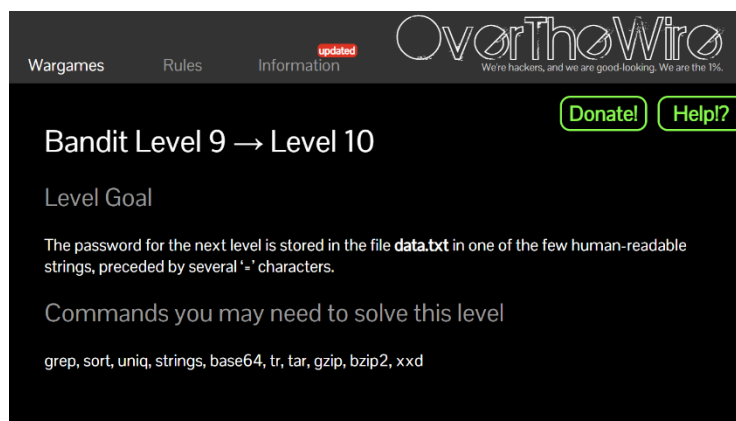


### 3회차 - 20. 케로로

## Bandit 레벨 9 ~ 12 라이트업

정보보호학부 2024111262 조현서

### Lv. 9 → 10



웹 사이트의 문제 설명을 보니, 다음 레벨의 비밀번호는 data.txt 파일에 저장되어 있으며, 몇 개의 사람이 읽을 수 있는 문자열 중 하나에 포함되어 있다고 한다. 또한 이 문자열은 여러 개의 = 문자로 시작된다고 한다.

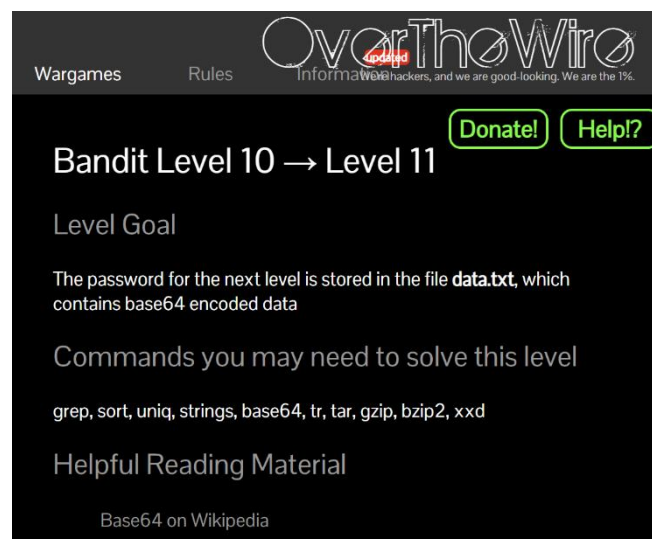
```
bandit9@bandit:~$ cat data.txt | grep "==="
grep: (standard input): binary file matches
bandit9@bandit:~$ cat data.txt | grep -a "==="
D=====
#####h#!#####JesseVozl7####POL%Y]eHoeae^oUovToeDe|e@T####eNe####8?gee}eebe}ee?
Q#egeeX===== theeAeeledHee^e)F1ee>)SeKee3ePZeetee&xs肉NB/e2eeÜBee eZ/eBjGe#####<eeee3e7ee<#####
eue/eed|
e-#####n
#eieiU=
ee7eene)eeU2ee5ee_bBKeeKee}x>}:ee4Rl_7gHeeD:ee27ee4eeeeCeeFey
Qeedqh0#####o0S####F 4Cqeyyzee#ee#61QW
ee6ee!ee8eezeBee$ee_e_GeepehqI.Xeeeb02eeeeH####SeTweemeo◆eeeo3emet0eepee~eLe3JprD===== passwordlee e
Le ~,ee<@eeEhe$eeeeQ5eeeeDeeke e|e3
~eTeeefee;eo9seeeP#te+Peee[?]
QqDfeee.8eeeeCzmnf&veeeL:eeFexeeeeKeeeebeMe
eCeeIeeeeBie>eeYe
eEkeee $enXeeT=~e)*4a2e?eeT0" 'e&eJe~FDV3===== isede5z(ee#e&seT!1e0e&peeeoqee
enRe eeeF
eeze|!e(eiefee+eeA6e4e+'eeFeeeT=eeb5eeAe)e
e#e)#9ee減eebeRe+ee~&eeOiu####VheeMe}^eeQp^eGee==e
eIeéT:ek#####A####Ue2eQceBee%#g+;YA_eokr####X53|ef8+e
]e.g:7ee#####npe##### e#####CDe`vooS0e-<e]e`e@#H Uum####BiAeej堵ee!O&#####D9===== F
GUW5iLLVJrxX9kMYMmLn4MgbpfMiqey
bandit9@bandit:~$ strings data.txt | grep "==="
}===== the
3JprD===== passwordl
~FDV3===== is
D9===== FGUW5iLLVJrxX9kMYMmLn4MgbpfMiqey
bandit9@bandit:~$
```

