Directory traversal vulnerability exists in uploaded and downloaded files #48

Closed marckwei opened this issue on Dec 1, 2020 · 1 comment

marckwei commented on Dec 1, 2020

Directory traversal

Through code audit, it is found that the file download function in flamingo has a problem with directory traversal. Through this vulnerability, files can be downloaded anywhere on the server through the directory.

Test environment

```
mysql> select version();
+-------------------------+
| version()               |
+-------------------------+
| 5.7.32-0ubuntu0.16.04.1 |
+-------------------------+
1 row in set (0.02 sec)
```

Vulnerability analysis

Flamingo is a C/S mode communication software.User A sends the file to user B. The server saves the file in A specific folder of the server and waits for User B to receive it. After User B sends the receive request, the server sends the corresponding file to user B.

When uploading files, use the result of file md5 encoding as the file name (unfortunately, the encryption process is on the client side).

The base directory of the cache file is hard-coded in the configuration file, and the corresponding file path is directly spliced through the base directory and the md5 result. The file has no identification for a specific user, all files exist together, and there is no distinction between different users (that is, the server does not know who the file belongs to, and it can be downloaded as long as the correct file path is provided to the server).

Poc

From the simple analysis above, it can be seen that this file transfer function has a lot of security issues. Only the most serious problems are demonstrated here.

It can be seen from the declaration of the `onDownloadFileResponse` function in `FileSession.cpp`

```
string filename = m_strFileBaseDir;
filename += filemd5;
m_fp = fopen(filename.c_str(), "rb+");
```

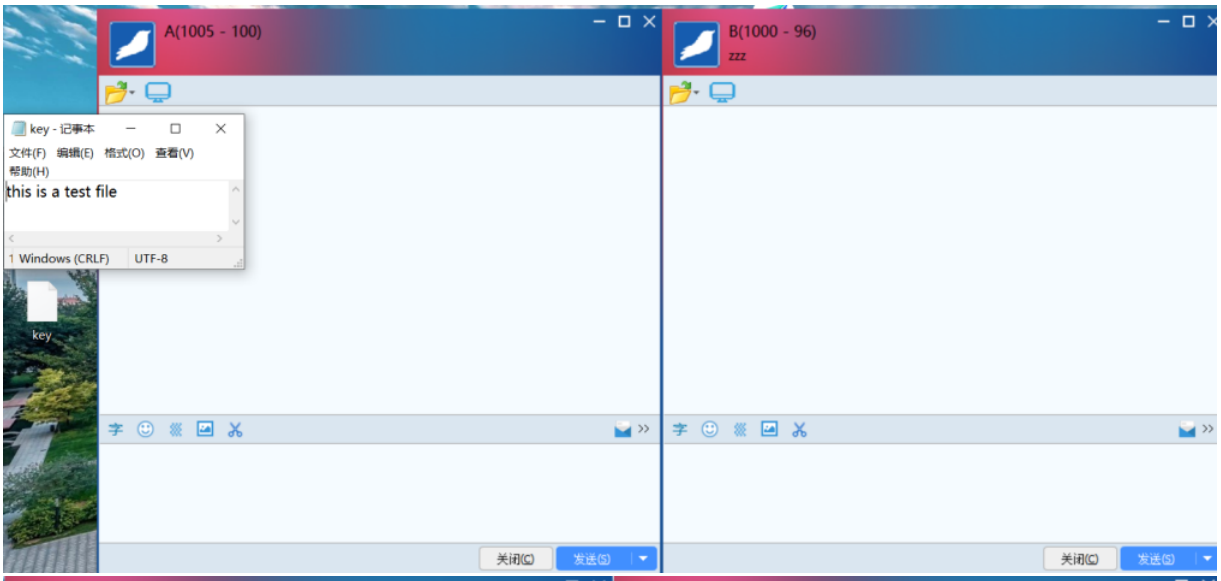
Since the download path is directly spliced by the base directory and the md5 result, as long as the file name can be controlled, the file name of the form ../../../pwd.txt can be used to achieve directory traversal and download any file .

