Information Disclosure: Username Enumeration via Login Endpoint Timing Side-Channel

Low **enchant97** published **GHSA-w6m9-39cv-2fwp** 5 days ago

Package

note-mark

Affected versions	Patched versions
v0.19.1	v0.19.2

Description

Summary

A timing side-channel in the login endpoint allows unauthenticated attackers to determine whether a username exists by measuring response time differences. Requests for valid usernames take noticeably longer because the server performs bcrypt password verification, while requests for nonexistent usernames return much faster. This enables reliable remote username enumeration and increases the risk of targeted credential attacks.

Details

The issue affects the login endpoint:

- `POST /api/auth/token`

The root cause is that authentication processing takes different code paths depending on whether the supplied username exists. When the username is found, the server performs `bcrypt.CompareHashAndPassword`, which adds substantial latency. When the username does not exist, the server returns immediately without performing an equivalent bcrypt operation.

Vulnerable flow:

```
user, err := db.Where("username = ?", username).First(&user)
if err != nil {
    return ErrUnauthorized
}
err = bcrypt.CompareHashAndPassword(user.PasswordHash, []byte(password))
```

This creates a measurable timing discrepancy between:

- **existing username + wrong password** requests, which incur bcrypt cost
- **nonexistent username + any password** requests, which avoid bcrypt entirely

Because no constant-time equalization is performed, the endpoint leaks account existence through timing behavior.

The measurements provided show a large and consistent gap between the two cases across repeated trials, making the difference distinguishable without requiring especially high request volume. In the supplied test results:

- existing user requests averaged about `0.0616s`
- nonexistent user requests averaged about `0.0027s`

That gap is large enough to support reliable username enumeration under typical testing conditions.

PoC

The issue can be reproduced by sending repeated authentication attempts to the login endpoint using the same invalid password while alternating between a known valid username and a nonexistent username, then comparing average response times. Valid usernames consistently take longer because bcrypt verification is performed.

Impact

- **Type:** Timing side-channel / username enumeration
- **Who is impacted:** Any deployment exposing the affected login endpoint
- **Security impact:** Unauthenticated attackers can confirm valid usernames one at a time, improving the effectiveness of credential stuffing, password spraying, phishing, and other targeted account attacks
- **Attack preconditions:** None beyond network access to the login endpoint
- **Confidentiality impact:** Low to moderate, depending on the sensitivity of account existence in the target environment

Severity

Low 3.7 / 10

CVSS v3 base metrics

Attack vector	Network
Attack complexity	High
Privileges required	None
User interaction	None
Scope	Unchanged
Confidentiality	Low
Integrity	None
Availability	None

Learn more about base metrics

CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N

CVE ID

CVE-2026-40263

Weaknesses

▶ CWE-208

Credits

QiaoNPC	Reporter
Across-Verticals-Malaysia	Reporter
enchant97	Remediation developer