지금까지 했던 대로 cat 명령어를 사용하였다. Cat data.txt | grep "===" 명령어를 입력하여, data.txt 파일 안에 ===를 포함한 부분을 출력하게끔 하였다. 하지만, data.txt가 바이너리 파일이라 grep 명령어가 먹히지 않았다. 이를 보완하기 위해 grep의 -a 옵션을 사용하였다. 이는 바이너리 파일을 텍스트 파일처럼 처리할 수 있게 해준다. 다시 -a 옵션을 사용하여 진행해보니, 너무 더럽게 나와서 비밀번호가 무엇인지 한 눈에 알 수가 없었다. 그래서, 파일에 포함된 문자열을 출력하는 명령어 strings 를 사용하였다. Strings data.txt | grep "===" 를 통해, data.txt 파일 안에 = 문자열 뒤에 나오는 human-readable 텍스트만 출력하게끔 하였다.

출력된 비밀번호는 "FGUW5ilLVJrxX9kMYMmlN4MgbpfMiqey" 이다.

Lv.9 → 10 해결

Lv. 10 → 11



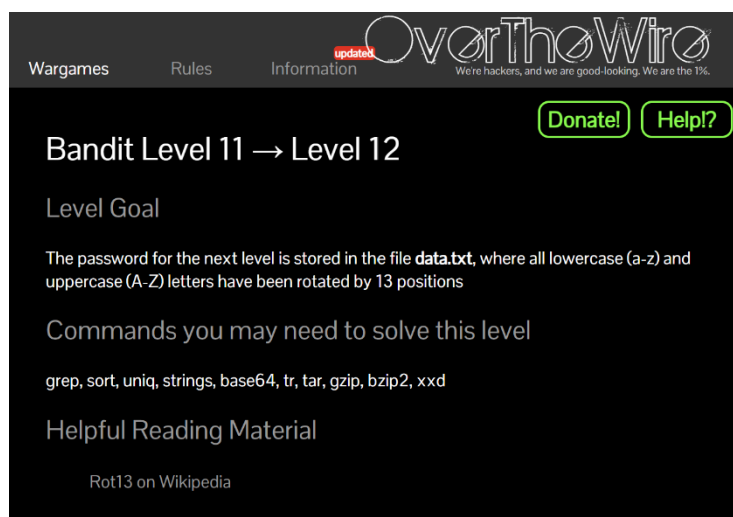
다음 레벨의 비밀번호는 data.txt 파일에 저장되어 있으며, 이 파일은 base64로 인코딩된 데이터를 포함하고 있다고 한다.

```
bandit10@bandit:~$ cat data.txt
VGhlIHBhc3N3b3JkIGlzIGR0UjE3M2ZaS2IwU1JzREZTR3NnMlJXbnBOVm9zcVJyCg==
bandit10@bandit:~$ base64 --decode data.txt
The password is dtR173fZKb0RRsDFSGsg2RWnpNVj3qRr
bandit10@bandit:~$
```

일단 cat 명령어를 사용해서, data.txt 파일을 열어보니, 인코딩되어있는 알 수 없는 문자열이 나오는 것을 확인할 수 있다. Base64 명령어를 사용하여 이를 디코딩하여 출력하게끔 해보려고 한다. Base64 -decode data.txt 명령어를 입력해보니, 아까 암호화되어 있던 문자열이 해독되어 출력되는 것을 확인할 수 있다.  
비밀번호는 "dtR173fZKb0RRsDFSGsg2RWnpNVj3qRr"이다.

