CVE RESEARCH

CVE-2026-37603 & CVE-2026-37604: Admin Brute-Force Protection Bypass Chain in pH7Builder

CVE: CVE-2026-37603 (CWE-307) · CVE-2026-37604 (CWE-290)CVSS v3.1: 6.5 (Medium) for each vulnerabilityAffected versions: ≤ 18.2.0 (both) · ≤ 19.2.0 (CVE-2026-37603 only)Fixed in: 18.3.0 for CVE-2026-37604. CVE-2026-37603 unfixed at time of writing An attacker who can reach the administration panel of an affected pH7Builder installation over the network can guess administrator passwords without any of the […]

Haluk Baran Akbulut September 19, 2026 42 Minutes Read

Cookie Usage

We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.

Cookie Policy



CVE: CVE-2026-37603 (CWE-307) · CVE-2026-37604 (CWE-290)

CVSS v3.1: 6.5 (Medium) for each vulnerability

Affected versions: ≤ 18.2.0 (both) · ≤ 19.2.0 (CVE-2026-37603 only)

Fixed in: 18.3.0 for CVE-2026-37604. CVE-2026-37603 unfixed at time of writing

An attacker who can reach the administration panel of an affected pH7Builder installation over the network can guess administrator passwords without any of the limits the application is meant to impose. No attacker account or prior authentication is required, and no action by a legitimate user is needed. The product ships three measures to make password guessing expensive: a challenge that appears after repeated failures, a counter of failed attempts, and a temporary block on the source of those attempts. All three can be made ineffective, leaving the attack bounded by network and server capacity rather than by any authentication control. In the laboratory, up to correct password was recovered without either intended control being triggered.

Cookie Usage
We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.
Cookie Policy

← Back to CyberMap

Both flaws are rate-limiting failures of the same shape: the state that decides whether to challenge or block a request is stored under a key the requester chooses. CVE-2026-37603 keys the challenge on the PHP session, which any client can obtain on demand. CVE-2026-37604 keys the block on a client IP address read from request headers. The findings were reproduced in an isolated lab against 18.1.0 and then traced through the published source of every tagged release, from 1.2.7 (December 2015) to 19.2.0 (see Section 9 and Section 10 for the root cause, Section 11 for the lab work, and Section 11.3 for what upstream has and has not fixed).

Contents

- 1. Executive Summary
- 2. Legal Notice and Scope
- 3. Disclosure Timeline and Vendor Coordination
- 4. Terminology
- 5. Vulnerability Identification
- 6. Background
- 7. Attack Surface and Preconditions
- 8. Attack Flow
- 9. Link 1 — Session-Bound Challenge State (CVE-2026-37603)
- 9.1 Principle Violated
- 9.2 The Vulnerable Code
- 9.3 Step-by-Step Trace with Concrete Values
- 9.4 Analogy
- 9.5 Architectural Root Cause
- 10. Link 2 — Attacker-Controlled Client-IP Resolution (CVE-2026-37604)
- 10.1 Principle Violated
- 10.2 The Vulnerable Code
- 10.3 Step-by-Step Trace with Concrete Values

Cookie Usage
We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.
Cookie Policy

- 10.4 Analogy
- 10.5 Architectural Root Cause
- 11. Hands-On Demonstration
- 11.1 Lab Environment
- 11.2 Step-by-Step Verification
- 11.3 Patch Status and Regression Testing
- 12. Impact Analysis
- 13. Detection and Log Analysis
- 14. Remediation and Secure Coding Lessons
- 14.1 Secure Coding Lessons
- 15. Real-World Context
- 16. Conclusion
- 17. References
- Appendix A — Glossary

← Back to CyberMap

---

1. Executive Summary

This is a chain of two independent flaws rather than a single defect. pH7Builder layers three defences over the administration login: a CAPTCHA challenge that appears after a failure, a counter of failed attempts, and a timed lockout of the offending source. All three collapse into two decisions, and each of those decisions is made from data the client supplies. Each credential guess can therefore be presented under a fresh session and, on versions affected by the second flaw, under a fresh rate-limit key as well. The two links are not interchangeable: CVE-2026-37603 on its own removes the CAPTCHA escalation but leaves the per-address attempt counter intact, and this CVE-2026-37604 that removes that remaining bound.

Cookie Usage

We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.

Cookie Policy

← Back to CyberMap

```
Fresh session per attempt  →  no challenge, no escalation  →  fresh claimed IP
(CVE-2026-37603)                    (CWE-307)                    (CVE-2026-37604)
```

#	LINK	CLASS (CWE)
1	**Session-bound challenge state** (CVE-2026-37603)	CWE-307 (Improper Res
2	**Attacker-controlled client-IP resolution** (CVE-2026-37604)	CWE-290 (Authenticati

Key insight: neither link is a probabilistic race or a timing artefact. Both are deterministic lookups against a key the attacker obtains at no cost, which is why the outcome is not a weakened throttle but an inapplicable one: on every request, the state the application consults has no record of the attacker to escalate against.

---

2. Legal Notice and Scope

Legal notice: this article is published solely for security awareness, defence, vulnerability management, and education. The techniques described must only be reproduced in authorised, isolated laboratory environments. Scanning, exploitation attempts, unauthorised access, or any action that may cause service disruption against systems you do not own or have explicit written permission to test may constitute a criminal offence. In Türkiye, unauthorised access to and interference with information systems is addressed under the Turkish Penal Code No. 5237 (Articles 243–245).

Cookie Usage
We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.
Cookie Policy

Scope and PoC policy: this analysis explains the root cause at source level, with the raw HTTP request and response pairs needed to understand and detect the attack. End-to-end automation tooling is not published. Only its operating logic is summarised (see Section 11.2). All testing was performed on an isolated laboratory network with no internet egress, against

an installation of pH7Builder 18.1.0 deliberately left unpatched for the purpose. No third-party system was touched, and the only account attacked was one created by the researcher. Source-level claims about releases other than 18.1.0 were made by reading the public repository at each release tag, not by attacking any live installation.

3. Disclosure Timeline and Vendor Coordination

This section is an addition to the standard structure, included because this article is the disclosure itself rather than a commentary on someone else's. Individual contact addresses are omitted deliberately.

DATE	EVENT
2026-03-04	Both issues identified during security research on pH7Builder 18.1.0.
2026-03-05	Vendor notified through the project's published contact address, the maintainer's
2026-03-12	CVE assignment requested from MITRE after seven days without a vendor respons
2026-04-06	CVE-2026-37603 and CVE-2026-37604 reserved by MITRE.
2026-06-08	Release 18.3.0 published. It contains a fix for the IP-resolution flaw (CVE-2026-376
2026-09-19	Public disclosure. The 90-day window has long elapsed, the session-reset flaw (C
2026-09-22	Both CVE Records published by MITRE CNA-LR.

Stated as observation rather than inference: a fix for one of the two reported issues shipped roughly three months after notification, and no advisory or changelog entry identified the

← Back to CyberMap

Cookie Usage

We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.

Cookie Policy

security relevance of that change. No correspondence was received at any point. The CVE records are reserved but not yet published, so the `cve.org` links in Section 17 may return an unpopulated record until MITRE publishes them.

← Back to CyberMap

4. Terminology

TERM	DEFINITION
CVE	Common Vulnerabilities and Exposures — the standard identifier scheme for pub
CVSS	Common Vulnerability Scoring System — a scheme for scoring technical severity
CWE	Common Weakness Enumeration — a taxonomy of software and hardware weakn
PoC	Proof of Concept — a controlled demonstration that a flaw exists
XFF	`X-Forwarded-For` — an HTTP header conventionally used by proxies to record th
Rate-limit key	The value a rate limiter groups requests under when deciding whether a threshol

5. Vulnerability Identification

Cookie Usage

We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.

Cookie Policy

Back to CyberMap

FIELD	CVE-2026-37603
Product / component	pH7Builder (pH7 Social Dating CMS) — admin login form, `admin123`
Vendor	pH7Software / Pierre-Henry Soria
Affected versions	All versions through 19.2.0
Fixed in	Not fixed at time of writing
Vulnerability class	Authentication throttling bypass via session reset
CWE	CWE-307 — Improper Restriction of Excessive Authentication Att
CVSS v3.1 base score	6.5 — Medium
CVSS vector	`CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N`
Attack vector	Network
Privileges required	None
User interaction	None
MITRE ATT&CK	T1110 — Brute Force
Reservation / disclosure date	Reserved 2026-04-06 / disclosed 2026-09-19 / record publishe
Public exploit available	No public exploit at time of writing
Researcher	Haluk Baran Akbulut (CyberMap Group)

Cookie Usage
We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.

Cookie Policy

← Back to CyberMap

Note on corroboration: at the time of writing both CVE Records were published by MITRE CNA-LR on 2026-05-22 and cite this article as a reference. MITRE CNA-LR does not record CVSS scores, CWE mappings, product details or researcher credit in the records it publishes, and directed that those be documented in the public reference instead; they are therefore given in the table above. Both IDs were requested and reserved by the author. The version ranges were established by checking both code paths at every one of the 141 release tags in the vendor's repository, the earliest being 1.2.7 dated 2015-12-24; both are present unchanged from that tag onward. Scores are the author's own assessment under CVSS v3.1 and have not been reviewed by a CNA; NVD may assign its own.

---

6. Background

pH7Builder (formerly pH7CMS) is an open-source social networking and dating CMS written in PHP. Its administration panel lives, by default, under the `/admin123/` path, and the login form at `/admin123/main/login` is the component under analysis.

What this component does under normal conditions:

1. A `GET` request to the login URL starts a PHP session, issues a `PHS7SESS` cookie, and renders the login form with a CSRF token bound to that session.
2. The form is assembled conditionally: an email field, a username field, a password field, and, *only if the current session is marked as having already failed a login*, a CAPTCHA field.
3. A `POST` is validated element by element against the form definition, then the submitted credentials are checked against the `admins` table.
4. On failure, the application records the attempt in a per-IP counter table, marks the session so the next render of the form includes a CAPTCHA, and sleeps briefly before responding.
5. If the per-IP counter has reached the configured maximum, the login handler refuses to evaluate credentials at all until a timeout expires.

Cookie Usage

We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.

Cookie Policy

← Back to CyberMap

```
// Intended shape of the control, as expressed in LoginFormProcess.php
if ($bIsLoginAttempt && !$oSecurityModel->checkLoginAttempt(/* ... */)) {
    \PFBC\Form::setError('form_admin_login', Form::loginAttemptsExceededMsg($
    return; // credentials are never evaluated
}
// ... credentials checked; on failure:
$oSecurityModel->addLoginAttempt(DbTableName::ADMIN_ATTEMPT_LOGIN);
$this->enableCaptcha();
```

The design is sound in outline: escalate to a challenge after one failure, refuse entirely after several. The break is not in the escalation logic but one level below it, in how the application decides *which* counter and *which* challenge state a given request belongs to. See Section 9 and Section 10 for the source-level detail.

---

7. Attack Surface and Preconditions

PRECONDITION	REQUIRED?	NOTES
pH7Builder within the affected range	Yes	Version is shown in the admin foo
Admin login endpoint reachable	Yes	Default path `/admin123/main/lo`
Authentication	None	Both flaws are exercised by an er
User interaction	None	No administrator action is neede
`isAdminLoginAttempt` enabled	Not required	Default: `1` (enabled). The chain

Cookie Usage
We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.
Cookie Policy

← Back to CyberMap

PRECONDITION	REQUIRED?	NOTES
Application behind a reverse proxy	No (for ≤ 18.2.0)	Header trust is unconditional in t
Network position	Network-reachable	Any client with network access t

Default configuration assessment: on an affected version, a default installation is vulnerable, and no non-default setting is needed to reach either flaw. The shipped defaults are `isAdminLoginAttempt = 1`, `maxAdminLoginAttempts = 10` and `loginAdminAttemptTime = 120` minutes. In other words the product ships with brute-force protection *enabled*, and the chain defeats it in that configuration. No unusual setting, plugin or deployment topology is required.

Non-destructive exposure check:

```
# Confirms the admin panel is reachable and issues a session cookie.
# Sends no credentials and creates no failed-login record.
curl -sSI "https://TARGET_HOST/admin123/main/login" | grep -Ei '^(HTTP/|set-c
```

A response carrying a `Set-Cookie: PHS7SESS=...` header confirms the endpoint is exposed and that a fresh session is handed out on request. Whether the installation is *affected* then depends on the version.

