251 – Find the suspect

Team Information

Team Name: kimbabasaksaksak

Team Member: Jaeheon Kim, Donghyun Kim, Soyoung Yoo, Minhee Lee

Email Address : uaaoong@gmail.com

Instructions

Description Investigators impound a MacBook believed to have been used in the crime at the crime scene. However, it is difficult to solve the case because the suspect of the crime is not specified. Analyze the evidence file to identity of the criminal.

Target	Hash (MD5)
Users.zip	56E3072B8D12D449B57E47A90BB35CAF

Questions

1) Find all files that appear to be related to the crime, and identify the file name and upload or download time (UTC+9) (20 points)
2) Identify suspect's name and email address. (80 points)
3) Submit the tool to decrypt the dbx. (150 points)

Teams must:

- Develop and document the step-by-step approach used to solve this problem to allow another examiner to replicate team actions and results.
- Specify all tools used in deriving the conclusion(s).
-

Tools used:

Name:	DB Browser for SQLite	Publisher:	sqlitebrowser
Version:	3.12.0.0		
URL:	https://sqlitebrowser.org/		

Name:	sqlite3-dbx	Publisher:	newsoft
Version:	-		
URL:	https://github.com/newsoft/sqlite3-dbx		

Name:	Sublime Text 4	Publisher:	Sublime Text
Version:	4143		
URL:	https://www.sublimetext.com/		

Name:	Sqlite-dissect	Publisher:	dod-cyber-crime-center
Version:	V0.2.0		
URL:	https://github.com/dod-cyber-crime-center/sqlite-dissect/		

Step-by-step methodology:

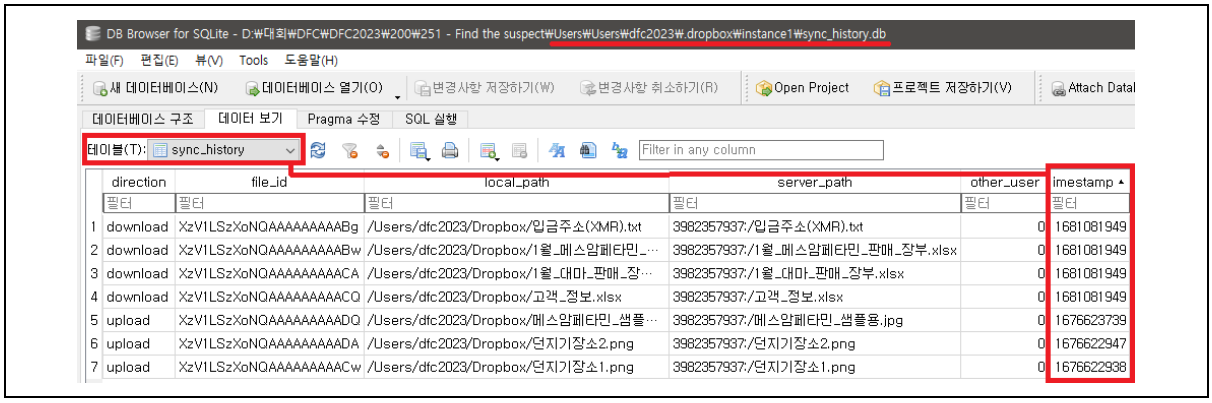
1. Find all files that appear to be related to the crime, and identify the file name and upload or download time (UTC+9)

문제에서 주어진 Users.zip 압축 파일을 해제하면 dropbox 아티팩트들을 확인할 수 있다.

```
PS D:\대회\DFC\DFC2023\200\251 - Find the suspect\Users\Users\dfc2023\.dropbox> ls


    디렉터리: D:\대회\DFC\DFC2023\200\251 - Find the suspect\Users\Users\dfc2023\.dropbox


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        2023-04-25   오후 6:00                Crashpad
d-----        2023-04-25   오후 6:00                events
d-----        2023-04-25   오후 6:00                instance1
d-----        2023-04-25   오후 6:00                logs
d-----        2023-04-25   오후 6:00                machine_storage
d-----        2023-04-25   오후 6:00                metrics
d-----        2023-04-25   오후 6:00                QuitReports
-a----        2023-04-25   오후 6:00           4096 apex.sqlite3
-a----        2023-04-25   오후 6:00              3 dropbox.pid
-a----        2023-04-25   오후 6:00            117 info.json


PS D:\대회\DFC\DFC2023\200\251 - Find the suspect\Users\Users\dfc2023\.dropbox>
```

