

datavane / datavines Public

<> Code Issues 73 Pull requests 1 Discussions Actions Projects

[Fix][Security] Fix JWT authentication bypass via hardcoded secret and self-comparison logic #579

New issue

Open

jackieya wants to merge 3 commits into datavane:dev from jackieya:fix/jwt-auth-bypass

Conversation 6 Commits 3 Checks 4 Files changed 4

Changes from 1 commit File filter Conversations Jump to

[Fix][Security] Add null guard for tokenPassword in validateToken to ... < Prev Next >

...prevent NPE

jackieya committed 3 weeks ago

commit e540d6dc04e2e6ad11907fb655f3728a13e7b939

Some comments aren't visible on the classic Files Changed page.

3 datavines-core/src/main/java/io/datavines/core/utils/TokenManager.java

```

175     public boolean validateToken(String token, String username, String
176         password) {
177         String tokenUsername = getUsername(token);

```

Comment on lines 175 to 176

Copilot AI 3 weeks ago

validateToken(..., String password) now expects the stored BCrypt hash (from user.getPassword()), not a plaintext password. Renaming this parameter (e.g., passwordHash) would reduce confusion and help prevent accidental misuse in future call sites.

Copilot uses AI. Check for mistakes.

177	177	String tokenPassword = getPassword(token);
	178	+ if (StringUtils.isEmpty(tokenUsername)
		StringUtils.isEmpty(tokenPassword)) {
	179	+ return false;
	180	+ }
178	181	return (username.equals(tokenUsername) && BCrypt.checkpw(tokenPassword,
		password) && !(isExpired(token)));



Copilot AI 3 weeks ago

`BCrypt.checkpw(tokenPassword, password)` will likely throw if `tokenPassword` is null/empty (e.g., token missing the `up` claim, or `getPassword()` failed). Previously `password.equals(tokenPassword)` would safely return false. Consider guarding for missing/blank `tokenUsername` / `tokenPassword` and returning false instead of letting an exception bubble up as a 500.

Suggested change

```

181 - return (username.equals(tokenUsername) &&
      BCrypt.checkpw(tokenPassword, password) && !(isExpired(token)));
181 + if (StringUtils.isBlank(username) || StringUtils.isBlank(password)
182 +     || StringUtils.isBlank(tokenUsername) ||
      StringUtils.isBlank(tokenPassword)) {
183 +     return false;
184 + }
185 + return (StringUtils.equals(username, tokenUsername)
186 +         && BCrypt.checkpw(tokenPassword, password)
187 +         && !(isExpired(token)));

```



[Copilot](#) uses AI. Check for mistakes.

Comment on lines 175 to +181



Copilot AI 3 weeks ago

This validation approach still depends on the JWT carrying the user's plaintext password (`tokenPassword` comes from the token claims). That creates a high-impact credential exposure risk if a token is leaked, and it also makes per-request validation expensive (BCrypt on every request). Consider removing password from JWT claims entirely and validating via standard JWT claims (subject/userId) + signature/expiry, optionally with a server-side token revocation/version check.



[Copilot](#) uses AI. Check for mistakes.

179	182	}
180	183	