summary

SnapSmack maintains one shared ban table, `snap_ip_bans`. Four subsystems write to it and four read from it. Two of them resolve the client address from request headers that the client itself controls.

`probe-ban.php` (IP SMACKER) traps well-known scanner paths and records a 30-day ban. `snap-in.php` gates login, counts failed attempts, and issues a 7-day brute-force ban. Both derive "the client" from `X-Forwarded-For` without first establishing that the request arrived through a proxy entitled to set it. On a direct request that header is supplied by the attacker.

Three consequences follow, in ascending order of severity.

The ban table is being filled with addresses that are not real. This is confirmed live: 16 of 16 site databases hold private-range rows, including `127.0.0.1` on every single site.

Whenever a scanner sends its own `X-Forwarded-For`, IP SMACKER bans the forged value and **the actual scanner is never banned**. The control does not merely have a hole; against any attacker who sets one header it does not function at all.

And because `snap-in.php` keys its brute-force counter on the same spoofable value, an attacker who varies `X-Forwarded-For` on each login attempt never accumulates failures against any single key and is **never rate-limited or banned** - defeating the protection SECAUDIT 005 was written to establish. The same primitive run in reverse, with the header set to a victim's address, bans that address for seven days at the login gate.

The subsystem requires the address that actually connected. It accepts whatever the client claims. That is the trust boundary this report is about.

2. Finding A - login brute-force protection is bypassable (CRITICAL)

2.1 The code

`snap-in.php`:

```
function snap_client_ip(): string {
$fwd = $_SERVER['HTTP_X_FORWARDED_FOR'] ?? '';
if ($fwd !== '') {
$first = trim(explode(',', $fwd)[0]);
if (filter_var($first, FILTER_VALIDATE_IP)) return $first;
}
return filter_var($_SERVER['REMOTE_ADDR'] ?? '', FILTER_VALIDATE_IP)
? $_SERVER['REMOTE_ADDR'] : '0.0.0.0';
}
```

This validates that the header is a syntactically valid address, which is more than `probe-ban.php` does, but format validation is not authenticity. No check establishes whether the request came from a trusted proxy. Any client may set the header, and a spoofed value that parses as an address is preferred over the real peer.

The returned value feeds both the ban gate and the failure counter:

```
$_snap_ip = snap_client_ip();
// ... SELECT expires_at FROM snap_ip_bans WHERE ip = ? AND expires_at > NOW()
// ... $fail_count >= 5 -> INSERT ... 'auto:brute_force' ... INTERVAL 7 DAY
```

2.2 Evasion

An attacker sends a distinct `X-Forwarded-For` with each login attempt. Each attempt is counted against a different key, so no key ever reaches the five-failure threshold. Password guessing proceeds without limit, and no ban is ever issued. The five-attempt lockout, the 7-day ban, and the ban gate are all inert against an attacker who sets one header.

2.3 Targeted lockout

The same primitive inverted: send five failed logins carrying `X-Forwarded-For: <victim address>`. The victim is banned for seven days. The gate at the top of `snap-in.php` then refuses that address, so the target - including the site owner - cannot reach login. Combined with Finding B this is sustainable indefinitely.

2.4 Impact

Authentication hardening is the security property SECAUDIT 005 established. This finding removes it against any attacker aware of the header, while simultaneously providing a remote lockout primitive against the owner. That combination is why this report is rated CRITICAL rather than HIGH.

3. Finding B - IP SMACKER records an unvalidated, client-supplied address (HIGH)

`probe-ban.php`:

```
$ip = trim(explode(',', (
$_SERVER['HTTP_CF_CONNECTING_IP']
?? $_SERVER['HTTP_X_FORWARDED_FOR']
?? $_SERVER['REMOTE_ADDR']
?? '0.0.0.0'
))[0]);
```

`REMOTE_ADDR` is set by the web server from the actual TCP peer and cannot be forged. The two headers preferred above it can be set by anyone, and neither `CF-Connecting-IP` nor `X-Forwarded-For` is checked against a trusted-proxy allowlist. Unlike `snap_client_ip()`, this path applies **no** validation at all - the value is written to the database as received.

Taking element `[0]` of `X-Forwarded-For` is correct for a trusted proxy chain, where the leftmost entry is the original client. It is exactly wrong for untrusted input, because the leftmost entry is the part the client controls.