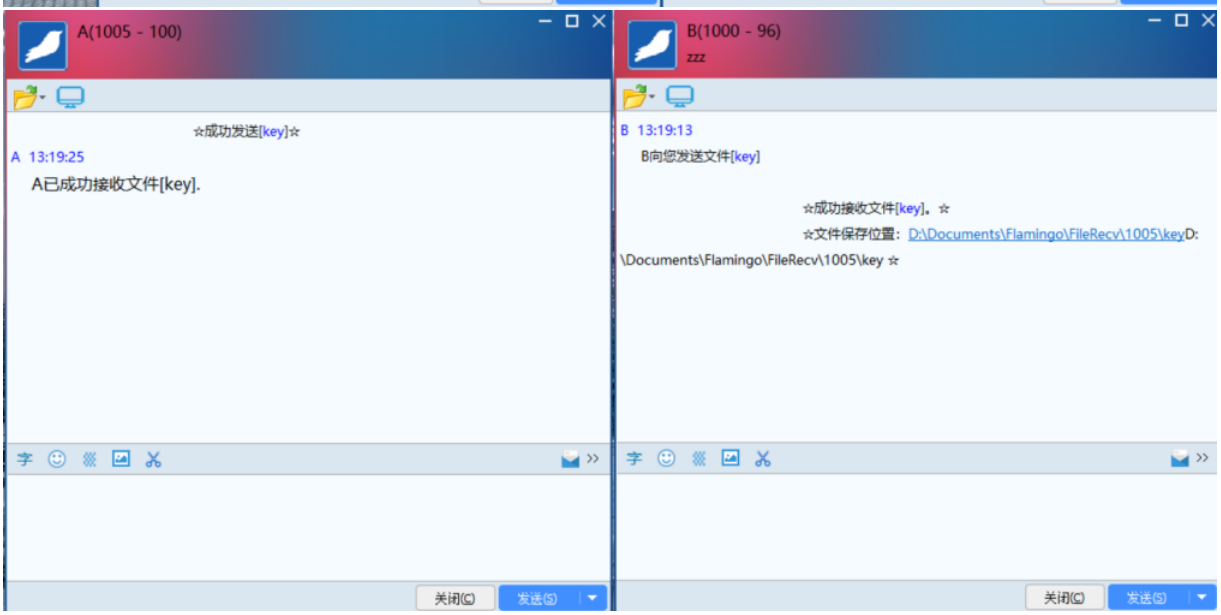
Flamingo's problem is that MD5 encryption is done on the client side, and because the communication protocol is open source, it is easy to forge.

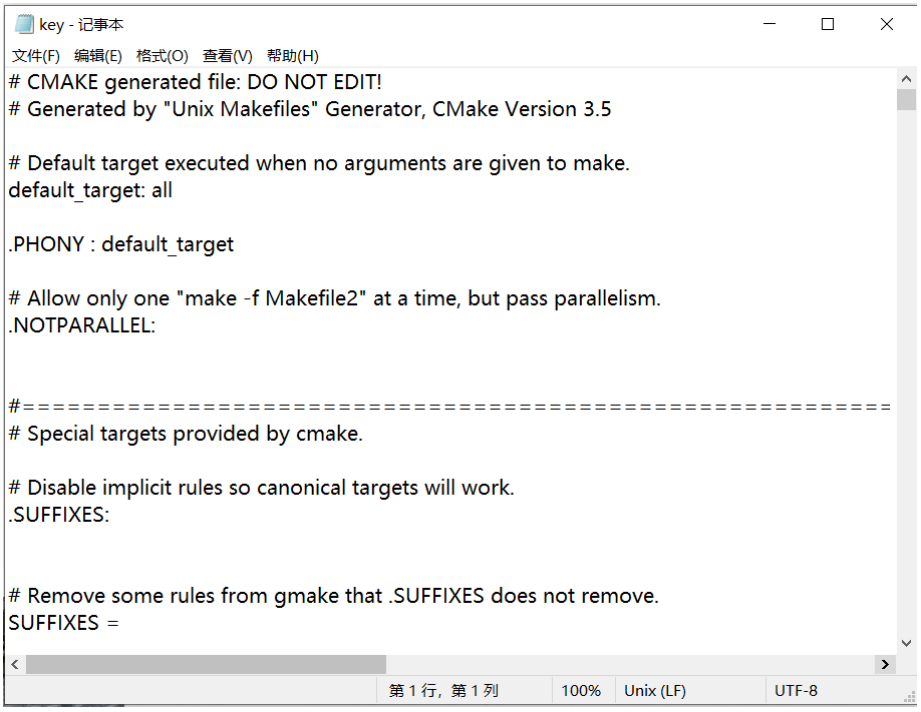
Find the location where the client sends the download command and tamper with the file name.