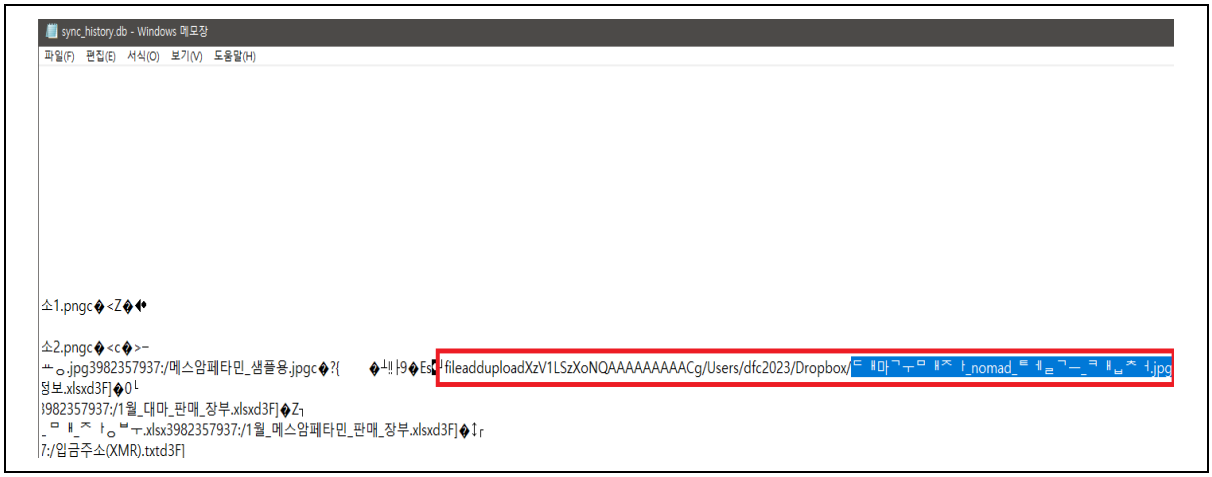
[그림 1-1] User.zip 압축 해제

sync_history.db(.dropbox₩instance1₩sync_history.db)를 확인하면 로컬에서 동기화 목적으로 upload, download 된 기록을 확인할 수 있다.



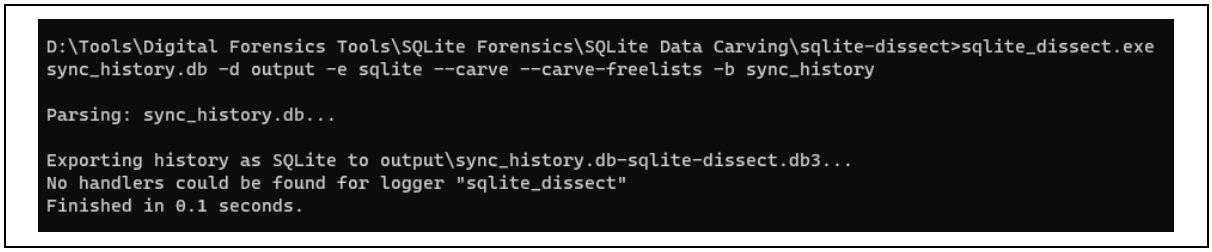
[그림 1-2] sync_history.db 확인

추가로 Notepad를 사용하여 해당 데이터베이스를 확인할 경우 숨겨진 데이터를 식별할 수 있다.

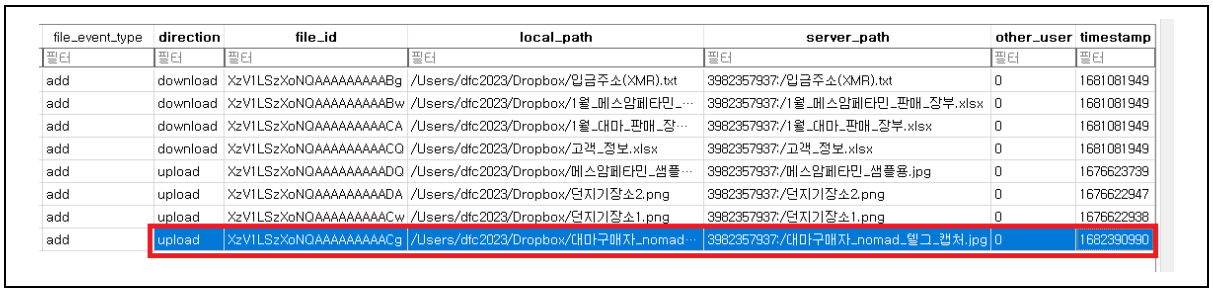


[그림 1-3] sync_history.db 숨겨진 데이터

SQL Database Carving 도구인 sqlite-dissect을 사용하여 복구하면 해당 데이터를 정상적으로 식별할 수 있다.



