Published in d3crypt

d3crypt Follow
May 1, 2021 · 2 min read · Listen

Timing Attack on openmptcprouter-vps-admin authentication CVE-2021–31245

Ysurac/openmptcprouter-vps-admin
OpenMPTCProuter VPS admin rest api. Contribute to Ysurac/openmptcprouter-vps-admin development by creating an account...
github.com

openmptcprouter-vps-admin Version 0.57.3 and before is vulnerable to timing attack during authentication based on 'Basic Authentication' mechanism.

The weakness exists in the file omr-admin.py on line 779 where the password supplied in the HTTP request is compared with the one in the configuration. The method uses python string comparison '==' method which internally compares the string one character at a time without checking the string length returning false only when a character differs. More analysis has already been done and can be found here

Let's look at the code snippet responsible for this weakness,

```
2 ■■□□□ omr-admin.py
@@ -776,7 +776,7 @@ def set_lastchange(sync=0):
776   fake_users_db = omr_config_data['users'][0]
777
778   def verify_password(plain_password, user_password):
779 -     if plain_password == user_password:
780           LOG.debug("password true")
781           return True
782       return False
```

I reported the issue to the author https://github.com/Ysurac and a fix was issue within hours the same day. After fix, the code snippet looks like the below,

```
776   fake_users_db = omr_config_data['users'][0]
777
778   def verify_password(plain_password, user_password):
779 +     if secrets.compare_digest(plain_password,user_password):
780           LOG.debug("password true")
781           return True
782       return False
```

(Link to the commit history)

Basically, the author used python secrets module's "compare_digest()" method to securely compare strings. Official document for this module can be found here

This issue has been reported to MITRE and they assigned CVE-2021–31245 to it.

> *Takeaways,*
>
> *Although timing attacks are difficult, they are not impossible. So one must account for these possibilities when implementing functions such as authentication/authorization or anything that has a security impact.*

Please subscribe to d3crypt on YouTube
www.youtube.com

Please follow me on Medium / twitter as well :)