Delivery is one unauthenticated request: `GET /xmlrpc.php` with `X-Forwarded-For: <target>`. The trapped paths are not secret; they are the standard scanner paths and are enumerable from the shipped `.htaccess`.

Because the table is shared (section 5), a ban injected here is enforced by the login gate, the SMACKVERSE inbox limiter, and the FLKR FCKR limiter.

4. Finding C - the control fails at its stated purpose (HIGH)

This is distinct from the security findings and is arguably the more important operational result.

When a scanner supplies its own `X-Forwarded-For`, IP SMACKER records the forged value. The scanner's real address is never written and never banned. It continues probing, unimpeded, while the ban table accumulates fictional entries.

The evidence in section 6 shows this happening: three documented bursts of randomised RFC1918 addresses, each burst spanning three to five minutes, each representing a scan during which the responsible party was not banned even once.

IP SMACKER therefore provides protection only against unsophisticated clients that send no forwarded header. Any scanner that sets one - which is common behaviour, both to obscure origin and to probe for header-handling flaws - is immune by default. The feature has been providing materially less protection than its presence implies.

5. Finding D - one shared ban table, four consumers, two trust levels (HIGH)

`snap_ip_bans` is written by `probe-ban.php` (`auto:probe`, 30 days), `snap-in.php` (`auto:brute_force`, 7 days), `core/smackverse.php` (`auto:smackverse_inbox`, 24 hours), and `core/flkrfckr-api.php` (`auto:flkrfckr_auth`, 7 days). It is read as a gate by `snap-in.php`, `sv_inbox_rate_ok()`, and `flkrfckr_ip_banned()`.

Two of those producers resolve the address safely and two do not:

Consumer	Resolution	Verdict
`snap-in.php` / `snap_client_ip()`	`X-Forwarded-For` preferred, format-validated, no proxy boundary	**Vulnerable**
`probe-ban.php`	`CF-Connecting-IP` then `X-Forwarded-For`, no validation, no proxy boundary	**Vulnerable**
`core/flkrfckr-api.php` / `flkrfckr_client_ip()`	`REMOTE_ADDR` only	**Correct**
`smackverse.php` inbox route	`REMOTE_ADDR` passed directly	**Correct**

The correct pattern already exists in the codebase twice. The defect is that two call sites diverge from it, and because the table is shared, the two vulnerable writers can poison the gate for all four readers. A forged ban written through a scanner path is enforced against login.

`smack-fingerprints.php` provides admin `fetch_ip_bans` and `lift_ip_ban` actions, so a release path exists - but it is not surfaced as a prompt when a ban removes the owner's own access, which is how the incident in section 6 became a prolonged outage rather than a noticed event.

6. Evidence

6.1 Operator self-ban

`squared.pixhellated.ca` returned a persistent bare 403 that outlived the previously diagnosed provisioning window. `snap_ip_bans` held one row:

```
2001:56a:7763:5000:6838:fc58:b813:6439 | auto:probe | 2026-07-27 02:31:34 | 2026-08-26 02:31:34
```

That is the operator's own public IPv6 address (Telus prefix), banned during overnight 2FA-lockout work. No hostile party was involved. Its value is as demonstration: the subsystem can remove administrative access to a site, did so, gave no notification, and set an expiry a month out.

6.2 Fleet sweep

Every site database on the shared MariaDB host was swept for private and reserved ranges. **16 of 16 databases returned rows**, approximately 279 in total:

```
bad 30, colour_less 49, craptasti_ca 2, fauxlaroid 6, forever 17,
foundtextures_ca 23, hekeepsdroningon_ca 26, hockney_joiner 2,
lightafter 4, photowalk_ing 42, pixhellated_ca 15, squared_straight 2,
strathmore_pics 9, theschoolof 34, unzucked 7, wateronthebrain_ca 11
```

The count is an upper bound: the sweep predicate matched `172.%`, which over-matches, since only `172.16.0.0/12` is private.

6.3 Two distinct patterns

`127.0.0.1` **appears in all 16 databases.** Loopback can only be the site's own infrastructure. The most likely mechanism is a probe arriving through a same-host reverse proxy or tunnel without `CF-Connecting-IP` set, so `REMOTE_ADDR` is loopback and the real origin is discarded. Every such event both fails to ban the attacker and bans the site's own local traffic path.

