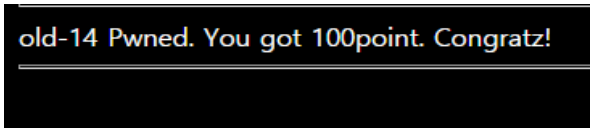
<웹해킹 14,17,34번 문제> 김수현

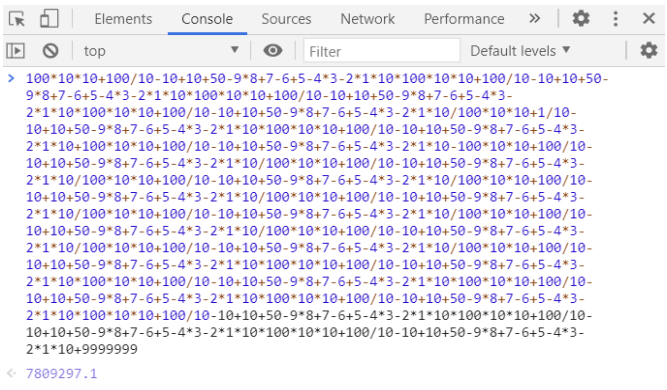
-14번 문제-

14번 문제는 그냥 아무 말 없이 간단히 입력창만 있었다. Ctrl+u를 통해 소스코드를 확인해보았다.


```
<html>
<head>
<title>Challenge 14</title>
<style type="text/css">
body { background:black; color:white; font-size:10pt; }
</style>
</head>
<body>
<br><br>
<form name=pw><input type=text name=input_pwd><input type=button value="check" onclick=ck()></form>
<script>
function ck(){
  var ul=document.URL;
  ul=ul.indexOf(".kr");
  ul=ul*30;
  if(ul==pw.input_pwd.value) { location.href="?"+ul*pw.input_pwd.value; }
  else { alert("Wrong"); }
}
</script>
</body>
</html>
```

지난 3주차에 자바 스크립트에 대해서 배웠기 때문에, script가 있는 부분을 집중적으로 살펴보았다. Var을 보고 자바 스크립트라고 확신을 했다. 그리고나서 확인한 것이 document.URL이다. 처음에는 이게 뭔가..라는 생각이 들어서 구글링을 해보았다. 결과는 '현주소'였다. Ul 이라는 변수에 현주소를 집어넣는 것인가라는 생각이 들었다. 또 indexOf가 뭔지 몰라서 구글링을 해봤더니 자바스크립트 함수의 일종인데, 문자열 안에 조건이 되는 문자열이 몇 번째 위치에 존재해 있는지를 확인하는 함수라고 한다. 따라서 해석해보니 .kr이 시작되는 위치를 숫자로 표현한 다음 다시 ul이라는 변수에 넣는 것이라고 생각이 들었다. 그 다음 ul*30한 값을 다시 ul에 넣고 ul과 pw(패스워드)가 같으면, 그 옆 문장을 실행하고 아니라면 Wrong이라는 경고창을 띄운다는 것이다. 이제 문제는 location.href="?"+ul*pw.input_pwd.value; 부분이다. 처음에는 무엇을 말하는지 몰랐었다. 그런데 조금더 생각해보니 현재 URL에서 ".kr"의 위치에 30이 곱해진 값이 저장되고, 이 ul 값과 pw.input_pwd.value(입력창에 넣은 값)이 일치하면 문제가 풀리는 것 같다는 생각이 들었다. 처음에 계산했을 때는 .kr의 위치가 19가 나와서 19*30한 값을 적었는데 틀렸었다. 하지만 내가 빼먹은 부분이 indexOF함수에서는 처음시작위치가 0이라는 사실을 깨달았다. 따라서, 18*30=540이 답이라고 생각했다. 결과는 성공





