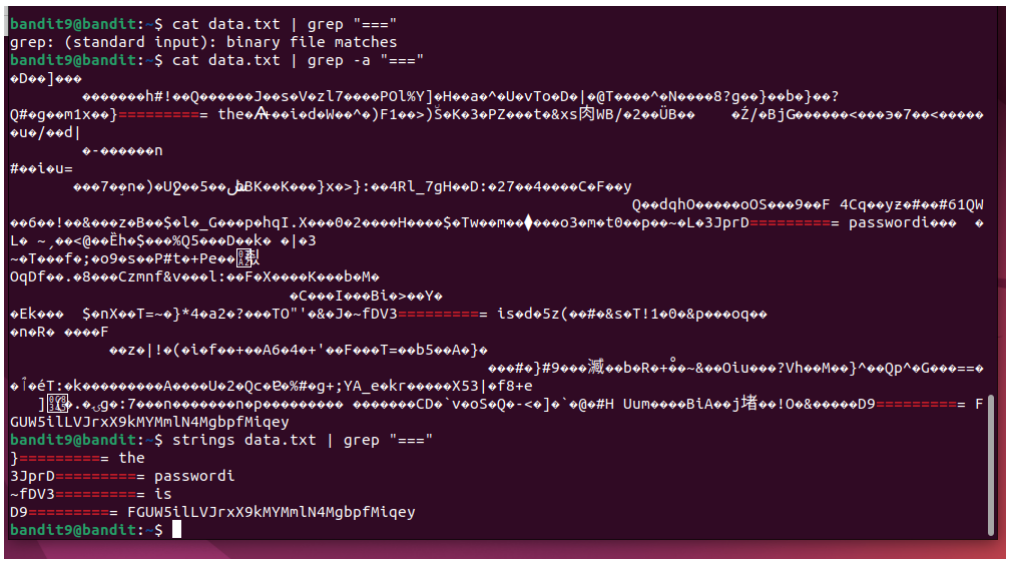
3회차 - 20. 케로로

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.

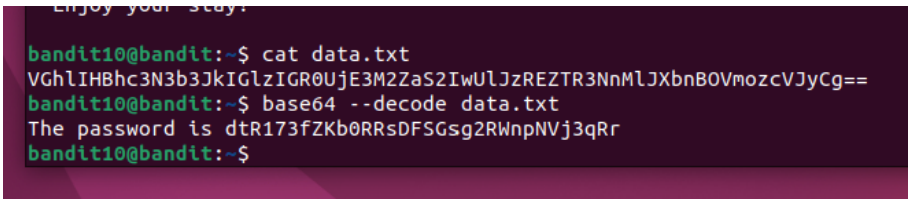


지금까지 했던 대로 cat 명령어를 사용하였다. Cat data.txt | grep "===" 명령어를 입력하여, data.txt 파일 안에 ===를 포함한 부분을 출력하게끔 하였다. 하지만, data.txt가 바이너리 파일이라 grep 명령어가 먹히지 않았다. 이를 보완하기 위해 grep의 -a 옵션을 사용하였다. 이는 바이너리 파일을 텍스트 파일처럼 처리할 수 있게 해준다. 다시 -a 옵션을 사용하여 진행해보니, 너무 더럽게 나와서 비밀번호가 무엇인지 한 눈에 알 수가 없었다. 그래서, 파일에 포함된 문자열을 출력하는 명령어 strings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

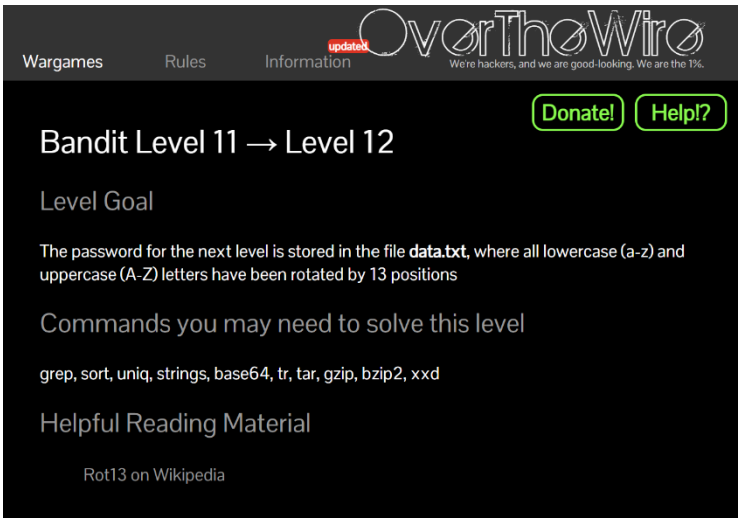
다음 레벨의 비밀번호는 data,txt 파일에 저장되어 있으며, 이 파일은 base64로 인코딩된 데이터를 포함하고 있다고 한다.