---

8. Attack Flow

Cookie Usage
We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.
Cookie Policy

Under normal operation the CAPTCHA state and the lockout counter are meant to accumulate across an attacker's attempts, so that the tenth request is treated differently

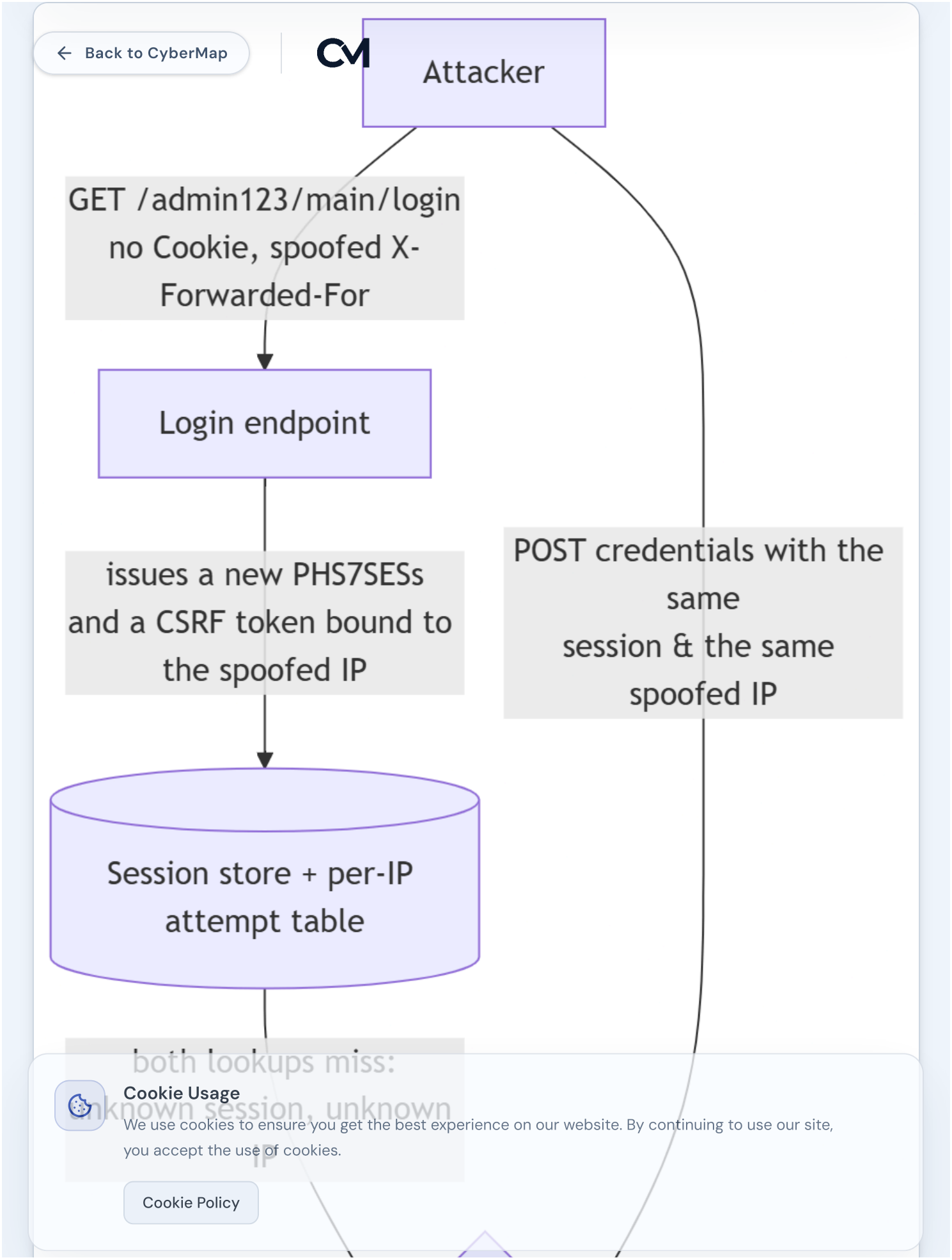
from the first. What connects the attacker's requests in the application's mind is not the attacker but a session cookie and an IP string, both of which travel in the request itself.

← Back to CyberMap

Cookie Usage

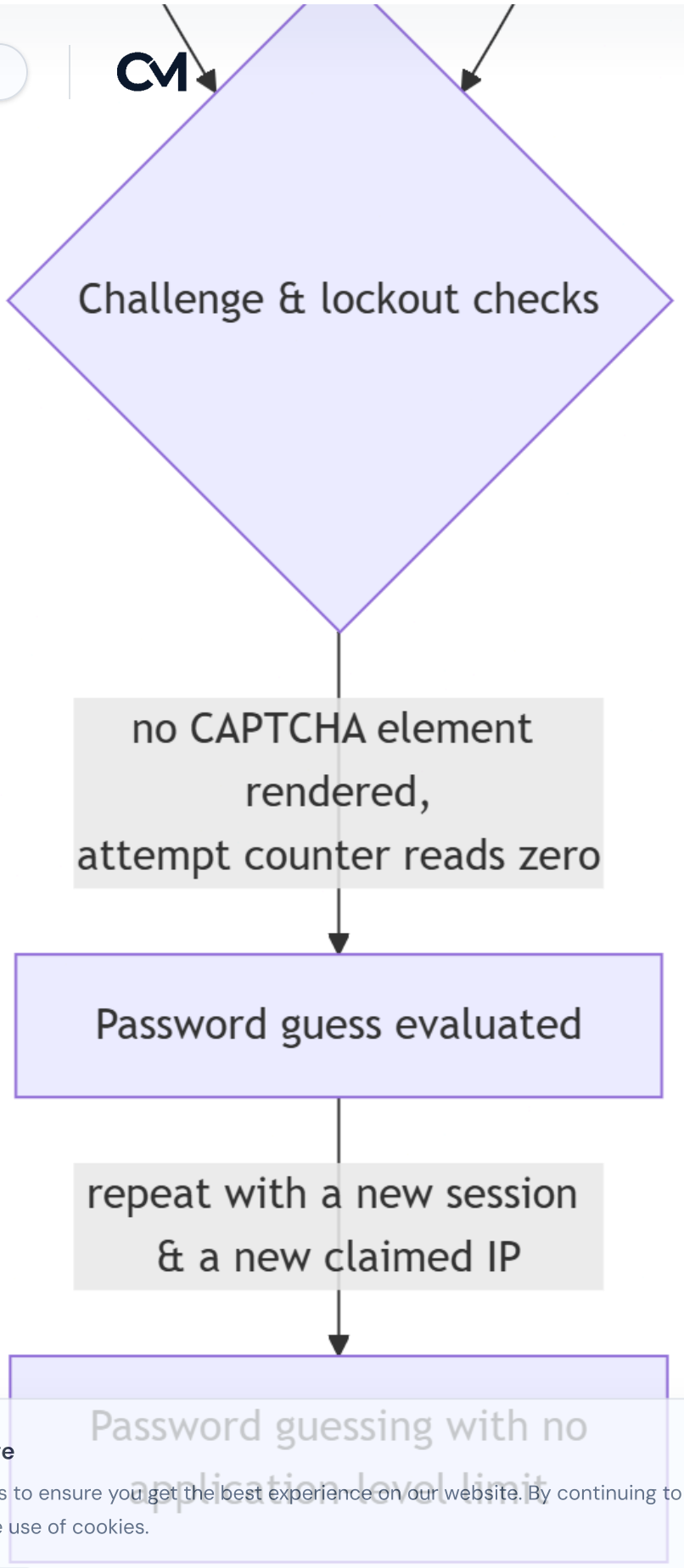
We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.

Cookie Policy



Cookie Usage
We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.
Cookie Policy

Back to CyberMap



Cookie Usage
We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.
Cookie Policy

Every guess is the application's first encounter with a brand-new visitor, so the escalation

Back to CyberMap

Cookie Usage

We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.

Cookie Policy

logic never has arbitrary to escalate against.

9. Link 1 — Session-Bound Challenge State (CVE-2026-37603)

← Back to CyberMap

9.1 Principle Violated

A control that escalates after failure must store its state under a key the failing party cannot choose. The key has to be bound to the thing being protected, such as the account, or to something the client cannot create at will. A session cookie is neither: for an unauthenticated visitor a session is free, unlimited, and issued on request.

pH7Builder stores the escalation state in `$_SESSION`, under the key `captcha_admin_enabled`, and reads it back when deciding whether to *build* a CAPTCHA field into the form. Because the session is selected by a cookie the client sends, the client also selects which escalation history applies to its request.

Expected behaviour:

```
POST attempt #1 (wrong password)  → "Password is incorrect"
POST attempt #2 (wrong password)  → CAPTCHA required, credentials not evalua
```

Vulnerable behaviour:

```
GET (no Cookie) → PHS7SESS=<S1>;  POST with <S1> (wrong password) → "Passwo
GET (no Cookie) → PHS7SESS=<S2>;  POST with <S2> (wrong password) → "Passwo
GET (no Cookie) → PHS7SESS=<S3>;  POST with <S3> (wrong password) → "Passwo
```

Technical nuance: the CAPTCHA is not being submitted and accepted incorrectly. It is never instantiated at all. The form engine validates only the elements present in the form definition, and the CAPTCHA element is added conditionally at render time. For a fresh session there is no CAPTCHA element, therefore no CAPTCHA validator, therefore nothing to bypass. This

Cookie Usage
We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.
Cookie Policy

matters for anyone testing the control: submitting a wrong CAPTCHA value produces a correct-looking rejection, and a tester who stops there will conclude the defence works.

← Back to CyberMap

9.2 The Vulnerable Code

```
// _protected/app/system/modules/admin123/forms/LoginForm.php

// ...
if (static::isCaptchaEligible()) {           // VULNERABILITY HERE:
    $oForm->addElement(new CCaptcha(/* ... */));  // the challenge exists only
}                                                 // client's own session says
// ...

public static function isCaptchaEligible(): bool
{
    return (new Session)->exists('captcha_admin_enabled');
}
```

```
// _protected/app/system/modules/admin123/forms/processing/LoginFormProcess.ph

public function enableCaptcha(): void
{
    // VULNERABILITY HERE: escalation state written to client-selected storage
    $this->session->set('captcha_admin_enabled', 1);
}
```

The developer assumption is that a session identifies a client. For an authenticated area that assumption is workable, because the session is the only way to hold the identity. For the *login* form it collapses, because the party being throttled has not authenticated yet and can therefore hold as many identities as it cares to ask for. The application writes "this visitor has failed once" into a token it hands to the visitor, and then trusts the visitor to bring that token back.

Cookie Usage
We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.
Cookie Policy

9.3 Step-by-Step Trace with Concrete Values

← Back to CyberMap

The values below are illustrative and were not taken from any live target.

INPUT	CONTENT	PURPOSE
Session S1	`PHS7SESS=0f2a7c9b1d4e6a83`	First convers
Session S2	`PHS7SESS=b71e05d8c3a94f20`	Second conv
Credentials	`mail=admin@TARGET_HOST`, `username=admin`, `password=<guess>`	The guess ur

Step 1 — first POST under S1 with a wrong password:

- `checkLoginAttempt()` → per-IP counter row absent → **TRUE** (authorised to try)
- `adminLogin()` → **FALSE**
- `addLoginAttempt()` → per-IP row created, `attempts = 1`
- `enableCaptcha()` → `captcha_admin_enabled` written into S1

```
$_SESSION[S1] = { captcha_admin_enabled: 1 }
$_SESSION[S2] = (does not exist)
ph7_admins_attempts_login = [ { ip: <client ip>, attempts: 1 } ]
```

Step 2 — second POST under S1:

- `LoginForm::isCaptchaEligible()` → **TRUE** → CAPTCHA element added to the form
- Form validation fails on the CAPTCHA element, and credentials are never evaluated
- Response: *"The code of Captcha entered was incorrect. Please re-try."*

Step 3 — `GET` with no `Cookie` header, then POST under S2:

Cookie Usage
We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.
Cookie Policy

- A new session is started, and the response carries a `Set-Cookie: PHS7SESS=b71e…` header
- `isCaptchaEligible()` reads S2 → **FALSE** → no CAPTCHA element is built
- Credentials are evaluated normally

```
$_SESSION[S1] = { captcha_admin_enabled: 1 }        ← still set, and now irre
$_SESSION[S2] = { }                                  ← the request is judged
ph7_admins_attempts_login = [ { ip: <client ip>, attempts: 2 } ]   ← the only
```

Result: the guess in Step 3 was evaluated under a rule set that had no record of Steps 1 and 2, and the only counter that still remembers them is keyed on the client's IP address, which is precisely the key Link 2 takes control of.

