BEISTLAB FOR SECURITY SINCE 2001

[micsland.com web wargame풀이]

Written by OldZombie

MSN: oldzombie@hotmail.com

EMAIL: tjrdmswhaql@naver.com

by beistlab(http://beist.org)

micsland.com 워게임 레벨1-1 풀이

개요

micsland.com 워게임은 php 코드에 대한 이해도를 평가하는 문제로, php 코드를 보여준 후 그 코드에서 원하는 값을 입력하면 해결코드를 얻을 수 있습니다. 레벨은 1에서 6까지 있으며 각 레벨당 4개 정도의 문제가 있습니다. 각 레벨의 모든 문제를 마칠 경우 plus 문제가 주어지는데, 이 문제는 그 레벨에서 나온 문제의 종합으로 난이도가 가장 높습니다.

레벨1의 1단계 문제는 자바스크립트의 prompt를 이용한 인증을 통과하는 문제입니다. Prompt로 사용자에게 입력을 받은 후, if문으로 입력한 문자와 패스워드를 비교하는데 이 때 비교하는 패스워드가 무엇인지 알아내야 됩니다.

내용

문제 페이지에 접근하면 비번은? 이라는 메시지가 써있는 프롬포트 창이 하나 뜨고, 아무 문자나 입력한 후 엔터를 누르면 틀렸습니다. 라는 메시지가 출력됩니다. 클라이언트 브라우저에서 해석되는 자바스크립트의 특성상, 소스보기를 통해 자바스크립트 소스 전체를 볼 수 있습니다.

```
<script>
password = prompt("비번은?","");
if(password == "thefirst") document.write("축하합니다! 1-1단계를 해결하셨습니다.<br>1-1단계 해결코드는 n7Dy8MXb입니다.<br><a href='1-2.php'>1-2단계로</a>");
else document.write("틀렸습니다. 새로고침해서 재시도해 주세요.");
</script>
```

프롬포트 창에 사용자의 입력을 받아서 password 변수에 저장하고,if문으로 이 password 변수의 값이 thefirst가 되면 해결코드를 보여주는 것을 알 수 있습니다.
즉,패스워드는 thefirst가 됩니다.

결론

자바스크립트는 클라이언트의 브라우저에서 실행되는 스크립트이므로 소스보기를 통해 어떤 식으로 작동하는지 정확하게 파악할 수 있습니다. 자바스크립트를 이용한 인증은 클라이언트가 얼마든지 스크립트 내용을 조작할 수 있기 때문에 이것을 이용한 인증은 믿을 수 없습니다.

micsland.com 워게임 레벨1-2 풀이

개요

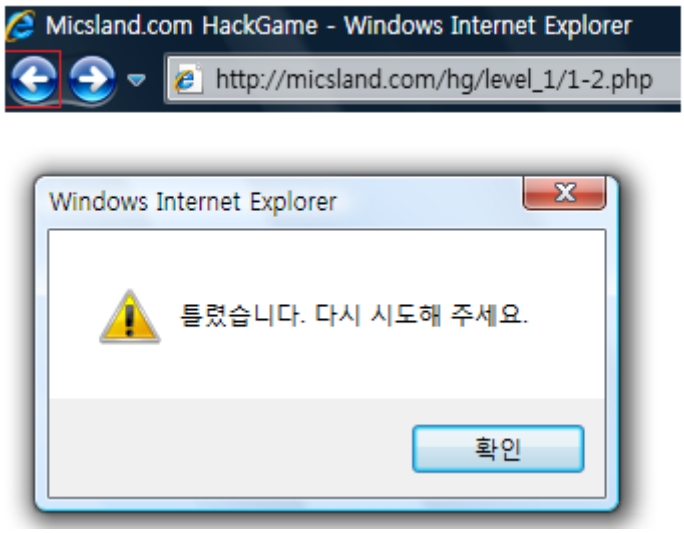
레벨1의 2단계 문제 역시 자바스크립트 prompt를 이용한 인증 문제입니다. 하지만 이번 문제는 단순한 방법으로는 소스보기를 할 수 없으므로 다른 방법을 사용해야 됩니다.

내용

문제 페이지에 접근하면 1단계와 똑같이 비번은? 이라는 메시지가 써있는 프롬포트 창이 하나 뜨고, 아무 문자나 입력한 후 엔터를 누르면 틀렸습니다. 라는 메시지가 경고창에 출력됩니다. 1단계와는 다르게 소스보기를 할 틈이 없이 바로 전 페이지로 강제 이동됩니다.
이렇게 소스보기를 막아놓았을 경우에는 문제 페이지를 직접 다운받아서 텍스트 에디터로 열어보면 소스를 볼 수 있습니다.

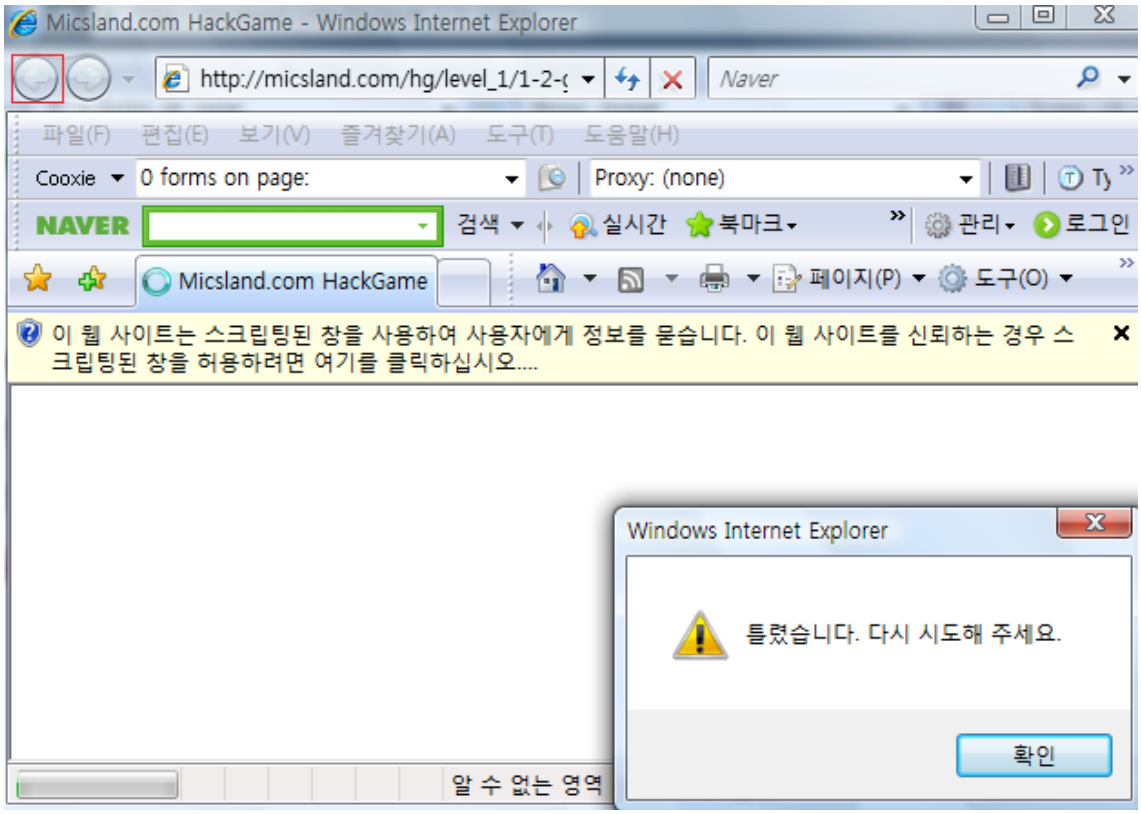
```
<script>
password = prompt("비번은?","");
if(password == "hahahaha") document.write("축하합니다! 1-2단계를 해결하셨습니다.<br>1-2단계 해결코드는 fF32TmaC입니다.<br><a href='1-3.php'>1-3단계로</a>");
else { alert("틀렸습니다. 다시 시도해 주세요."); history.back(); }
</script>
```

프롬포트 창에 사용자의 입력을 받아서 password 변수에 저장하고,if문으로 이 password 변수의 값이 hahahaha면 해결코드를 보여주고,아니면 history.back을 통해 바로 전 페이지로 이동하는 것을 확인할 수 있습니다. 만약에 이 문제 페이지에 접근할 때, shift+클릭(새 창으로 열기)을 했다면 저 history.back은 있으나 마나 제대로 역할을 하지 못하게 됩니다. History.back은 내 브라우저에서 뒤로 버튼을 누른것과 같은 효과를 주는데,브라우저로 새 창을 띄웠을 때에는 바로 전 페이지가 없기 때문입니다.
이해를 돕기 위해 그림을 살펴봅시다.



[그림1-2-1. 문제 페이지에 그냥 접속했을 때]

뒤로가기 버튼에 불이 켜져있는 것을 확인할 수 있습니다. 이 경우에 틀렸습니다. 라는 경고 박스에서 확인을 누르면 바로 전 페이지로 강제로 이동됩니다.



[그림1-2-2. 새 창을 띄워서 문제 페이지에 접속했을 때]

뒤로가기 버튼의 불이 켜지지 않았습니다. 여기서 확인을 눌러도 강제로 전 페이지로 이동되지 않는 것을 확인할 수 있습니다.

