
2021 하계 해킹캠프 CTF

Unique Crack write up

by arrester (Demon Team)



작성자: 김주원(arrester)

arresterloyal@gmail.com

<https://github.com/Demon-KR>

<https://blog.naver.com/lstarrlodyl>

목 차

I 문제 개요 및 목표	3
I -1. 문제 개요	3
I -2. 문제 목표	3
II 문제 풀이	4
II-1. 정보 수집	4
II-2. 환경 준비 및 풀이	7
III 결론	14

2021 하계 해킹캠프 CTF

I 문제 개요 및 목표

I -1. 문제 개요

문제 제목: Unique Crack

문제 내용: find the id and password

문제 출제자: arrester

I -2. 문제 목표

아이디와 비밀번호를 찾아라!

2021 하계 해킹캠프 CTF

II 문제 풀이

II-1. 정보 수집

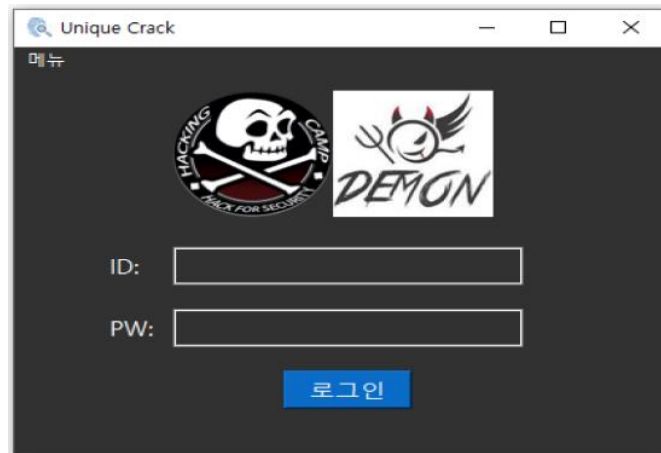


그림 1 UniqueCrack_starter.exe

주어진 프로그램을 실행하면 [그림 1]과 같다. ID와 PW 입력 칸이 있는 것을 확인할 수 있다.

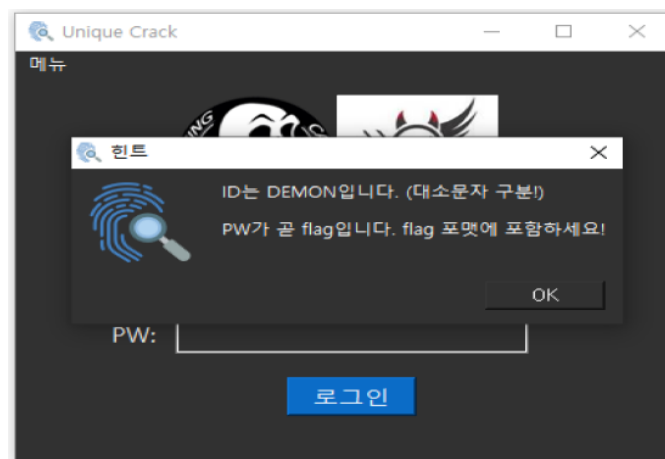


그림 2 Hint

메뉴 – 힌트를 선택하면 ID가 DEMON인 것과 PW가 곧 flag인 것을 확인할 수 있다.