[그림 1-4] sync_history.db 카빙(sqlite-dissect)



file_event_type	direction	file_id	local_path	server_path	other_user	timestamp
필터	필터	필터	필터	필터	필터	필터
add	download	XzV1LSzXoNQAAAAAAAAABg	/Users/dfc2023/Dropbox/입금주소(XMR).txt	3982357937:/입금주소(XMR).txt	0	1681081949
add	download	XzV1LSzXoNQAAAAAAAAABw	/Users/dfc2023/Dropbox/1월_메스암페타민_…	3982357937:/1월_메스암페타민_판매_장부.xlsx	0	1681081949
add	download	XzV1LSzXoNQAAAAAAAAACA	/Users/dfc2023/Dropbox/1월_대마_판매_장…	3982357937:/1월_대마_판매_장부.xlsx	0	1681081949
add	download	XzV1LSzXoNQAAAAAAAAACQ	/Users/dfc2023/Dropbox/고객_정보.xlsx	3982357937:/고객_정보.xlsx	0	1681081949
add	upload	XzV1LSzXoNQAAAAAAAADQ	/Users/dfc2023/Dropbox/메스암페타민_샘플…	3982357937:/메스암페타민_샘플용.jpg	0	1676623739
add	upload	XzV1LSzXoNQAAAAAAAADA	/Users/dfc2023/Dropbox/던지기장소2.png	3982357937:/던지기장소2.png	0	1676622947
add	upload	XzV1LSzXoNQAAAAAAAACw	/Users/dfc2023/Dropbox/던지기장소1.png	3982357937:/던지기장소1.png	0	1676622938
add	upload	XzV1LSzXoNQAAAAAAAACg	/Users/dfc2023/Dropbox/대마구매자_nomad…	3982357937:/대마구매자_nomad_텔그_캡처.jpg	0	1682390990

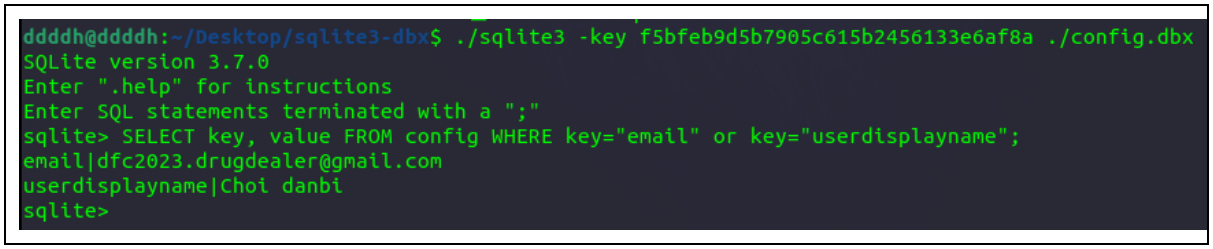
[그림 1-5] sync_history.db 모든 데이터 확인

다음은 범죄에 관련된 파일의 정보이다.

Type	File	Time(UTC+9)
Download	/Users/dfc2023/Dropbox/입금주소(XMR).txt	2023-04-10 08:12:29
Download	/Users/dfc2023/Dropbox/1월_메스암페타민_판매_장부.xlsx	2023-04-10 08:12:29
Download	/Users/dfc2023/Dropbox/1월_대마_판매_장부.xlsx	2023-04-10 08:12:29
Download	/Users/dfc2023/Dropbox/고객_정보.xlsx	2023-04-10 08:12:29
Upload	/Users/dfc2023/Dropbox/메스암페타민_샘플용.jpg	2023-02-17 17:48:59
Upload	/Users/dfc2023/Dropbox/던지기장소2.png	2023-02-17 17:35:47
Upload	/Users/dfc2023/Dropbox/던지기장소1.png	2023-02-17 17:35:38
Upload	/Users/dfc2023/Dropbox/대마구매자_nomad_텔그_캡처.jpg	2023-04-25 11:49:50

2. Identify suspect's name and email address.

해당 사용자의 이름과 이메일을 식별하기 위해서는 암호화된 데이터베이스(config.dbx)를 확인해야 한다. config.dbx는 .dropbox₩instance1₩config.dbx 경로에 위치하며 해당 데이터에 접근하기 위해서는 키 값이 필요하다. 키를 구하는 방법에 대해서는 3번 지문에서 확인할 수 있다. 해당 문제에서 주어진 config.dbx의 키는 "f5bfeb9d5b7905c615b2456133e6af8a"이며 sqlite3-dbx(https://github.com/newsoft/sqlite3-dbx)을 사용하여 복호화한 데이터베이스를 확인할 수 있다.



[그림 2-1] sqlite3-dbx을 사용하여 복호화한 config.dbx

다음은 config.dbx에서 이름과 이메일 정보를 확인한 결과이다.

Key	Value
userdisplayname	Choi danbi
email	dfc2023.drugdealer@gmail.com

3. Submit the tool to decrypt the dbx.

해당 문제에서 암호화되어 있는 대상은 config.dbx이다.