결론

자바스크립트로 클라이언트의 브라우저를 통제할려고 해봤자 자바스크립트는 클라이언트에서 해석되는 스크립트이므로 얼마든지 수정될 수 있습니다. 자바스크립트를 이용한 인증은 안전하지 않습니다. 그렇다고 서버측에서만 인증을 하면 서버에 많은 부담이 됩니다.
자바스크립트로 일차적으로 입력값 검증을 하고 서버에서 두 번째로 검증하는 식으로 만들어놓으면 서버의 부담도 줄일 수 있고 입력값 검증도 제대로 할 수 있습니다.

micsland.com 워게임 레벨1-3 풀이

개요

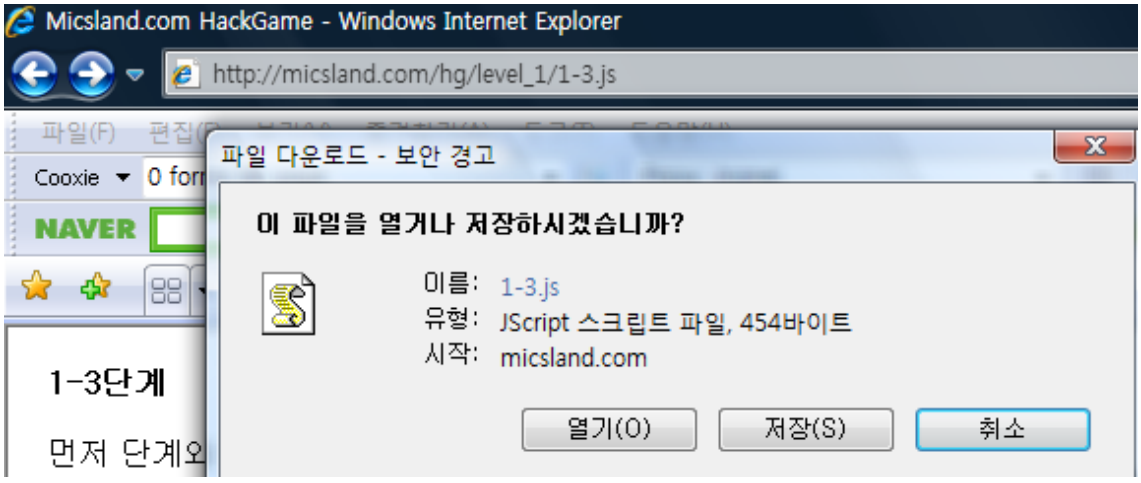
레벨1의 3단계 문제는 자바스크립트의 이해도를 알아보는 간단한 문제입니다.
자바스크립트에서 사용하는 php의 include같은 명령어에 대해 알고있어야 됩니다.

내용

문제 페이지에 접근하자마자 2단계와 비슷하게 경고창을 보여준 후, 바로 전 페이지로 강제로 이동시킵니다. 2단계에서 했던 방법으로 새 창을 띄운 다음에 소스를 살펴봅시다.

```
<body leftmargin='20' topmargin='20' marginwidth='0' marginheight='0' scroll='auto'><script src="1-3.js"></script>
</body>
```

이번 문제는 1,2단계와 다르게 소스가 그대로 나타나 있지 않습니다. 뒷부분을 살펴보면 script src라는 명령이 있는데, script src는 php의 include처럼 자바스크립트 파일을 첨부하여 실행하라는 명령입니다. 1-3.js를 첨부하므로 1-3.js를 다운받아 봅시다.



[그림1-3-1. js파일을 다운받는 모습]

다운받은 파일을 텍스트 에디터로 열어봅시다.

```
id = "nzeo";
pw = "kakaka";
id_i = prompt("ID는?","");
password = prompt("비번은?","");
if(id_i == id && password == pw) document.write("축하합니다! 1-3단계를 해결하셨습니다.<br>1-3단계 해결코드는 8gaD1fcL입니다.<br><a href='1-p.php'>1+ 단계로</a><br><a href='1-final.php'>1단계 바로 종료</a><br>* 1+ 단계를 깨지 않고 바로 종료하셔도 다음 단계로 진행하는 데 아무런 문제가 없습니다.");
else { alert("틀렸습니다. 다시 시도해 주세요."); history.back(); }
```

id 변수에 nzeo,pw 변수에 kakaka라는 값을 넣고 두 개의 프롬포트를 띄워서 사용자의 입력을 받습니다. 프롬포트에 입력된 값과 id,pw값을 비교하므로 첫 번째 프롬포트에는 id변수의 값은 nzeo,두 번째 프롬포트에는 pw 값 kakaka를 넣어주면 조건을 만족하게 되고 해결코드를 얻을 수 있게됩니다.

결론

자바스크립트는 어떤 방법을 써도 결국에는 클라이언트 브라우저에서 해석되므로 자바스크립트를 이용한 인증은 안전하지 않습니다.

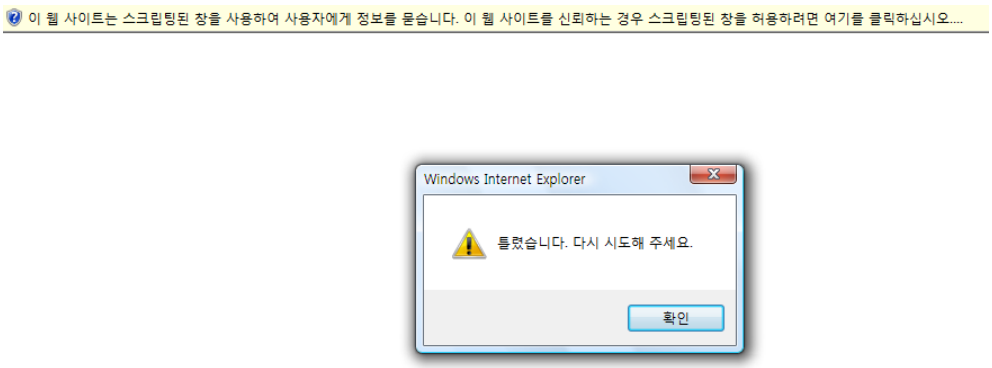
micsland.com 워게임 레벨1-plus 풀이

개요

각 레벨은 마지막에 plus 문제가 존재합니다. Plus 문제는 레벨에서 나왔던 문제들의 종합으로 그 레벨에서 난이도가 가장 높은 문제입니다.

내용

레벨1의 plus 문제의 소스를 살펴봅시다. 레벨1은 자바스크립트에 관한 레벨이었습니다. 자바스크립트 차단이 되어있는 상태에서 아래 페이지에 접속하면 경고메세지가 출력되고 바로 전 페이지로 강제이동 됩니다.



[그림1-4-1. 잘못된 값 입력시 나타나는 경고메세지]

페이지를 강제로 이동시키니까 소스를 볼 수 없습니다. 하지만 아무리 이렇게 소스보기를 막아놓았다고 해도 소스를 볼 방법은 얼마든지 있습니다. 가장 확실한 방법은 그 문제 페이지를 다운받는 방법인데, 어떻게 하는지 살펴봅시다.

1+단계

약간 생소한 경우일 수 있습니다.

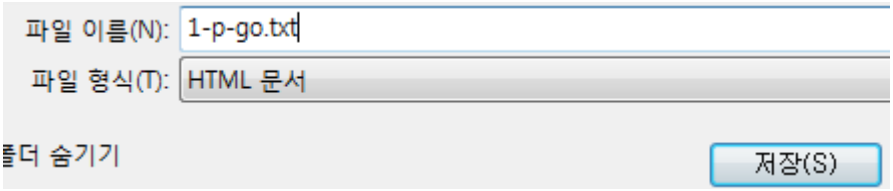
푸는 방식은 기존과 동일하지만, 상당히 난해합니다.

들어가기

결정적 힌트 보기

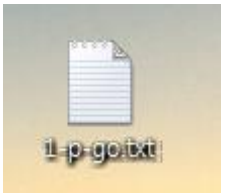
[그림1-4-2. 1+ 문제를 풀기 전에 나오는 화면]

들어가기를 클릭하면 문제 페이지로 접속됩니다. 이 들어가기에 마우스 오른쪽 버튼을 누르고 다른이름으로 저장을 합니다.



[그림1-4-3. 문제 파일을 텍스트문서로 저장하는 화면]

저장을 할 때에는 확장자를 txt로 하여 간편하게 파일을 열여볼 수 있도록 합니다.



[그림1-4-4. 텍스트문서로 저장된 문제파일]

위 그림처럼 문제 페이지가 텍스트파일로 바탕화면에 저장되고, 저 파일을 열람함으로써 소스를 볼 수 있게 됩니다.

또 다른 방법은,문제 페이지 주소 앞에 위처럼 view-source: 를 붙이는 겁니다.