0000005	C	pyi-	00...	00000026	C	Failed to get address for Py_SetPath\n
0000046	C	Failed to convert Wflag %s using mbstowcs (invalid m	00...	00000008	C	Py_GetPath
0000023	C	Failed to convert argv to wchar_t\n	00...	00000026	C	Failed to get address for Py_GetPath\n
0000027	C	Failed to convert progname to wchar_t\n	00...	00000012	C	Py_SetProgramName
0000025	C	Failed to convert pyhome to wchar_t\n	00...	0000002D	C	Failed to get address for Py_SetProgramName\n
0000019	C	%s%cbase_library.zip%c%s	00...	00000011	C	Py_SetPythonHome
0000031	C	sys.path (based on %s) exceeds buffer[%d] space\n	00...	0000002C	C	Failed to get address for Py_SetPythonHome\n
0000025	C	Failed to convert pypath to wchar_t\n	00...	00000015	C	PyDict_GetItemString
0000023	C	Error detected starting Python VM.	00...	00000030	C	Failed to get address for PyDict_GetItemString\n
0000007	C	strict	00...	0000000C	C	PyErr_Clear
0000006	C	utf-8	00...	00000027	C	Failed to get address for PyErr_Clear\n
0000025	C	Failed to get _MEIPASS as PyObject.\n	00...	0000000F	C	PyErr_Occurred
0000009	C	_MEIPASS	00...	0000002A	C	Failed to get address for PyErr_Occurred\n
0000008	C	marshal	00...	0000000C	C	PyErr_Print
0000006	C	loads	00...	00000027	C	Failed to get address for PyErr_Print\n
0000011	C	mod is NULL - %s	00...	0000000C	C	PyErr_Fetch
0000007	C	%U?%zu	00...	00000027	C	Failed to get address for PyErr_Fetch\n
0000005	C	path	00...	0000000E	C	PyErr_Restore
0000028	C	Installing PYZ: Could not get sys.path\n	00...	00000029	C	Failed to get address for PyErr_Restore\n
000001E	C	Failed to append to sys.path\n	00...	00000013	C	PyImport_AddModule
000003C	C	LOADER: Failed to convert runtime-tmpdir to a wide s	00...	0000002E	C	Failed to get address for PyImport_AddModule\n
0000047	C	LOADER: Failed to expand environment variables in th	00...	00000018	C	PyImport_ExecCodeModule
0000043	C	LOADER: Failed to obtain the absolute path of the run	00...	00000033	C	Failed to get address for PyImport_ExecCodeMo
0000035	C	LOADER: Failed to set the TMP environment variable.\n	00...	00000016	C	PyImport_ImportModule
0000013	C	pyi-runtime-tmpdir	00...	00000031	C	Failed to get address for PyImport_ImportModul
0000034	C	INTERNAL ERROR: cannot create temporary directory!	00...	0000000E	C	PyList_Append
0000031	C	WARNING: file already exists but should not: %s\n	00...	00000029	C	Failed to get address for PyList_Append\n
000001F	C	Error creating child process!\n	00...	0000000B	C	PyList_New
000000F	C	CreateProcessW	00...	00000026	C	Failed to get address for PyList_New\n
000001E	C	No error messages generated.\n	00...	0000000E	C	PyLong_AsLong
000000F	C	FormatMessageW	00...	00000029	C	Failed to get address for PyLong_AsLong\n
0000024	C	PyInstaller: FormatMessageW failed.	00...	00000011	C	PyModule_GetDict
000002D	C	PyInstaller: pyi_win32_utils_to_utf8 failed.	00...	0000002C	C	Failed to get address for PyModule_GetDict\n
000002D	C	PyInstaller: pyi_win32_utils_to_utf8 failed.	00...	00000016	C	PyObject_CallFunction

그림 3 IDA check

IDA로 분석하면 해당 프로그램은 Python으로 작성된 것을 확인할 수 있으며 [그림 3]과 같이 PyInstaller 모듈로 제작된 것을 확인할 수 있다.

추가적으로 해당 암호화 과정을 확인할 수 없는 것을 IDA 또는 다른 도구를 활용하여 분석하면 알 수 있게 된다. 즉, python decompile을 통해 python 파일을 확인해서 해결해야 한다.

그러면 python decompile 혹은 GUI 관련 PyQt Decompile 등 검색을 통해 도구를 찾을 수 있게 되는데 해당 보고서에서는 python-decompile3라는 도구를 사용한다. 이유는 아래에서 추가로 설명한다.

