Biometric fingerprint recognition USB3.0 memory (PD065) replay attack vulnerability

High **bosslabdcu** published **GHSA-xvrv-w76r-gh28** on Jun 30, 2021

Package	
fingertool (Commercial Software)	
Affected versions	Patched versions
1.19	Not patched

Description

Biometric fingerprint recognition USB3.0 memory (PD065) replay attack vulnerability

Target

PD065 v.1.19

Impact

Through this replay attack, it can be verified that an attacker who does not know the password bypasses the authentication even if an incorrect password is entered. Additionally, a function of deleting a user or adding a new malicious user can be used through a replay attack. Finally, the encrypted data can also be stolen, and the **data stored inside safely can be stolen even if the encrypted password itself is not stolen.**

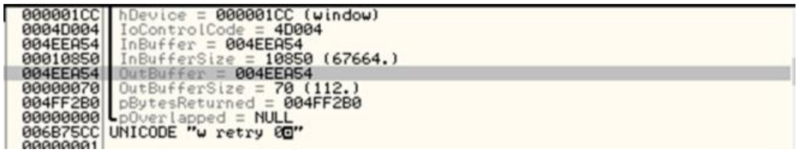
Summary

Storage security technologies such as secure flash memory have emerged for secure storage data.
Such secure technologies include user authentication and access control technologies, and dual user authentication technologies are primarily used.
Password authentication is typical in user authentication technology and is used in conjunction with iris and fingerprint authentication. For this reason, we recently selected Wizflat's DM PD065 secure flash memory, one of the most used secure flash memories, to analyze the vulnerability of replay attacks of the password authentication feature applied to the product.
The security flash memory (DM PD065) uses a total of two user authentication technologies: password authentication and fingerprint authentication.
In other words, a user is identified through their fingerprint and password, and if any information is exposed, the protected data of the authorized user can be accessed maliciously.
In password authentication technology, users who use the product set their own passwords.

Analysis

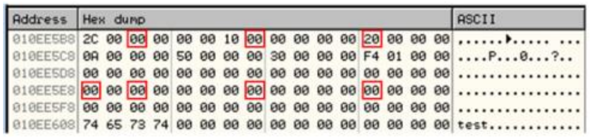
The method applied to fingerprint recognition USB, which is an analysis target, entails the input password being transmitted to the *Authentication Module, and the *Authentication Module compares the registered password with the input password. Additionally, the authentication result (the validity of the password), which is the result of the comparison, is determined and transmitted to the *management program (Fingertool).
Owing to this method, vulnerabilities of existing password authentication technology, such as the hard-coded password vulnerability in which the password is exposed in the source code as it is, do not appear. However, in this authentication method, the result is transmitted to the *management program, and the user is authenticated based on the result. This implies that even if a malicious attacker collects the correct (normal) authentication result first and enters an incorrect (abnormal) password, replay attack that can bypass the authentication based on the collected result can occur.
Considering the vulnerability of the replay attack through the analysis result, the *Authentication Module is the hardware, the USB in this context, to be analyzed and the authentication result is received by the *management program. Therefore, to obtain the outcome of receiving the authentication result from the hardware, the analysis was performed using the DeviceIoControl function. Because of the function analysis, a total of 8 parameters were in place when calling the DeviceIoControl function, and important information among them was stored in the IN Buffer and OutBuffer. In addition, data transferred to the driver was stored in the InBuffer and data transferred from the driver is stored in OutBuffer. Based on this analysis result, after entering the correct password, the data stored in OutBuffer can be recorded, and even if an incorrect password is entered, the recorded data can be forcibly injected. Consequently, it can be confirmed that the authentication was performed normally.

DeviceIoControl Function



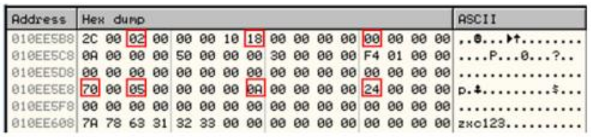
```
000001CC  hDevice = 000001CC (window)
0004D004  IoControlCode = 4D004
004EEA54  InBuffer = 004EEA54
00010850  InBufferSize = 10850 (67664.)
004EEA54  OutBuffer = 004EEA54
00000070  OutBufferSize = 70 (112.)
004FF2B0  pBytesReturned = 004FF2B0
00000000  pOverlapped = NULL
006B75CC UNICODE "w retry 0"
00000001
```

correct password



```
Address  Hex dump                                         ASCII
010EE5B8 2C 00 00 00 00 00 10 00 00 00 00 00 20 00 00 00 ......▶..... ...
010EE5C8 0A 00 00 00 50 00 00 00 30 00 00 00 F4 01 00 00 ....P...0...?..
010EE5D8 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
010EE5E8 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
010EE5F8 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
010EE608 74 65 73 74 00 00 00 00 00 00 00 00 00 00 00 00 test............
```

incorrect password



```
Address  Hex dump                                         ASCII
010EE5B8 2C 00 02 00 00 00 10 18 00 00 00 00 00 00 00 00 ..☻...▶↑........
010EE5C8 0A 00 00 00 50 00 00 00 30 00 00 00 F4 01 00 00 ....P...0...?..
010EE5D8 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
010EE5E8 70 00 05 00 00 00 00 0A 00 00 00 00 24 00 00 00 p.♣.........$...
010EE5F8 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
010EE608 7A 78 63 31 32 33 00 00 00 00 00 00 00 00 00 00 zxc123..........
```

Patches

Not patched

Workarounds

Code Obfuscation, Code Encryption, Digital Watermarking etc..

Discoverer(s)/Credits

Kyungroul Lee/South Korea/carpedm@mnu.ac.kr
Jae hyuk Lee/south korea/gurmggg@cu.ac.kr

For more information

If you have any questions or comments about this advisory:

- Open an issue in example link to repo
- Email us at boss lab email address

Severity

High 7.8 / 10

CVSS base metrics	
Attack vector	Local
Attack complexity	Low
Privileges required	Low
User interaction	None
Scope	Unchanged
Confidentiality	High
Integrity	High
Availability	High

CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H

CVE ID

CVE-2021-26824

Weaknesses

CWE-259 CWE-321 CWE-798

Credits

jh1113