2023 코드큐어 여름방학 공부

202321316 정보보안공학과 이승훈

내용: dreamhack.io에 존재하는 웹해킹 문제를 풀어보기.(1단계)

1. flying chars

이 문제의 사이트를 들어가보면 날라다니는 글자들을 멈춰서 문자열을 알아내는 문제다.

F12를 키고, source 항목의 맨 아래를 보면

for(var i = 0; i < 20; i++){

anim(imgs[i], 0, Math.random()\*60+20);

}

이런 항목이 보이는데, 실제로 0부터 19까지의 png를 보면 어떤 문자가 있고 화면에는 그 문자들이 무작위로 돌아다니는 모습을 볼수 있다. 우리는 거기서 그 함수를 멈춰줘야 하는데,

Math.random()\*60+20을 \*0으로 바꿔주면 문자들이 알아서 순서대로 정렬을 하게 된다.

거기서 우리는 답을 볼수 있는데, Too\_H4rd\_to\_sEe\_th3\_Ch4rs\_x.x 이 답이다.

2. ex-reg-ex

문제에서 요구하는 형식의 문자열을 입력하여 flag를 획득.

d r\ w{5,7} e \d + am @ [a-z] {3,7}\ .\ w+

dr(알파벳)e(숫자)am@(영어)(3개부터 7개까지).알파벳 or 숫자

그러면?

dr(a부터 z까지 5개~7개)e(숫자 0~)am@ (a부터 z까지 3개~7개).(a or 0)

근데 일반적인 에메일 형식은 example@???.??? 이럼.

draaaaae0am@aaa.a

draaaaaae1am@aaa.0

이런식으로 넘어감.

답은 draaaaae0am@aaa.a 임. Flag: DH{e64a267ab73ae3cea7ff1255b5f08f3e5761defbfa6b99f71cbda74b7a717db3}

텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명

3. command injection chatgpt

특정 호스트에 핑 패킷을 보내는 서비스임.

Command injection 을 통해 플래그를 획득하셈.(flag는 flag.py에 있다고 했다.)

말 그대로 chatgpt를 통해 풀어보세요.

우선 이 문제를 풀어보기 전에 command injection이 뭔지 chatgpt에게 물어봤다.

텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

이런 답을 내놓았는데, 거기서 ls이라는 명령어가 나왔다.

그 ls라는 명령어가 뭔지 이 코드에서 대충이나마 알게 되었는데, 대충 그 파일에서 뭐가 있는지 알아내는 그런것이다.

더 자세히 알기 위해서 ls가 뭔지도 물어보았다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

음…. 내가 생각한 것이 맞다.

ls라고 치면 파일이 뭐가 있는지 알아내는 그런거다…

그러면 뭐가 있는지 알아내는 그런 명령어는 알아냈으니. 우리가 알아내고자 하는 답은 flag.py안에 있는 답을 빼오는것이다.

그것도 물어봤다.

텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

라고 나오는데, cat, head, tail, more, less, type 라는 명령어가 있고, 예제에서는 cat에 대해서 예를 들었다.

이제 알아냈으니 해당 문제의 코드로 가볼까?

텍스트, 스크린샷, 디스플레이, 폰트이(가) 표시된 사진

자동 생성된 설명

이런 코드가 있는데, /ping 이라는 루트의 코드와 사이트 안에 있는 결과와 비교해보면,

텍스트, 폰트, 스크린샷, 화이트이(가) 표시된 사진

자동 생성된 설명

일부러 오류를 내서 확인해보면,

Ping -c {host} 와 8.8.8.8ㅂ 를 비교했을 때, 8.8.8.8ㅂ가 host라는 것을 알아낼수 있다.

우리가 거기서 무작정 host부분에 ls를 치고 해보면 될것이라고 생각하지만 어림도 없다.

텍스트, 폰트, 스크린샷, 화이트이(가) 표시된 사진

자동 생성된 설명

이런 오류가 생긴다.

이 ls 라는 명령어만 작동되게 하던가 명령어를 끊어서(2개로 분활해서 실행)해야 하는데,

그것도 물어봤다.

텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명

~~음…똑똑하네…귀여운 것…~~

앞에 보이는대로 ; 기호를 사용해서 두개의 명령어를 실행할수 있다고 한다.