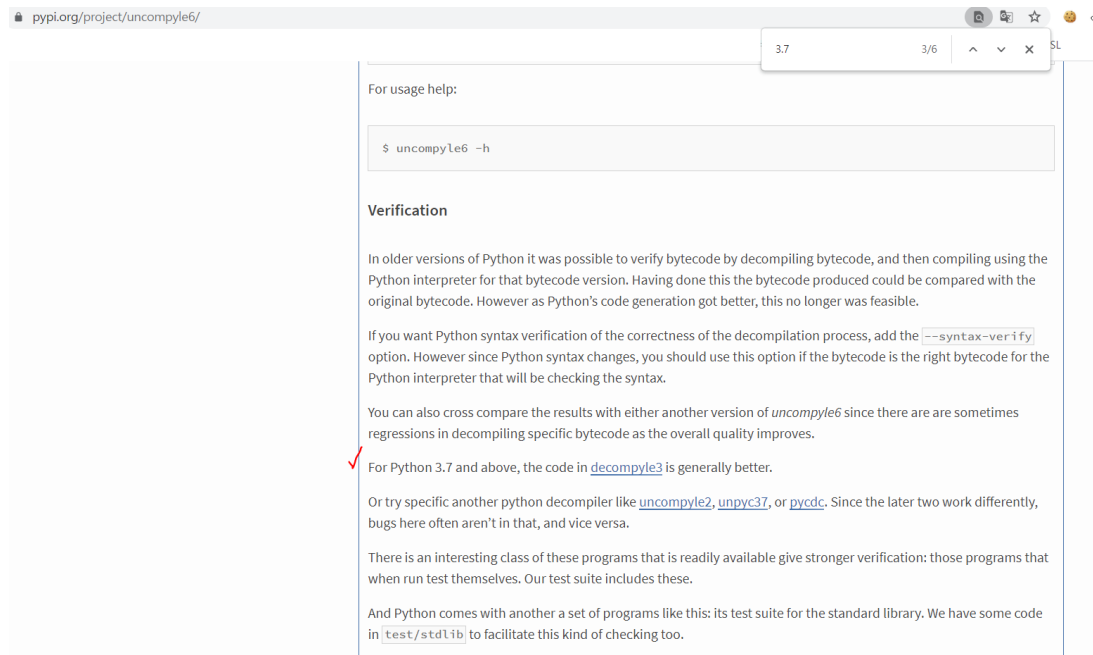


그림 4 python-decompile3 선택

python decompile 관련 모듈 중 대표적으로 uncompyle6가 있다. uncompyle6 공식 문서에 가보면 현재 업데이트가 되지 않아서 python 3.7 이상의 경우 일반적으로 decompyle3 = python-decompile3가 더 좋다고 되어있다. pyc 추출할 때 3.7임을 확인할 수 있으므로 python-decompile3를 사용하여 복원한다.