1. 맥버전 복호화 키 생성(https://github.com/dnicolson/dbx-keygen-macos/blob/master/dbx-keygen-macos.py)
2. 윈도우 복호화 키 생성(https://github.com/newsoft/dbx-keygen-windows/blob/master/dbx-keygen-windows.py)
3. 리눅스 복호화 키 생성(https://github.com/newsoft/dbx-keygen-linux/blob/master/dbx-keygen-linux.py)

복호화 키를 구하는 방법은 운영체제에 따라 조금씩 차이가 있으며 문제 지문에서 맥북을 사용하고 있는 점을 확인할 수 있다. 맥버전 복호화 키 생성 스크립트를 확인해보면 로컬에 설치된 dropbox의 config.dbx에 한해서만 동작한다. 하지만 스크립트 내 무차별 대입 공격 요소를 이용하면 타 시스템에 존재하는 config.dbx 키를 구할 수 있다.

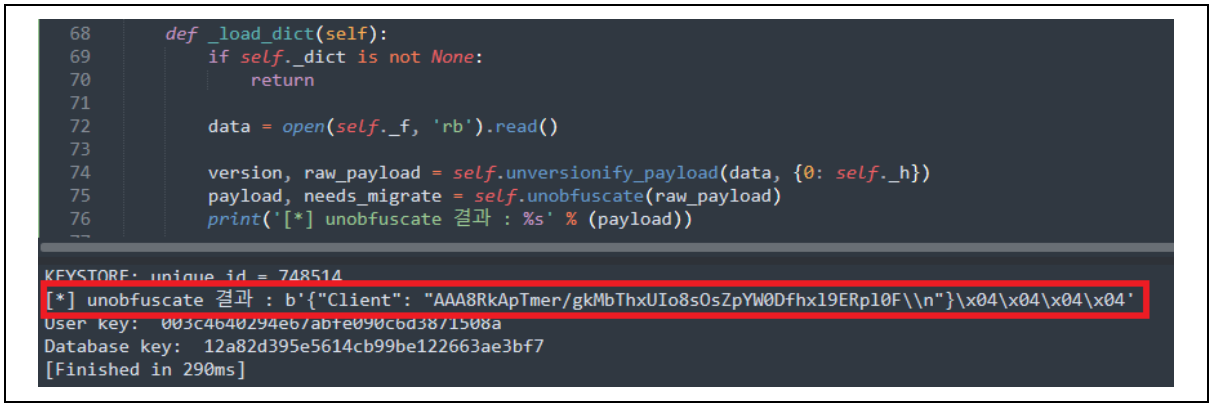
키 생성 스크립트의 unique_id 함수를 확인하여 보면 현재 로컬 시스템의 inode를 구하여 AES 알고리즘의 키로 사용하는 것을 확인할 수 있다. 이 때 사용되는 inode 값은 64비트 크기의 정수 형태로서 무차별 대입 공격이 가능하다.

```
38  class KeyStoreFileBacked(object):
39
40      def __init__(self, appdata_path):
41          self._appdata_path = appdata_path
42          self._s = self.id2s(self.unique_id(self._appdata_path))
43          self._i = 'l\x078\x014$sX\x03\xffri3\x13aQ'
44          self._h = '\x8f\xf4\xf2\xbb\xad\xe9G\xea\x1f\xdfim\x80[5>'
45          self._f = os.path.join(self._appdata_path, u'hostkeys')
46          self._dict = None
47          # simplified version
48
49      def id2s(self, _id):
50          return hashlib.md5('ia9%sX' % _id + 'a|ui20').digest()
51
52      def obfuscate(self, data):
53          encryptor = AES.new(key=self._s, mode=AES.MODE_CBC, IV=self._i)
54          return encryptor.encrypt(data)
55
56      def _unobfuscate_low(self, data, key):
57          decryptor = AES.new(key, mode=AES.MODE_CBC, IV=self._i)
58          return decryptor.decrypt(data)
59
60      def unobfuscate(self, data):
61          return (self._unobfuscate_low(data, self._s), False)
62

104 class KeyStore(KeyStoreFileBacked):
105
106     def unique_id(self, path):
107         inode = os.stat(path).st_ino
108         print 'KEYSTORE: unique_id = %r %s'  % (path, '%d' % inode)
109         return ('%d' % inode).encode('ascii')
110
```