[그림1-4-5. 문제 페이지의 소스를 보기 위해 view-source를 붙이는 화면]

여러가지 방법이 있지만 여기서는 문제 1-2를 풀 때 사용했던 새 창을 띄우는 방법을 써서 문제의 소스를 보겠습니다.

```
<script
language='JScript.Encode'>#@~^MwEAAA==@#@&k[,',J/1Db2Yri@#@&ah~x,J+
        ^W9+Jp@#@&k9{bPxP2.K:2YvEqG는gESrJ#I@#@&wC/kAKD[,',wDK:aO`r비
번은gE~rJbI@#@&k6ck9{k,xxPbN,['P2Ck/AWM[P{'~2S#P[G1Eh+  Och.bY口J축하합
니다eP8Q단계를P해결하셨습니다c@!8D@*FQ단계~해결코드는,ymoAGN口버입니다
c@!8D@*@!mP4DnW{BqO6rxmV                                    24wB@*q단계,끝
"@!zm@*r#I@#@&口V/口PP~I^+.YvJ틀렸습니다                    ~다시,시도해P주세요
Rrbi,tkkOGDHR(I^3cbpPN@#@&jkQAAA==^#~@
</script>
```

자바스크립트는 맞긴 맞는데 글씨가 이상하게 깨져서 나옵니다. Script 부분에 있는 Jscript.Encode를 보면 자바스크립트 인코딩을 사용했다는 것을 알 수 있습니다.

자바스크립트 디코더를 사용하기 전에, 스크립트에 XMP 태그를 붙여서 소스가 나타나는지 확인해봅시다.

Xmp 태그는 소스코드를 그대로 보여주는 태그입니다.

#@~^MwEAAA==@#@&k[,',J/1Db2Yri@#@&ah~x,J+ ^W9+Jp@#@&k9{bPxP2.K:2YvEqG는gESrJ#I@#@&wC/kAKD[,',wDK:aO`r비번은gE~rJbI@#@&k6ck9{k,xxPbN,

[그림1-4-6. xmp태그를 써서 디코딩된 값을 볼려고 하는 화면]

음..이건 안돼는군요.

한 가지 작업을 더 해봅시다. 레벨1 plus 문제 페이지에 접속하면 아이디를 묻는 프롬포트 하나, 패스워드를 묻는 프롬포트 창이 하나 뜹니다. 그리고 아이디와 패스워드를 각각 입력하면 패스워드가 틀렸다는 경고메세지를 띄운 후, 바로 전 페이지로 강제이동 시킵니다. 사

용자가 아이디와 패스워드를 입력하면 if문으로 입력한 아이디와 패스워드가 맞는지 비교하는 것 같습니다. 이것을 예상하여 코드를 적어보겠습니다.

```
id=1;
pw=1;
input_id=prompt("아이디를 입력하세요");
input_pw=prompt("패스워드를 입력하세요");
if(input_id==id && input_pw==pw) { alert("패스워드는 ~~ 입니다.") }
else { alert("땡!"); }
```

이런식으로 id,pw 변수에 아이디와 패스워드 값을 입력해놓고 사용자가 입력한 값을 input id,inputpw 변수에 넣은 다음 이 두가지를 비교하여 일치하면 패스워드를 출력해주는 식으로 구성이 되어있을 것입니다. id,pw 변수는 가장 일반적인 변수명을 추측한 것입니다.
그럼 실제 문제에서도 이 방식을 사용했는지 확인해봅시다.

문제 스크립트 밑에 아래와 같은 코드를 입력했습니다.

```
<script language='JScript.Encode'>#@~^MwEAAA==@#@&k[,',J/1Db2Yri@#@&ah~x,J+
^W9+Jp@#@&k9{bPxP2.K:2YvEqG는gESrJ#I@#@&wC/kAKD[,',wDK:aO`r비번은gE~rJbI@#@&k6ck9
{k,xxPbN,['P2Ck/AWM[P{'~2S#P[G1Eh+ Och.bY`J축하합니다eP8Q단계를P해결하셨습니
다c@!8D@*FQ단계~해결코드는,ymoAGNH입니다c@!8D@*@!mP4DnW{BqO6rxmV
24wB@*q단계,끝"@!zm@*r#I@#@&V/PP~I^+.YvJ를렸습니다
~다시,시도해P주세요Rrbi,tkkOGDHR(I^3cbpPN@#@&jkQAAA==^#~@

document.write(id+"<br>"+pw);

</script>
```

[그림1-4-7. 아이디와 패스워드 변수명을 추측하는 화면]

만약에 추측한 id와 pw 변수가 맞다면 그 변수안에 있는 내용이 출력될것입니다.

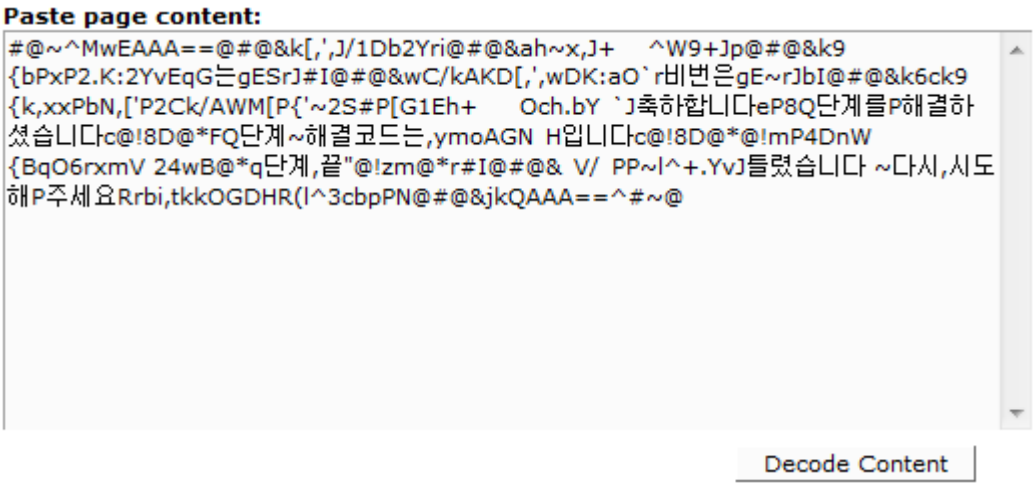
script
encode

[그림1-4-8. 화면에 표시된 아이디와 패스워드]

위와 같이 출력되었습니다. 예상대로 id,pw 변수를 사용했군요.
이런 간단한 추측으로 아이디와 패스워드를 알아낼수도 있습니다.
이번에는 간단하게 푸는방법으로 자바스크립트 디코더를 사용해보겠습니다.

url : http://www.greymagic.com/security/tools/decoder/decoder.asp

이곳으로 이동하여 인코딩 되어있는 자바스크립트 코드를 붙여넣습니다.



[그림1-4-9. 자바스크립트 디코더를 사용 장면]

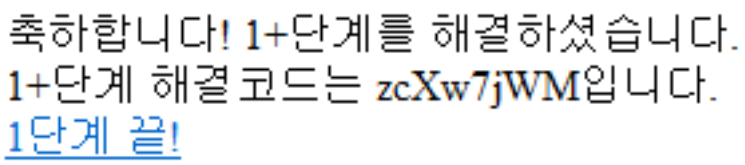
디코딩 버튼을 누르면 아래와 같이 출력됩니다.

GreyMagic Online Script Decoder Results

```
id = "script";
pw = "encode";
id_i = prompt("ID3VhQ^NOuNu,O/VY
FH iBor#[= p]TT#t{"3)hb^k}qIVh}3xhq/)q9kfbqN","");
iO(sJ7i =J )j9#= 3HsCB4Gd == pw) #oN'went(wzA]W{"/VQEOEdu3`YhOcxq/VQbOk+
else { alert("/VQkJQpqI)bv9OfC3`bD}hqC3VbQ          x1q/VhY^DOC2 3`hQ^DfqI)b
```

[그림1-4-10. 디코딩 결과]

한글은 깨져서 알아볼 수 없지만 id,pw 변수에 저장되있는 값을 확인할 수 있습니다.
이제 아이디와 패스워드를 입력해봅시다.



[그림1-4-11. 레벨1 plus 문제의 해결코드]

해결코드가 성공적으로 출력되었습니다.

결론

자바스크립트의 내용을 인코딩하는 방법이 존재하지만, 디코딩 하는 방법도 존재하므로 자바스크립트 인코딩 역시 안전하다고 할 수 없습니다.

micsland.com 워게임 레벨2-1 풀이

개요

micsland.com 워게임은 php 코드에 대한 이해도를 평가하는 문제로, php 코드를 보여준 후 그 코드에서 원하는 값을 입력하면 해결코드를 얻을 수 있습니다. 레벨은 1에서 6까지 있으며 각 레벨당 4개 정도의 문제가 있습니다. 각 레벨의 모든 문제를 마칠 경우 plus 문제가 주어지는데, 이 문제는 그 레벨에서 나온 문제의 종합으로 난이도가 가장 높습니다.

레벨2부터는 본격적으로 php코드를 보여주고, 코드에서 원하는 조건(if문)을 만족시키는게 목적입니다. 레벨2는 GET method를 많이 사용합니다. http에서 사용하는 전송 방식은 크게 GET,POST가 있는데 GET은 전송되는 데이터가 URL에 표시되며 글자 수의 제한이 있고 전송되는 데이터가 쉽게 변조될 수 있습니다. 쓸모없는 것처럼 보이지만 전송되는 데이터를 쉽게 알아볼 수 있으므로 게시판,방명록에서 많이 사용됩니다.