9.4 Analogy

A club stamps the back of your hand when you are turned away at the door, and the bouncer checks for a stamp before letting anyone argue their case again. The stamp is on *your* hand, not in the bouncer's notebook. Wash your hands on the way round the block and you are, as far as the door is concerned, a person who has never been refused.

9.5 Architectural Root Cause

This is an instance of **CWE-307 — Improper Restriction of Excessive Authentication Attempts**, and it is a design assumption rather than an implementation slip: there is no line to correct here, because the escalation logic does exactly what it was written to do. The assumption that fails is *"a session is a client."* A session identifies a conversation, and an unauthenticated party can open as many conversations as it likes at no cost. Any control whose purpose is to make repetition expensive must therefore be keyed on something repetition cannot renew, such as the account being targeted or a server-side record of the source, and must be consulted before credentials are evaluated rather than rendered into the form afterwards. The same assumption shows up wherever a failure counter, a cooldown, a one-time-code attempt limit, or a "you must solve a challenge now" flag is written to `$_SESSION` on a pre-authentication path.

← Back to CyberMap

Cookie Usage
We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.
Cookie Policy

Back to CyberMap

10. Link 2 — Attacker-Controlled Client-IP Resolution (CVE-2026-37604)

10.1 Principle Violated

The client address used in a security decision must come from the transport layer, meaning the peer address of the TCP connection, exposed to PHP as `REMOTE_ADDR`. Forwarded-address headers are hints written by whoever spoke last, and they may only be believed when the immediate peer is a proxy the operator controls.

In the affected range, `Ip::get()` resolves the client address by returning the first populated value among `HTTP_CLIENT_IP`, `HTTP_X_FORWARDED_FOR` and `REMOTE_ADDR`. Since the first two are populated directly from client-supplied request headers, any client can declare its own address, and `REMOTE_ADDR` is consulted only when the client chooses not to.

Expected behaviour:

```
X-Forwarded-For: 203.0.113.42   →  Ip::get() returns the TCP peer address
```

Vulnerable behaviour:

```
X-Forwarded-For: 203.0.113.42   →  Ip::get() returns 203.0.113.42
```

Technical nuance: the value must be a syntactically valid, non-private address to be useful. `Ip::get()` passes its result through `isPrivate()`, which returns TRUE for anything that is *not* a valid public address, including malformed strings and comma-separated lists, and rewrites it to `127.0.0.1`. Spoofing `192.168.x.x`, or sending a realistic proxy chain such as `X-Forwarded-For: 203.0.113.42, 198.51.100.7`, therefore lands every request in one shared

Cookie Usage
We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.
Cookie Policy

← Back to CyberMap

`127.0.0.1` bucket and *triggers* the lockout faster rather than evading it. Only a single well-formed public address per request produces a fresh counter, which is why the observable signature of this attack is a sequence of clean, distinct, public XFF values rather than a chain.

Technical nuance: the same spoofed value must appear on the `GET` that mints the CSRF token and on the `POST` that spends it. The token is stored alongside `Ip::get()` at generation time and compared against `Ip::get()` at validation time, so a session obtained under one claimed IP cannot be used to submit a form under another. This is why the two links are not merely additive but interlocked: each attempt needs its own *pair* of session and claimed address, consistently applied across both requests.

10.2 The Vulnerable Code

```
// _protected/framework/Ip/Ip.class.php (18.1.0)

private static function getClientIp()
{
    // VULNERABILITY HERE: two of these three sources are request headers
    $aVars = [Server::HTTP_CLIENT_IP, Server::HTTP_X_FORWARDED_FOR, Server::RE

    foreach ($aVars as $sVar) {
        if (Server::getVar($sVar) !== null) {
            return Server::getVar($sVar);   // first match wins, so the reques
        }
    }

    return '';
}
```

Cookie Usage
We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.
Cookie Policy

```
// _protected/framework/Mvc/Model/Security.class.php

public function __construct()
{
```

```
        $this->sIp = Ip::get();   // the rate-limit key, taken from the request
        //
    }

    public function checkLoginAttempt(/* ... */): bool
    {
        // ...
        $rStmt = Db::getInstance()->prepare(
            'SELECT * FROM' . Db::prefix($sAttemptTable) . 'WHERE ip = :ip LIMIT 1
        );
        $rStmt->bindValue(':ip', $this->sIp, PDO::PARAM_STR);   // VULNERABILITY H
        // ...
        return true;   // no row for this key → authorised
    }
```

The SQL is parameterised and the counter arithmetic is correct. Nothing here is an injection or an off-by-one. The defect is upstream of the query: `$this->sIp` is not an observation about the request, it is a claim made by the request. Every lookup, every increment and every lockout in this class inherits that.

10.3 Step-by-Step Trace with Concrete Values

The values below are illustrative and were not taken from any live target.

INPUT	CONTENT	PURPOSE
Header A	`X-Forwarded-For: 198.51.100.7`	Claimed adc
Header B	`X-Forwarded-For: 203.0.113.42`	Claimed adc
Configuration	`maxAdminLoginAttempts = 10`, `loginAdminAttemptTime = 120`	Shipped def

Step 1 — ten failed attempts sent with Header A:

← Back to CyberMap

Cookie Usage
We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.
Cookie Policy

- Each request: `Ip::get()` → `198.51.100.7` → counter row for that key incremented
- On the tenth failure the row reaches `attempts = 10`

← Back to CyberMap

```
ph7_admins_attempts_login = [ { ip: "198.51.100.7", attempts: 10, lastLogin: "
```

Step 2 — eleventh attempt, still with Header A:

- `checkLoginAttempt()` → row found, `attempts ⩾ 10`, lockout window still open → **FALSE**
- Credentials are not evaluated, and the request is refused

Step 3 — same eleventh guess, resent with Header B:

- `Ip::get()` → `203.0.113.42`
- `SELECT * FROM ph7_admins_attempts_login WHERE ip = '203.0.113.42'` → **0 rows**
- `checkLoginAttempt()` → **TRUE** (authorised)

```
ph7_admins_attempts_login = [
  { ip: "198.51.100.7",  attempts: 10 },   ← locked out, and abandoned
  { ip: "203.0.113.42",  attempts: 1  }    ← the request is judged against th
]                                           ← the lockout has been stepped aro
```

Result: the eleventh guess was evaluated against a counter that had existed for one request, and the attacker can continue simply by selecting another public address the table has not seen. The lockout table does not restrain the attack. It records it, one row per guess.

Cookie Usage
We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.
Cookie Policy

10.4 Analogy

A doorman keeps a list of names that are not welcome tonight, and asks each arrival for their name rather than for identification. The list fills up correctly and is consulted correctly. It just

never contains the name of the person currently standing at the door.

← Back to CyberMap

10.5 Architectural Root Cause

This is an instance of **CWE-290 — Authentication Bypass by Spoofing**, and unlike Link 1 it is an implementation defect with a well-known correct form. The developer assumption is that `X-Forwarded-For` means “the real client”, because in a deployment behind a reverse proxy that is what an operator sees in the logs. The header actually means “the party before me said this”, and it carries no authority unless the party before you is one you trust. The corollary is easy to miss and is the reason this class recurs: the fix is not a better parser but a *trust boundary*. The application must know which peers are allowed to speak about addresses at all, and treat every other claim as noise. Any code that reads `HTTP_X_FORWARDED_FOR`, `HTTP_CLIENT_IP`, `HTTP_X_REAL_IP` or `HTTP_CF_CONNECTING_IP` and uses the result for anything other than logging carries the same defect, including IP allowlists, geo-restrictions, and audit trails.

---

11. Hands-On Demonstration

11.1 Lab Environment

COMPONENT	CONFIGURATION
Target OS	Ubuntu 22.04.5 LTS
Web stack	Apache 2.4.52, PHP 8.1.2, MySQL
Vulnerable build	pH7Builder 18.1.0, default installation, admin path left at `/admin123/`
Brute-force settings	Shipped defaults: enabled, 10 attempts, 120-minute lockout

Cookie Usage
We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.
Cookie Policy

← Back to CyberMap

COMPONENT	CONFIGURATION
Patched comparison build	Not built — see Section 11.3
Attacker host	Linux workstation; Burp Suite for manual request analysis, Python 3 fo
Network	Isolated host-only segment, no internet egress
Reverse proxy	None — requests reach Apache directly
Logging	Apache access and error logs enabled; application login tables inspe
Snapshot	Taken before testing; environment reverted afterwards

Throughout this section the target is referred to as `TARGET_HOST`, and the administrator account under test as `admin@TARGET_HOST` / `admin`.

11.2 Step-by-Step Verification

Step 1 — baseline: a single failed login:

```
POST /admin123/main/login HTTP/1.1
Host: TARGET_HOST
Cookie: PHS7SESS=<S1>
Content-Type: application/x-www-form-urlencoded

submit_admin_login=form_admin_login&security_token=<T1>&mail=admin%40TARGET_H
```

Cookie Usage
We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.
Cookie Policy

```
HTTP/1.1 301 Moved Permanently
Location: https://TARGET_HOST/admin123/main/login
```

Following the redirect renders the login page with the error *"Password is incorrect"*, confirming the credentials were evaluated.

← Back to CyberMap

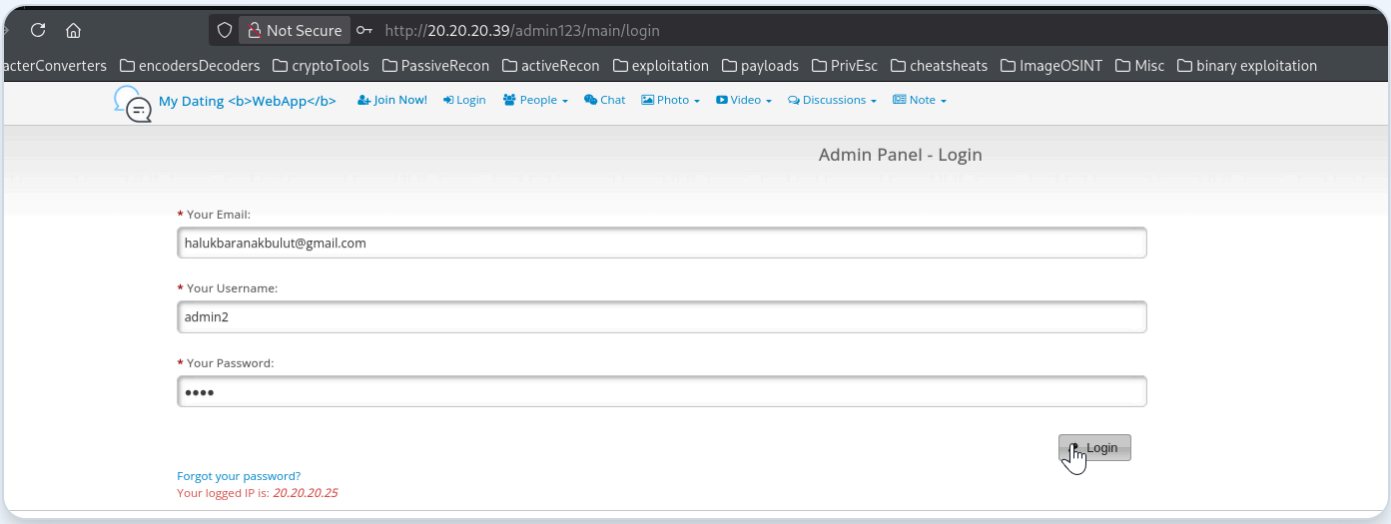


Figure 1: The admin login panel with an incorrect password submitted.