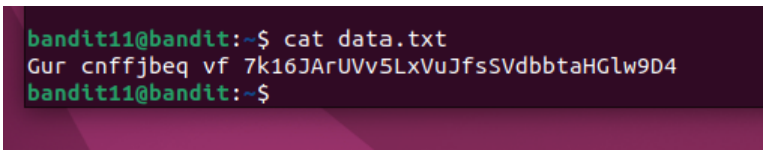
일단 cat 명령어를 사용해서, data.txt 파일을 열어보니, 인코딩되어있는 알 수 없는 문자열이 나오는 것을 확인할 수 있다. Base64 명령어를 사용하여 이를 디코딩하여 출력하게끔 해보려고 한다. Base64 –decode data.txt 명령어를 입력해보니, 아까 암호화되어 있던 문자열이 해독되어 출력되는 것을 확인할 수 있다.
비밀번호는 "**dtR173fZKb0RRsDFSGsg2RWnpNVj3qRr**"이다.

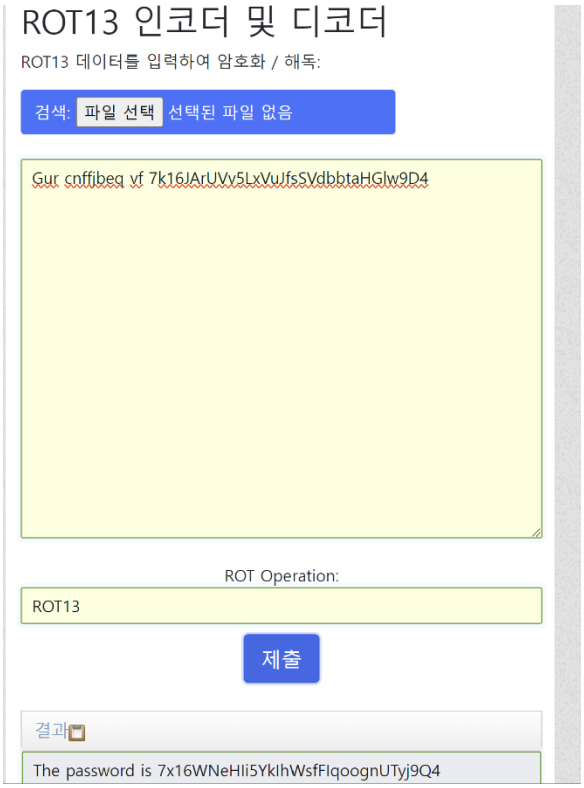
Lv.10 → 11 해결

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.





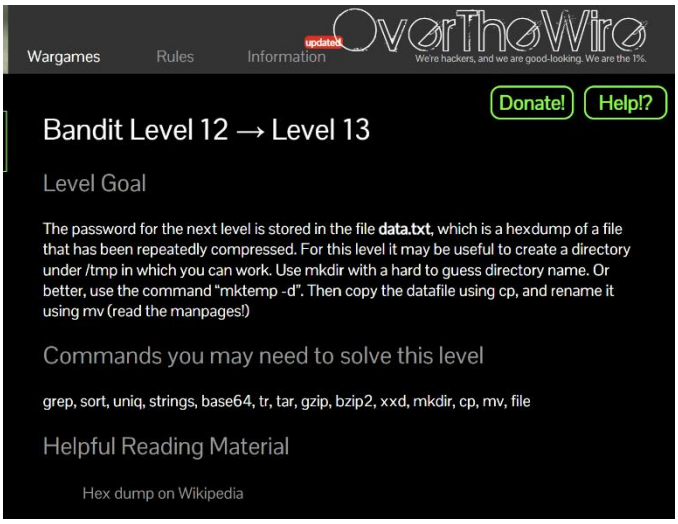
Cat 명령어로 data.txt 파일을 불러오면, rot13으로 암호화된 것 같은 문자열이 출력된다. 이를 ROT13 디코딩 사이트에 입력하여, 해독해보니,

The password is 7x16WNeHIi5YkIhWsfFIqoognUTyj9Q4 라는 문구였다.

