써니나타스 문제로 해킹의 기초 익히기

장 준 호 (humit)

2018. 05. 24.

1. [Web] Level 1

1.1. 문제 분석

먼저 문제에 나와있는 소스코드를 보도록 하자.

```
<%
    str = Request("str")

    If not str = "" Then
        result = Replace(str,"a","aad")
        result = Replace(result,"i","in")
        result1 = Mid(result,2,2)
        result2 = Mid(result,4,6)
        result = result1 & result2
        Response.write result
        If result = "admin" Then
            pw = "????????"
        End if
    End if
%>
```

먼저 폼에서 str로 되어있는 부분의 값을 가져온 다음, Replace와 Mid, &연산자를 통해 값을 가공하고 있는 것을 볼 수 있다. 그리고 result 변수의 값이 'admin'이면 된다고 볼 수 있다.

1.2. 이론적 배경

이번 문제에서는 asp에서 사용하는 함수들과 연산자에 대해서 알면 충분히 해결이 가능하다.

- Request 함수
  - 사용법 : Request([입력 필드의 이름])
  - GET이나 POST요청에서 해당하는 입력 필드의 값을 반환한다.

- Replace 함수
  - 사용법 : Replace([원본 문자열], [바꾸려는 문자열], [바꾸려는 문자열의 변경 값])
  - 바꾼 문자열을 반환한다.

- Mid 함수
  - 사용법 : Mid([원본 문자열], [문자열 시작 위치], [가져오려는 문자열의 길이])

    - 원본 문자열에서 해당 시작 위치로부터 가져오려는 문자열의 길이만큼 뽑아낸 문자열을 반환한다.
    - 만약 가져오려는 문자열의 길이가 범위를 벗어난 경우 원본 문자열의 마지막까지만 가져온다.

- & 연산자
    - 사용법 : [첫 번째 문자열] & [두 번째 문자열]
    - 첫 번째 문자열과 두 번째 문자열을 서로 연결한 값을 반환한다.

- not 연산자
    - 사용법 : not [논리값]
    - 피연산자가 True인 경우에는 False를, 피연산자가 False인 경우에는 True를 반환한다.

1.3. 문제 해결

result1은 최대 2글자이고, result2는 최대 6글자가 가능하다. result1과 result2를 결합한 문자열이 admin이 되어야 하기 때문에, result1은 ad가 되어야 하고, result2는 min이 되어야 한다.

따라서 7번째 줄에서 result의 값이 ?admin이 되어야 한다. 이제 Replace 함수를 역으로 추적하면, 6번째 줄에서 result 값이 ?admi이어야 하고, str 값이 ami가 되어야 함을 알 수 있다.

그러면 아래와 같이 flag 값을 얻을 수 있다.

<사진첨부>

2. [Web] Level 2

2.1. 문제 분석

문제를 보면 매우 간단한 회원가입 창을 볼 수 있다. 소스를 확인하면 힌트로 ID와 PW의 값이 일치하도록 가입을 하면 된다고 한다.
<사진첨부>

일단 ID와 PW의 값을 서로 다르게 하여 Join 버튼을 누르면 아무런 변화가 없어보인다.
하지만 ID와 PW의 값을 같게 해서 Join 버튼을 누르면 'You can't join! Try again'이라는 경고창이 뜨고 진행이 되지 않는다.
<사진첨부>

ID와 PW가 같게하면서 경고창을 띄우지 않도록 하는 방법을 찾으면 된다. Join 버튼에 해당하는 HTML코드를 보면 아래와 같다.

```
<input type="button" value="Join" style="width:60" onclick="chk_form()">
```

즉 버튼을 클릭했을 때 chk_form이라는 자바스크립트가 실행된다는 사실을 알 수 있다. 해당 함수의 소스코드를 보면 아래와 같다.

```
function chk_form(){
	var id = document.web02.id.value ;
	var pw = document.web02.pw.value ;
	if ( id == pw )
	{
		alert("You can't join! Try again");
		document.web02.id.focus();
		document.web02.id.value = "";
		document.web02.pw.value = "";
	}
	else
	{
		document.web02.submit();
	}
}
```

즉 id와 pw의 값이 같으면 경고창을 띄워주고 다른 경우에는 폼을 전송함을 알 수 있다.

2.2. 이론적 배경

이번 문제에서는 앞의 문제 분석에서도 보았듯이 HTML 코드와 Javascript에 대한 지식만 있으면 충분히 해결이 가능하다.

- HTML 이벤트 속성[1]
  - onerror : 에러가 발생했을 때 실행
  - onload : 페이지가 전부 로드된 다음 실행
  - onfocus : 커서가 해당 원소에 있는 경우 실행
  - onsubmit : 폼이 전송되었을 때 실행
  - onkeydown : 사용자가 키보드를 눌렀을 때 실행
  - onkeypress : 사용자가 키보드를 누르고 떼었을 때 실행
  - onkeyup : 사용자가 키보드를 떼었을 때 실행
  - onclick : 사용자가 마우스로 클릭했을 때 실행
  - ondbclick : 사용자가 마우스로 더블클릭했을 때 실행

이밖에도 많은 이벤트 속성이 있으니 필요한 속성이 있는 경우 주석 사이트를 참조하면 된다.