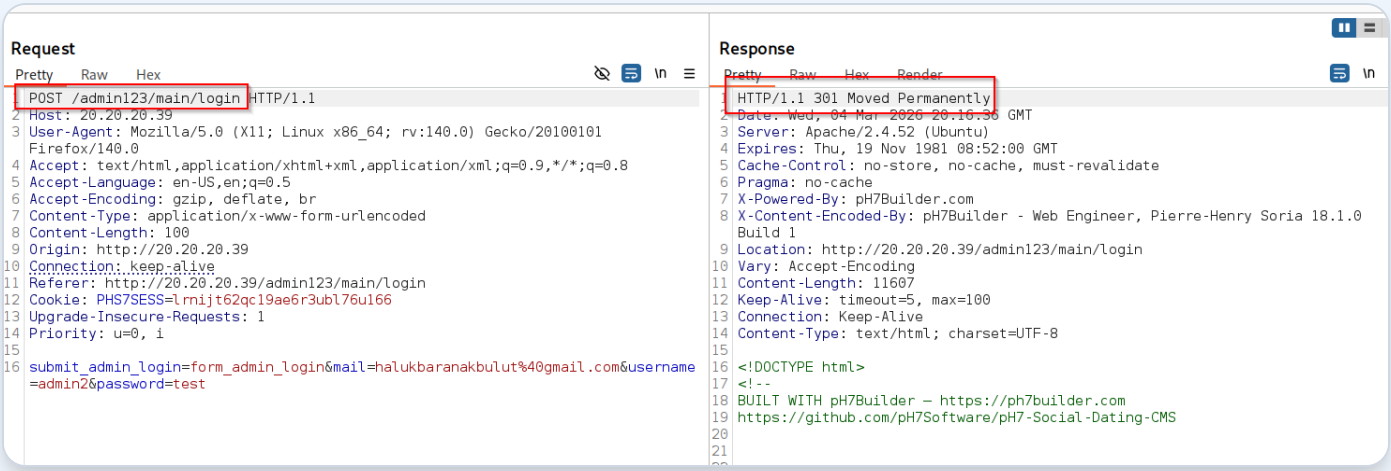


Figure 2: The intercepted `POST` and its 301 response.

Cookie Usage

We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.

Cookie Policy

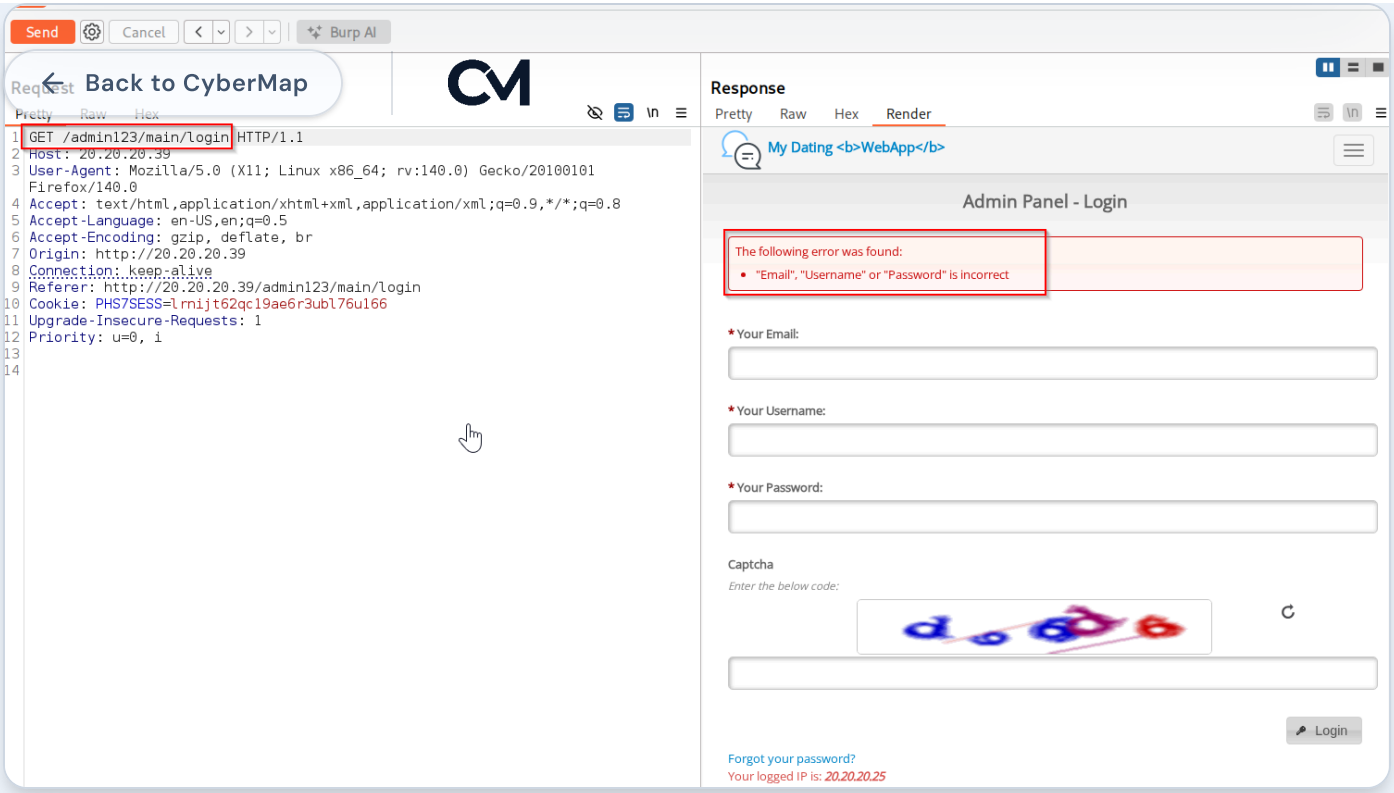


Figure 3: After following the redirect: the incorrect-password error.

Step 2 — expected CAPTCHA escalation under the same session:

Resending the identical request with the same `PHS7SESS` value produces a different outcome. The response no longer reports a password error. It reports a CAPTCHA error, because the session is now marked and the re-rendered form contains a CAPTCHA element.

```
The code of Captcha entered was incorrect. Please re-try.
```

Cookie Usage

We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.

Cookie Policy

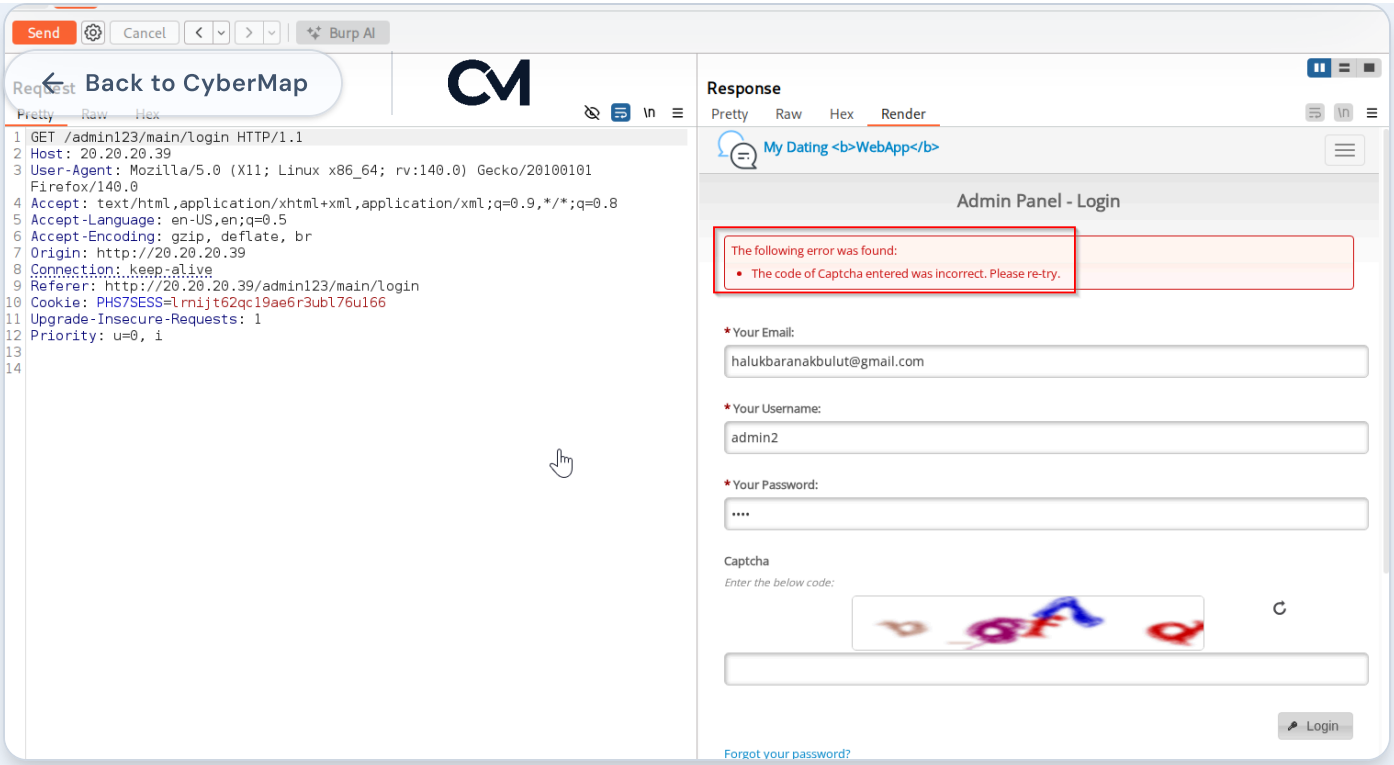


Figure 4: CAPTCHA validation triggered on the repeat attempt under the same session.

This step matters: it establishes that the defence is present and functioning, so that what follows is a bypass rather than a misconfigured target.

Step 3 — obtaining a fresh session:

```
GET /admin123/main/login HTTP/1.1
Host: TARGET_HOST
```

```
HTTP/1.1 200 OK
Set-Cookie: PHS7SESS=<S2>; path=/; HttpOnly; SameSite=Lax
```

Cookie Usage
We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.
Cookie Policy

With no `PHS7SESS` header, the server starts a new session and returns a new `PHS7SESS` value, together with a freshly minted CSRF token in the form body.

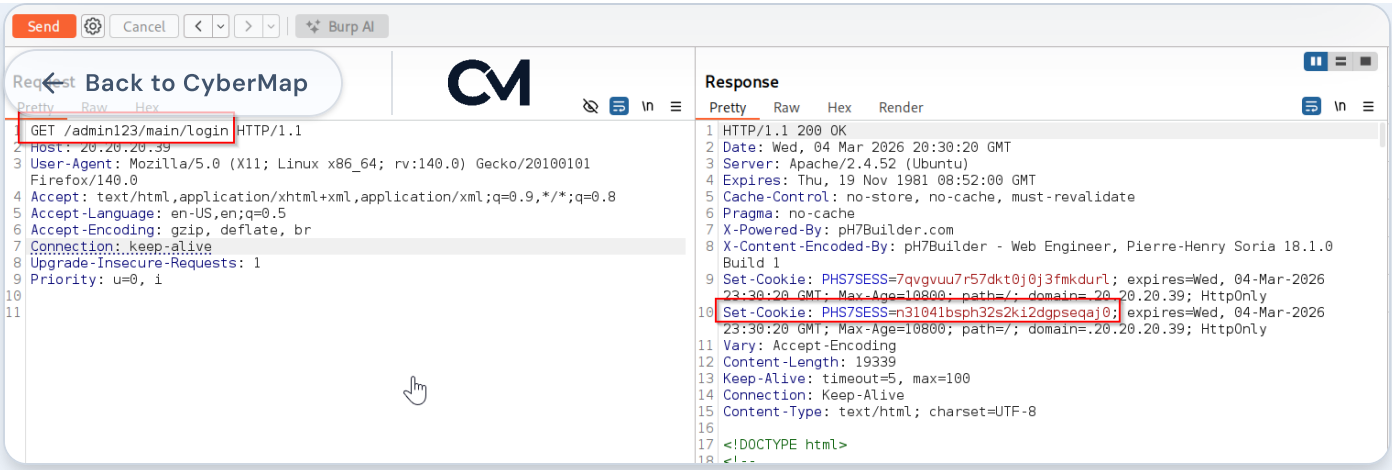


Figure 5: `GET` with the `Cookie` header removed; the server issues new `PHS7SESS` values.