II-2. 환경 준비 및 풀이

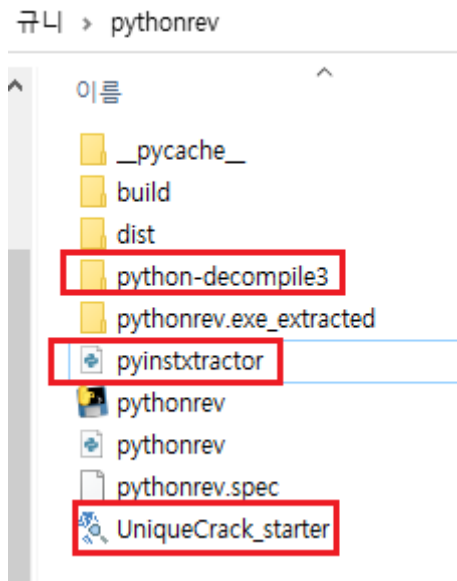


그림 5 파이썬 디컴파일을 위한 준비

■ 환경 준비

1. Python (version 3.7)
2. python-decompile3 //decompile3
3. pyinstxtractor.py // pyc 추출
4. UniqueCrack_starter.exe

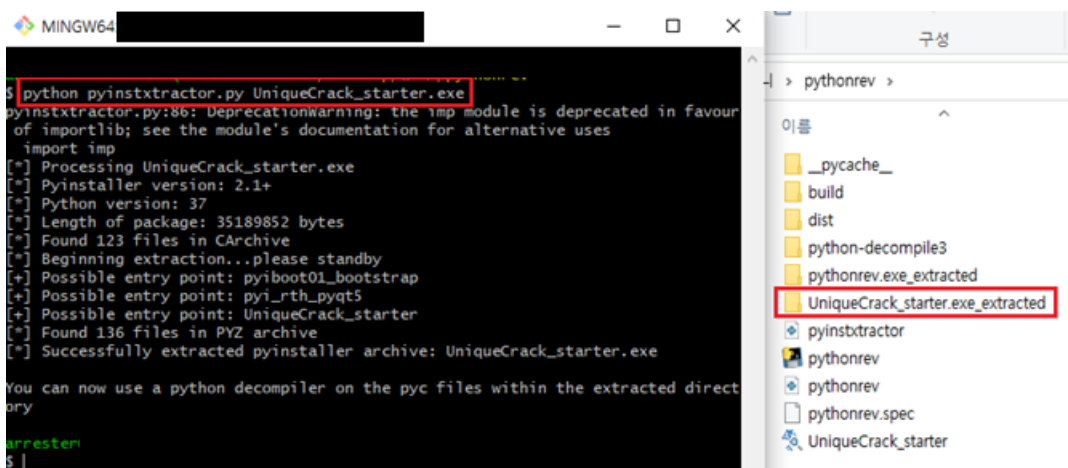


그림 6 pyinstxtractor.py

pyinstxtractor.py를 사용하면 exe file안에서 .pyc 파일들을 수집할 수 있다.

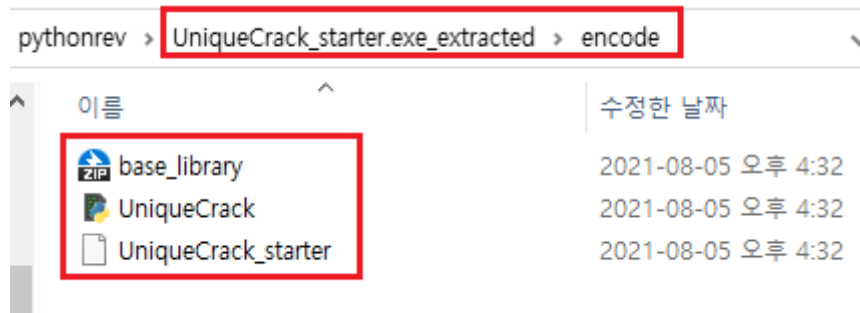


그림 7 patch

.pyc 파일을 이용하여 py 파일로 변경해야 한다.

우선 안에 포함된 파일들 중 기본 이름인데 확장자가 없는 파일이 있다. UniqueCrack_starter이다.

해당 파일과 PYZ-00.pyz_extracted라는 디렉토리 안에 UniqueCrack.py가 있다. 그리고 기본 라이브러리 파일인 base_library.zip을 가져와서 encode라는 디렉토리를 생성하여 필요한 파일들을 보기 편하게 정리한다.

위 파일들을 가져온 이유는 아래와 같다.

1. base_library.zip // 기본 라이브러리들이 pyc 파일로 되어 있는데 헤더를 가져오기 위함
2. UniqueCrack_starter // 해당 exe 파일의 pyc 파일로 디컴파일 되면서 헤더가 지워진 상태임
3. UniqueCrack // 이 파일은 원래 풀이 순서로는 2 번까지 해서 복원하고 안에 import 내용에 이 파일이 포함된 것을 보고 가져와야 하는 것인데 풀이 보고서에서는 중복되는 복원 과정을 한 번에 보여주기 위함으로 가져온 것

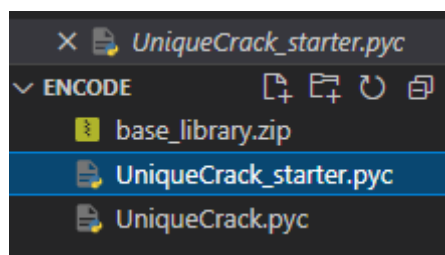


그림 8 UniqueCrack_starter -> UniqueCrack_starter.pyc

확장자를 .pyc로 변경한다.