Lv.10 → 11 해결

Lv. 11 → 12



다음 레벨의 비밀번호는 data.txt 파일에 저장되어 있으며, 이 파일 안의 모든 소문자와 대문자는 13글자씩 회전되어 있다고 한다. Rotated by 13 positions 이 무슨 뜻일까 찾아보니, ROT13을 뜻하는 것 같다. 이는 문자를 알파벳 뒤에 13번째 문자로 대체하는 간단한 문자 대체 암호를 뜻한다. 예를 들어, A는 N으로, B는 O로 이동한 것이다.

```
bandit11@bandit:~$ cat data.txt
Gur cnffjbeq vf 7k16JARUVv5LxVuJfsSVdbbtaHGlw9D4
bandit11@bandit:~$
```

## ROT13 인코더 및 디코더

ROT13 데이터를 입력하여 암호화 / 해독:

검색:  파일 선택

Gur cnffibeq vf 7k16JArUVv5LxVuJfsSVdbbtaHGlw9D4

ROT Operation:

결과

The password is 7x16WNeHli5YklhWsfFlqoognUTyj9Q4

Cat 명령어로 data.txt 파일을 불러오면, rot13으로 암호화된 것 같은 문자열이 출력된다. 이를 ROT13 디코딩 사이트에 입력하여, 해독해보니, The password is 7x16WNeHli5YklhWsfFlqoognUTyj9Q4 라는 문구였다. 다음 단계의 비밀번호는 “7x16WNeHli5YklhWsfFlqoognUTyj9Q4”이다.

Lv.11 → 12 해결

Lv. 12 → 13

Wargames
Rules
Information

OverTheWire

We're hackers, and we are good-looking. We are the 1%.

## Bandit Level 12 → Level 13

### Level Goal

The password for the next level is stored in the file **data.txt**, which is a hexdump of a file that has been repeatedly compressed. For this level it may be useful to create a directory under /tmp in which you can work. Use `mkdir` with a hard to guess directory name. Or better, use the command `"mktemp -d"`. Then copy the datafile using `cp`, and rename it using `mv` (read the manpages!)

### Commands you may need to solve this level

`grep`, `sort`, `uniq`, `strings`, `base64`, `tr`, `tar`, `gzip`, `bzip2`, `xxd`, `mkdir`, `cp`, `mv`, `file`

### Helpful Reading Material

Hex dump on Wikipedia

다음 레벨의 비밀번호는 data.txt 파일에 저장되어 있으며, 이 파일은 여러 번 압축된 파일의 hex스 덤프이다. 이 레벨에서는 /tmp 디렉토리 안에 작업할 수 있는 임시 디렉토리를 만드는 것이 유용할 수 있다고 한다. Mkdir 명령을 사용하여 추측하기 어려운 이름으로 디렉토리를 생성할 수 있다고 한다. 또는 mktemp -d 명령어를 사용하고, cp 명령을 사용하여 데이터 파일을 복사하고, mv 명령을 사용하여 파일 이름을 바꾸라고 한다.

```
bandit12@bandit:~$ cat data.txt
00000000: 1f8b 0808 8e0b bf63 0203 6461 7461 322e .....c..data2.
00000010: 6269 6e00 013c 02c3 fd42 5a68 3931 4159 bin..<...BZh91AY
00000020: 2653 598c b471 f700 0014 ffff fa59 c6c5 &SY..q.....Y..
00000030: af63 cfff af73 ffff bdb7 7c9f b1fb eafa .c...s....|.....
00000040: bfff fb9f f9fe bdbf ffeb ffef b001 3b2c .....;,,
00000050: 5900 0341 a064 007a 8003 40d0 6869 a068 Y..A.d.z..@.hi.h
00000060: 3464 007a 81a0 0680 3401 90d0 6800 00d1 4d.z....4...h...
00000070: a0c9 a680 f51e 9a83 27a4 3d4f 4991 0000 .....'.=OI...
00000080: 69a6 803d 4001 9001 a686 8c40 0d00 d3d2 i..=@.....@....
00000090: 0d00 0d06 08f5 0323 4034 069e a340 0da8 .....#@4...@..
000000a0: 3d46 83ca 0343 41a0 3400 e9a0 d07a 8680 =F...CA.4.....z..
000000b0: 3ca3 d47a 8068 0079 4006 8d0c 8034 d068 <..z.h.y@....4.h
000000c0: d03d 401a 0680 0d00 0683 407a 8680 6834 .=@.....@z..h4
000000d0: 0034 0003 ca64 00b9 6862 e5be 0fc5 ac97 .4...d..hb.....
000000e0: 996a 03e6 d176 bda4 7989 5466 5357 2377 .j...v...y.TfSW#w
```