Step 4 — the same guess under the new session:

```
POST /admin123/main/login HTTP/1.1
Host: TARGET_HOST
Cookie: PHS7SESS=<S2>
Content-Type: application/x-www-form-urlencoded

submit_admin_login=form_admin_login&security_token=<T2>&mail=admin%40TARGET_H
```

The response is a 301 and, after following it, the standard *"Email, Username or Password is incorrect"* error. No CAPTCHA was presented, and no CAPTCHA value was submitted.

Cookie Usage

We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.

Cookie Policy

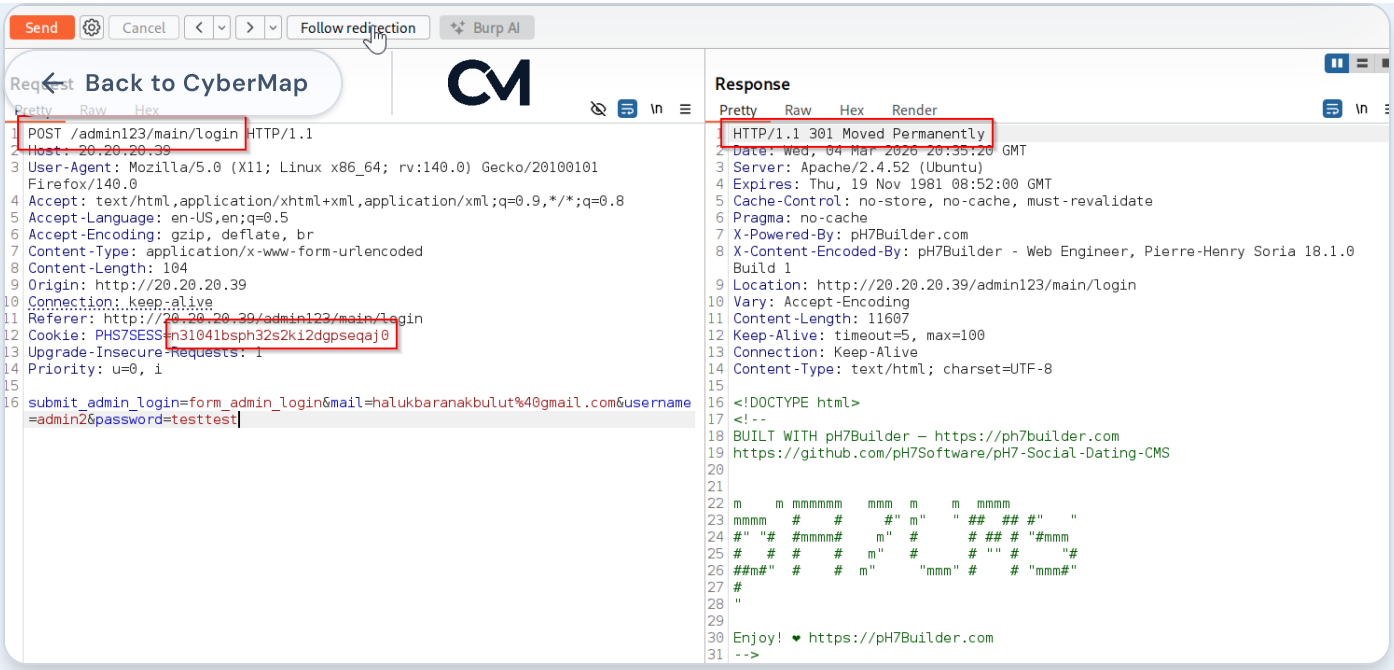


Figure 6: `POST` with the newly obtained session cookie.

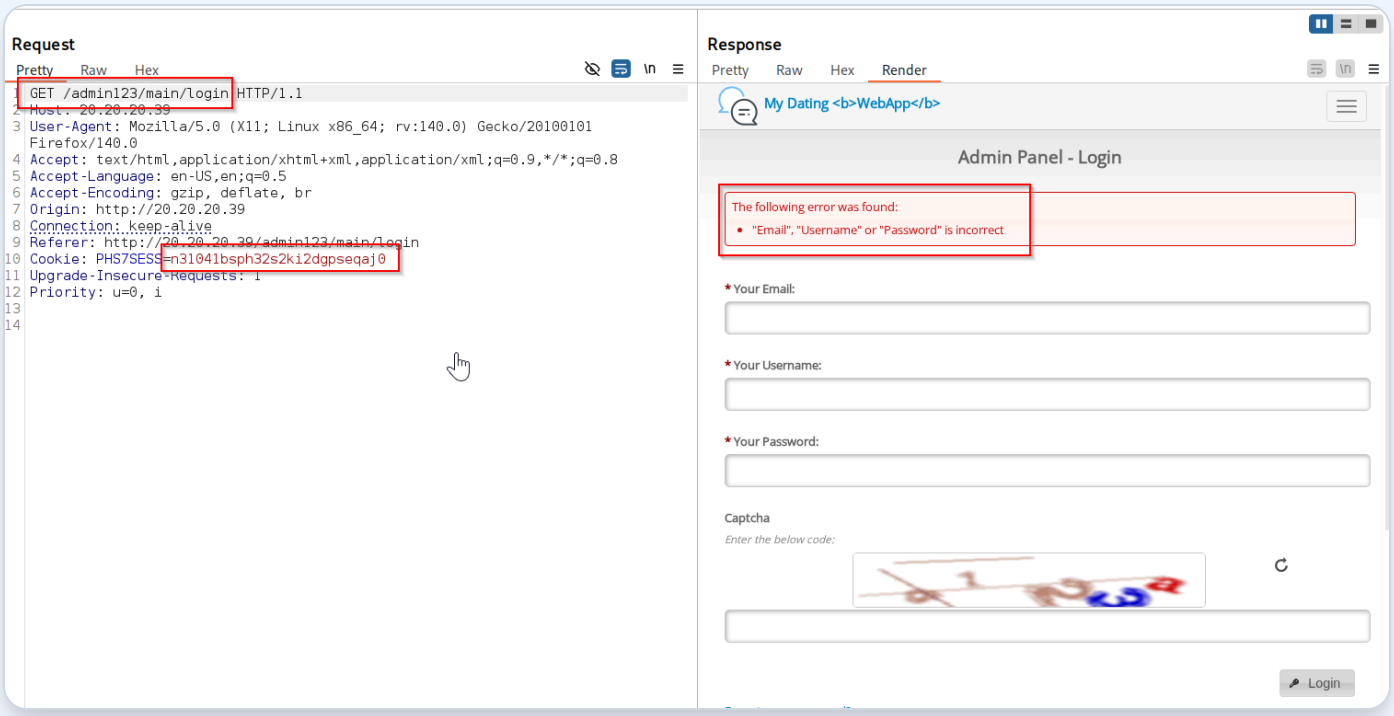


Figure 7: Challenge absent: only the credential error is returned.

Cookie Usage

We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.

Cookie Policy

Step 5 — confirming the authentication-success oracle:

Repeating Step 4 with the correct password produces a redirect to a different location. This is the oracle the automation relies on: a failed authentication redirects back to

`/admin123/main/login`, while a successful one redirects to `/admin123/main/index`. The successful response also carries a new `Set-Cookie`, because the application regenerates the session identifier on login.

← Back to CyberMap

```
HTTP/1.1 301 Moved Permanently
Location: https://TARGET_HOST/admin123/main/index
```

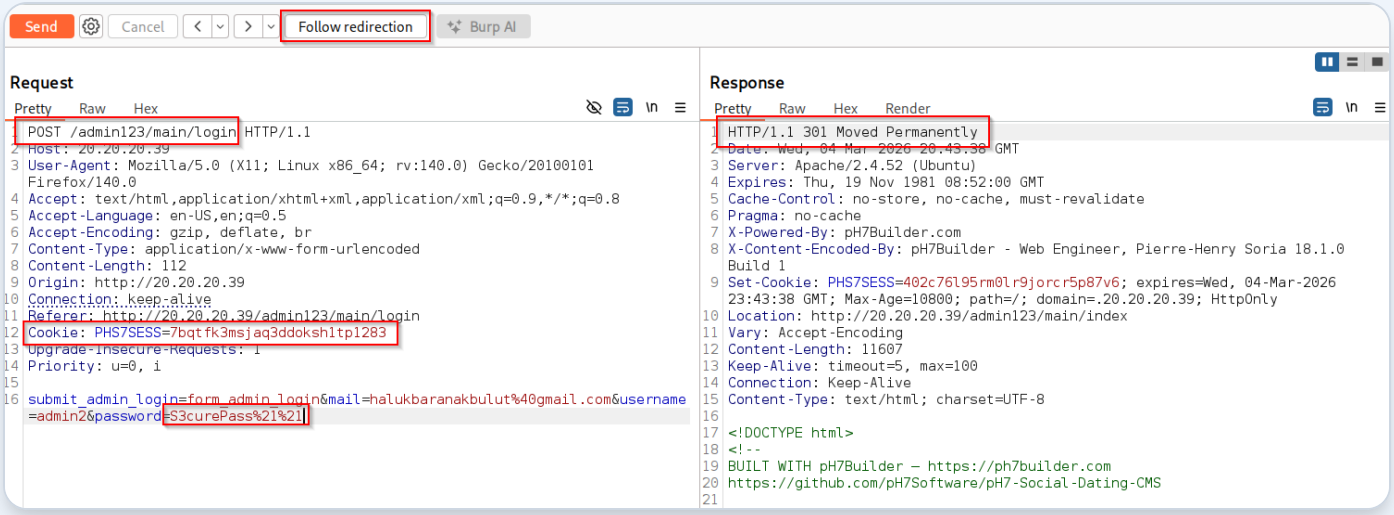


Figure 8: `POST` with a fresh session cookie and the correct password.



Figure 9: Following the redirect with the old session cookie returns to the login page rather than the admin panel.

Cookie Usage
We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.
Cookie Policy

Step 6 — reaching the IP lockout:

Back to CyberMap

Continuing to guess with the session-reset technique alone eventually crosses the per-IP threshold. In this environment the lockout path returned a generic *Internal Server Error* page rather than the intended lockout message. The underlying cause was not investigated, since the observable effect of further attempts being refused was what mattered for the test.

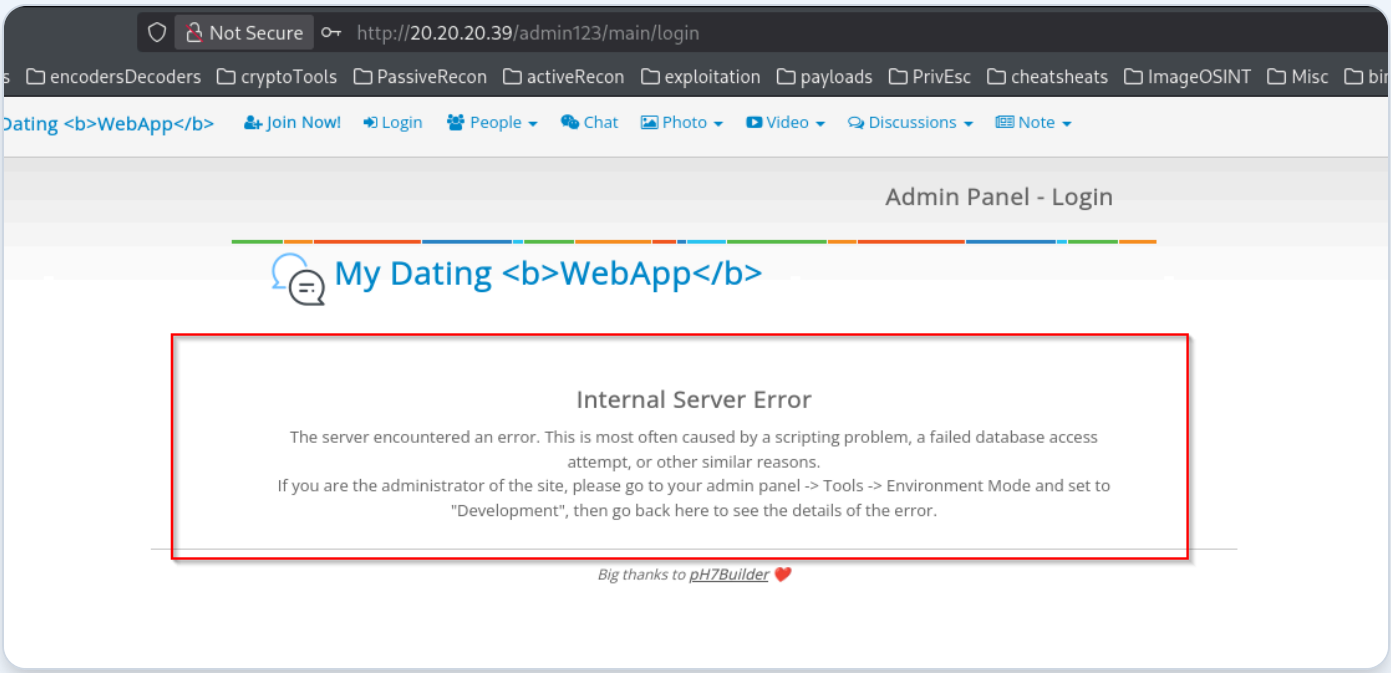


Figure 10: Requests refused after the per-IP threshold is crossed.