[그림 3-1] 스크립트 내 unique_id 함수

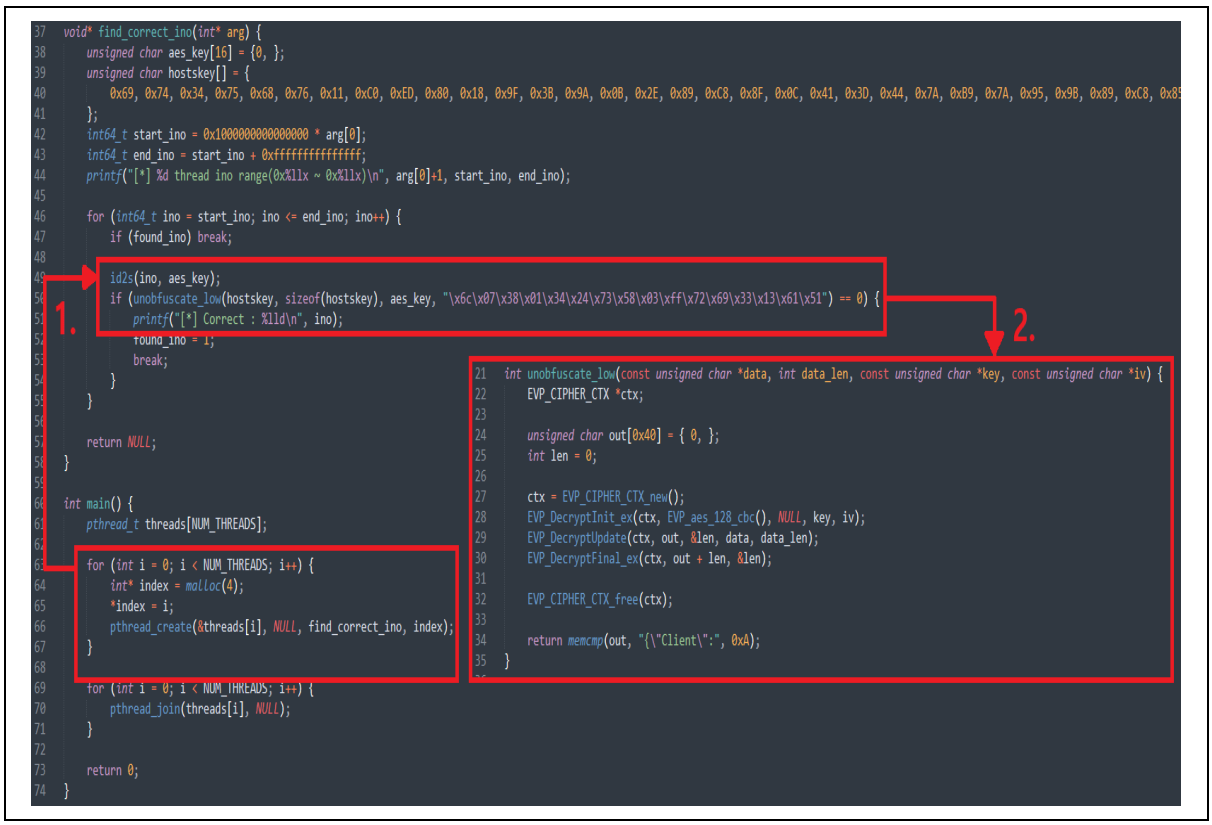
무차별 대입 공격의 성공, 실패 판별을 위해선 unobfuscate 함수의 정상적인 결과를 알아야 한다. [그림 3-2]에서 임의로 만든 dropbox를 사용한 정상적인 unobfuscate *결과 값을 확인할 수 있다.

*unobfuscate 결과: {"Client": "AAA8RkApTmer/gkMbThxUIo8sOsZpYW0Dfhxl9ERpl0F\\\n"}\x04\x04\x04\x04