그러면 바로 답을 찾아볼까?

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

우선 8.8.8.8;ls 를 입력해봤는데, 정상적인 값이 나오고, 그 아래 .py가 몇 개가 있는것을 볼수 있다. 우리가 찾고 있는 flag.py도 여기에 있는 것을 확인할수 있다.

그럼 바로 cat 을 이용해서 열어볼까?

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

나이스. 답은

FLAG = 'DH{chatGPT\_knows\_what\_the\_ping\_is}'

이거다.

이걸로 1레벨 문제는 40개정도가 남았네…?

한번 해보자.

4. command injection-1

3번 문제에서 변형문제 인것 같다….

3번 문제에서 본 것 처럼 flag.py의 위치는 같다. 한번 ;ls를 쳐서 확인해보자.

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

이런 형식이 뜨는데, 보기엔 검색항목에서 걸러지는 것 같다.

F12를 열어서 한번 확인해보자.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

Class=”form-control”, pattern=”[A-Za-z0-9.]{5,20}”이라는 항목이 있다.

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

아마도 a부터 z까지 대문자, 소문자, 숫자를 넣을수 있는 것으로 보인다.

그리고 5부터 20자이내를 넣을 수 있는 것 으로 보인다.

하지만 burp 로 우회를 해서 flag 르 찾을것이다.

cmd = f'ping -c 3 "{host}"'

이런 코드에서 {host} 부분에 내가 입력한 코드가 들어가는데,

그냥 8.8.8.8;ls 를 입력하고 우회하면 나올줄 알았는데 안된다.

그래서 생각해낸 방법이 코드를 변조해서 우회하는 방식으로 할것이다.

8.8.8.8;ls를 입력하고 우회했을 때 안된다고 한적이 있다.

여기서 “”를 이용해서 우회할것인데, “8.8.8.8”;”ls”를 해서 우회를 해볼것이다.

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

우회에 성공했으니, flag.py를 열어보도록 하자.

그런데, 내가 아는 방법으로 8.8.8.8”;”cat flag.py 를 입력하고 우회했는데 안된다.

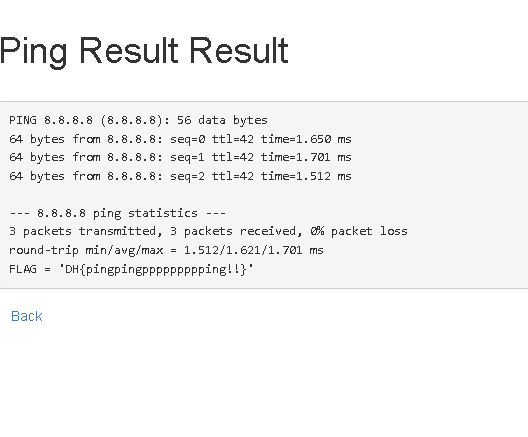
여기서 의문이 들었다. 설마 띄어쓰기 때문에 안되는건가?

근데 명령을 실행할려면 띄어쓰기는 꼭 있어야 하는데….

그래서 8.8.8.8”;”cat””flag.py 로 시도했는데 안된다.