Add the following statement to the `filetaskThread.cpp`

```
274     uploadFileResult.m_nFileSize = nFileSize;
275     strcpy_s(szMd5, ARRAYSIZE(szMd5), "../../../../Desktop/message"); //Tamper with filename
276     strcpy_s(uploadFileResult.m_szMd5, ARRAYSIZE(uploadFileResult.m_szMd5), szMd5);
277
```



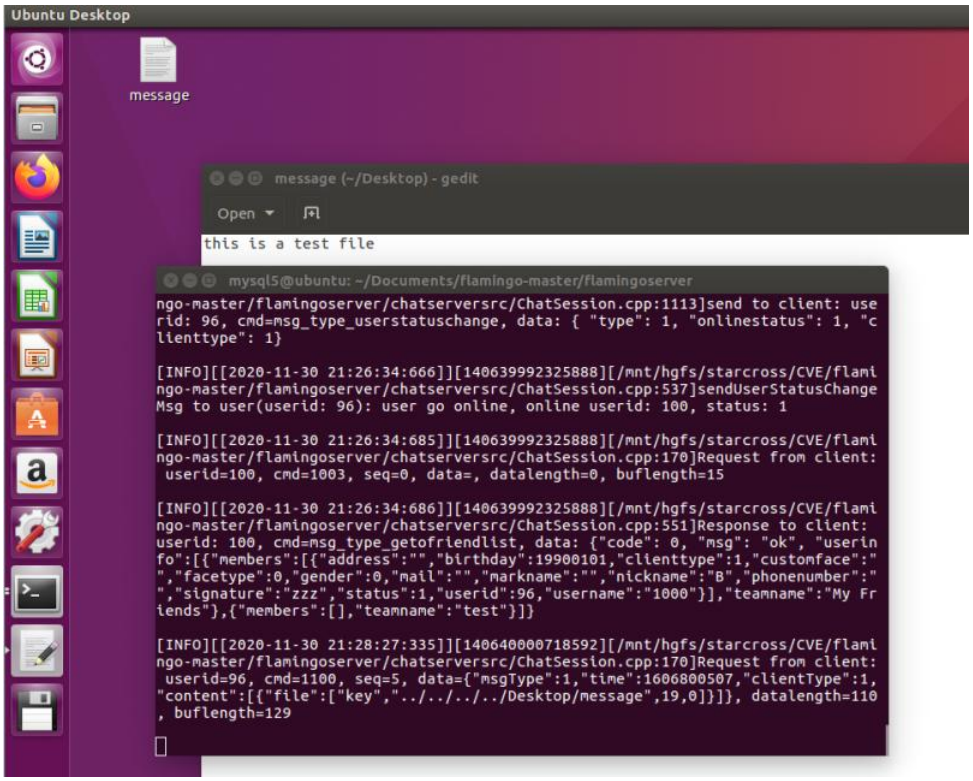




During the test, it is found that when the tampered file path does not exist, the server will first create the file, then write the contents of the sent file, and then download it for the recipient.So using this vulnerability can also achieve arbitrary location write (can be multi-level directory traversal).

```
274     uploadFileResult.m_nFileSize = nFileSize;
275     strcpy_s(szMd5, ARRAYSIZE(szMd5), "../../../../Desktop/message"); //篡改文件名
276     strcpy_s(uploadFileResult.m_szMd5, ARRAYSIZE(uploadFileResult.m_szMd5), szMd5);
277
```

Send the file again here.The file is written on Desktop.



balloonwj commented on Dec 1, 2020 — Owner

@marckwei yes， you are right. If use flamingo for commercial use, remember to enhance this. Not adding this additional checks and enhancement are just for simplicity for users who study it.

balloonwj closed this as completed on Dec 1, 2020

Assignees
No one assigned

Labels
None yet

Projects
None yet

Milestone
No milestone

Development
No branches or pull requests

2 participants