Step 7 — stepping around the lockout:

Adding a single header to the refused request, and to the `GET` that precedes it, restores normal processing:

```
X-Forwarded-For: 1.2.3.4
```

Cookie Usage
We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.
Cookie Policy

The response returns to the ordinary credential error, and the login page's own footer, which prints the address the application believes it is talking to, displays the spoofed value.

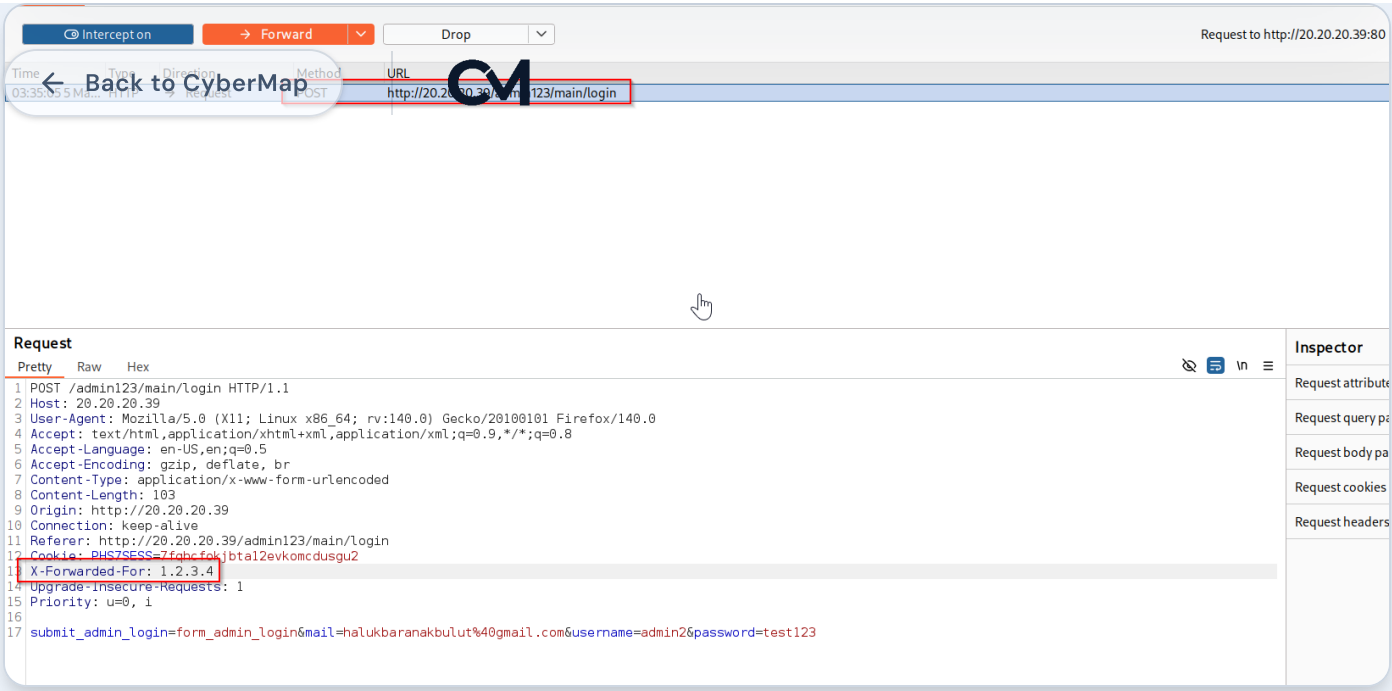


Figure 11: The spoofed `X-Forwarded-For` header added to the request.

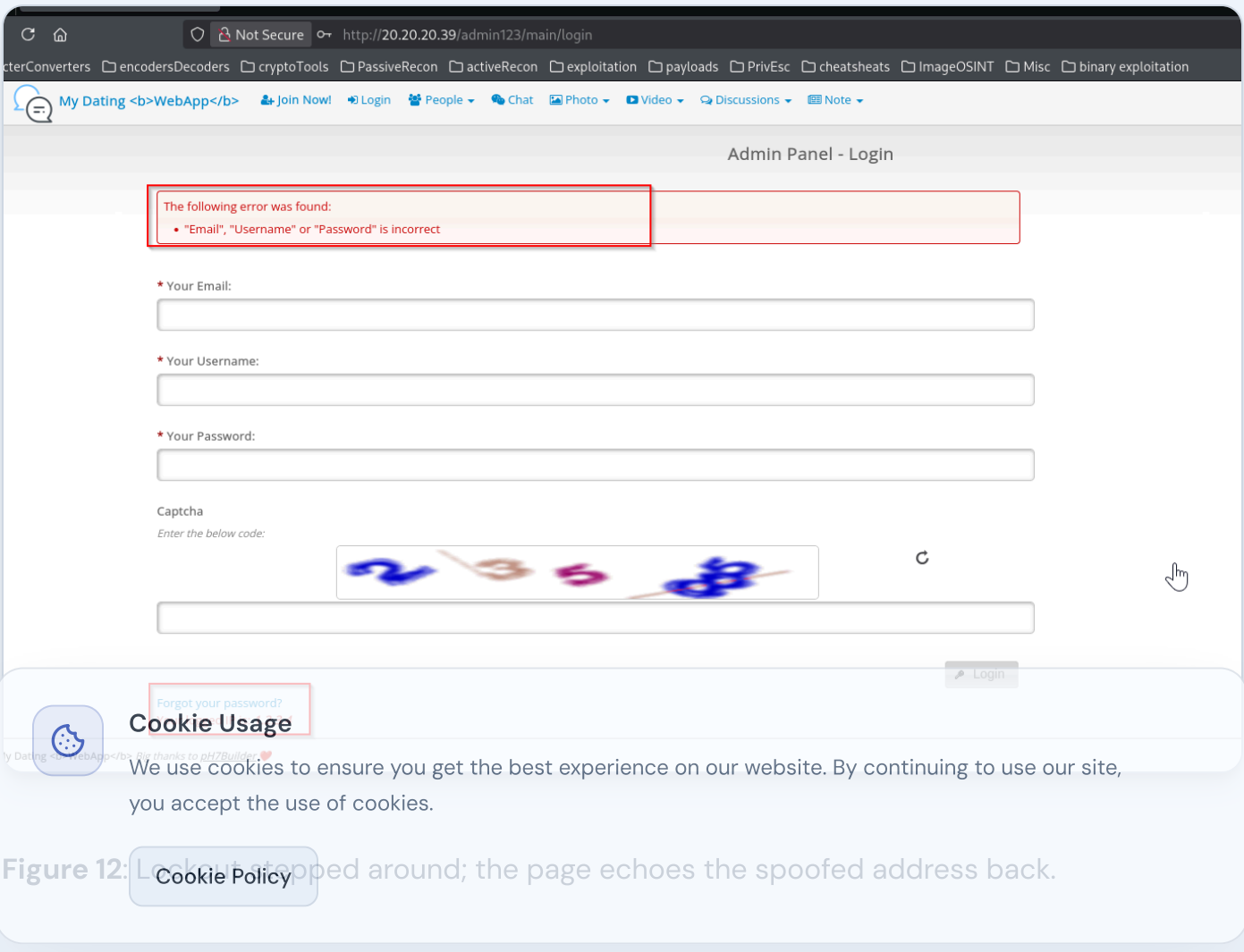


Figure 12: Lockout stopped around; the page echoes the spoofed address back.

That echo is the clearest single piece of evidence in the run: it confirms that the application accepted the attacker-supplied header value as its resolved client address.

← Back to CyberMap

Step 8 — the chain, automated: <details> <summary>Operating logic of the automation tooling</summary>

1. **Per-attempt setup.** For each candidate password, generate one random public IPv4 address and send a `GET` to the login URL with no `Cookie` header and that address in `X-Forwarded-For`, then take the issued `PHS7SESS` value and the CSRF token from the response.
2. **Guess.** `POST` the credentials using that session cookie, that CSRF token, and the *same* `X-Forwarded-For` value. The token is bound to the claimed address, so the pair must be consistent (see Section 10.1).
3. **Oracle.** Treat a `Location` header pointing at `/admin123/main/index` as success and a redirect back to the login page as failure, stopping on the first success.
4. **Concurrency.** Run the above in a small thread pool, which is what makes the per-failure server-side delay irrelevant (see below).
5. **Cleanup.** Nothing is written to the target beyond the rows the application itself creates. The lab was restored from snapshot afterwards.

No weaponised tooling is published with this article. </details>

Cookie Usage

We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.

Cookie Policy

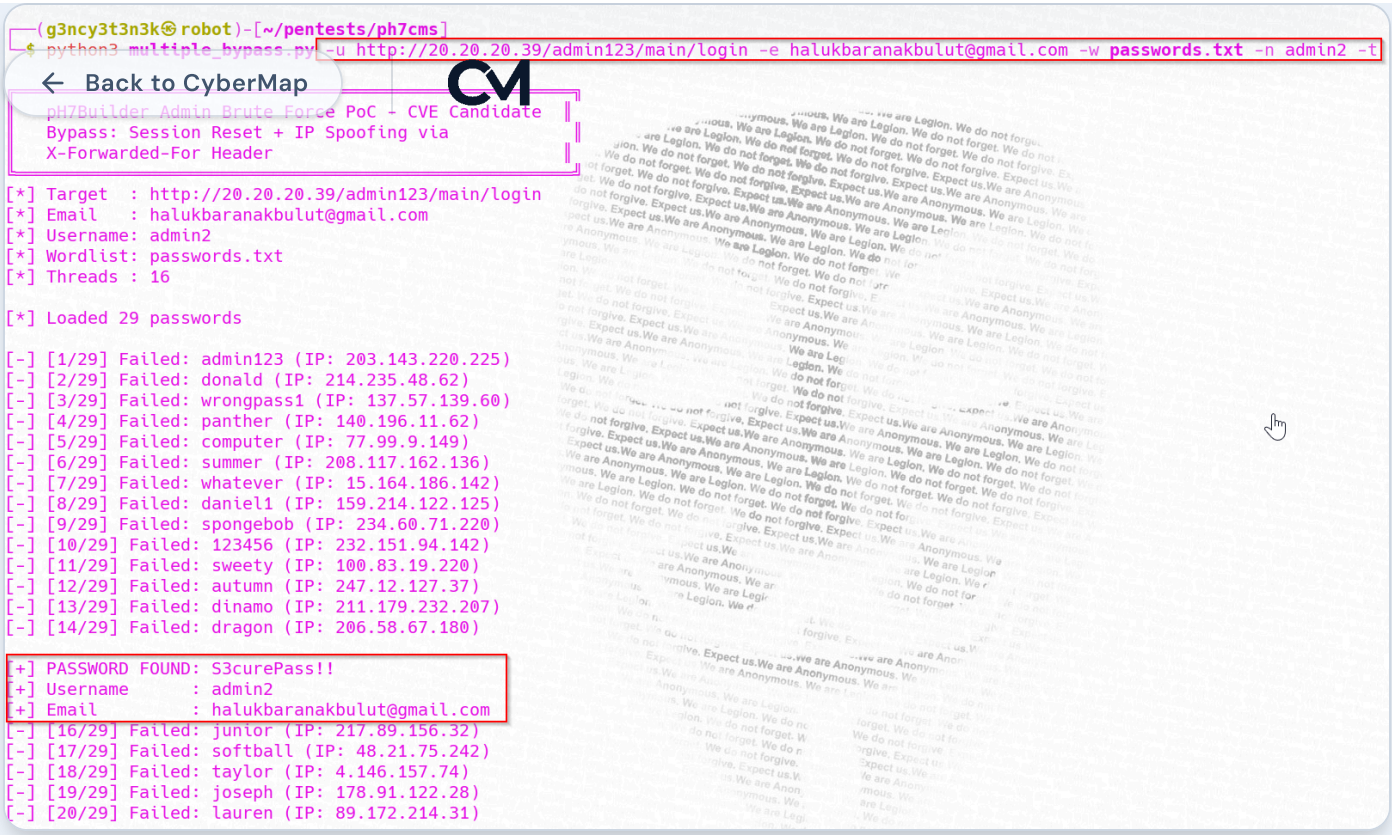


Figure 13: Automated run: the correct password recovered without any protection triggering.