내용

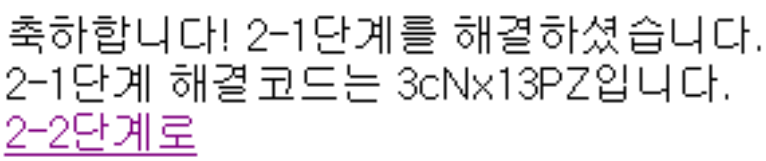
일단 문제의 코드를 살펴봅시다.

```
<? if($_GET[data] == "123") { ?>

축하합니다! 2-1단계를 해결하셨습니다.<br>
2-1단계 해결코드는 ########입니다.<br>
<a href='2-2.php'>2-2단계로</a>

<? } else { ?>

자~어서 풀어보세요~
```

맨 첫번째 줄에 있는 if문을 살펴보면, $_GET[data] == "123" 이라는 부분이 있습니다. 이것은 GET method로 전송된 data 라는 이름의 변수값이 123 이면 이라는 뜻이 됩니다. http의 GET method는 전송되는 값이 url에 표시되며 길이가 제한되어있고 수정이 post에 비해 간편합니다. 전송되는 값이 url에 표시되므로 간단하게 url을 수정함으로써 전송되는 값을 추가하거나 수정할 수 있습니다.

http://micsland.com/hg/level_2/2-1-go.php?data=123

이렇게 data라는 변수를 만들고 그 값을 123으로 해주면 문제의 if문을 만족하게 되고 해결코드가 출력됩니다.



[그림2-1-1. 해결코드]

결론

GET method는 url에 전송되는 값이 표시되어서 사용자에게 현재 위치(예를들어 게시판의 경우에는 어느 게시판의 몇 번째 글을 읽고있는지)를 쉽게 확인시켜 줄 수 있지만,그만큼 전송되는 데이터가 쉽게 변조될 수 있습니다. 따라서 위 php코드처럼 간단한 방법을 통한 인증은 사용하면 안됩니다. 전송받은 변수를 체크하지 않는다고 가정하고 잘못된 예를 들면

http://test.com/member/modify.php?id=guest

위와 같은 회원정보 수정 페이지에서 id 변수의 값을 admin으로 수정하면 admin의 회원정보를 열람,수정 할 수 있게 됩니다. 그러므로 이 문제처럼 get,post method를 너무 믿으면 안되며 회원정보수정같이 중요한 페이지는 세션을 이용한 인증을 사용해야 될 것입니다.

micsland.com 워게임 레벨2-2 풀이

개요

레벨2-1과 비슷하게 Get method로 변수를 추가해주면 풀 수 있는 문제입니다.
Php에서 어떤 것을 변수로 인식하는지 알고있다면 수월하게 문제를 풀 수 있습니다.

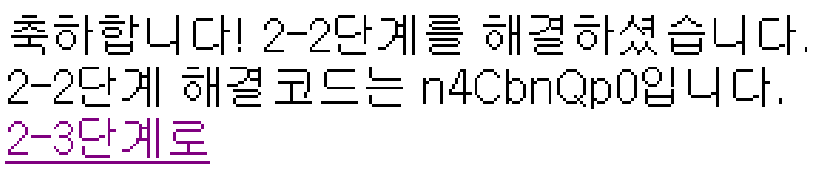
내용

일단 문제의 코드를 살펴봅시다.

```
<? if($super and $man) { ?>
축하합니다! 2-2단계를 해결하셨습니다.<br>
2-2단계 해결코드는 ########입니다.<br>
<a href='2-3.php'>2-3단계로</a>
<? } else { ?>
여기서 기죽으면 안되지!
<? } ?>
<? } ?>
```

문제의 핵심이 되는 첫 번째 줄을 살펴보면, if문으로 $super와 $man 변수를 검사하는 것을 알 수 있습니다. 지금까지 본 if문은 변수명=값 으로 이루어져 있었는데, 이 문제는 변수명만 존재합니다. 이런 경우에는 해당 변수가 존재하는지 아닌지를 검사한다는 뜻입니다.
이런경우에는 get,post,cookie 중 어느 것을 사용하여도 상관없습니다. 간편하게 풀기 위해 get method를 사용합시다.
if문으로 super,man 변수를 검사하지만 그 변수에 어느 값이 있는지는 검사하지 않습니다. 그러므로 super와 man 변수의 값은 아무렇게나 지정해줘도 됩니다.

http://micsland.com/hg/level_2/2-2-go.php?super=1&man=2



[그림2-2-1. 해결코드]

결론

2-2 문제는 변수가 있는지 없는지만 확인하는 문제였습니다.

micsland.com 워게임 레벨2-3 풀이

개요

레벨2-3문제는 변수와 일반 문자에 대한 이해가 필요한 문제입니다.

내용

일단 문제의 코드를 살펴봅시다.

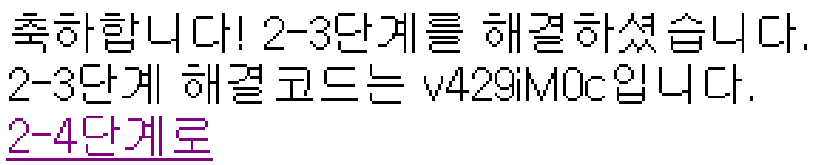
```
<? if($catch == $me and $catch == "themics") { ?>

축하합니다! 2-3단계를 해결하셨습니다.<br>
2-3단계 해결코드는 ########입니다.<br>
<a href='2-4.php'>2-4단계로</a>

<? } else { ?>

Fighting!
```

if문은 한 개가 있습니다. 하나하나 정리해보면

catch라는 이름의 변수가 me와 같아야 된다.
그리고
catch라는 이름의 변수의 값이 themics가 되야한다.

if문 두 번째 비교문에서 catch의 값이 themics가 되야된다고 했으므로 catch에는 themics가 들어가야됩니다. 그리고 catch는 me와 같아야 된다고 했으므로 me의 값 역시 themics가 되야됩니다.

http://micsland.com/hg/level_2/2-3-go.php?catch=themics&me=themics



[그림2-3-1. 해결코드]

결론

2-3 문제는 if문에 두 개의 조건이 있었고 논리연산자가 and 였으므로 두 개의 조건을 모두 만족해야 참이 되는 문제였습니다. If에서 비교하는 대상이 많아질수록 코드를 꼼꼼히 관찰해야됩니다.

micsland.com 워게임 레벨2-4 풀이

개요

레벨2의 마지막 문제입니다. 마지막 문제답게 여러가지 변수를 비교합니다. 비교하는 값이 많지만 심하게 꼬여있는 문제는 아니므로 코드를 꼼꼼히 살펴보면 쉽게 풀 수 있습니다.

내용

일단 문제의 코드를 살펴봅시다.

```
<? if(($think == "reverse" and $running != "stop") and ($_GET[stop] > $go or $_P
OST[move] != 0) and $_GET[running] == $_POST[move]) { ?>

축하합니다! 2-4단계를 해결하셨습니다.<br>
2-4단계 해결코드는 ########입니다.<br>
<a href='2-p.php'>2+단계로</a><br>
<br><a href='2-final.php'>2단계 바로 종료</a>

<? } else { ?>
```

if문 처음부터 하나하나 살펴봅시다.

think 변수의 값이 reverse 이고,
Running 변수의 값이 stop 이 아니고, (!=은 논리연산자 Not을 의미합니다)

GET method로 전송된 stop 변수가 go 변수보다 크거나
POST method로 전송된 move 변수가 0이 아니고,

GET method로 전송된 running 변수가 POST method로 전송된 move 변수와 같으면
조건을 만족하게 됩니다.

일단 문제에서 처음 나온 것들을 살펴봅시다.

!= 는 논리연산자로 Not을 의미하므로 일치하지 않아야 참이 됩니다. 그리고 or은 말 그대로 또는 이라는 뜻을 갖고있으며 둘 중 하나만 참이 되면 참이 됩니다.
$_POST[변수명] 는 $_GET 처럼 POST method로 전송된 변수명이 있는지 확인하는 것으로, http POST method에 대해 알아야됩니다. POST 전송방식은 GET과 다르게 전송되는 값이 표시되지 않으며 길이에 제한이 없이 데이터를 전송할 수 있습니다.(get은 url에 전송되는 데이터가 표시되기 때문에 글자제한이 있습니다) 아무래도 직접적으로 url에 데이터가 나타나지 않으므로 get보다는 안전하며, POST 방식을 사용할려면 아래와 같은 폼을 만들어야 됩니다.

```
<form name=test action=login.asp method=POST>
<input type=text name=id>
<input type=submit>
</form>
```

action은 폼에서 입력한 값을 전송할 url을 입력하는 것이고 method는 전송방식을 결정하는 것으로 get이나 post 중 하나를 선택합니다.
Input 박스를 보면 name이라는 것이 있는데, 이 name이 전송되는 변수명이 됩니다. Name=id 이므로 여기에 입력한 값은 id라는 변수명으로 전송이 됩니다. 아래쪽에 있는 submit은 폼에 입력한 값들을 action에 쓰여있는 곳으로 전송시켜주는 역할을 합니다.