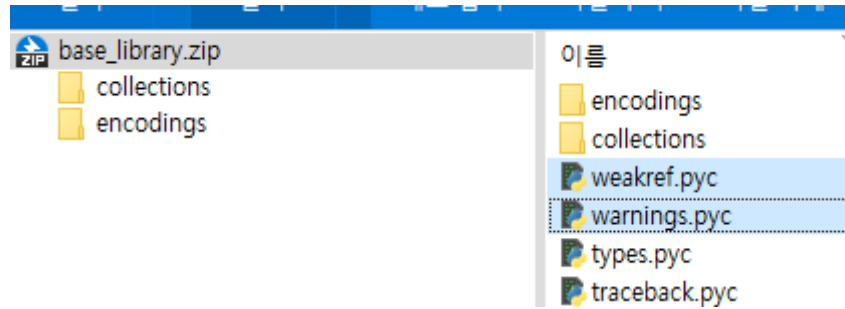


그림 9 base_library.zip에서 비교할 2개 임의 pyc 파일 가져오기

[그림 9]와 같이 헤더 값을 위해 임의 pyc 파일 2개를 가져온다.

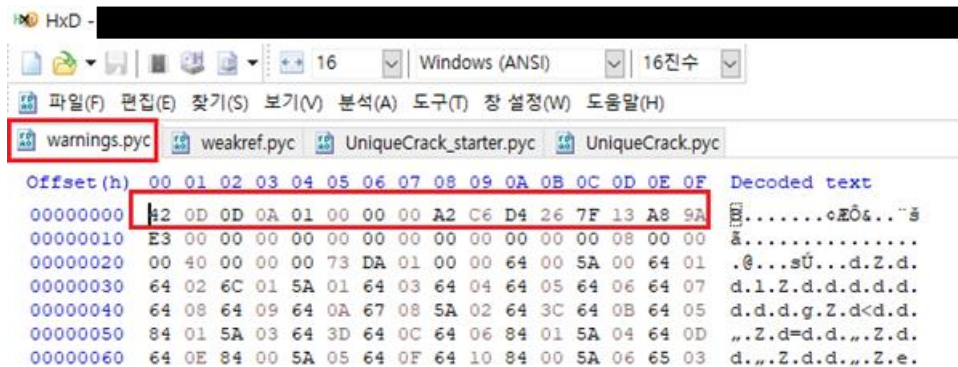


그림 10 header

warnings.pyc 위 헤더 값을 보고 weakref.pyc의 헤더 값도 본다. 보면 중간부터 값이 다름을 확인할 수 있다.

