COMMUNITY CONTENT

Home › Submit › 785631

Submit #785631: hcengineering platform v0.7.382 Authentication Bypass Issues

Title hcengineering platform v0.7.382 Authentication Bypass Issues

Description Description

The JWT token signing/verification module uses a hardcoded fallback value of the literal string `"secret"` when the `SERVER_SECRET` environment variable is not configured. Any attacker who knows this default (which is public in the open-source codebase) can forge arbitrary JWT tokens, including tokens with `admin: "true"` in the `extra` field, gaining full administrative control over the platform.

Vulnerable Code

File: `foundations/core/packages/token/src/token.ts` (Lines 48–50)

```typescript
const getSecret = (): string => {
  return getMetadata(serverPlugin.metadata.Secret) ?? 'secret'
}
```

This secret is used to both sign and verify all JWTs in the platform:

```typescript
// Line 100-111 — Token signing
return encode(
  {
    ...(extra !== undefined ? { extra } : {}),
    account: accountUuid,
    workspace: workspaceUuid,
    grant: sanitizedGrant,
    sub, exp, nbf
  },
  secret ?? getSecret()  // Falls back to 'secret'
)

// Line 117-123 — Token verification
export function decodeToken (token: string, verify: boolean = true, secret?: string): Token
{
  try {
    return decode(token, secret ?? getSecret(), !verify) // Falls back to 'secret'
  } catch (err: any) {
    throw new TokenError(err.message)
```

Community Content

Submissions are made by VulDB community users. VulDB is *not responsible* for their content nor the links to external sources.

Please use the raw information shown and the links listed *with caution*. They might contain malicious and harmful actions, code or data.

The corresponding VulDB entries contain the moderated, verified, and normalized information provided within the raw submission.

Documentation

- Submission Policy
- Data Processing
- CVE Handling

```
  }
}
```

The admin privilege is checked directly from the token's `extra` field without any server-side role lookup:

File: `server/account-service/src/index.ts` (Line 341)

```typescript
const payload = decodeToken(token)
if (payload.extra?.admin !== 'true') { // Admin check relies solely on JWT claim
  req.res.writeHead(404, {})
  ...
```

Attack Scenario

1. Attacker reads the open-source code and identifies the default secret is `'secret'`.
2. Attacker crafts a JWT: `jwt.encode({ account: "<any-uuid>", extra: { admin: "true" } }, "secret")`.
3. Attacker calls admin-only APIs (e.g., `deleteAccount`, `/api/v1/manage`) with the forged token.
4. Attacker can delete workspaces, delete accounts, impersonate any user, access any workspace data.

Impact

Complete platform compromise. Any self-hosted Huly instance that fails to set `SERVER_SECRET` (common in quick deployments, Docker Compose defaults, or development environments) is fully compromised. Since the default is embedded in public source code, this is a zero-click takeover for exposed instances.

Recommendation

1. **Remove the fallback entirely.** The application should refuse to start if the secret is not explicitly configured:

```typescript
const getSecret = (): string => {
  const secret = getMetadata(serverPlugin.metadata.Secret)
  if (secret == null || secret === '' || secret === 'secret') {
    throw new Error('FATAL: SERVER_SECRET is not configured. Refusing to start.')
  }
  return secret
}
```

2. Generate a cryptographically random secret on first deployment and persist it.
3. Enforce a minimum secret length (e.g., 32+ characters).
4. Add a startup health check that warns if common weak secrets are detected.

User Ghufran Khan (UID 95493)

Submission 03/22/2026 04:26 PM (15 days ago)

Moderation 04/05/2026 06:06 PM (14 days later)

Status	Accepted
VulDB entry	355412 [hcengineering Huly Platform 0.7.382 JWT Token token.ts SERVER_SECRET hard-coded key]
Points	17