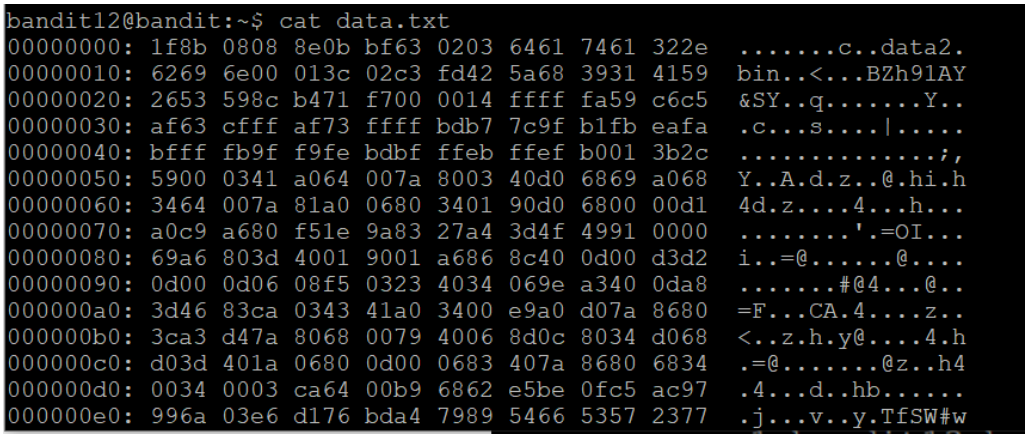
다음 단계의 비밀번호는 "**7x16WNeHIi5YkIhWsfFIqoognUTyj9Q4**"이다.

Lv.11 → 12 해결

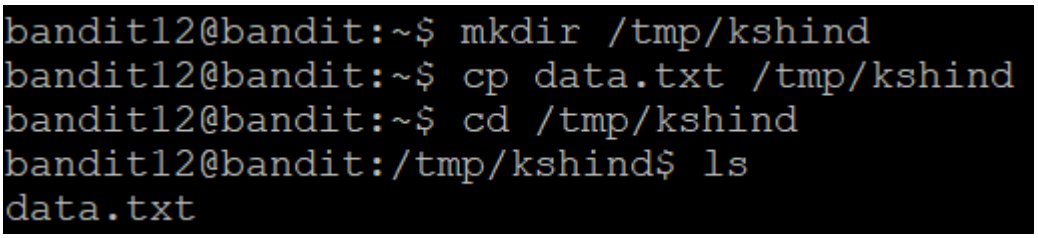
Lv. 12 → 13

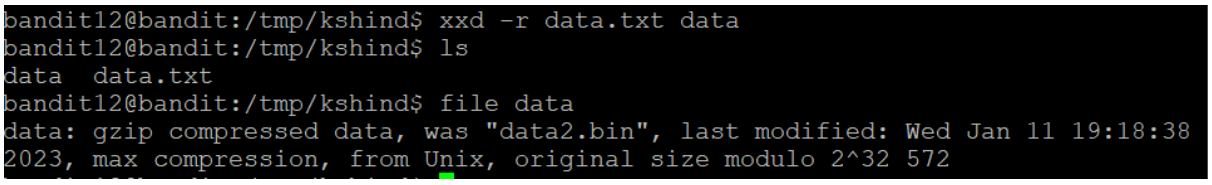


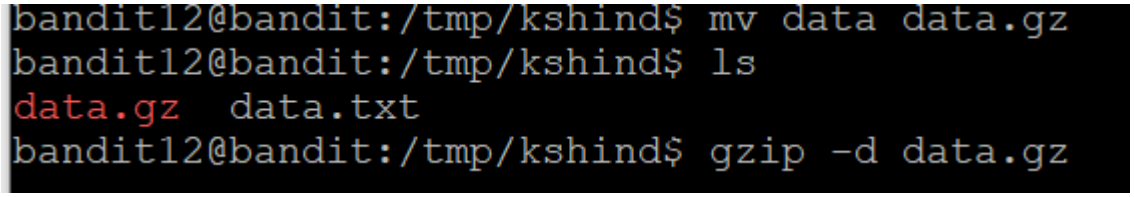
다음 레벨의 비밀번호는 data.txt 파일에 저장되어 있으며, 이 파일은 여러 번 압축된 파일의 헥스 덤프이다. 이 레벨에서는 /tmp 디렉토리 안에 작업할 수 있는 임시 디렉토리를 만드는 것이 유용할 수 있다고 한다. Mkdir 명령을 사용하여 추측하기 어려운 이름으로 디렉토리를 생성할 수 있다고 한다. 또는 mktemp -d 명령어를 사용하고, cp 명령을 사용하여 데이터 파일을 복사하고, mv 명령을 사용하여 파일 이름을 바꾸라고 한다.