-17번 문제-

17번 문제도 14번 문제와 같이 검은 색 화면에 입력창 하나만 있었다. 또 같은 방법으로 ctrl+u를 통해 소스코드를 확인해보았다.

```
<html>
<head>
<title>Challenge 17</title>
</head>
<body bgcolor=black>
<font color=red size=10></font>
<p>
<form name=login>
<input type=passwd name=pw><input type=button onclick=sub() value="check">
</form>
<script>
unlock=100*10*10+100/10-10+10+50-9*8+7-6+5-4*3-2*1*10*100*10*10+100/10-10+10+50-9*8+7-6+5-4*3-2*1*10*100*10*10+100/10-10+10+50-9*8+7-6+5-4*3-2*1*10*100*10*10+100/10-10+10+50-9*8+7-6+5-4*3-2*1*10/100*10*10+1/10-10+10+50-9*8+7-6+5-4*3-
2*1*10*100*10*10+100/10-10+10+50-9*8+7-6+5-4*3-2*1*10+100*10*10+100/10-10+10+50-9*8+7-6+5-4*3-2*1*10-100*10*10+100/10-10+10+50-9*8+7-6+5-4*3-2*1*10/100*10*10+100/10-10+10+50-9*8+7-6+5-4*3-2*1*10/100*10*10+100/10-10+10+50-9*8+7-6+5-
4*3-2*1*10/100*10*10+100/10-10+10+50-9*8+7-6+5-4*3-2*1*10/100*10*10+100/10-10+10+50-9*8+7-6+5-4*3-2*1*10/100*10*10+100/10-10+10+50-9*8+7-6+5-4*3-2*1*10/100*10*10+100/10-10+10+50-9*8+7-6+5-4*3-2*1*10/100*10*10+100/10-10+10+50-9*8+7-
6+5-4*3-2*1*10/100*10*10+100/10-10+10+50-9*8+7-6+5-4*3-2*1*10/100*10*10+100/10-10+10+50-9*8+7-6+5-4*3-2*1*10*100*10*10+100/10-10+10+50-9*8+7-6+5-4*3-2*1*10*100*10*10+100/10-10+10+50-9*8+7-6+5-4*3-2*1*10*100*10*10+100/10-10+10+50-
9*8+7-6+5-4*3-2*1*10*100*10*10+100/10-10+10+50-9*8+7-6+5-4*3-2*1*10*100*10*10+100/10-10+10+50-9*8+7-6+5-4*3-2*1*10*100*10*10+100/10-10+10+50-9*8+7-6+5-4*3-2*1*10*100*10*10+100/10-10+10+50-9*8+7-6+5-4*3-2*1*10+9999999;
function sub(){ if(login.pw.value==unlock){ location.href="?"+unlock/10; } else{ alert("Wrong"); } }
</script>
```

Script부분을 보면 unlock에대한 식이 엄청나게 길게 있음을 알 수 있다. 보기만 해도 끔찍한 계산식이지만 계산할 방법이 분명히 있을 것이라고 생각이 들었다. 그리고나서 밑에 살펴보니 function sub() 식이 있는데 만약 login.pw.value 값이 위에 있는 unlock이라면 unlock을 /10한 값을 대입하라고 되어있고 아니라면 wrong 창을 띄우라는 식을 볼 수 있다. 일단, 어떻게 저 식을 계산해야 될 지 몰라서 구글링을 해보았다. 그랬더니 크롬의 상단에 개발자 도구를 찾아 콘솔창에 입력하면 자동으로 계산된다고 하는 것을 알게 되었다.

```
> 100*10*10+100/10-10+10+50-9*8+7-6+5-4*3-2*1*10*100*10*10+100/10-10+10+50-
  9*8+7-6+5-4*3-2*1*10*100*10*10+100/10-10+10+50-9*8+7-6+5-4*3-
  2*1*10*100*10*10+100/10-10+10+50-9*8+7-6+5-4*3-2*1*10/100*10*10+1/10-
  10+10+50-9*8+7-6+5-4*3-2*1*10*100*10*10+100/10-10+10+50-9*8+7-6+5-4*3-
  2*1*10+100*10*10+100/10-10+10+50-9*8+7-6+5-4*3-2*1*10-100*10*10+100/10-
  10+10+50-9*8+7-6+5-4*3-2*1*10/100*10*10+100/10-10+10+50-9*8+7-6+5-4*3-
  2*1*10/100*10*10+100/10-10+10+50-9*8+7-6+5-4*3-2*1*10/100*10*10+100/10-
  10+10+50-9*8+7-6+5-4*3-2*1*10/100*10*10+100/10-10+10+50-9*8+7-6+5-4*3-
  2*1*10/100*10*10+100/10-10+10+50-9*8+7-6+5-4*3-2*1*10/100*10*10+100/10-
  10+10+50-9*8+7-6+5-4*3-2*1*10/100*10*10+100/10-10+10+50-9*8+7-6+5-4*3-
  2*1*10/100*10*10+100/10-10+10+50-9*8+7-6+5-4*3-2*1*10/100*10*10+100/10-
  10+10+50-9*8+7-6+5-4*3-2*1*10*100*10*10+100/10-10+10+50-9*8+7-6+5-4*3-
  2*1*10*100*10*10+100/10-10+10+50-9*8+7-6+5-4*3-2*1*10*100*10*10+100/10-
  10+10+50-9*8+7-6+5-4*3-2*1*10*100*10*10+100/10-10+10+50-9*8+7-6+5-4*3-
  2*1*10*100*10*10+100/10-10+10+50-9*8+7-6+5-4*3-2*1*10*100*10*10+100/10-
  10+10+50-9*8+7-6+5-4*3-2*1*10*100*10*10+100/10-10+10+50-9*8+7-6+5-4*3-
  2*1*10+9999999
< 7809297.1
```