자 이제 이것들이 무슨 역할을 하는것인지 알았으니 문제를 다시 한 번 살펴봅시다.

```
<? if(($think == "reverse" and $running != "stop") and ($_GET[stop] > $go or $_POST[move]
!= 0) and $_GET[running] == $_POST[move]) { ?>
```

변수명 앞에 그냥 $ 표시가 붙어있으면 어느 방식을 사용해도 상관이 없습니다.

Think는 reverse가 되야되고 running은 stop이 아니면 됩니다. 그러면 첫 번째 비교문을 통과하게 됩니다.

그리고

GET으로 전송된 stop 변수의 값은 go변수보다 커야됩니다.
또는 POST로 전송된 move 변수의 값이 0이 아니어야 됩니다.

여기서 논리연산자가 or이라는것을 알 수 있습니다. ($_GET[stop > $go **or** $_POST ...]
즉 한 개의 조건만 일치해도 참이 됩니다. Get,post method를 섞어서 전송하는건 아직까지 사용할 필요가 없어보이므로 $_POST[move] != 0 은 건너뛰겠습니다. 이 조건을 만족시키지 않아도 앞에있는 $_GET[stop] > $go 만 일치하면 or 덕분에 참이 됩니다.

그리고

GET으로 전송된 running변수가 POST로 전송된 move와 같아야됩니다.

이제 이 조건들을 조합하여 입력해봅시다.

http://micsland.com/hg/level_2/2-4-go.php?think=reverse&stop=1

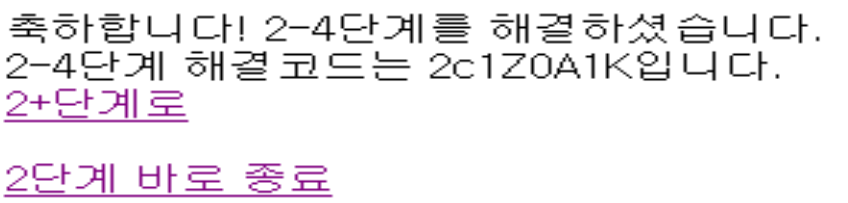
```
<? if(($think == "reverse" and $running != "stop") and ($_GET[stop] > $go or $_POST[move] != 0) and $_GET[running] == $_POST[move]) { ?>
```

think=reverse, running은 stop만 아니면 되므로 그냥 입력하지 않았습니다. 아무것도 입력하지 않았으므로 공백이 됩니다. 어쨌든 이걸로 첫 번째 비교문은 만족하게 됩니다.

GET으로 전송된 stop 변수는 go변수보다 커야된다. Stop 변수의 값을 1로 해주고 go는 입력하지 않았으므로 0 이 됩니다. 0은 1보다 작으므로 이 조건을 만족하게 됩니다.

or 덕분에 $_post[move] != 0 은 만족시키지 않아도 됩니다.

마지막으로 get으로 전송된 running의 값이 post로 전송된 move와 값이 같아야 된다는 조건이 있습니다. Running변수는 처음부터 만들어주지 않아서 공백이고, move 변수도 만들어주지 않아서 공백이므로 둘 다 공백이 됩니다. 그러니까 둘 다 같은 값이므로 참이 되어 조건을 만족하게 됩니다.
그래서 모든 조건을 만족하게 되고, 해결코드가 출력됩니다.



[그림2-4-1. 해결코드]

결론

if문을 자세히 관찰하고 논리연산자가 무슨역할을 하는지 등을 알고있으면 풀 수 있는 문제였습니다. Form에 대해 보충설명이 필요한 분들은 링크를 참조하세요.

micsland.com 워게임 레벨2-plus 풀이

개요

각 레벨은 마지막에 plus 문제가 존재합니다. 이 문제는 그 레벨에서 나왔던 문제들의 종합으로 그 레벨에서 난이도가 가장 높은 문제입니다.

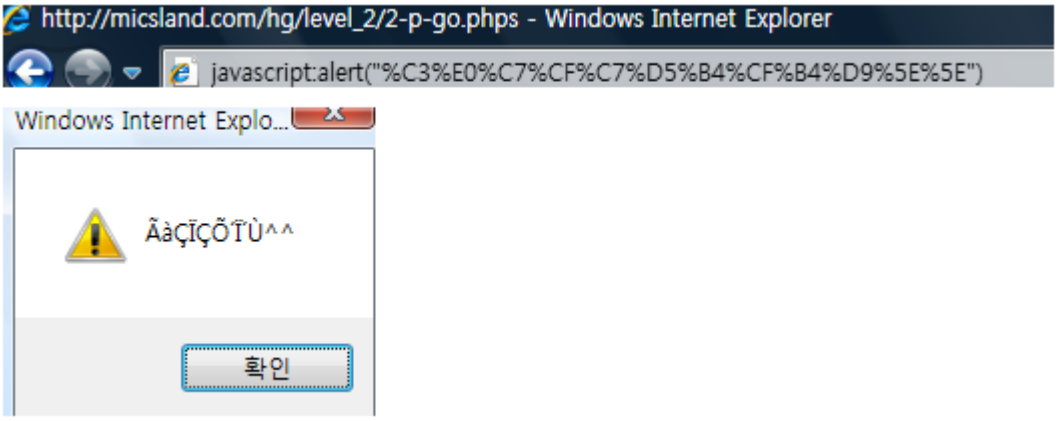
레벨2의 plus문제는 레벨1의 plus문제와 비슷하게 인코딩된 값을 디코딩해야됩니다. 인코딩을 처음 접하는 분들은 난해하실수도 있지만 의외로 간단하게 풀 수 있는 문제입니다.

내용

생각보다 소스가 간단합니다. 한 번 살펴봅시다.

```
<?
$wow = ####($_GET[wow]);        //어떤 함수로 $_GET[wow]를 가공합니다. 뭘까요?
if($wow == "%C3%E0%C7%CF%C7%D5%B4%CF%B4%D9%5E%5E") {
?>

축하합니다! 2+단계를 해결하셨습니다.<br>
2+단계 해결코드는 ########입니다.<br>
<a href='2-final.php'>2단계 끝!</a>

<? } else { ?>

자자, 이것만 끝내면 됩니다!

<? } ?>
```

wow 함수에 무언가 값을 넣습니다. 이 wow의 값이 어떤 인코딩된 글자와 같으면 해결코드를 보여줍니다. URL인코딩이 된 것 같습니다. 이 인코딩된 무언가를 찾기 위해 주소창에 아래와 같이 입력해봅시다.



[그림2-5-1. 정답을 디코딩 했지만 깨져서 나온 화면]

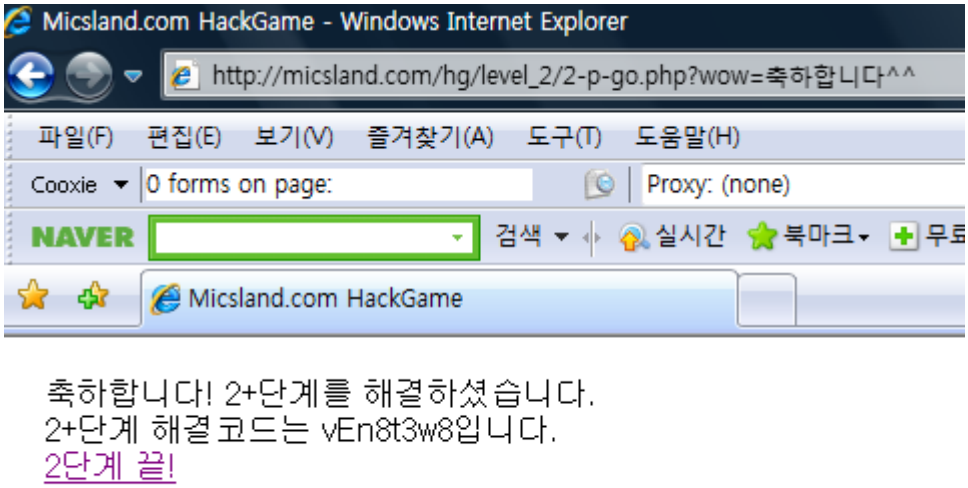
오잉. 뭔가가 나오긴 나오는데 깨져서 나옵니다. 아마 한글이 들어갔나봅니다.
한글을 보기 위해 아래와 같이 php의 urldecode 함수를 사용하였습니다.

```
<? $a=urldecode("%C3%E0%C7%CF%C7%D5%B4%CF%B4%D9%5E%5E"); echo($a); ?>
```

축하합니다^^

[그림2-5-2. php의 urldecode함수 실행결과]

확인해보니 축하합니다^^ 가 출력되었습니다.
그럼 알아낸 패스워드를 입력해보겠습니다.



[그림2-5-3. 해결코드]

해결코드가 출력되었습니다.

결론

URL디코딩에 관한 문제였습니다.

링크

http://kr2.php.net/manual/en/function.urldecode.php

micsland.com 워게임 레벨3-1 풀이

개요

micsland.com 워게임은 php 코드에 대한 이해도를 평가하는 문제로, php 코드를 보여준 후 그 코드에서 원하는 값을 입력하면 해결코드를 얻을 수 있습니다. 레벨은 1에서 6까지 있으며 각 레벨당 4개 정도의 문제가 있습니다. 각 레벨의 모든 문제를 마칠 경우 plus 문제가 주어지는데, 이 문제는 그 레벨에서 나온 문제의 종합으로 난이도가 가장 높습니다.