Cat 명령어로 data.txt 파일을 출력해보니, dump 형식으로 되어있는 것을 확인할 수 있다.

```
bandit12@bandit:~$ mkdir /tmp/kshind
bandit12@bandit:~$ cp data.txt /tmp/kshind
bandit12@bandit:~$ cd /tmp/kshind
bandit12@bandit:/tmp/kshind$ ls
data.txt
```

```
bandit12@bandit:/tmp/kshind$ xxd -r data.txt data
bandit12@bandit:/tmp/kshind$ ls
data data.txt
bandit12@bandit:/tmp/kshind$ file data
data: gzip compressed data, was "data2.bin", last modified: Wed Jan 11 19:18:38
2023, max compression, from Unix, original size modulo 2^32 572
```

```
bandit12@bandit:/tmp/kshind$ mv data data.gz
bandit12@bandit:/tmp/kshind$ ls
data.gz data.txt
bandit12@bandit:/tmp/kshind$ gzip -d data.gz
```

문제 설명에 따라 /tmp 디렉토리를 만들고 data.txt를 cp를 이용하여 복사해서 이동을 시켰다. 그리고, 해당 디렉토리에 xxd -r을 이용해서 data라는 파일로 새로 만들고, file 명

명령어를 이용해서 확인해보니 gzip으로 압축된 파일임을 알 수 있었다. Gzip 명령어를 통해 gzip으로 압축된 파일을 풀 수 있는데 이것 하기 위해선 이름이 .gz로 끝나야 되서 mv 명령어를 통해 이름을 변경하고, 압축을 풀었다.

```
bandit12@bandit:/tmp/kshind$ file data
data: bzip2 compressed data, block size = 900k
```

```
bandit12@bandit:/tmp/kshind$ mv data data.bz2
bandit12@bandit:/tmp/kshind$ ls
data.bz2  data.txt
bandit12@bandit:/tmp/kshind$ bzip2 -d data.bz2
bandit12@bandit:/tmp/kshind$ ls
data  data.txt
bandit12@bandit:/tmp/kshind$ file data
data: gzip compressed data, was "data4.bin", last modified: Wed Jan 11 19:18:38 2023, max compression, from Unix, original size modulo 2^32 20480
```

```
bandit12@bandit:/tmp/kshind$ ls
data.gz  data.txt
bandit12@bandit:/tmp/kshind$ gzip -d data.gz
bandit12@bandit:/tmp/kshind$ file data
data: POSIX tar archive (GNU)
```

```
bandit12@bandit:/tmp/kshind$ tar -xvf data
data5.bin
bandit12@bandit:/tmp/kshind$ ls
data  data5.bin  data.txt
```

```
bandit12@bandit:/tmp/kshind$ file data8.bin
data8.bin: ASCII text
```

```
bandit12@bandit:/tmp/kshind$ cat data8.bin
The password is wbWdlBxEir4CaE8LaPhauuOo6pwRmrDw
```

압축을 풀고 나니, 이번엔 bzip2로 압축이 됐다고 하니까 한 번 더 압축을 풀었다. Bzip2로 압축 해제하기 위해, data.bz2로 이름을 바꾸고 또 다시 압축을 풀었다. 그리고 또 확인해보니, 다시 gzip으로 압축되어 있다고 한다. 이를 또 압축해제 하니, 이번에는 tar 아카이브로 압축되어 있다고 한다. 이를 풀기 위해 tar -vxf 명령어를 사용해서 풀어 주었다. 이런 형식으로 계속되는 압축들을 풀어주다가 보니, 이제 ASCII text 형식의 파일이 드디어 나오게 된다. 이를 cat으로 출력해보니, 비밀번호를 얻어낼 수 있었다.

다음 단계 비밀번호는 "**wbWdlBxEir4CaE8LaPhauuOo6pwRmrDw**"이다

**Lv.12 → 13 해결**