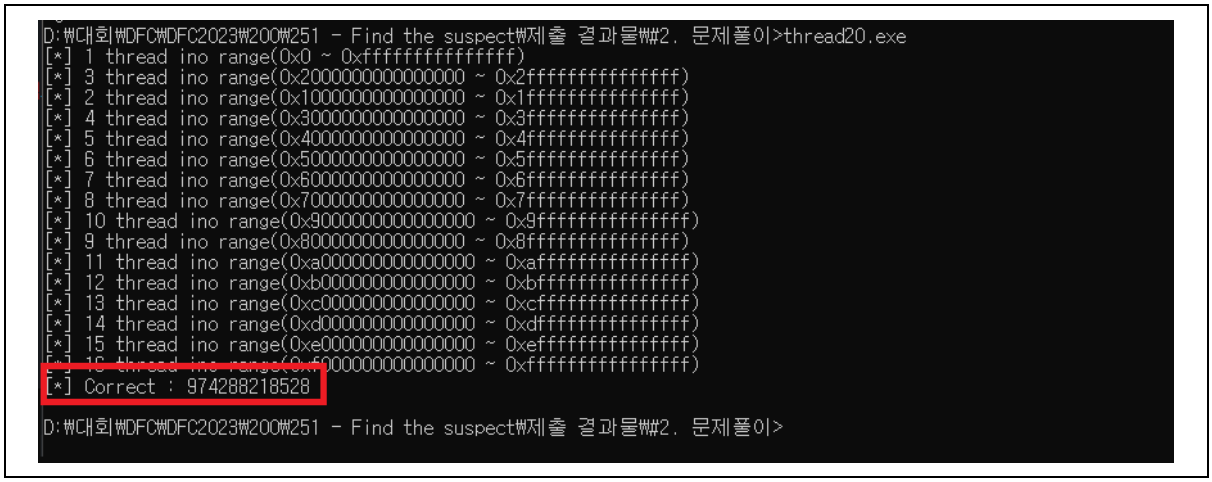
[그림 3-2] unobfuscate 결과

이로써 inode 값을 증가하며 unobfuscate의 결과가 Client 키를 가진 JSON 데이터임을 검증하는 무차별 대입 공격 프로그램을 만들면 된다. 한 번의 연산으로 고정 값을 가지는 부분들을 하드코딩하여 최적화하고 쓰레드를 사용하여 inode를 구한다. 여러 쓰레드 중 AES 디크립트 데이터의 결과가 {"Client" 문자열을 가지고 있을 경우 올바른 inode로 간주하며 프로그램을 종료한다.



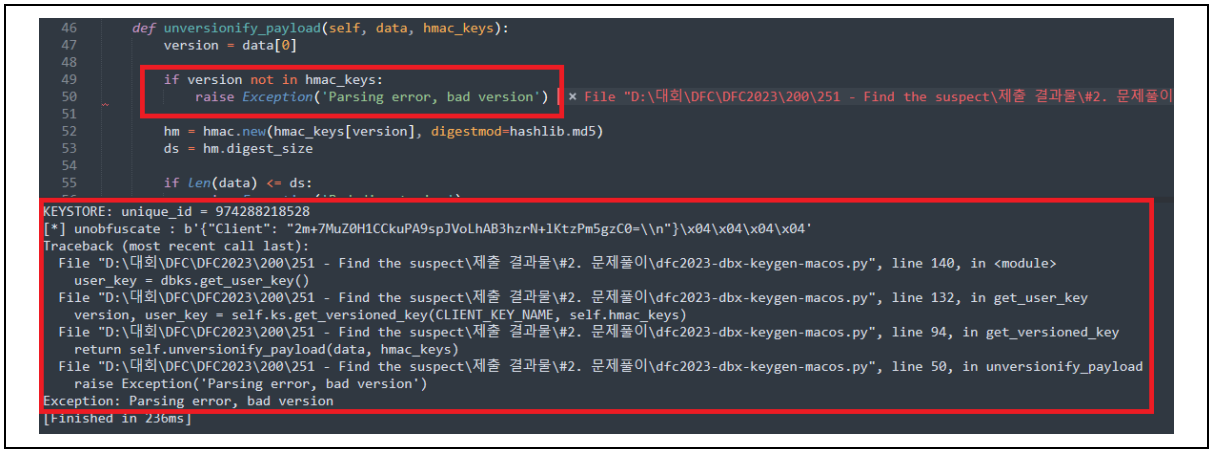
[그림 3-2] 무차별 대입 공격 코드 일부

프로그램을 일정 시간 실행하면 올바른 값을 얻을 수 있으며, 문제에서 사용된 환경의 inode 값은 974288218528 임을 확인할 수 있다.