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

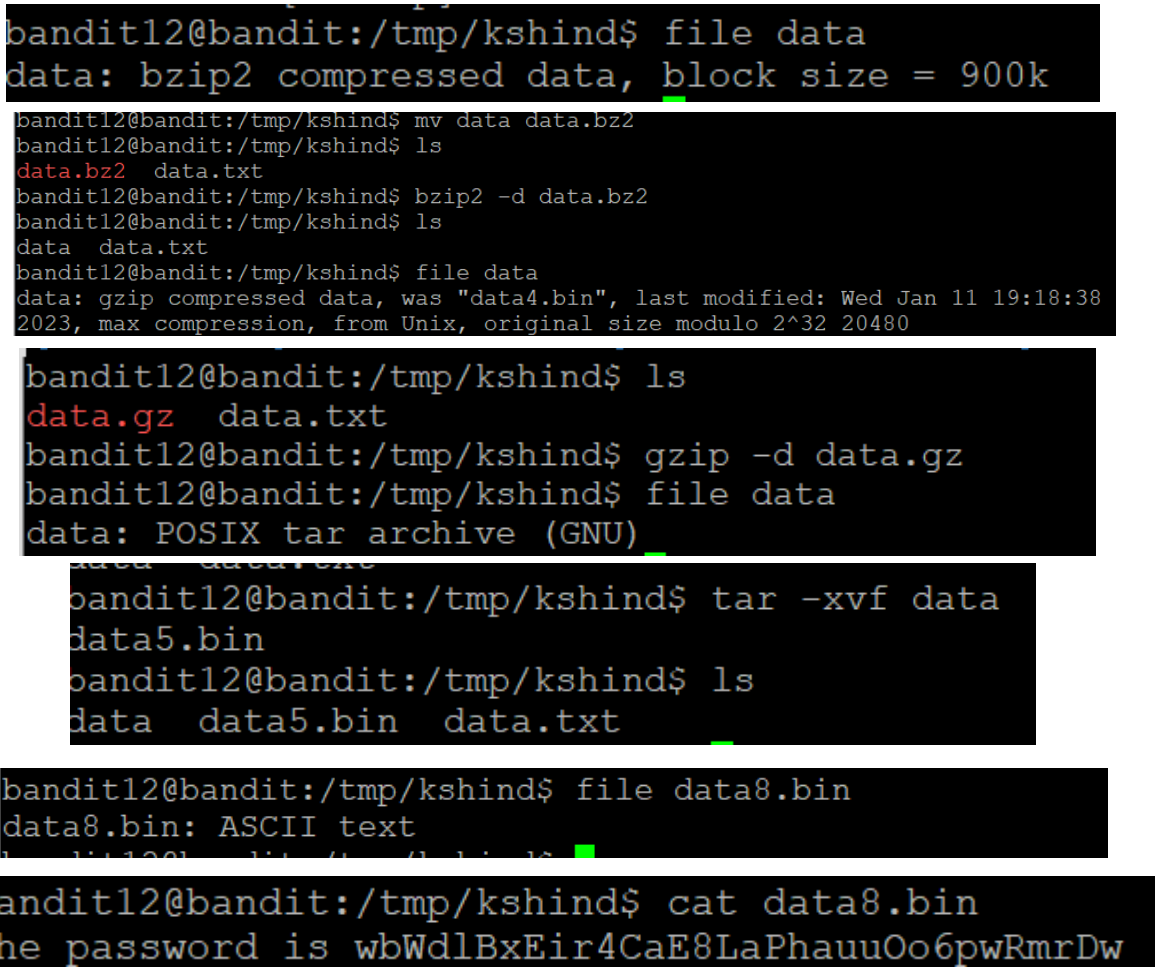






문제 설명에 따라 /tmp 디렉토리를 만들고 data.txt를 cp를 이용하여 복사해서 이동을 시켰다. 그리고, 해당 디렉토리에 xxd -r을 이용해서 data라는 파일로 새로 만들고, file 명

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



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 풀어주었다. 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