In the recorded run the tool worked through a 29-entry wordlist using 16 threads and recovered the correct password at attempt 15, with no CAPTCHA presented and no lockout encountered.

Technical nuance: one throttle does survive both bypasses. A `sleep(2)` on the failure path holds each failed request open for two seconds. It is a per-request delay rather than a per-attacker budget, so with 16 concurrent workers the fixed two-second cost is largely amortised across requests in flight rather than imposing two seconds of wall-clock time per candidate. Additional concurrency amortises it further, until server-side resource limits become the bottleneck. This is why the automation uses a thread pool at all: the delay is the only cost left once both links are in place. A delay throttles an attacker only when it is enforced against a key the attacker cannot multiply, which is the same property both links of this chain remove.

Cookie Usage
We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.
Cookie Policy

11.3 Patch Status and Regression Testing

← Back to CyberMap

The blueprint this article follows calls for the verification steps to be re-run against a patched build and shown to fail. That test is only partly available here, and the reason is itself part of the finding.

CVE-2026-37604 was fixed in release 18.3.0 (2026-06-08). `getClientIp()` was rewritten to consult forwarded headers only when the TCP peer appears in a `$trustedProxies` allowlist, defaulting to loopback addresses:

```
// _protected/framework/Ip/Ip.class.php (18.3.0 and later)

$trustedProxies = [
    '127.0.0.1', '::1',
    '::ffff:127.0.0.1',
    // Add your reverse proxy IPs here (e.g., Cloudflare, nginx, etc)
];

$remoteAddr = (string)Server::getVar(Server::REMOTE_ADDR, '');
$isTrustedProxy = in_array($remoteAddr, $trustedProxies, true);

$ipHeaderVars = $isTrustedProxy
    ? ['HTTP_CF_CONNECTING_IP', 'HTTP_TRUE_CLIENT_IP', 'HTTP_X_REAL_IP',
       Server::HTTP_CLIENT_IP, Server::HTTP_X_FORWARDED_FOR, Server::REMOTE_AD
    : [Server::REMOTE_ADDR];
```

For a deployment where requests reach PHP directly, as in the configuration tested in Section 11.2, the source-level change appears to close Link 2 along the lines recommended in the report: a spoofed header from a non-loopback peer is ignored and the counter is keyed on the transport address again. The release carried no advisory and no changelog entry identifying it as a security fix.

Not lab-verified. The statements above are based on reading the vendor's published source at release tags 18.0.0 and 18.3.0 and diffing them. The fix has not been re-tested in the laboratory, and the regression run the blueprint asks for is outstanding. Readers validating their own patching should reproduce Step 7 against 18.3.0 rather than take this on trust.

Cookie Usage
We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.
Cookie Policy

← Back to CyberMap

What the patch does not cover. CVE-2026-37603 is untouched. The CAPTCHA flag still lives in `$_SESSION` and `isCaptchaEligible()` still reads only that flag, in every release through 19.2.0. On a patched build the session-reset technique still removes the challenge from the form; what now constrains the attacker is the per-IP counter, which on a direct-to-PHP deployment is finally doing its job. The practical result is a reduction from “no application-level limit” to “ten guesses per source address, with no challenge”, which is a real improvement but not a fix. Upgrading therefore closes one link of the chain and leaves the other in place.

---

12. Impact Analysis

DIMENSION	IMPACT	JUSTIFICATION
Confidentiality	Low at the vulnerability level	The flaws disclose no data themselves; they re
Integrity	Low at the vulnerability level	Likewise indirect: after successful authenticat
Availability	None directly	Neither flaw causes service disruption by itsel

Both vectors assign Low impact to Confidentiality and Integrity and None to Availability, which reflects the flaws as scored in isolation: bypassing a throttle is not, by itself, a compromise. That framing understates the operational consequence and should not be read as a measure of urgency. What these flaws remove is the assumption every password policy quietly depends on, that an attacker gets a small, bounded number of attempts. Once the application's own limits no longer apply, the security of the admin account rests on password strength alone.

Organisational risk: the administration panel of a dating and social platform is a concentration point for exactly the data that carries the highest regulatory weight: identity

Cookie Usage
We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.
Cookie Policy

information, private messages, photographs, and in many deployments payment configuration. Depending on the data actually processed by a given deployment and on the legal framework that applies to it, compromise of an administrative account may expose personal data subject to obligations under frameworks such as Türkiye's KVKK or the GDPR. Administrative access also carries the full set of privileges the panel is designed to grant, from settings and content to accounts and templates, so a recovered password is not a partial compromise of the platform but a complete one. A single weak or reused administrator password is the entire distance between "exposed endpoint" and that outcome.

Chaining potential: the two flaws in this article are already a chain, and the table below is the comparison that matters, not because either link is severe alone, but because the combination is what removes the last bound on the attack.

SCENARIO	REQUIREMENTS	FINAL IMPAC
CVE-2026-37603 alone	Attacker limited to the addresses it actually controls	No challeng
CVE-2026-37604 alone	Attacker must solve or evade the CAPTCHA	Unlimited *re*
Both combined	None beyond network access	Unchallenge

The wider lesson generalises past this product. Two findings rated Medium, each reasonably scored, compose into an outcome that neither vector describes, because CVSS scores a flaw while an attacker exploits a system. This illustrates a limitation of evaluating chained vulnerabilities solely by their individual base scores: the combined attack path can carry materially greater operational impact than either score suggests.

Cookie Usage
We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.
Cookie Policy

13. Detection and Log Analysis

Network and application layer:

← Back to CyberMap

- **A `GET` / `POST` pair per credential guess, with a new session each time.** The chain's signature is structural rather than payload-based: in the demonstrated attack flow each credential guess is preceded by a cookie-less `GET` to the login URL, so the ratio of sessions issued to login attempts approaches 1:1. Legitimate use produces one session and many requests. This produces one session per request. `# GET vs POST volume on the login endpoint, per source. A ratio approaching 1:1 # from a single source is the structural indicator; normal use is GET-light. awk '$7 ~ /admin123\/main\/login/ {gsub("/", "", $6); print $1, $6}' /var/log/apache2/access.log \ | sort | uniq -c | sort -rn | head -40` The *cookie-less* property of those `GET` requests is not visible in the default combined log format, which records no request headers. Confirming it requires logging the inbound `Cookie` header or the outbound `Set-Cookie`, as in the format below.
- **High cardinality of `X-Forwarded-For` values against a single `REMOTE_ADDR`.** This is the strongest available indicator, and it is only visible where the transport address and the claimed address are logged side by side. A large number of distinct public claimed addresses associated with one transport source is highly suspicious and should be investigated as potential header spoofing. Benign causes such as a misconfigured intermediary exist, but they do not normally produce a distinct value per request. `# Log both, so the two can be compared at all. # %{c}a is the connection peer, %{X-Forwarded-For}i is what the client claimed. LogFormat "%{c}a %{X-Forwarded-For}i %t \"%r\" %>s %b" xff_transport` `%{c}a` is used rather than `%a` deliberately. Where `mod_remoteip` is active, `%a` already reflects the rewritten client address, which is the value under the attacker's control. `%{c}a` preserves the underlying connection peer, and the comparison only works if both are recorded.
- **A burst of 301 responses to `/admin123/main/login` with no successes.** Failed and successful logins both redirect, so status codes alone do not separate them. The `Location` target does. A concentration of redirects back to the login page, ending in a single redirect to `/admin123/main/index`, is the attack succeeding.

Host and database layer (SIEM / audit):

- `ph7_admins_attempts_login` **filling with single-attempt rows.** Under normal operation this table holds a handful of rows with rising counters. Under this attack it accumulates one row per guess, each with `attempts = 1`. A count of rows created per minute is a cheap, high-signal query. `SELECT COUNT(*) AS rows_created, MIN(lastLogin), MAX(lastLogin) FROM ph7_admins_attempts_login WHERE lastLogin > NOW() - INTERVAL 1 HOUR;`

Cookie Usage
We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.
Cookie Policy

- `ph7_admins_log_login` **recording addresses that never connected.** Because CVE-2026-37604 places the resolved address under the client's control, the addresses this table stores are the ones the client claimed. On an affected version, do not treat them as the source of an attempt: correlate against the web server's transport-level log instead, and read disagreement between the two as the indicator.

Detection indicators:

TYPE	VALUE
URI pattern	`/admin123/main/login`
Request pattern	Cookie-less `GET` immediately followed by a login `POST` using the newly issu
Header anomaly	Many distinct, well-formed public `X-Forwarded-For` values from one `REMOTE`
Success marker	`Location: /admin123/main/index` following a run of failures
Database anomaly	High row-creation rate in `ph7_admins_attempts_login` with `attempts = 1`

Limits of detection: every indicator above depends on observing the transport-level source address. An attacker who distributes the same attack across a proxy pool or botnet, rather than spoofing a header from one connection, produces traffic that is not structurally distinguishable from many unrelated visitors failing to log in, and the header-correlation rule finds nothing because each claim matches its connection. Detection then falls back to the aggregate: total failed administrative logins per unit time, regardless of source. Equally, an installation that logs only `X-Forwarded-For` as the client address, a common reverse-proxy configuration, has no ground truth to correlate against and cannot detect this attack from its own logs at all.

Cookie Usage

We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.

Cookie Policy

14. Remediation and Secure Coding Lessons

Back to CyberMap

Permanent fix:

- **Upgrade to 18.3.0 or later.** This closes CVE-2026-37604 for deployments where PHP is reached directly, by keying the counter on the transport address again. It does not close CVE-2026-37603, and Section 11.3 describes what it leaves open for proxied deployments.
- **No vendor fix exists for CVE-2026-37603.** Until one does, the compensating controls below are the available response, and the first of them is the one that matters.

Compensating controls (where patching is not sufficient or not immediately possible):

- **Enforce rate limiting outside the application, keyed on the transport address.** A reverse proxy or WAF rule on the admin login path, such as nginx `limit_req` keyed on `$binary_remote_addr` or fail2ban reading the web server access log, is the only control here that an attacker cannot re-key, because it never consults a request header. *Does not cover:* distributed guessing from many real source addresses.
- **Enable two-factor authentication for administrator accounts.** pH7Builder supports it, and it breaks the chain at the point where a correct password would otherwise become a session. *Does not cover:* the guessing itself. The attacker still learns the password, which matters wherever it has been reused.
- **Restrict the admin panel at the network edge.** An IP allowlist in the web server, a VPN requirement, or client-certificate authentication in front of `/admin123/` removes the attack surface outright. *Does not cover:* nothing, provided the restriction is enforced at the edge on the transport address rather than inside the application.
- **Rename the administration path.** Raises the cost of discovery. *Does not cover:* anything once the path is known, which it will be for any installation that has ever exposed it. Treat this as noise reduction, not as a control.
- **Set a long, unique administrator password now, and confirm the account's own history.** With guessing unbounded, entropy is the only remaining bound. *Does not cover:* a password already recovered before the control was applied, which is why the log review in Section 13 should be done before, not after.

Cookie Usage

We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.

