Write up

팀: I want money

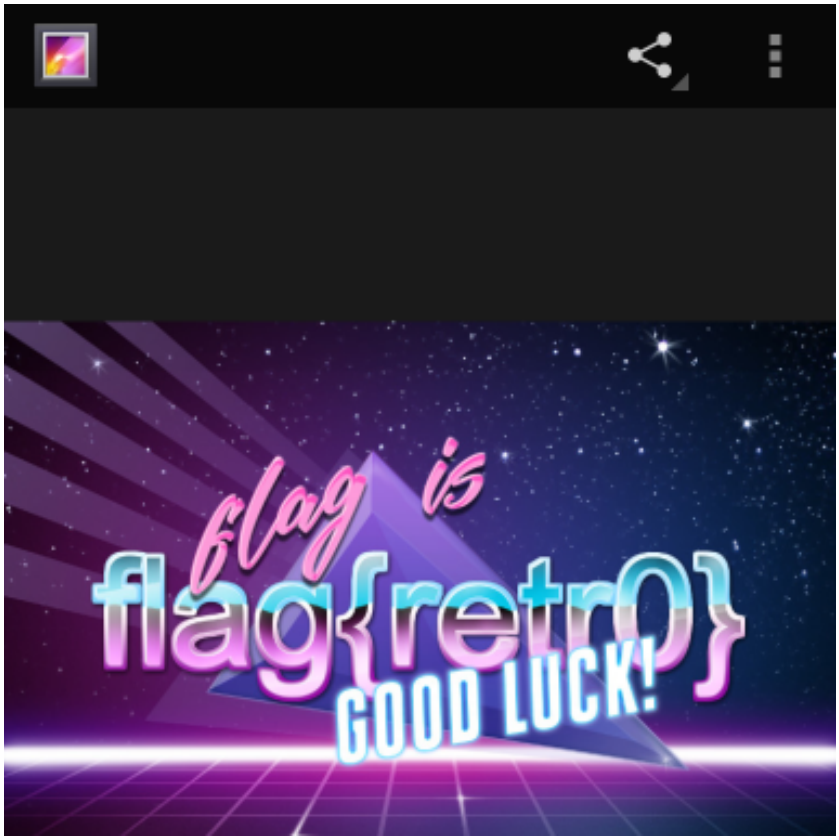
이름: 서희원
아이디: Hacods

1. Mobile dump - 3

```
[ Mobile dump - 3 ]
https://drive.google.com/open?id=1I-TDn_gojX8vDbh-FrPUpO1aU6aPnKIm
```

파일을 다운로드 해보니 Android.7z 이라는 파일이 나오고 압축을 풀어보면 android.img 라는 안드로이드 덤프 파일이 나온다. 이 파일을 원래는 프로그램을 이용해 마운팅 해야하지만 7zip 이라는 프로그램을 이용하면 압축을 해재하면 여러개의 폴더가 나온다.

이 여러개의 폴더에서 system > recent_images 라는 파일에 flag가 있다.



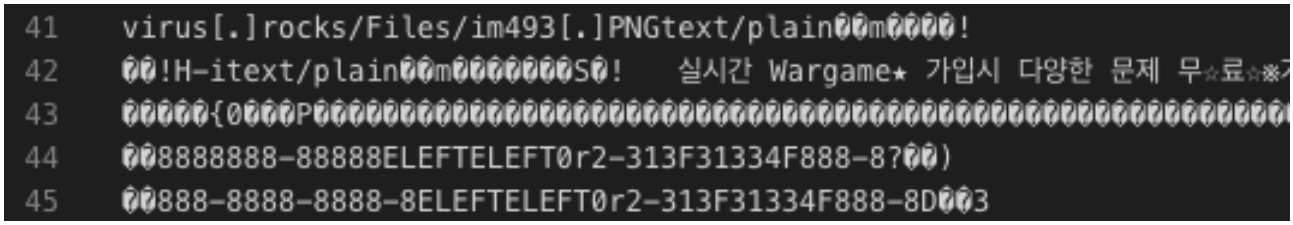
Flag: flag{retr0}

2. Mobile dump - 4

```
[ Mobile dump - 4 ]
https://drive.google.com/open?id=1I-TDn_gojX8vDbh-FrPUpO1aU6aPnKIm
```

파일을 다운로드 해보니 Android.7z 이라는 파일이 나오는데 이 파일은 위에 파일과 같다.

여기에서 data > com.android.messaging > dataases > bugle_db 라는 파일에 들어가면 문자 내역이 있다. 여기에 내용에 virus[.]rocks/Files/im493[.]PNG 라는 사이트가 있고 **https://virus.rocks/Files/im493.PNG**에 접속하여 사진을 다운로드한다.



hex로 열어보면 스테가노그래피로 flag가 숨겨져 있다.