문득 든 생각이 “”안에서 띄어쓰기를 포함하면 안되고, “” 밖에서는 띄어쓰기가 있어도 상관없는게 아닐까 싶어서

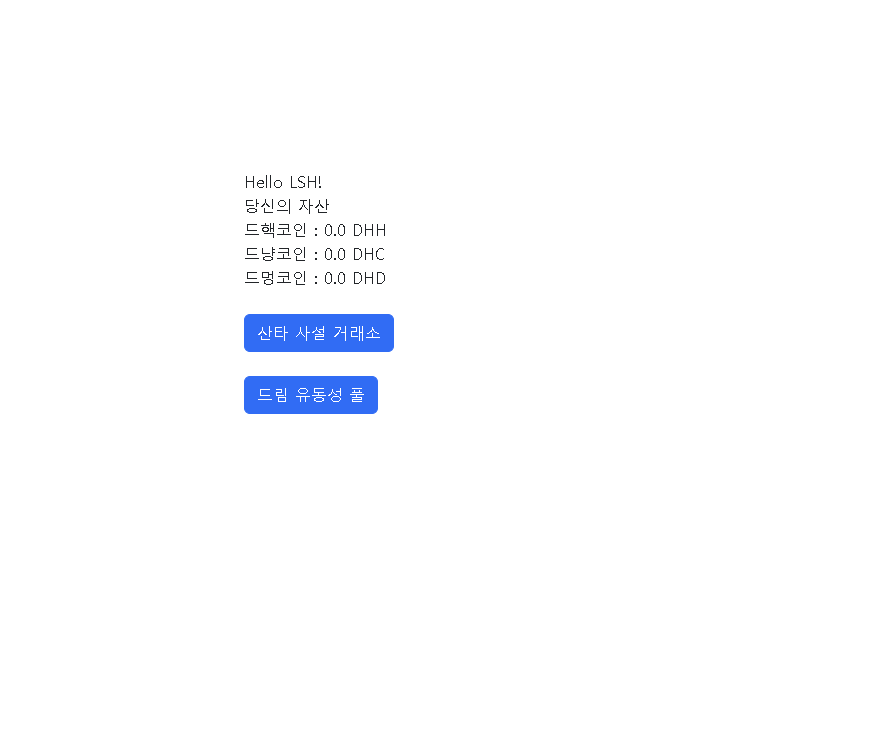
8.8.8.8";"cat" "flag.py 로 시도해봤다.



그렇게 하니까 됐다.

그래서 답은 DH{pingpingppppppppping!!} 이다.

5. out of money



이런 화면에서 산타 사설 거래소와 드림 유동성 풀 이라는 항목이 있는데, 우선 산타 거래소에는 텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

이런 화면이 있고, 드림 유동성에는

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

이런 화면이 있다.

이 문제에서는 DHC, DHD, DHH가 있다. 코드를 보면,

텍스트, 스크린샷, 폰트, 디스플레이이(가) 표시된 사진

자동 생성된 설명

Flag를 얻기 위해서는 DHH가 1000 이상이 있어야 하고, DHH 빚이 0이면 flag를 얻는 방식이다.

우선 3000을 빌린 다음에 DHH DHC DHD에 1000을 반환해주고, 드림에 가서 DHC 담보에 500을 내면 DHD를 500만큼 빌릴수 있다, flag를 얻을때는 DHD 빚은 따지지 않는다.

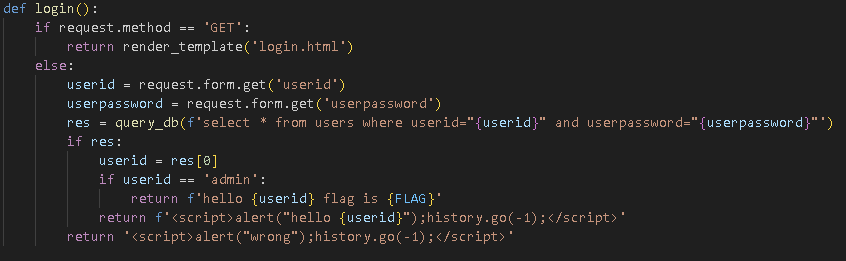
그러니 DHD를 왕창 빌리고 DHH로 반환해서 DHH 빚을 갚고, flag를 사면 된다.

DH{https://etherscan.io/tx/0x958236266991bc3fe3b77feaacea120f172c0708ad01c7a715b255f218f9313c}

개인적으로 조금 이상한 문제였지 않나 싶다.

6. simple sqli

말 그대로 sql 취약점을 이용해서 flag를 얻는 문제다.



로그인 항목을 보면, 아이디가 admin 이라는것을 알수 있다.

보통 admin’ ‘1’=’1 – 을 통해서 획득한다.

하지만 1=1을 빼고 –(주석)만을 이용해서 flag를 획득할수 있는 경우가 있다.

아이디에 admin” –을 넣고 비번에 12345(또는 아무거나)를 넣고 로그인 하면 flag를 획득할수 있다.