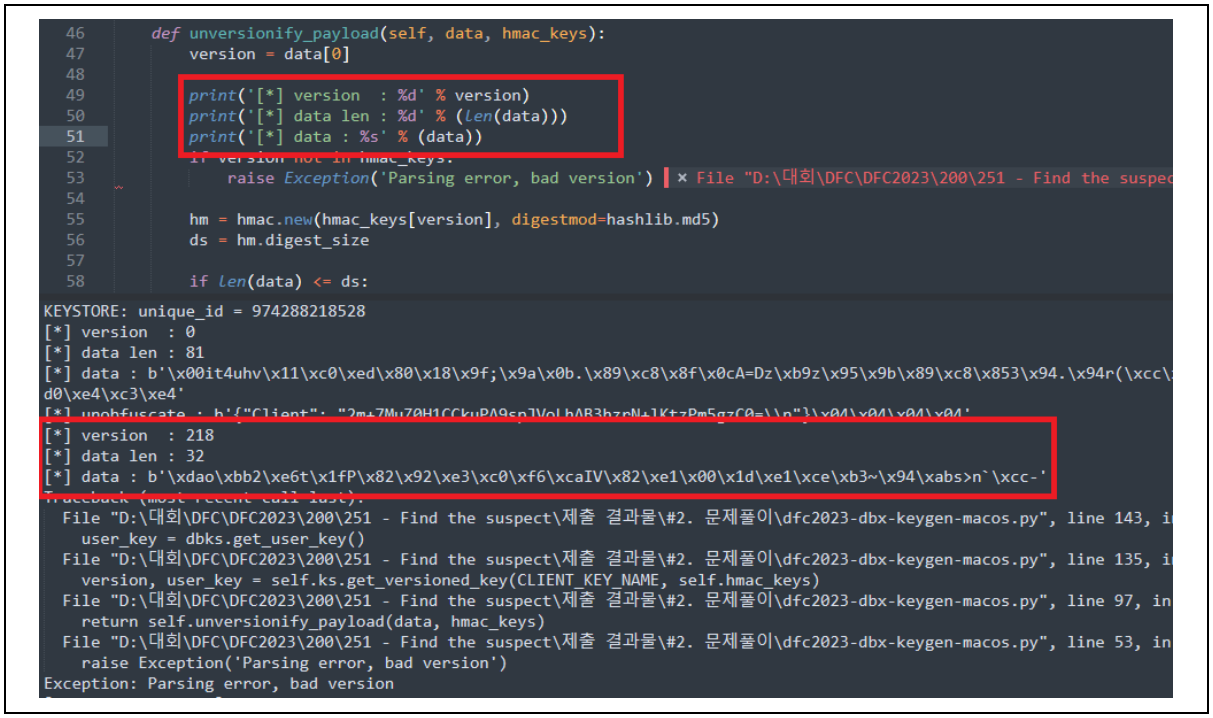
[그림 3-3] 무차별 대입 공격 실행 결과

올바른 inode 값을 하드코딩하여 결과를 확인해보면 정상적으로 Client 키가 존재하는 JSON 데이터를 복호화했지만 도중에 버전 오류가 발생한다.

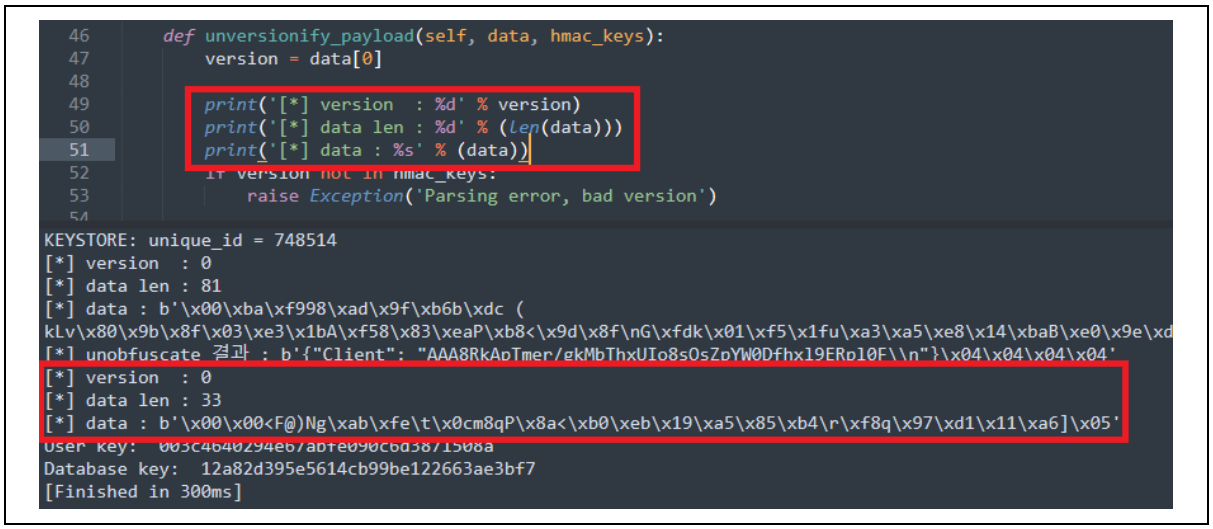


[그림 3-4] 버전 오류

오류를 확인하고자 실험파일과 비교하면 버전과 데이터 길이가 잘못된 것을 확인할 수 있다.