Cookie Policy

Defence matrix

← Back to CyberMap

CONTROL	LINK IT BREAKS
Upgrade to ≥ 18.3.0 (direct-to-PHP deployment)	Link 2 — the counter key becomes the transpo
Edge rate limiting on `REMOTE_ADDR`	Link 2, independently of the application versior
Server-side per-account attempt tracking	Link 1 — the challenge no longer depends on wh
Two-factor authentication	Neither link; breaks the path from a recovered p
Network-level access restriction	Both, by removing reachability

14.1 Secure Coding Lessons

1) A throttle must be keyed on the resource it protects, not on the conversation:

```
// Vulnerable pattern — state lives where the client can renew it
$this→session→set('captcha_admin_enabled', 1);
if ((new Session)→exists('captcha_admin_enabled')) {
    $form→addElement(new Captcha(/* ... */));
}

// Safe pattern — state lives with the account, and the check precedes evalua
$attempts = $store→failedAttemptsFor($username, $windowMinutes);   // server
if ($attempts ⩾ self::CHALLENGE_THRESHOLD && !$challenge→verify($request)) {
    return AuthResult::challengeRequired();   // fails closed; no element to
}
```

The structural difference is not the storage backend but the direction of the dependency: the safe version asks the account how many times it has been attacked; the vulnerable version asks the attacker.

Cookie Usage

We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.

Cookie Policy

2) Read the client address from the transport, and trust headers only from peers you named.

← Back to CyberMap

```
// Vulnerable pattern
$ip = $_SERVER['HTTP_CLIENT_IP']
   ?? $_SERVER['HTTP_X_FORWARDED_FOR']
   ?? $_SERVER['REMOTE_ADDR'];

// Safe pattern
$ip = $_SERVER['REMOTE_ADDR'];
if (in_array($ip, $trustedProxies, true)) {
    // walk the chain from the RIGHT, discarding hops you trust,
    // and stop at the first address a trusted hop did not contribute
    $ip = self::rightmostUntrusted($_SERVER['HTTP_X_FORWARDED_FOR'] ?? '', $tr
}
```

Parsing direction is not a detail. In a forwarded chain the leftmost entry is the one the original client wrote and each trusted hop appends to the right, so reading the list left to right hands the key back to the client even when the allowlist itself is correct.

3) Controls must fail closed, and be verifiable when they do:

A challenge that is *conditionally constructed* can be omitted. A challenge that is *unconditionally verified* can only be passed or failed. Where a control is built into a form at render time, its absence is indistinguishable from its success, which is why the flaw in Section 9 survives casual testing. Write the check so that the only way past it is to satisfy it, then confirm it by asserting on the negative case rather than the positive one. The question to ask of any throttle under review is the shortest one available: *what value is this keyed on, and can the client change it?*

Cookie Usage

We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.

Cookie Policy

15. Real-World Context

Back to CyberMap

- **Bludit ≤ 3.9.2 (CVE-2019-17240):** `bl-kernel/security.class.php` counted failed logins against an address taken from `X-Forwarded-For`, so supplying a different value per request bypassed the lockout entirely. Public exploits followed quickly, and the flaw became a standard teaching example of the class.
- **Keycloak ≤ 24.0.3 (CVE-2024-4629):** brute-force protection validated the attempt count before the account was locked, so a burst of *parallel* login attempts obtained more guesses than the configured threshold permitted, the same outcome as Section 11.2's thread pool reached by a different route.
- **Ghost (CVE-2024-34451):** multiple `X-Forwarded-For` headers with differing values circumvented login rate limiting. The advisory was later revoked on the argument that the fault lay in proxy configuration rather than the application. That dispute is worth noting here, because pH7Builder's own fix resolves the same question by handing the trust decision to the deployer.

The pattern across these is not that developers forget to rate-limit. All three products had a throttle, implemented deliberately, with sensible thresholds. What they shared is a key the attacker could choose, a header in two cases and a scheduling gap in the third. A key the attacker can choose turns a rate limiter into an accountant: it records the attempts faithfully and prevents none of them. The exploit primitive this class yields is unbounded retries against any check that assumes it will only be run a few times, which is a far larger surface than login forms alone: password reset codes, one-time passwords, coupon and voucher redemption, API keys, and email verification tokens are all defended primarily by the assumption that guessing is expensive.

---

16. Conclusion

Cookie Usage
We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.
Cookie Policy

Two independently reported flaws in pH7Builder's administration login combine into an unauthenticated, unbounded password-guessing attack against administrator accounts. Neither is exotic: one stores escalation state in the client's own session, the other reads the client's address from a header the client writes. The root cause is the same in both, and it is

not a missing control but a mis-keyed one. In each case the application asked the attacker which bucket to count them in. As of this writing the header flaw has been fixed upstream, silently, in release 18.3.0. The session flaw has not been fixed at all.

- **A rate limiter is only as good as its key.** Thresholds, backoff curves and lockout windows are refinements on top of one decision that has to be right first: what value groups these requests together, and can the requester change it? Every other parameter is tuning.
- **An unauthenticated client can hold as many sessions and claimed addresses as it asks for.** Sessions are issued on request, forwarded headers are free to assert, and both are free to discard. Any pre-authentication control that stores its memory in either has, in practice, no memory at all.
- **Two Mediums compose into something neither vector describes.** CVSS scores a flaw in isolation, and it does so correctly here. It has no mechanism for expressing that these two remove the same bound from opposite sides. Chained findings therefore warrant assessment as a path, not only as a pair of base scores.
- **Silent patching reduces defenders' ability to judge urgency.** The fix in 18.3.0 is a real improvement, but shipped without an advisory, a changelog entry, or a CVE reference it gave administrators no basis on which to prioritise the upgrade, and upgrade-averse installations are still running without it.
- **Go and check what your own throttles count.** Find every failed-attempt counter, lockout, and challenge-escalation flag in code you own, and for each one, write down the expression that produces its key. If that expression is derived from a client-controlled value such as a cookie, a session or a request header, the control should be reviewed for this class of rate-limit bypass before anyone else reviews it for you.

---

17. References

Official records

- MITRE — CVE-2026-37603: https://www.cve.org/CVERecord?id=CVE-2026-37603
- MITRE — CVE-2026-37604: https://www.cve.org/CVERecord?id=CVE-2026-37604
- CWE-307 — Improper Restriction of Excessive Authentication Attempts: https://cwe.mitre.org/data/definitions/307.html

← Back to CyberMap

Cookie Usage
We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.
Cookie Policy

- CWE-290 — Authentication Bypass by Spoofing: https://cwe.mitre.org/data/definitions/290.html
- CVSS v3.1 calculator: https://www.first.org/cvss/calculator/3.1
- MITRE ATT&CK T1110 — Brute Force: https://attack.mitre.org/techniques/T1110/

Vendor

- pH7Builder repository: https://github.com/pH7Software/pH7-Social-Dating-CMS
- Affected source, 18.1.0 — `Ip.class.php`: https://github.com/pH7Software/pH7-Social-Dating-CMS/blob/v18.1.0/_protected/framework/Ip/Ip.class.php
- Fix comparison, 18.2.0 → 18.3.0: https://github.com/pH7Software/pH7-Social-Dating-CMS/compare/v18.2.0...v18.3.0
- Current source — `LoginForm.php` (CAPTCHA state, still session-bound): https://github.com/pH7Software/pH7-Social-Dating-CMS/blob/master/_protected/app/system/modules/admin123/forms/LoginForm.php
- Vendor security policy (added in 18.5.0): https://github.com/pH7Software/pH7-Social-Dating-CMS/blob/master/SECURITY.md

Public exploit status

- No exploit tooling is included with this publication. The article publishes request/response detail and the operating logic of the researcher's own lab automation, but not the automation itself (see Section 2). No public exploit for either CVE was known to the researcher at the time of publication.

Prior art in the same class

- Bludit brute-force mitigation bypass (CVE-2019-17240): https://www.cvedetails.com/cve/CVE-2019-17240/
- Keycloak brute-force protection bypass (CVE-2024-4629): https://bugzilla.redhat.com/show_bug.cgi?id=2276761
- Ghost rate-limit bypass via multiple XFF headers (CVE-2024-34451, advisory later revoked): https://security.snyk.io/vuln/SNYK-JS-GHOST-7266556

Back to CyberMap

Cookie Usage
We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.
Cookie Policy

Technical guidance

← Back to CyberMap

- OWASP — Blocking Brute Force Attacks: https://owasp.org/www-community/controls/Blocking_Brute_Force_Attacks
- OWASP — Authentication Cheat Sheet: https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html
- OWASP — Proxy Header Security Cheat Sheet: https://cheatsheetseries.owasp.org/cheatsheets/Proxy_Header_Security_Cheat_Sheet.html
- NIST SP 800-63B (Revision 4), Digital Identity Guidelines: Authentication and Lifecycle Management. See §3.2.2, “Rate Limiting (Throttling)”: https://pages.nist.gov/800-63-4/sp800-63b.html

Appendix A — Glossary

TERM	DEFINITION
CVE-2026-37603	Improper restriction of excessive authentication attempts (CWE-
CVE-2026-37604	Authentication bypass by spoofing (CWE-290) in `Ip::get()`: the
`PHS7SESS`	The PHP session cookie name shipped in pH7Builder’s installer co
`Ip::get()` / `getClientIp()`	The framework helper resolving the client address; the single poir
`captcha_admin_enabled`	The session key whose presence causes the CAPTCHA element t
`ph7_admins_attempts_login`	The per-address failed-attempt counter table consulted before
Session reset	Discarding the session cookie and requesting a new one before e
Rate-limit key	The value a throttle groups requests under. Where the requester

Cookie Usage
We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.
Cookie Policy

← Back to CyberMap



Haluk Baran Akbulut

Administrator

Subscribe to the newsletter

New cybersecurity articles, research and hardware-lab notes — straight to your inbox.

Your email address

Subscribe →

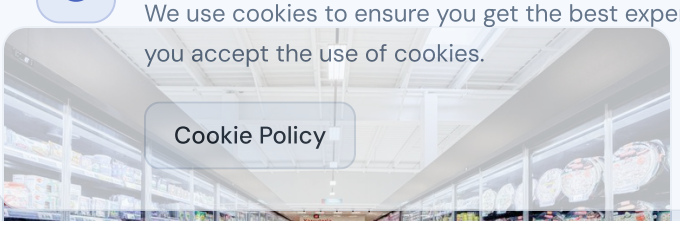
I agree to receive the newsletter and have read the Privacy Policy.

Related articles

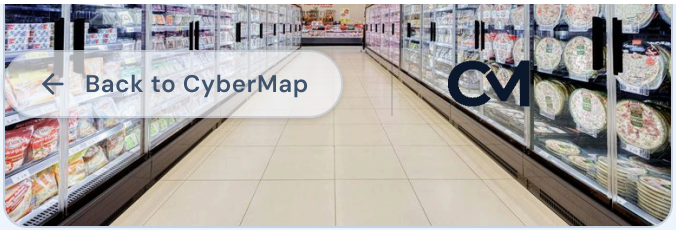
Cookie Usage

We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.

Cookie Policy







October 22, 2026

How Was the Supermarket Refrigeration System Sabotaged Remotely?



July 27, 2026

CVE-2026-25769 Analysis: Remote Code Execution in Wazuh Cluster via Insecure...



July 27, 2026

CVE-2026-31431 Technical Analysis: The “Copy Fail” Local Privilege Escalation...

All articles →

Let's assess your organization's security posture together.

Start with a no-commitment consultation; we listen to your needs and build a roadmap tailored to you.

Cookie Usage

We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.

Cookie Policy

← Back to CyberMap

Request a consultation ↗

CYBERMAP GROUP

A results-driven cybersecurity ecosystem with an adversary mindset: offensive testing, compliance consultancy, corporate training and R&D.

info@cybermapgroup.com

YDA Center — Kızılırmak Mah. Dumlupınar Bul. No: 9A, Çankaya / Ankara

Cookie Usage

We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.

Cookie Policy

Brand Assets
Disclaimer
Back to CyberMap
CyberMap Blog
Contact

© 2026 CyberMap Group
Developed by CyberMap Group.

Cookie Usage
We use cookies to ensure you get the best experience on our website. By continuing to use our site, you accept the use of cookies.
Cookie Policy