7. cookie

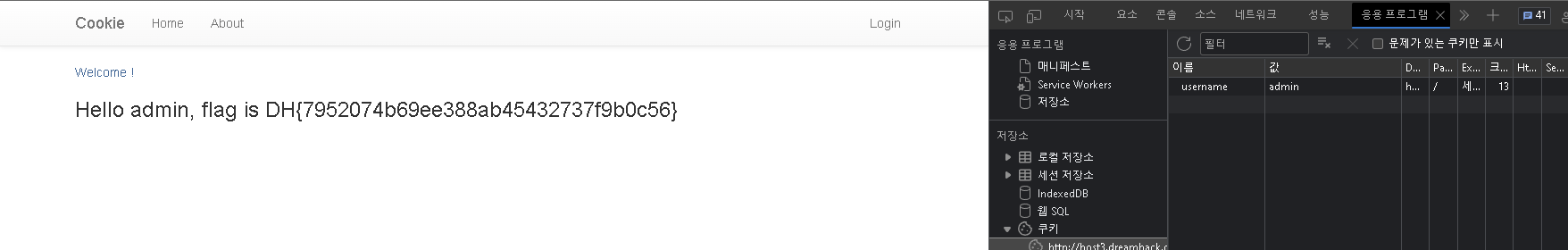
관리자 계정에 있는 쿠키를 획득해서 로그인 해서 flag를 획득하면 되는 문제다.

이 문제는 아무리 /admin을 봐도 아무런 단서를 얻지 못했고, 코드를 봐도 단서를 얻지 못했다.

스크린샷, 소프트웨어, 멀티미디어 소프트웨어, 텍스트이(가) 표시된 사진

자동 생성된 설명

먼저 guest 계정으로 로그인 하면 admin이 아니라는 문장이 나오고 그 옆에는 개발지 도구를 열어서 쿠키를 본 화면인데, 게스트일때 쿠키에 guest가 있다. Guest를 admin으로 바꾸면 어떻게 될까 해서 admin으로 바꿔보았다.