2.3. 문제 해결

먼저 아래 사진과 같이 ID와 PW를 똑같이 해서 입력을 한다.
<사진첨부>

그 다음 콘솔 창을 연 다음, 명령어로 document.web02.submit()을 입력해서 실행하면 된다.
이 때 주의할 점이 실행공간이 main(web02.asp)로 되어있어야 에러가 발생하지 않는다. 그러면 아래와 같이 Flag를 얻을 수 있다.
<사진첨부>

[1] https://www.w3schools.com/tags/ref_eventattributes.asp

3. [Web] Level 3

3.1. 문제 분석

문제에 접속하면 아래의 사진과 같은 페이지를 볼 수 있다.
<사진첨부>

즉 공지사항 게시판에 글을 작성하면 flag 값을 준다고 볼 수 있다. 공지사항 게시판에 들어가보면 Q&A 게시판과 달리 따로 WRITE 버튼이 없다는 것을 알 수 있다.
<사진첨부>

3.2. 이론적 배경

과거에 많이 사용했던 제로보드에서의 글쓰기 과정을 알아보자.

3.3. 문제 해결

Q&A 게시판에서 글쓰기 링크가 http://suninatas.com/board/write.asp?page=1&divi=Free처럼 되어 있는 것을 확인할 수 있고 이 정보를 통해 공지사항 게시판에 글을 쓸 수 있을 것이다. http://suninatas.com/board/write.asp?page=1&divi=notice로 접속을 해서 글을 작성한 후 submit 버튼을 누르면 아래와 같이 Flag 값을 경고창으로 띄워줌을 알 수 있다.

4. [Web] Level 4

4.1. 문제 분석

문제에 접속하면 아래의 사진과 같은 페이지를 볼 수 있다.
<사진첨부>

4.2. 이론적 배경

4.3. 문제 해결

5. [Web] Level 5

5.1. 문제 분석

5.2. 이론적 배경

5.3. 문제 해결

6. [Web] Level 6

6.1. 문제 분석

6.2. 이론적 배경

6.3. 문제 해결

7. [Web] Level 7

7.1. 문제 분석

7.2. 이론적 배경

7.3. 문제 해결

8. [Web] Level 8

8.1. 문제 분석

8.2. 이론적 배경

8.3. 문제 해결

9. [Binary] Level 9

9.1. 문제 분석

9.2. 이론적 배경

9.3. 문제 해결

10. [Binary] Level 10

10.1. 문제 분석

10.2. 이론적 배경

10.3. 문제 해결

11. [Binary] Level 11

11.1. 문제 분석

11.2. 이론적 배경

11.3. 문제 해결

12. [Misc] Level 12

12.1. 문제 분석

12.2. 이론적 배경

12.3. 문제 해결

13. [Misc] Level 13

13.1. 문제 분석

13.2. 이론적 배경

13.3. 문제 해결

14. [Forensic] Level 14

14.1. 문제 분석

14.2. 이론적 배경

14.3. 문제 해결

15. [Forensic] Level 15

15.1. 문제 분석

15.2. 이론적 배경

15.3. 문제 해결

16. [System] Level 16

16.1. 문제 분석

16.2. 이론적 배경

16.3. 문제 해결

17. [Misc] Level 17

17.1. 문제 분석

17.2. 이론적 배경

17.3. 문제 해결

18. [Forensic] Level 18

18.1. 문제 분석

18.2. 이론적 배경

18.3. 문제 해결

19. [Forensic] Level 19

19.1. 문제 분석

19.2. 이론적 배경

19.3. 문제 해결

20. [System] Level 20

20.1. 문제 분석

20.2. 이론적 배경

20.3. 문제 해결

21. [Forensic] Level 21

21.1. 문제 분석

21.2. 이론적 배경

21.3. 문제 해결

22. [Web] Level 22

22.1. 문제 분석

22.2. 이론적 배경

22.3. 문제 해결

23. [Web] Level 23

23.1. 문제 분석

23.2. 이론적 배경

23.3. 문제 해결

24. [System] Level 24

24.1. 문제 분석

24.2. 이론적 배경

24.3. 문제 해결

25. [System] Level 25

25.1. 문제 분석

25.2. 이론적 배경

25.3. 문제 해결

26. [Forensic] Level 26

26.1. 문제 분석

26.2. 이론적 배경

26.3. 문제 해결

27. [System] Level 27

27.1. 문제 분석

27.2. 이론적 배경

27.3. 문제 해결

28. [Forensic] Level 28

28.1. 문제 분석

28.2. 이론적 배경

28.3. 문제 해결

29. [Forensic] Level 29

29.1. 문제 분석

29.2. 이론적 배경

29.3. 문제 해결

30. [Forensic] Level 30

30.1. 문제 분석

30.2. 이론적 배경

30.3. 문제 해결

31. [Forensic] Level 31

31.1. 문제 분석

31.2. 이론적 배경

31.3. 문제 해결

32. [Forensic] Level 32

32.1. 문제 분석

32.2. 이론적 배경

32.3. 문제 해결