level09

태그	Shell
상태	완료

풀이과정
- 겪었던 어려움
- 풀이과정
  - xxd
- 풀이과정(2)
- 정답

출처

풀이과정

겪었던 어려움

등차수열은 차갑다..

풀이과정

xxd

strings로 주어진 level09를 보니 아래와 같은 문구가 있었다.

```
You should not reverse this
LD_PRELOAD
Injection Linked lib detected exit..
/etc/ld.so.preload
/proc/self/maps
/proc/self/maps is unaccessible, probably a LD_PRELOAD attempt exit..
libc
You need to provied only one arg.
00000000 00:00 0
LD_PRELOAD detected through memory maps exit .
```

이해가 잘 안되서 옆에 있던 token을 읽어보니 nonprintable한 문자가 여럿 있었고 level09 파일을 실행시켜보니 `You need to provied only one arg.` 와 같은 에러메시지, 그래서 token을 인자로 실행시켜보니 `tpmhr` 와 같은 문자열이 나왔고 `./level09 aaaaaaaaaaaa...` 와 같이 실행시켜보니 `abcdefg...` 와 같은 문자열이 결과값으로 나왔다.

이와 같이 여러 입력을 통해 입력값 길이 == 출력값 길이, 맨 처음 입력값은 그대로이고 `(dest[0] = src[0] + 0, dest[1] = src[1] + 1, dest[2] = src[2] + 2, ...)` 와 같은 형태로 문자열 변화가 생긴다는 것을 알았다.

`xxd -c 1 token` 를 통해서 token 내부 값이 아래와 같다는 것을 알았다.

```
0000000: 66  f
0000001: 34  4
0000002: 6b  k
0000003: 6d  m
0000004: 6d  m
0000005: 36  6
0000006: 70  p
0000007: 7c  |
0000008: 3d  =
0000009: 82  .
000000a: 7f  .
000000b: 70  p
000000c: 82  .
000000d: 6e  n
000000e: 83  .
000000f: 82  .
0000010: 44  D
0000011: 42  B
0000012: 83  .
0000013: 44  D
0000014: 75  u
0000015: 7b  {
0000016: 7f  .
0000017: 8c  .
0000018: 89  .
0000019: 0a  .
```

이를 통해서 연산한 결과는 아래와 같다

```
> #       CODE    ASCII          CODE   ASCII
  ================================================
  0       102     f       >      102     f
  1       52      4       >      51      3
  2       107     k       >      105     i
  3       109     m       >      106     j
  4       109     m       >      105     i
  5       54      6       >      49      1
  6       112     p       >      106     j
  7       124     |       >      117     u
  8       61      =       >      53      5
  9       130     .       >      121     y
  10      127     .       >      117     u
  11      112     p       >      101     e
  12      130     .       >      118     v
  13      110     n       >      97      a
```

```
  14      131     .       >       117     u
  15      130     .       >       115     s
  16      68      D       >       52      4
  17      66      B       >       49      1
  18      131     .       >       113     q
  19      68      D       >       49      1
  20      117     u       >       97      a
  21      123     {       >       102     f
  22      127     .       >       105     i
  23      140     .       >       117     u
  24      137     .       >       113     q
```

이를 통해 결과값을 구할 수 있다. `f3iji1ju5yuevaus41q1afiuq`

풀이과정(2)

아래 c코드를 /tmp/에 작성후, cd /tmp/경로로 이동해서 gcc를 해준다.
그리고 cat token | xargs /tmp/<실행파일>로 토큰 탈취

```c
int main(int ac, char **av)
{
    char *str;
    int i = 0;
    str = av[1];

    while (*str)
    {
        printf("%c", *str - i)
        i++;
        str++;
    }
    printf("\n");
    return(0);
}
```

정답

```
level09@SnowCrash:~$ su flag09
Password:
Don't forget to launch getflag !
flag09@SnowCrash:~$ getflag
Check flag.Here is your token : s5cAJpM8ev6XHw998pRWG728z
flag09@SnowCrash:~$
```

`'s5cAJpM8ev6XHw998pRWG728z'`

출처

xxd 명령어로 hex dump출력하기 – 한윤석 개발 블로그

파일에서 hex dump를 출력하는 방법

https://hannut91.github.io/blogs/bash/xxd