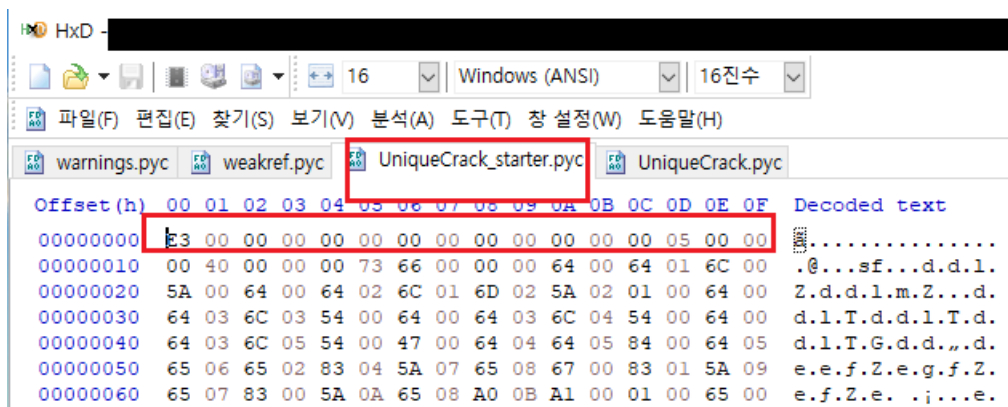
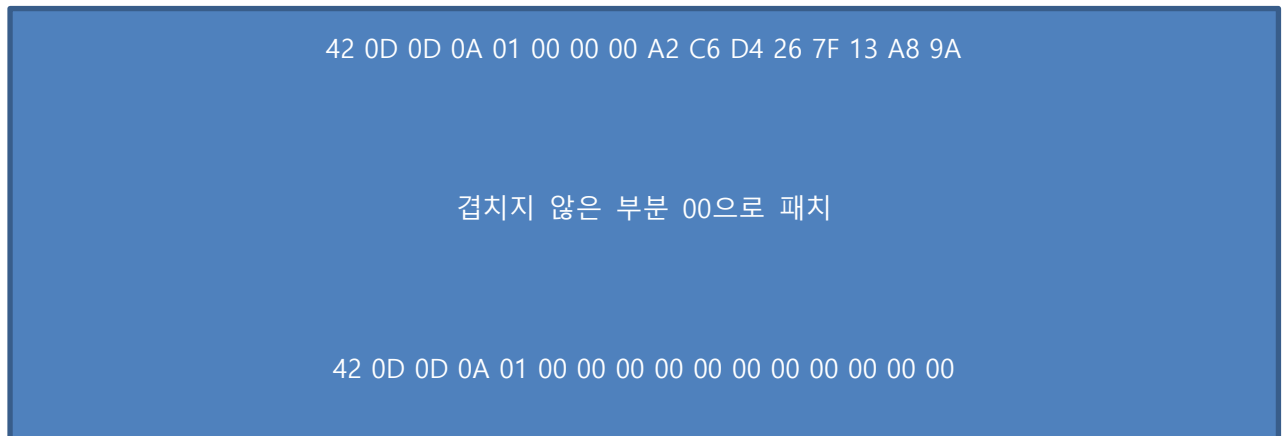


그림 11 UniqueCrack_starter.pyc 헤더

매직 넘버인 헤더 값이 다름을 확인할 수 있는데 [그림 10]과 [그림 11]을 보면 [그림 11]의 현재 헤

더 값은 [그림 10]의 2번째 줄이다. 즉, Pyinstaller로 인해서 [그림 11]과 같이 헤더 값이 지워진 상태다. 그러면 이 값만 패치하면 해결할 수 있다. 패치 기준은 [그림 10]에서 확인한 것처럼 겹치지 않은 부분은 00으로 수정하여 패치하면 된다.



위와 같이 말이다.

기본 정상적인 pyc 파일 2개에서 비교한 값 중 겹치지 않은 부분만 00으로 패치한 값으로 UniqueCrack_starter.pyc에 추가하면 된다.

여기까지 하고 decompyle3(python-decompile3)를 이용하여 복원하면 UniqueCrack_starter.py 파일을 복원할 수 있다. 그리고 소스 코드를 보면 UniqueCrack.py를 import하는 것을 볼 수 있는데 PYZ에서 해당 UniqueCrack.pyc를 가져와서 위와 같이 복원하면 내부 암호화 과정 코드까지 확인할 수 있다. 본 풀이 보고서에서는 중복 과정 자체를 한 번에 보여주기 위해 이어서 진행한다.

