

01 漏洞简述

3月30日，GitLab 官方修复了CE/EE版本产品中硬编码密码导致的接管用户账户的安全问题。

由于使用 OmniAuth 注册的代码逻辑中存在硬编码密码，导致账号可被攻击者直接登录。该漏洞受影响版本为 14.7.0 ~ 14.7.6，14.8.0 ~ 14.8.4，14.9.0 ~ 14.9.1，均为较高版本，且该漏洞依赖于开启 OmniAuth 注册登录（用于CAS等三方登录场景），整体对企业实际影响较小。

02 漏洞时间线

- 3月22日，GitLab 表示到当天还未有用户帐户遭到破坏。
- 3月30日，GitLab 修复了硬编码密码的问题。
- 3月31日，GitLab 推出 14.9.2，14.8.5，14.7.7 三个安全版本。

03 漏洞分析

针对此次安全问题的修复 commit 将多处对Password.test_default 的调用修改为固定字符串，还删除了定义 test_default 方法的 lib/gitlab/password.rb 文件。

可见漏洞根源是 Password.test_default，这是在两个月前为了增加密码强度而引入的。从 password.rb 的注释得知其本意是为了测试构造出的强密码，但可能被误用于正常业务逻辑当中。

lib/gitlab/password.rb

0 — 100644

+14 -0

View file @a5a3a41a

```
1  + # frozen_string_literal: true
2  +
3  + # This module is used to return fake strong password for tests
4  +
5  + module Gitlab
6  +   module Password
7  +     DEFAULT_LENGTH = 12
8  +     TEST_DEFAULT = "123qweQWE!@#" + "0" * (User.password_length.max - DEFAULT_LENGTH)
9  +     def self.test_default(length = 12)
10 +       password_length = [[User.password_length.min, length].max, User.password_length.max].min
11 +       TEST_DEFAULT[...password_length]
12 +     end
13 +   end
14 + end
```

通过 test_default 的引用分析可以发现，lib/gitlab/auth/o_auth/user.rb 中通过 OAuth 方式创建用户时设置了 21 位的默认密码，即 “123qweQWE!@#0000000000” 。

lib/gitlab/auth/o_auth/user.rb +2 -2 View file @e2fb87ec

```
...  ... @@ -218,20+218,20 @@
218 218     def build_new_user(skip_confirmation: true)
219 219       user_params = user_attributes.merge(skip_confirmation: skip_confirmation)
220 220       Users::AuthorizedBuildService.new(nil, user_params).execute
221 221     end
222 222
223 223     def user_attributes
224 224       # Give preference to LDAP for sensitive information when creating a linked account
225 225       if creating_linked_ldap_user?
226 226         username = ldap_person.username.presence
227 227         name = ldap_person.name.presence
228 228         email = ldap_person.email.first.presence
229 229       end
230 230
231 231       username ||= auth_hash.username
232 232       name ||= auth_hash.name
233 233       email ||= auth_hash.email
234 234
235 235       valid_username = ::Namespace.clean_path(username)
236 236       valid_username = Uniquify.new.string(valid_username) { |s| !NamespacePathValidator.valid_path?(s)
237 237     }
238 238     {
239 239       name:      name.strip.presence || valid_username,
240 240       username:  valid_username,
241 241       email:     email,
242 242       password:  Gitlab::Password.test_default(21),
243 243       password_confirmation: Gitlab::Password.test_default(21),
244 244       password:  auth_hash.password,
245 245       password_confirmation: auth_hash.password,
246 246       password_automatically_set: true
247 247     }
248 248   end
end
```

CN-SEC | 中文网

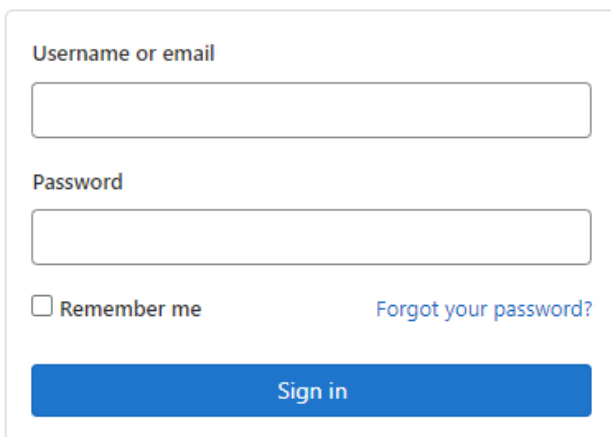
04

漏洞验证

- 1、拉取 gitlab-ce 受影响版本的 docker 镜像
- 2、注册 Github App，callback 链接设置为：
http://your-ip/users/auth/github/callback
- 3、开启 gitlab 中 OmniAuth 相关配置

```
1. gitlab_rails['omniauth_enabled'] = truegitlab_rails['omniauth_allow_single_sign_on'] = ['github']gitlab_rails['omniauth_block_auto_created_users'] = falsegitlab_rails['omniauth_providers'] = [{"name" => "github", "app_id" => "github_app_id", "app_secret" => "github_app_secret", "args" => { "scope" => "user:email" }}]
```

4、通过 OAuth github 登录（开启了 Github 和 Twitter 的 OAuth 登录如图）



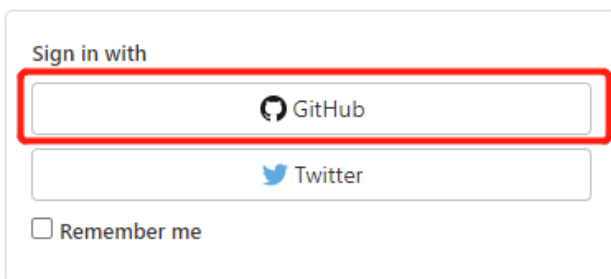
Username or email

Password


☐ Remember me [Forgot your password?](#)


Sign in

Don't have an account yet? [Register now](#)



Sign in with

 GitHub

 Twitter

☐ Remember me

CN-SEC | 中文网

5、用户被自动注册成功后，可以通过用户名和硬编码密码直接登录

径

/users/sign_in

/

/assets/application_utilities-4f92cbaa2387cd4a07d7edd3dde

/assets/application-cfa6748598b5e507db0e53906a7639e2c1

/assets/highlight/themes/white-462afd27a080b5e09a63dc7a

/assets/webpack/runtime.8a18edb1.bundle.js

× 标头 载荷 预览 响应 启动器 时间 Cookie

▼ 表单数据 查看源代码 查看网址编码格式的数据

authenticity_token: RkJtf30HxZBDSJe1UZvgSYUhSG2DY9T

user[login]:

user[password]: 123qweQWE!@#000000000


user[remember_me]: 0

Search GitLab

🔍 📄 📁 📧 ⚙️ 🌐


Welcome to GitLab

Faster releases. Better code. Less pain.




Create a project

Projects are where you store your code, access issues, wiki and other features of GitLab.




Create a group

Groups are the best way to manage projects and members.



Explore public projects

Public projects are an easy way to allow everyone to have read-only access.



Learn more about GitLab

Take a look at the documentation to discover all of GitLab's capabilities.

@

Set status

Edit profile

Preferences

Sign out

CN-SEC | 中文网