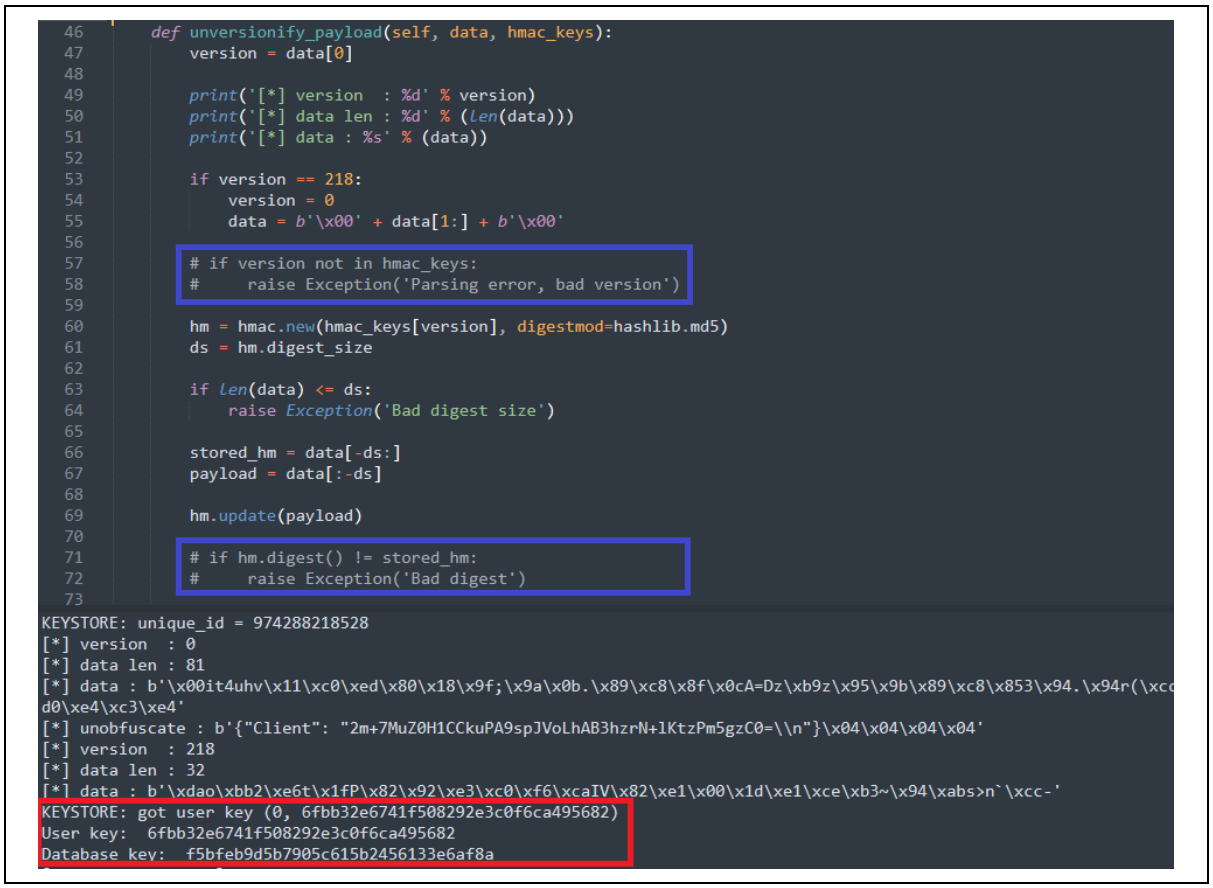


[그림 3-5] 버전 오류 출력 결과 (문제파일)



[그림 3-6] 버전 오류 출력 결과 (실험파일)

config.dbx 를 복호화 하기 위해서는 버전 값이 항상 0이어야 하기 때문에 해당 값을 하드 코딩하고 길이를 맞춰준다. 추가로 임의 수정한 부분이 있기에 검증 부분을 주석 처리하면 정상적으로 키를 구할 수 있다. 데이터베이스(config.dbx)의 올바른 키는 f5bfeb9d5b7905c615b2456133e6af8a 이다. 해당 키를 sqlite3-dbx의 인자로 전달하여 복호화된 데이터베이스(config.dbx)를 확인할 수 있다.



[그림 3-7] 올바른 config.dbx 키

다음은 첨부된 파일 목록과 설명이다.

Name	Summary
#1. 테스트용/config.dbx	결과 확인용 임의로 만든 테스트케이스
#1. 테스트용/hostkeys	결과 확인용 임의로 만든 테스트케이스
#1. 테스트용/test-dbx-keygen-macos.py	테스트케이스용 Inode를 지정해둔 코드
#2. 문제풀이/brute.c	문제의 올바른 indoe 값을 구하는 소스코드
#2. 문제풀이/thread20.exe	문제의 올바른 inode 값을 구하는 실행파일
#2. 문제파일/cconfig.dbx	문제에서 사용된 config.dbx
#2. 문제파일/hostkeys	문제에서 사용된 hostkeys

#2. 문제파일/dfc2023-dbx-keygen-macos.py	문제에서 사용된 inode 값으로 데이터베이스 키를 구하는 코드
#2. 문제파일/sqlite3-dbx-master.zip	복호화 키를 사용해 암호화된 config.dbx를 볼 수 있는 툴