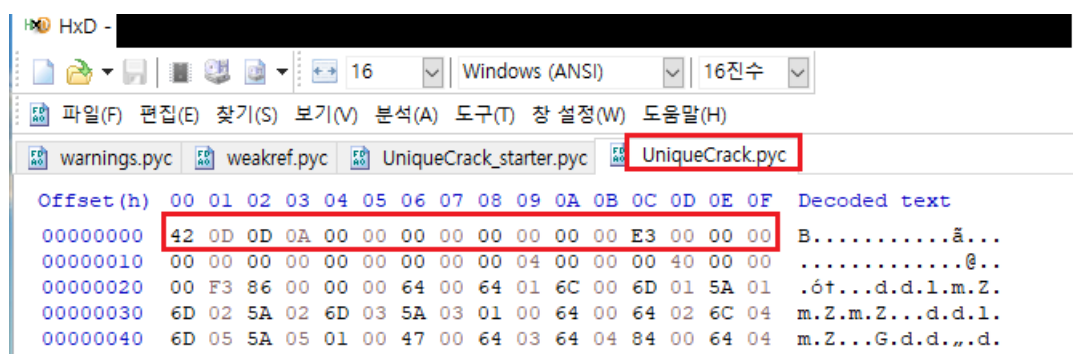


그림 12 UniqueCrack.pyc 헤더

UniqueCrack.pyc의 경우 4byte만 지워진 것을 알 수 있는데 4byte 만큼만 채우면 된다.





	warnings.pyc		weakref.pyc		UniqueCrack_starter.pyc		UniqueCrack.pyc										
Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	42	0D	0D	0A	01	00	00	00	00	00	00	00	00	00	00	00	B.....
00000010	E3	00	00	00	00	00	00	00	00	00	00	00	00	05	00	00
00000020	00	40	00	00	00	73	66	00	00	00	64	00	64	01	6C	00	.@...sf...d.d.l.
00000030	5A	00	64	00	64	02	6C	01	6D	02	5A	02	01	00	64	00	Z.d.d.l.m.Z...d.
00000040	64	03	6C	03	54	00	64	00	64	03	6C	04	54	00	64	00	d.l.T.d.d.l.T.d.
00000050	64	03	6C	05	54	00	47	00	64	04	64	05	84	00	64	05	d.l.T.G.d.d.,.d.
00000060	65	06	65	02	83	04	5A	07	65	08	67	00	83	01	5A	09	e.e.f.Z.e.g.f.Z.

그림 13 UniqueCrack_starter.pyc header patch

먼저 UniqueCrack_starter.pyc 파일을 [그림 13]과 같이 수정하면 된다.

FD RO	warnings.pyc	FD RO	weakref.pyc	FD RO	UniqueCrack_starter.pyc	FD RO	UniqueCrack.pyc										
Offset (h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	42	0D	0D	0A	00	00	00	00	00	00	00	00	00	00	00	00	B.....
00000010	E3	00	00	00	00	00	00	00	00	00	00	00	00	04	00	00
00000020	00	40	00	00	00	F3	86	00	00	00	64	00	64	01	6C	00	.@...ó†...d.d.l.
00000030	6D	01	5A	01	6D	02	5A	02	6D	03	5A	03	01	00	64	00	m.Z.m.Z.m.Z...d.
00000040	64	02	6C	04	6D	05	5A	05	01	00	47	00	64	03	64	04	d.l.m.Z...G.d.d.

그림 14 UniqueCrack.pyc patch

UniqueCrack.pyc도 [그림 12]에서 본 것과 같이 4byte만 패치한다.

```

arrester
e_extracted/encode
$ decompile3 UniqueCrack_starter.pyc > UniqueCrack_starter.py

arrester
e_extracted/encode
$ decompile3 UniqueCrack.pyc > UniqueCrack.py

arrester
e_extracted/encode
$ |

```

그림 15 decompile

python-decompile3를 정상적으로 설치했다면 decompile3 도구를 사용할 수 있다.

이것을 이용하여 각 pyc 파일을 py파일로 디컴파일한다.