Flag: flag{Mr_robots_net_hunter}

이름: 배상혁
아이디: Devleo

Mobile dump – 2

* 해당 문제의 Flag 포맷은 "flag{}" 가 아닙니다. *
https://drive.google.com/open?id=1I-TDn_gojX8vDbh-FrPUpO1aU6aPnKIm

---

Android.7z 파일을 7-zip 으로 압축을 풀고 ../misc/wifi/networkHistory 를 실행하면 플래그가 나온다.

```
&CONFIG:  "FL49{N0_C3llul4r_d4t4}"NONE
  SSID:  "FL49{N0_C3llul4r_d4t4}"
```

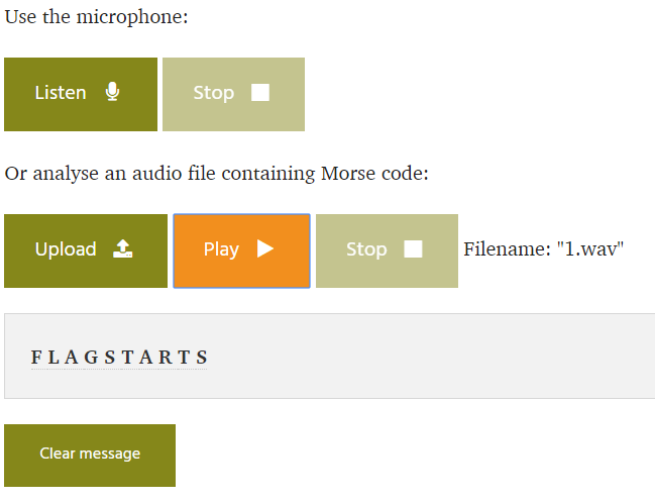
`FL49{NO_C3llul4r_d4t4}`

MagicBox

1, 2, 3, 4 각 파일에 숨겨진 flag 를 조합하자
https://drive.google.com/open?id=1W_MP0Wn7oDfA0RJ6h8sC9mJ3W97KpQ1r

---

1.wav 파일은 모스 부호이다. https://morsecode.scphillips.com/labs/audio-decoder-adaptive/ 이곳에서 모스 부호를 해독하면 `F L A G S T A R T`



2.png 파일을 분석해보면 hex 코드가 바이트 단위로 뒤집어져 있는 것을 볼 수 있다.

```cs
f=open('2.png','rb')
data = f.read()
rev_data = data[::-1]
e=open('test2.png','wb')
e.write(rev_data)
e.close()
```

간단한 python 코드로 다시 뒤집어주면 두번째 플래그 사진이 나온다. `_the_`

3.docx 파일은 확장자를 .zip으로 변경한 뒤 압축을 풀면 xml 형태로 파일을 볼 수 있는데, ../docProps/core.xml 에 들어가면

```cs
flag는 2019/10/31
```

라고 쓰여져있다. `2019/10/31`

4.PNG 파일은 초록색 부분을 검은색과 하얀색으로 그림판을 이용해 바꾸고 QR코드를 인증하면 된다. `_DSEC`

1 2 3 4 파일에서 나온 플래그 조각들을 합치면

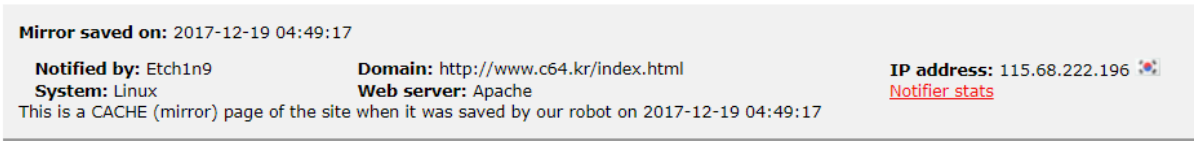
`FLAG{STARTS_the_2019/10/31_DSEC}`

해커가 해킹을 당했다!

출제자 본인이 예전에 운영하던 c64.kr 이라는 도메인을 가진 사이트가 실수로 인해 디페이스를 당한 적이 있다. 기억하고 싶진 않지만 언제 디페이스를 당했는지 찾아보자.
format: flag{yyyy-mm-dd hh:mm:ss}

디페이스 (Deface) 란? 해커가 홈페이지를 마음대로 바꾸고 해킹을 성공했음을 알리는 기법이다. (출처-보안뉴스)

디페이스 한 사이트를 공유하는 해커 커뮤니티인 zone-h.org 에 접속하여 오른쪽 검색창에 c64.kr 을 검색하면 해당 사이트 디페이스 이력이 나온다.



디페이스 당한 날짜와 시간이 플래그이므로

`flag{2017-12-19 04:49:17}`

이름 : 박현 , 닉네임 : 아이즈원

xorman

파일 을 열어보면 다음과 같은 문자열이 나온다.

```
c = 97C192CFCAD7DAFC92D0FCC790C0CAC797C7FCC1DAFC93CDC684D0FC93D4CDFC90C5C593D1D7
parallels@parallels-vm:~$
```

그래서 일정 범위 안으로 브루트포스를 돌려서 xor을 했다.