레벨2가 get방식으로 전송되는 변수들을 만드는 문제였다면 레벨3은 post방식으로 변수들을 전송해야 되는 문제입니다. 직접 폼을 만들어서 문제를 풀어야되므로 귀찮을수도 있지만 폼을 수정하는 것은 웹해킹에서 꼭 알아야되는 것이므로 연습하는 기분으로 풀어봅시다.

내용

일단 문제의 코드를 살펴봅시다.

```
<? if($_POST[postdata] == "form") { ?>

축하합니다! 3-1단계를 해결하셨습니다.<br>
3-1단계 해결코드는 ########입니다.<br>
<a href='3-2.php'>3-2단계로</a>

<? } else { ?>

설마 직접 폼을 짜시는건 아니겠죠?
```

POST method로 전송되는 postdata 변수의 값이 form이면 문제를 통과하게 됩니다.
폼을 작성하는 방법은 html 파일을 하나 만들고,아래와 같이 입력해주면 됩니다.

```
<form method=post action=위치>
<input type=text name=변수명 value=값>
<input type=submit>
</form>
```

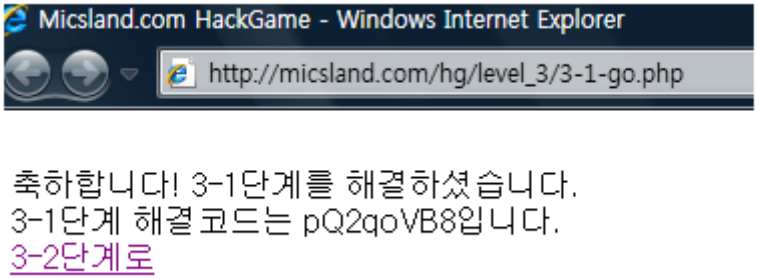
action에는 목표 url을 쓰는데,http://부터 정확히 입력해야됩니다. 그리고 name=변수명 에는 문제에서 요구하는 변수명을 입력해주며 value=값 에는 문제에서 요구하는 값을 입력해주면 됩니다. 이 문제에 맞춰서 폼을 작성해보았습니다.

```
<form method=post action=http://micsland.com/hg/level_3/3-1-go.php>
<input type=text name=postdata value=form>
<input type=submit>
</form>
```

이 파일을 html로 저장하고 열어보면 아래와 같은 화면이 보입니다.

[그림3-1-1. post method를 사용하기 위한 준비]

쿼리 전송 버튼을 눌러봅시다.



[그림3-1-2. 해결코드]

여기서 post 방식의 특징을 볼 수 있습니다. url에는 아무것도 표시되지 않습니다.
어쨌든 이 방법으로 폼을 작성하고 전송하면 간단하게 문제를 풀 수 있습니다.

결론

히든폼의 값만 믿는 웹애플이케이션은 간단한 폼 조작에 속수무책으로 당할 수밖에 없습니다. 예를들어 취약점이 있는 회원관리 수정 페이지의 소스를 보면

<form name=member action=member_modify_ok.asp method=post>
<input type=hidden name=id value=guest>
변경할 패스워드 : <input type=password name=pw value=1234>
</form>

이런식으로 히든 폼을 만들어놓고 이 폼에서 전송된 값만 믿고 회원정보를 수정하는 경우에는 간단한 조작으로 다른 사용자의 패스워드를 변경시킬 수 있습니다.
이런 폼 조작을 통한 회원정보 수정 등을 막기 위해서는 회원정보 수정 페이지는
http 헤더의 referer 값을 검사하고 히든폼 대신 세션을 사용해야 될 것입니다.

micsland.com 워게임 레벨3-2 풀이

개요

3-2 문제는 get과 post 방식 두 가지를 모두 사용해야 풀 수 있는 문제입니다.
이 문제를 풀면서 get,post method 두 가지를 한번에 사용하는 방법을 알아봅시다.

내용

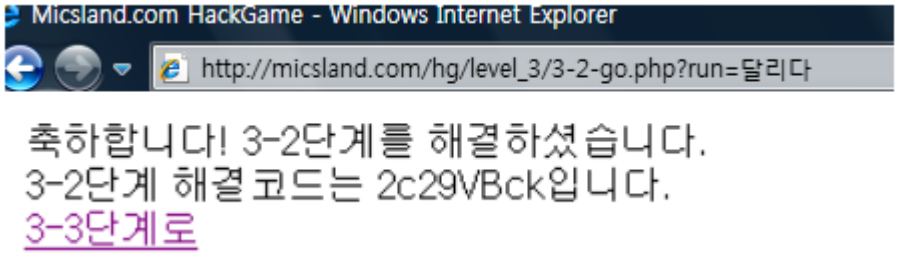
일단 문제의 코드를 살펴봅시다.

```
<? if($_POST[try] == "시도하다" and $_GET[run] == "달리다") { ?>

축하합니다! 3-2단계를 해결하셨습니다.<br>
3-2단계 해결코드는 ########입니다.<br>
<a href='3-3.php'>3-3단계로</a>

<? } else { ?>

이왕 오신 김에 제 홈페이지도 좀 들려주세요;ㅁ;/
```

if문은 간단합니다. POST로 전송된 try변수의 값이 시도하다 ,
GET으로 전송된 run 변수의 값이 달리다 가 되면 해결코드를 보여줍니다.
그런데 문제를 풀려고 하면 문제가 생깁니다. 어떻게 get과 post를 같이 사용할까요? 방법은 action에 있습니다. 폼을 만들 때, action에 직접 변수명을 넣어줌으로써 get방식을 사용하게 하면 됩니다. 즉 아래와 같이 폼을 짜면 됩니다.

```
<form action=http://micsland.com/hg/level_3/3-2-go.php?run=달리다>
<input type=text name=try value=시도하다>
<input type=submit>
</form>
```

이런식으로 action에 get로 전송될 변수를 넣어주고 폼에는 post로 전송될 변수를 넣어주면 두 가지 method를 동시에 전송할 수 있습니다.



[그림3-2-1. 해결코드]

결론

폼 action을 수정하면 get과 post 둘 다 사용하여 값을 전송할 수 있습니다.

micsland.com 워게임 레벨3-3 풀이

개요

php.ini의 gpc_order 에 대한 이해가 필요한 문제입니다.
설명을 보며 gpc_order에 대해 알아봅시다.

내용

문제코드를 살펴보기에 앞서 php.ini의 gpc_order에 대해 알아봅시다.
Php.ini의 gpc_order는 각각 **G**et,**P**ost,**C**ookie 의 약자로 이 세 가지의 우선순위를 정하는 옵션입니다. 기본값은 GPC이며 만약에 GP로 설정되어 있다면 Cookie가 없으므로 쿠키는 무시되고,동일한 이름의 Post 변수는 Get 변수로 덮어쓰이게 됩니다. 앞에 있는 애가 뒤에있는 동일한 이름의 변수를 먹는다고 생각하면 간단합니다.

문제 페이지에서 Gpc_order의 설정을 바꾸지 않았다는 힌트를 줬으므로 기본값은 gpc가 되고,동일한 이름의 변수가 있을 경우에는 Get이 우선이 됩니다.

이제 문제의 코드를 살펴봅시다.

```
<? if(($_POST[php4] == "zend" or $_GET[php4] == "mysql") and $php4 == "mysql
" and $php4 == $_GET[php4]) { ?>
축하합니다! 3-3단계를 해결하셨습니다.<br>
3-3단계 해결코드는 ########입니다.<br>
<a href='3-p.php'>3+단계로</a><br>
<a href='3-final.php'>3단계 바로 종료</a>

<? } else { ?>

홈페이지에 글 안쓰고 가면 데이트신청할껍니다;

<? } ?>
```

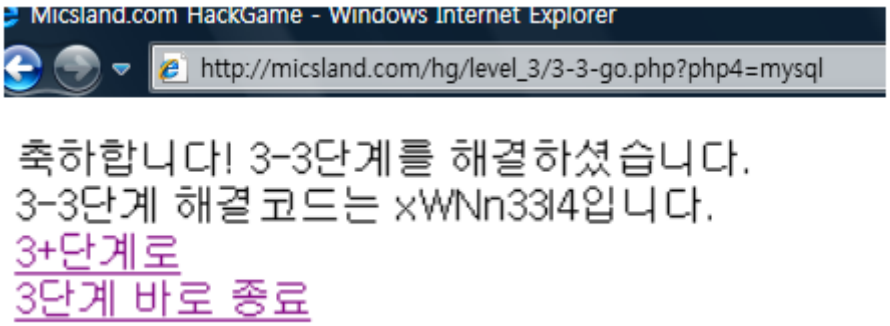
```
<? if(($_POST[php4] == "zend" or $_GET[php4] == "mysql") and $php4 == "mysql" and $ph
p4 == $_GET[php4]) { ?>
```

POST로 전송된 php4 변수값이 zend거나 GET으로 전송된 php4의 변수값이 mysql 이고 Php4 변수값이 mysql 이고 php4 변수가 get으로 전송된 php4 변수와 같으면

조건이 만족합니다. 간단해보이지만 문제는 변수명에 모두 php4 로 동일하다는데 있습니다. 변수명이 동일한 경우에는 gpc_order 설정에 따르는데 기본값이므로 GPC, Get이 가장 높은 우선순위를 갖게됩니다. 동일한 이름의 POST로 전송되는 변수는 무시하고 GET 변수만 보겠습니다.