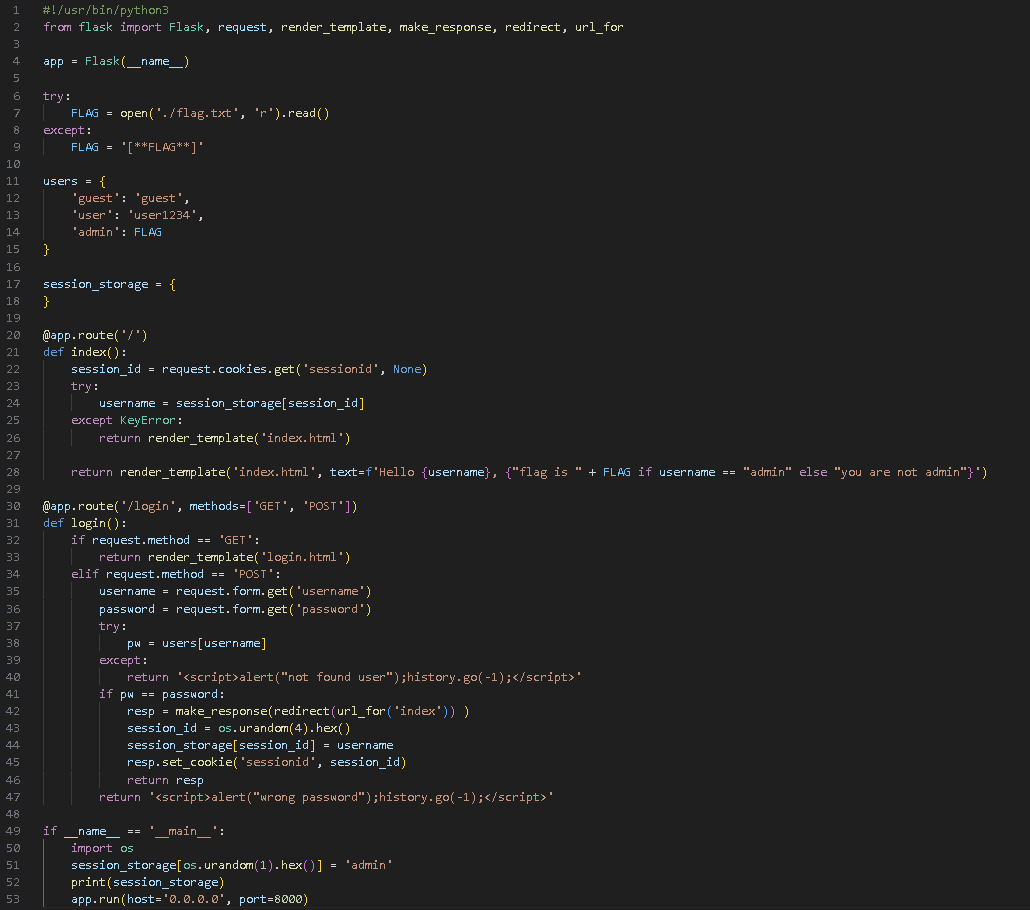


이런 화면이 뜨고 flag를 얻을수 있었다.

8. session

쿠키와 세션으로 인증 상태를 관리하는 간단한 로그인 서비스입니다.  
admin 계정으로 로그인에 성공하면 플래그를 획득할 수 있습니다.

라는 설명이 있다.

  
코드를 살펴보면

Guest/guest, user/user1234, admin/”FLAG” 로 로그인 할수 있고, 경로가 / , /login이 있다.

그리고 당연하게도 admin으로 로그인 하면 flag가 나오게 된다.

Pw==password의 함수에서

Session\_storage[session\_id]=username 의 함수를 볼수 있다. 이 함수에서 username의 session\_id 값은 session\_id = os.random(4).hex()이다.

즉 4글자로 된 hex 값이 username의 비번이다.

if \_\_name\_\_ == '\_\_main\_\_':

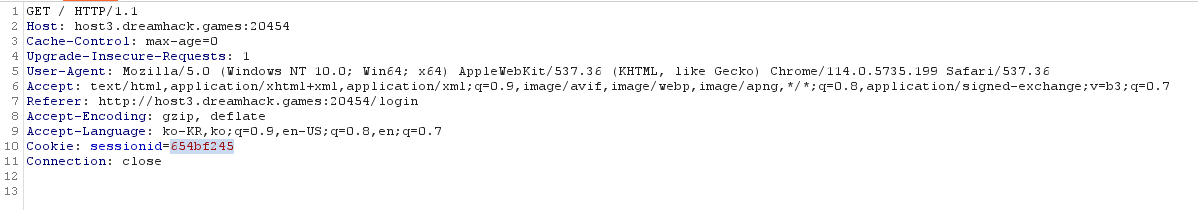
    import os

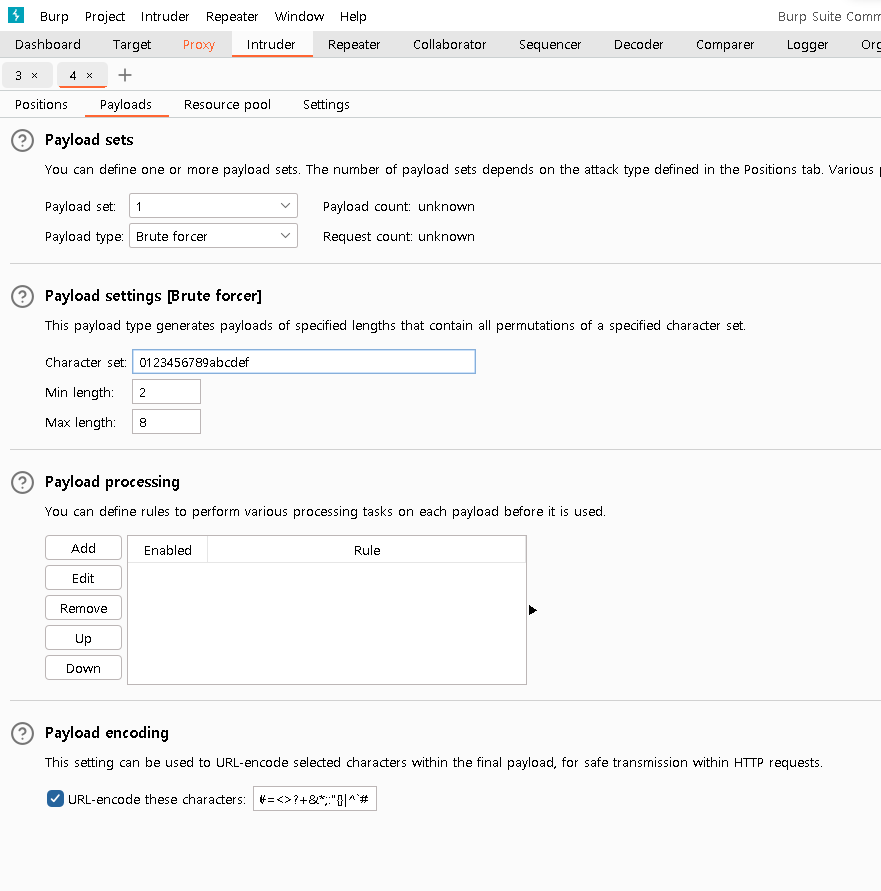
    session\_storage[os.urandom(1).hex()] = 'admin'