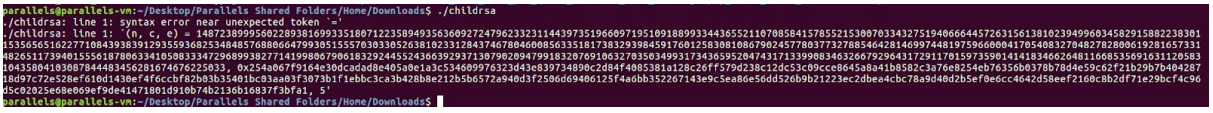
```
c = "97C192CFCAD7DAFC92D0FCC790C0CAC797C7FCC1DAFC93CDC684D0FC93D4CDFC90C5C593D1D7".decode('hex')
for x in range(255):
    text = ''

    for l in c:
        m = l.encode('hex')
        m = int(m,16)^x
        if 31<m and m<128:
            text += chr(m)

    if len(text)==len(c):
        print text
parallels@parallels-vm:~$
```

(스크립트)

```
parallels@parallels-vm:~$ python xor.py
7a2ojwz\2p\g0`jg7g\az\3mf$p\3tm\0ee3qw
6`3nkv{]3q]f1akf6f]`{]2lg%q]2ul]1dd2pv
5c0mhux^0r^e2bhe5e^cx^1od&r^1vo^2gg1su
4b1lity_1s_d3cid4d_by_0ne's_0wn_3ff0rt
```

flag{4b1lity_1s_d3cid4d_by_0ne's_0wn_3ff0rt}

childrsa



다음과 같이 n, c, e가 주어지는데, n이 매우 크고 e가 5정도로 작기때문에 5 제곱근 하여
평문을 구했다.

```
>>> from gmpy2 import *
>>> c =  0x254a067f9164e30dcadad8e405a0e1a3c534609976323d43e839734890c2d84f4085381a128c26ff579d238c12dc53c09cce8645a8a41b8582c3a76e8254eb76356b0378b78d4e59c62f21b29b7b40428718d97c72e528ef610d1430ef4f6ccbf82b03b35401bc03aa03f3073b1f1ebbc3ca3b428b8e212b5b6572a940d3f2506d69406125f4a6bb352267143e9c5ea86e56dd526b9b21223ec2dbea4cbc78a9d40d2b5ef0e6cc4642d58eef2160c8b2df71e29bcf4c96d5c02025e68e069ef9de41471801d910b74b2136b16837f3bfa1
>>> e = 5
>>> with local_context() as ctf:
...
KeyboardInterrupt
>>> with local_context() as ctx:
...     ctx.precision = 3000
...     m = iroot(c, 5)[0]
...     print("%x" , %int(m)).decode("hex")
  File "<stdin>", line 4
    print("%x", %int(m)).decode("hex")
                 ^
SyntaxError: invalid syntax
>>> with local_context() as ctx:
...     ctx.precision = 3000
...     m = iroot(c, 5)[0]
...     print("%x" %int(m)).decode("hex")
...
9r347_9r34t_br1lli4nt_l0w_3xp0n3nt_4tt4ck!
>>>
```

평문(플래그)가 나왔다.

Random_ROP

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  int v4; // [rsp+8h] [rbp-8h]
  int v5; // [rsp+Ch] [rbp-4h]

  v4 = 0;
  v5 = random_key(*(_QWORD *)&argc, argv, envp);
  set_buf();
  start_menu();
  printf("INPUT ID : ");
  read(0, log, 0xAuLL);
  printf("INPUT PW : ", log);
  read(0, &unk_6020DA, 0xAuLL);
  puts(&byte_400C60);
  puts(&byte_400C80);
  __isoc99_scanf(&unk_400CA1, &v4);
  getchar();
  if ( v5 == v4 )
  {
    printf(&byte_400CA4, log);
    vuln();
  }
  else
  {
    puts(&byte_400CBC);
    puts(&byte_400CD0);
  }
  return 0;
}
```

처음에 랜덤키(v5)를 생성하고, ID, PW를 입력 받는다.
그다음 v4를 입력받아서 v5 == v4면

vuln()를 실행 시킨다.

```
__int64 vuln()
{
  char v1; // [rsp+0h] [rbp-400h]

  puts(&byte_400C18);
  gets(&v1);
  write(1, "OK!\n", 4uLL);
  return 0LL;
}
```

gets() 함수에 취약점이 존재한다.
그럼 랜덤키를 생성하는 함수를 보자.

```
__int64 random_key()
{
  unsigned int v0; // eax

  v0 = time(0LL);
  srand(v0);
  return (unsigned int)rand();
}
```

srand(time(0))으로 시드를 생성한다.
하지만 libc가 주어졌기때문에 ctypes 모듈로 libc의 seed를 구해서 넣으면 통과할수 있다.

ASLR, NX만 존재하기 때문에 ROP하면 된다.

```
$ id
uid=1000(pwner) gid=1000(pwner) groups=1000(pwner)
$ cat flag
flag{ASHKDANLVAKLSHD}
$
```

플래그가 나왔다.