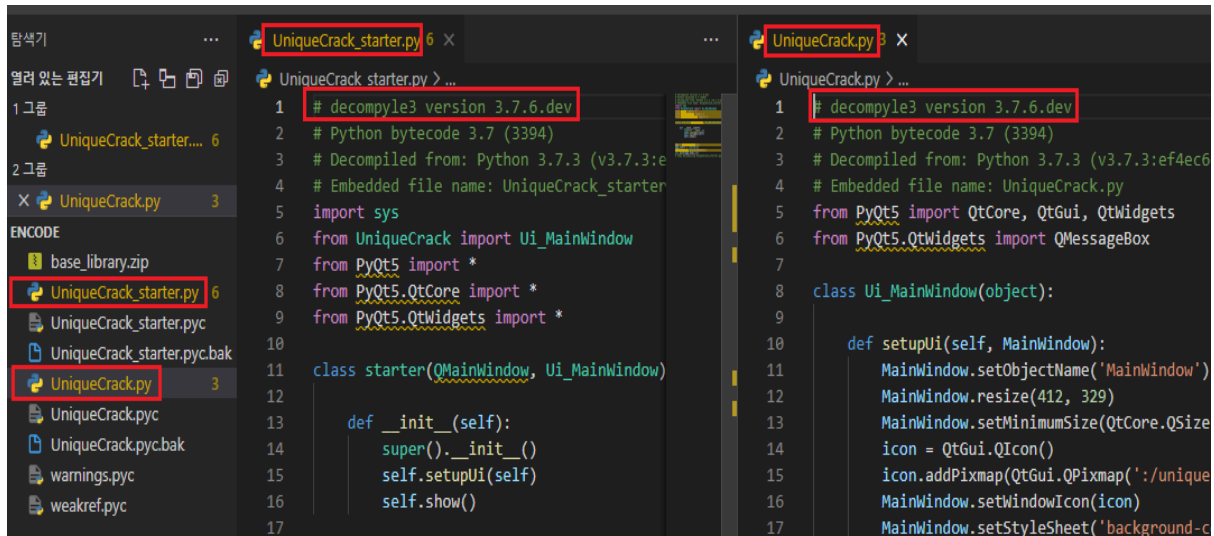


그림 16 파일 내용 확인

디컴파일된 UniqueCrack_starter.py와 UniqueCrack.py 파일 내용을 확인하면 python code를 그대로 볼 수 있다.

위에서 base_library 등 파일 정리할 때 각 파일을 가져온 이유에서 설명한 것과 동일하게 [그림 16]을 보면 UniqueCrack_starter.py에서 from Unique import Ui_MainWindow를 확인할 수 있기 때문에 풀이자 입장으로 대회 당시에서는 UniqueCrack_starter.py를 복원 후 내용을 확인하고 다시 같은 과정으로 UniqueCrack.py도 복원하게 되는 것이다. 풀이 보고서에서는 이 중복 과정을 한 번에 보여준 점을 참고하면 된다.

```

def login(self):
    main_id_value = [
        'D', 'E', 'M', 'O', 'N']
    main_pw_value = []
    main_pw_value.append(chr(ord(main_id_value[0]) - 23))
    main_pw_value.append(chr(ord(main_id_value[1]) - 24))
    main_pw_value.append(chr(ord(main_id_value[2]) + 27))
    main_pw_value.append(chr(ord(main_id_value[3]) + 18))
    main_pw_value.append(chr(ord(main_id_value[4]) + 21))
    main_pw_value.append(chr(ord('k')))
    id_value = []
    pw_value = []
    id_value = self.ID_lineEdit.text()
    pw_value = list(self.PW_lineEdit.text())
    if id_value == list('') or pw_value == list(''):
        QMessageBox.about(self, '💎:💎', '💎💎💎💎💎💎💎💎💎💎')
    else:
        try:
            if id_value == 'DEMON' and main_pw_value[0] == pw_value[0]:
                QMessageBox.about(self, 'flag', 'Login Success')
            else:

```

그림 17 finish

UniqueCrack.py 내용에서 PW 정보를 확인할 수 있다.

flag: HCAMP{--hack}

2021 하계 해킹캠프 CTF

Ⅲ 결론

Python으로 제작된 실행 파일을 분석하여 어떤 정보들로 제작된 건지 확인 및 디컴파일 과정을 학습할 수 있는 기회가 되는 것이 해당 문제의 출제 의도였습니다.

by. Demon Team arrester