

HCTF: Hanyang Capture the Flag 2018

Beginners Division Write-Up

Hanyang Univ.

Computer Science and Engineering

18. 김영중

1. rev_equation

```

17 printf("Inut Serial) ", argv, envp);
18 fflush(_bss_start);
19 read(0, &buf, 0x14uLL);
20 for ( i = 0; i <= 0x13; ++i )
21     *((_BYTE *)&buf + (signed int)i) -= 48;
22 if ( (unsigned int)check_serial((__int64)&buf) )
23     puts("It is not flag");
24 else
25     puts("Good Job");
26 result = 0;
27 v4 = *MK_FP(__FS__, 40LL) ^ v10;
28 return result;
29 }

```

```

1 signed __int64 __fastcall check_serial(__int64 a1)
2 {
3     signed __int64 result; // rax@2
4     signed __int64 v2; // rtt@23
5     signed __int64 v3; // rtt@35
6     signed __int64 v4; // rtt@39
7
8     if ( *(_BYTE *)(a1 + 15) + *(_BYTE *)(a1 + 4) == 10 )
9     {
10         if ( *(_BYTE *)(a1 + 1) * *(_BYTE *)(a1 + 18) == 2 )
11         {
12             if ( *(_BYTE *)(a1 + 17) - *(_BYTE *)a1 == 4 )
13             {
14                 if ( *(_BYTE *)(a1 + 5) - *(_BYTE *)(a1 + 17) == -1 )
15                 {
16                     if ( *(_BYTE *)(a1 + 15) - *(_BYTE *)(a1 + 1) == 5 )
17                     {

```

```

eqn.py x
1 import angr
2
3 p = angr.Project('equation')
4
5 initial_state = p.factory.entry_state()
6 sm = p.factory.simulation_manager(initial_state)
7 sm.explore(find=0x400b65, avoid=[0x400b71])
8
9 print(sm.found[0].posix.dumps(0))
10

```

- 풀이 : check_serial 루틴 통과
- 방식 : Angr 이용 스크립트 작성

2. rev_DecompileMe

```
MainActivity.class ✕
package icewall.decompileme;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;

public class MainActivity
    extends AppCompatActivity
{
    String real_flag = "FLAG{Trust_this_String}";

    protected void onCreate(Bundle paramBundle)
    {
        super.onCreate(paramBundle);
        setContentView(2131296284);
    }
}
```

- 풀이 : dex2jar + JD-GUI 통해 주어진 apk 파일 분해

3. pwn_bf

```
16 int compare(char *pwd)
17 {
18     for(int i = 0; strlen(origin); i++)
19     {
20         sleep(1);
21         if(pwd[i] == origin[i])
22         {
23             continue;
24         }
25         else
26         {
27             printf("Wrong Password!\n");
28             return 0;
29         }
30     }
31     printf("Correct Password!\n");
32 }
33
```

```

1  from socket import *
2  import time
3
4  host = '54.180.60.212'
5  port = 1002
6
7  lib = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789_{}'
8  ans = ''
9
10 while True:
11     for l in lib:
12         s = socket(AF_INET, SOCK_STREAM)
13         s.connect((host, port))
14
15         time.sleep(0.1)
16         s.recv(1024)
17
18         s.send((ans + l).encode('utf-8'))
19
20         start = time.time()
21         res = s.recv(1024)
22         end = time.time()
23
24         if end - start > len(ans) + 1.5:
25             print(ans + l)
26             ans += l
27             break
28
29     if not res.startswith(b'Wrong'):
30         break
31

```

- 풀이 : compare 루틴 통과
- 방식 :
 - I. compare 내부적으로 1 초에 한번 비교를 시행함.
 - II. 한글자씩 대입하여 응답에 걸리는 시간을 비교
 - III. 응답 시간이 긴 문자열을 선택해 나가며 플래그 확인

4. rev_script

The screenshot displays the IDA Pro interface with the main function selected. The assembly code for the main function is as follows:

```

1 int __cdecl main(int argc, const char **argv, const char **envp)
2 {
3     char buf; // [sp+0h] [bp-410h]@1
4     __int64 v6; // [sp+408h] [bp-8h]@1
5
6     v6 = *MK_FP(__FS__, 40LL);
7     read(0, &buf, 0x400uLL);
8     A0(&buf);
9     return *MK_FP(__FS__, 40LL) ^ v6;
10 }

```

Below the main function, the A1101 routine is shown:

```

1 int __fastcall A1101(__int64 a1)
2 {
3     int result; // eax@1
4
5     result = *(a1 + 0x440);
6     if ( result == 10 )
7         result = puts("TANGJINJAM-TANGJINJAM-TANGJINJAM~!");
8     return result;
9 }

```

At the bottom, a Python script named script.py is shown, which reads the output of the program and processes it:

```

1 import re
2
3 with open('./script', 'rb') as f:
4     data = f.read()
5
6 res = ''
7 for i in range(len(data)):
8     if data[i:].startswith(b'\x0F\xB6\x00\x3c'):
9         res += chr(data[i+4])
10
11 res = ''.join(a[-1] if len(a) > 0 else '' for a in res.split('\n'))
12 print(res)
13

```

- 풀이 : A0 ~ A1101 을 통과할 1102 개 길이의 문자열 탐색
- 방식 :
 - I. 모든 A 루틴은 'mov eax, byte ptr [rax]; cmp al, ##'의 명령어를 가짐
 - II. 파일을 읽어 해당 instruction 표현에 대응되는 모든 ##을 찾아냄
 - III. 결과는 BTS 의 'Go Go' 가사이며, 마지막 글자들을 연결하면 플래그가 나옴.