GET으로 전송되는 php4 변수의 값에 mysql 을 입력해주면 $php4 역시 mysql이 됩니다. (get이 가장 먼저이므로)

폼을 사용할 필요도 없이 GET 방식으로 php4 변수를 만들어주고 mysql이라는 값을 넣어주면 간단하게 문제를 풀 수 있습니다.



[그림3-3-1. 해결코드]

결론

php.ini의 gpc_order에 대한 이해가 필요한 문제였습니다.
Php 코딩을 할 때 GET은 $_GET[변수명], POST는 $_POST[변수명] 으로 전달된 값을 가져옵니다. 하지만 몇몇 프로그래머들은 이렇게 일일이 입력해주는게 귀찮아서 단순하게 $변수명으로 전달된 변수들을 가져오는데,이건 매우 좋지못한 습관입니다. Php 옵션중에 register_global 옵션이 켜있을 경우에는 사용자로부터 전달받은 변수를 전역변수로 등록하기 때문에 php코드 안에서 미리 선언된 변수들을 바꿀 수 있는 문제가 생기게됩니다.
예를들어 $test="123"; 이라는 변수가 미리 선언되어 있을 경우 POST method로 전송된 test 변수가 미리 선언되있던 test변수의 값을 바꿀 수 있게됩니다. 그러므로 귀찮더라도
$변수명 으로 전달된 값을 받아오는건 자제해야됩니다.

제로보드 같은 공개 게시판은 register_global옵션이 off로 되어있을 때 이것을 임의로 on으로 바꾼것과 같은 효과를 주기 위해 extract함수를 사용했는데,이 함수로 인해 lib.php에 취약점이 생기기도 했습니다.

micsland.com 워게임 레벨3-plus 풀이

개요

각 레벨은 마지막에 plus 문제가 존재합니다. 이 문제는 그 레벨에서 나왔던 문제들의 종합으로 그 레벨에서 난이도가 가장 높은 문제입니다.

레벨3의 plus문제는 if문에서 많은 것들을 비교하므로 if문을 자세히 봐야됩니다.

내용

레벨3은 폼을 작성하는게 많아서 귀찮았던 레벨이었습니다.
이제 레벨3의 마지막인 plus 문제를 살펴봅시다.

```
<?
if($_POST[trap] or $_GET[trap] == "mine") {
    echo("이건 안돼요-_-;");
    exit;
}
if(!$_GET[trap]) $trap = "mine";
if((!$_POST[real] or $_GET[trap] == "lucky" or $_POST[real] == $trap) and ($real !
= $_GET[real] or $_GET[trap]) and (!$_GET[real] or !$_POST[real])) {
?>

축하합니다! 3+단계를 해결하셨습니다.<br>
3+단계 해결코드는 ########입니다.<br>
<a href='3-final.php'>3단계 끝!</a>

<? } else { ?>

아리송~다리송~

<? } ?>
```

If문으로 비교하는 대상이 다른 문제와는 다르게 많이 있습니다.
먼저 첫 번째 if문을 살펴봅시다.

```
if($_POST[trap] or $_GET[trap] == "mine") {
    echo("이건 안돼요-_-;");
    exit;
}
```

POST방식으로 전송되는 trap 변수가 있거나, GET방식으로 전송되는 trap변수의 값이 mine 이면 이건 안돼요를 출력하며 php를 종료시킵니다.
그럼 다음 if문을 살펴봅시다.

```
if(!$_GET[trap]) $trap = "mine";
```

GET방식으로 전송된 trap 변수가 **없으면** trap 함수에 mine 이라는 값을 넣습니다.

```
if((!$_POST[real] or $_GET[trap] == "lucky" or $_POST[real] == $trap) and ($real != $_GET[real
] or $_GET[trap]) and (!$_GET[real] or !$_POST[real])) {
```

POST로 전송된 real 변수가 **없거나** GET으로 전송된 trap변수의 값이 lucky 거나
POST로 전송된 real변수의 값이 trap변수와 같고,

Real변수의 값이 GET으로 전송된 real변수와 같지 않거나 GET으로 전송된 trap변수가 있거고,

GET으로 전송된 real 함수가 없거나 POST로 전송된 real 변수가 없으면 통과합니다.
헥헥..

if문이 많으니 하나하나 헤처가며 변수를 만들어봅시다.

```
<?
if($_POST[trap] or $_GET[trap] == "mine") {
    echo("이건 안돼요-_-;");
    exit;
}
if(!$_GET[trap]) $trap = "mine";
if((!$_POST[real] or $_GET[trap] == "lucky" or $_POST[real] == $trap) and ($real != $_GET[real
] or $_GET[trap]) and (!$_GET[real] or !$_POST[real])) {
?>
```

처음 if문을 피하기 위해 POST trap변수가 있으면 안됩니다. 그리고 GET trap변수의 값이 mine 되서도 안됩니다.
두 번째 if문에서 GET trap함수가 없으면 trap함수의 값을 mine으로 만듦니다.
정리해보면 아무것도 하지 않고 페이지에 접속만 하면 trap="mine" 이라는 변수가 생성됩니다.

세 번째 if문이 가장 복잡합니다. 하나하나 나눠서 보겠습니다.

```
(!$_POST[real] or $_GET[trap] == "lucky" or $_POST[real] == $trap)
```

논리연산자가 or 이므로 세 가지 중에 하나만 만족하면 참이 됩니다. POST real 변수가 없거나 GET trap 변수값이 lucky거나 POST real 변수값이 trap변수와 같으면 입니다.
여기서 첫 번째 POST real이 없으면이 가장 간편하므로 이것을 선택하겠습니다. 조건이 POST로 전송되는 real 변수가 없으면 이므로 아무것도 하지 않아도 통과할 수 있습니다.
그럼 다음 비교문을 보겠습니다.

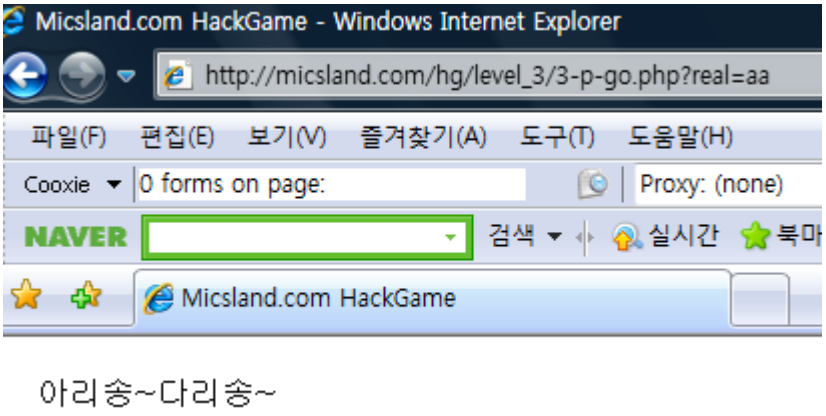
```
($real != $_GET[real] or $_GET[trap])
```

어느 방식으로 전송되어도 상관없는 real 변수의 값은 GET으로 전송된 real변수와 같으면 안됩니다. 또는 GET으로 전송된 trap변수가 있으면 됩니다. 여기서도 간편해보이는 첫 번째 조건, real변수의 값과 GET real변수의 값을 다르게 하는 것을 선택하겠습니다. 둘 다 존재하지 않는 변수이므로 처음에 공백을 갖게됩니다. 그럼 둘 다 같은값이 되므로 다르게 해주기 위해 GET real의 값을 아무거나 입력하여 전송해줍니다.

```
(!$_GET[real] or !$_POST[real])
```

드디어 마지막 비교문. GET으로 전송된 real 변수가 없거나 POST real 변수가 없어야됩니다. GET real 변수는 필요하지만, POST real 변수는 쓸 일이 없습니다. POST real 변수는 사용하지 않을것이므로 이 조건은 만족하게 됩니다.

그럼 이것들을 종합하여 만들어봅시다.