    print(session\_storage)

app.run(host='0.0.0.0', port=8000

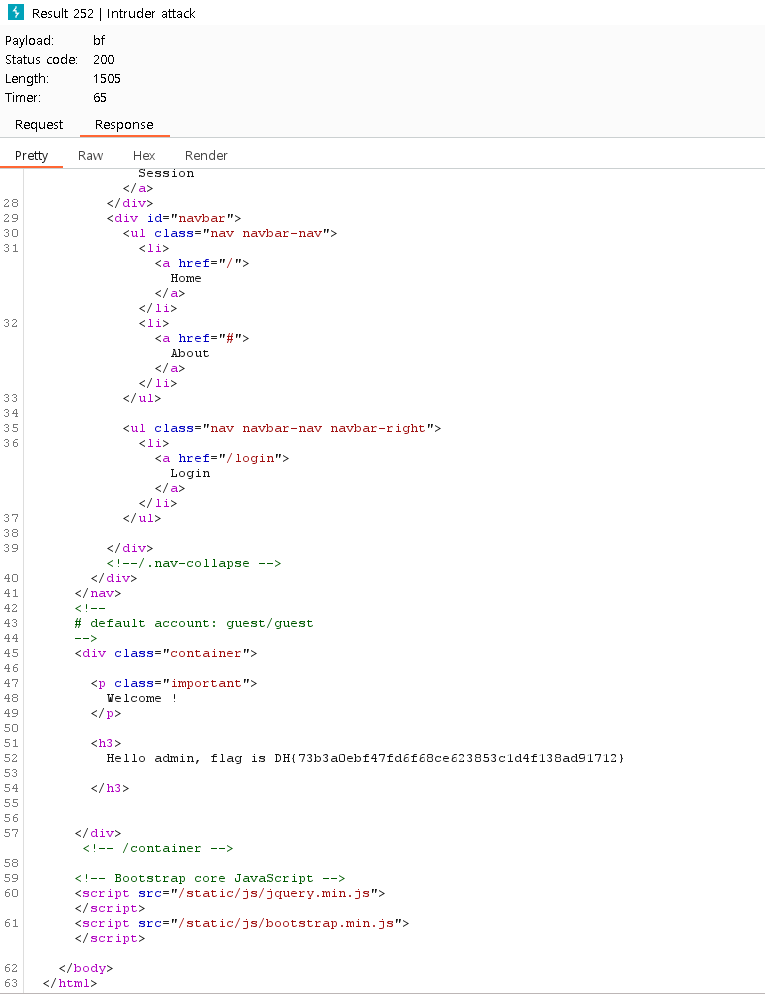
맨 마지막 코드를 보면 admin의 storage 의 값은 hex의 2글자이다.

Dlj 코드를 토대로 burp에서 

그 문제의 사이트를 intercept해서 그 id의 값을 ㄴ

2글자로 0123456789abcdef까지 해서 hex값을 만들어야 한다.

일일이 대입을 하는것이다.



이렇게 일일이 대입해서 답을 찾았다.

252번째에 답이 나왔는데,

DH{73b3a0ebf47fd6f68ce623853c1d4f138ad91712}이다.

8. php-1

먼저