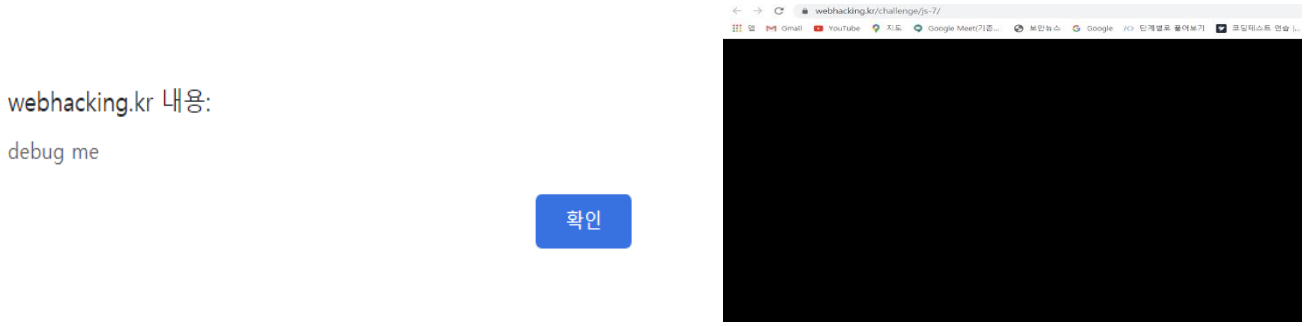
저기 밑에 있는 7809297.1 값을 입력했더니, 성공했다는 창이 떴다. 비교적 어렵지 않은 문제였던 것 같다.

webhacking.kr 내용:

old-17 Pwned!

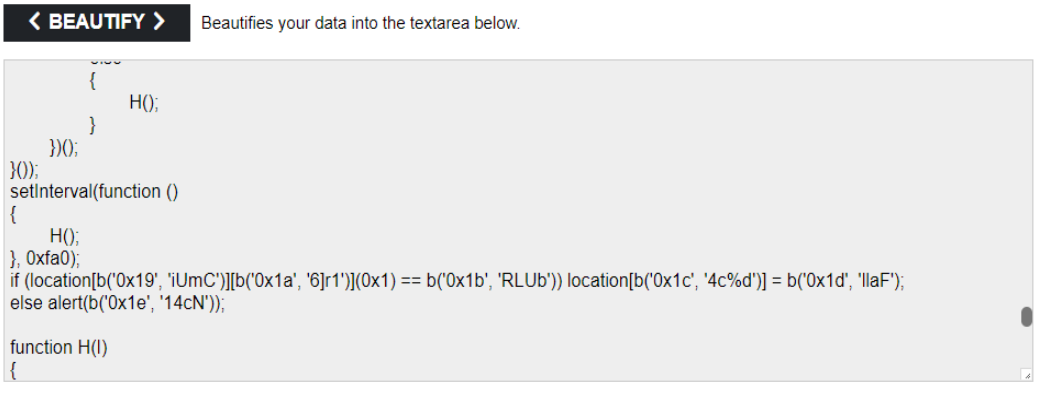
-34번 문제-

34번 문제는 눈이 빠질 뻔한 문제였다. 34번 문제를 클릭하면 갑자기 debug me 라는 창과 함께 아무것도 없는 검은 화면이 나타난다. 그래서 나는 바로 ctrl+u를 통해 소스코드를 확인해보았다.



이와 같은 엄청난 길이의 코드가 다닥다닥 붙어 있어서 확인하기가 어려웠다. 코드를 나눠서 보여주는 사이트에 들어가서 확인해보았다.

나는 처음에 경고창이 뜬게 이상했다. 수업시간 때 alert와 관련된 문제를 과제로 내줬다고 그래서 이 문제가 alert와 관련된 문제인가 생각했다. 코드를 쭉(인내심을 가지고)살펴보다가 발견했다. Alert가 있는 코드를 발견했다.



코드가 너무 복잡해서 무슨 말인지 몰랐다. If ~else 구문인데, 도대체 뭘 입력해야 할 지 몰랐다.

그래서 사실 코드가 될 만한 것을 다 정답 칸에 입력해봤다. 그 중 location[b('0x1c', '4c%d')]=b('0x1d', 'llaF') 가 정답이었다. 사실 이 문제는 내가 노가다(?)로 푼 거라 얼떨떨하긴 했다. 하지만 왜 저게 답이 되는 지 아직도 잘 모르겠다. 운이 좋았나보다.