Randomised RFC1918 addresses in tight bursts. Examples:

```
foundtextures_ca 192.168.253.99 / 192.168.232.116 / 192.168.63.61 /
192.168.109.184 / 10.112.241.1 2026-06-29 15:13-15:16
hekeepsdroningon 10.87.122.1 / 10.172.207.1 / 192.168.28.96 /
10.233.67.1 / 192.168.251.137 2026-06-29 18:49-18:52
photowalk_ing 10.18.142.1 / 10.190.2.1 / 10.37.42.1 /
192.168.112.33 / 192.168.128.98 2026-06-30 16:12-16:17
```

Scattered addresses across unrelated /8 and /16 ranges, many ending `.1`, arriving within minutes of each other, are not internal traffic. They are randomised `X-Forwarded-For` values from a scanning tool, recorded verbatim. Each burst is a scan that IP SMACKER did not stop.

7. Enforcement risk from the loopback bans

Because `127.0.0.1` carries an active ban on every site and the gates check `snap_ip_bans` before doing work, any component whose requests present as loopback can be refused. `sv_inbox_rate_ok()` is called with `REMOTE_ADDR` directly, so a local delivery worker or same-host inbox POST would be rejected while the ban is live. This should be treated as a probable cause for unexplained internal failures during the affected windows and verified against the SMACKVERSE delivery backlog history.

8. Remediation

1. **One shared, trustworthy resolver.** Introduce a single

`snap_trusted_client_ip()` and route all four subsystems through it. Duplicated resolution logic is what allowed two call sites to diverge.

2. **Establish a proxy trust boundary.** Honour `CF-Connecting-IP` or

`X-Forwarded-For` only when `REMOTE_ADDR` is a member of a configured trusted-proxy allowlist. Otherwise use `REMOTE_ADDR`. An unconfigured install must default to `REMOTE_ADDR` alone. This single change closes Findings A, B, and C.

3. **Validate before recording.** Require

`filter_var($ip, FILTER_VALIDATE_IP, FILTER_FLAG_NO_PRIV_RANGE | FILTER_FLAG_NO_RES_RANGE)` before any ban insert. On failure still return 403 - refusing the request is correct - but record no ban.

4. **Never-ban allowlist.** Loopback, the configured proxy and tunnel

addresses, and hub/spoke mesh members must be unbannable regardless of path.

5. **Stop renewing on duplicate.** All four writers use

`ON DUPLICATE KEY UPDATE ... banned_at = NOW()`, so a ban is a rolling window rather than a fixed one and cannot age out unattended. Preserve the original `banned_at`, or cap total extension.

6. **Bound the table.** Prune expired rows on a schedule and rate-limit

insertion of new distinct addresses per interval, or an attacker rotating the header can grow the table without limit - shared-fate on a consolidated database host.

7. **Surface owner lockout.** An active ban matching the owner's address should

be visible and releasable without database access. `smack-fingerprints.php` already has `fetch_ip_bans` and `lift_ip_ban`; the gap is notification.

8. **Purge existing rows** for private, reserved, and malformed values across

the fleet, and audit remaining `auto:brute_force` entries, which may be forged lockouts rather than real attackers.

Remediation must not weaken the traps. Scanner paths should still be refused and failed logins still counted. Only the identity the ban is recorded against changes.

9. Disposition

CLOSED in 0.7.454. The 0.7.453 trust boundary gates all four known ban writers and both shared-address rate limiters. In 0.7.454 those producers moved to one bounded writer: active duplicates preserve their original fixed expiry, expired and unsafe rows are pruned, stale limiter counters are removed, and ten-minute plus total-row caps prevent rotating-address storage exhaustion.

The normal updater creates a recoverable database backup before the new code goes live. On the first request after update, a one-time marked cleanup clears all potentially forged pre-fix `auto:*` rows while preserving manual moderation; valid new hostile traffic is immediately eligible to be banned again through the corrected resolver. A newly imposed administrator-login ban also sends a best-effort owner email with expiry and recovery directions. Permanent regression tests cover the trust boundary, unsafe targets, all four writer integrations, fixed-lifetime routing, cleanup, bounds, alerts, and tunnel-safe limiter keying.