[그림3-4-1. 문제풀이에 실패한 화면]

음..안타깝게 틀렸나고 나오네요. If문을 다시 한 번 살펴봅시다.

```
<?
if($_POST[trap] or $_GET[trap] == "mine") {
    echo("이건 안돼요-_-;");
    exit;
}
if(!$_GET[trap]) $trap = "mine";
if((!$_POST[real] or $_GET[trap] == "lucky" or $_POST[real] == $trap) and ($real != $_GET[real
] or $_GET[trap]) and (!$_GET[real] or !$_POST[real])) {
?>
```

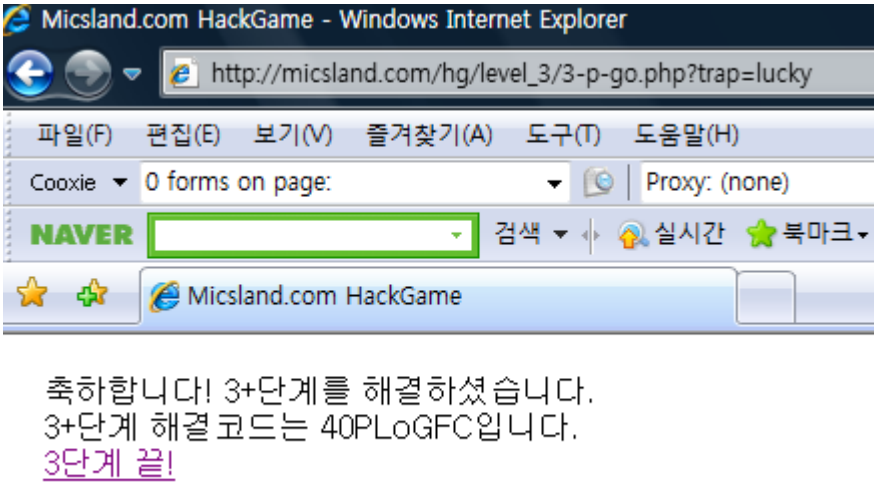
post trap이 있으면 안되고 get trap이 mine이면 안됩니다. Get trap이 **없으면** trap값을 mine으로 변경합니다. 아마 이 if문은 문제 푸는 사람을 골탕먹이기 위한 함정처럼 보입니다. 이 if문을 피하기 위해 trap 변수를 만들겠습니다. Trap 변수에는 세번째 if문의 조건을 만족하도록 lucky라는 값을 넣어주겠습니다.

trap=lucky

trap값이 lucky가 되면 첫번째 두번째 if문을 모두 무사히 통과합니다.
그럼 세 번째 if문을 봅시다.

```
if((!$_POST[real] or $_GET[trap] == "lucky" or $_POST[real] == $trap) and ($real != $_GET[real ] or $_GET[trap]) and (!$_GET[real] or !$_POST[real])) {
```

GET trap의 값이 lucky이므로 첫번째 비교문은 통과합니다. 그리고 real 변수 값이 get real 과 다르거나 trap변수가 존재하면 조건이 참이 됩니다. Trap변수가 있으므로 통과합시다. 마지막 비교문은 GET real변수가 없거나 post real 변수가 없으면 참이 됩니다. 두 개의 변수는 만들지 않아도 된다는 뜻입니다.
If문은 복잡하게 꼬여있지만 결국 만들어야 되는 변수는 trap=lucky 하나입니다.



[그림3-4-2. 해결코드]

결론

이렇게 if문이 많이 있는 문제는 if문을 분리해놓고 하나하나 보면서 해석하면 쉽게 풀 수 있습니다.

micsland.com 워게임 레벨4-1 풀이

개요

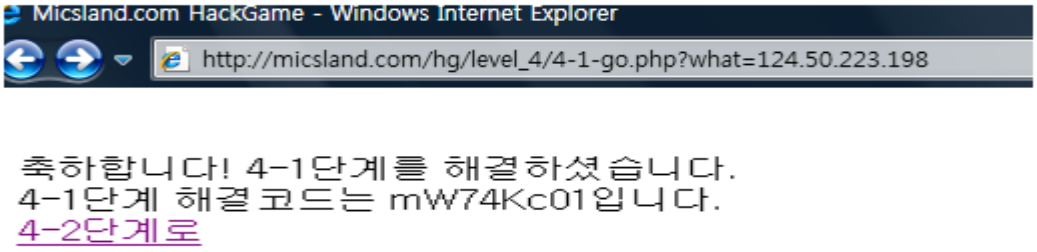
레벨4는 php의 환경변수에 관한 다양한 문제들이 나옵니다.
환경변수를 알아두는 것은 중요한것이므로 문제를 꼼꼼히 살펴봅시다.

내용

문제 코드를 살펴봅시다.

```
<? if($_GET[what] == $REMOTE_ADDR) { ?>

축하합니다! 4-1단계를 해결하셨습니다.<br>
4-1단계 해결코드는 ########입니다.<br>
<a href='4-2.php'>4-2단계로</a>

<? } else { ?>

자자, 아주 쉬워요~

<? } ?>
```

GET으로 전송되는 what 이라는 변수의 값이 $REMOTE_ADDR 이면 조건을 만족하게 됩니다.
$REMOTE_ADDR 은 php의 환경변수로 클라이언트의 아이피를 출력해주는 역할을 합니다.
그러니까 what 변수에 자신의 아이피를 넣어주면 조건을 만족하게 됩니다.



[그림4-1-1. 해결코드]

결론

$REMOTE_ADDR은 자신의 아이피를 출력해주는 역할을 하는 환경변수 입니다.
Register_global옵션의 영향으로 $REMOTE_ADDR을 사용할 수 없다면, getenv함수를 사용하면 됩니다.

개요

4-2는 문제를 풀기 전에 가변변수($$) 에 대한 힌트가 나옵니다.
이 문제는 가변변수에 대한 이해가 필요합니다.

내용

문제 코드를 살펴봅시다.

```
<?
$var = "address";
$data = $REMOTE_ADDR;
if($$var == $data and $_GET[data2] == $HTTP_HOST) {
?>
축하합니다! 4-2단계를 해결하셨습니다.<br>
4-2단계 해결코드는 ########입니다.<br>
<a href='4-3.php'>4-3단계로</a>
<? } else { ?>
챌린저 정신!
<? } ?>
```

가변변수($$) 가 있으면 그 변수에 있던 값이 변수가 되버립니다.

```
<?
echo("$a<hr>");
echo("$$a");
?>
```

a에 test를 넣으면

test

$test

[그림4-2-1. 가변변수]

---

test , $test 가 출력됩니다. 위에는 echo로 $a를 출력한것이고 아래는 가변변수를 보여주기 위해 $$a 를 출력한것입니다. a변수에 test를 입력했으므로 $a를 하면 test가 출력됩니다. 그런데 가변변수를 사용하면 앞에 변수를 의미하는 $가 붙어서 출력됩니다.
간단하게 변수 안에 있는 값이 변수가 된다라고 생각하시면 됩니다.
그럼 이제 문제 코드를 다시 살펴봅시다.

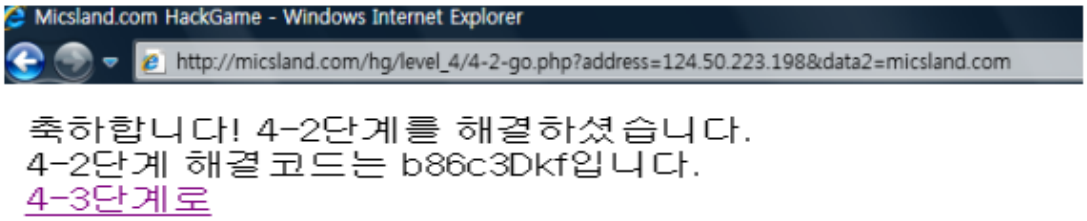
```
<?
$var = "address";
$data = $REMOTE_ADDR;
if($$var == $data and $_GET[data2] == $HTTP_HOST) {
?>
```

var 변수에는 address 라는 값이 들어갑니다.
data 변수에는 $REMOTE_ADDR , 나의 아이피가 들어갑니다.

$$var 변수의 값이 $data 와 같고 GET으로 전송된 data2의 값이 $HTTP_HOST와 같으면 조건이 참이 됩니다.

가변변수에 대해 보기 전에, $HTTO_HOST는 호스트를 출력해주는 php 환경변수로,여기서는 문제 서버의 호스트인 micsland.com 가 출력이 됩니다.

다시 가변변수가 있는 쪽으로 돌아와서, $$var이 data변수와 같으면 이 조건인데, var은 가변변수 입니다. var에 있는 값은 address 이므로 address가 변수가 됩니다.
즉 $$var은 $address 를 가르킵니다. 이 address가 data,즉 내 아이피가 되어야 첫 번째 조건을 만족하고 GET으로 전송되는 data2가 이 서버의 호스트 micsland.com 이 되면 두 번째 조건도 만족하게 됩니다.



[그림4-2-2. 해결코드]

결론

가변변수와 php의 환경변수 $HTTP_HOST에 대한 이해가 필요한 문제였습니다.

micsland.com 워게임 레벨4-3 풀이

개요

php 환경변수 세션에 관한 문제입니다.
세션을 확인하고 수정하는 방법을 알아야 됩니다.

내용

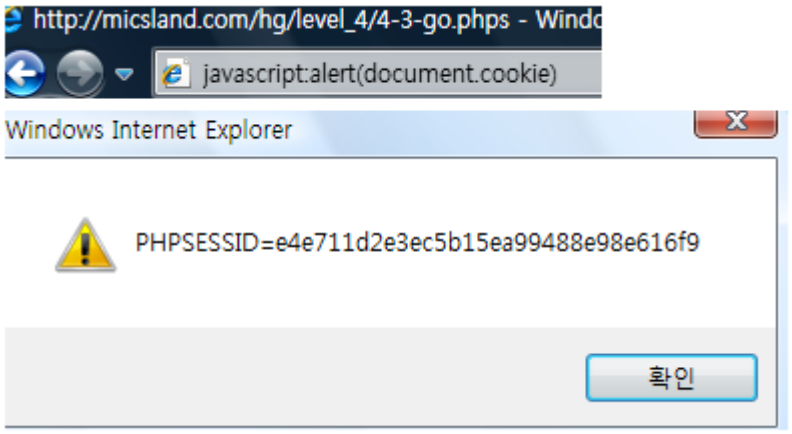
문제 코드를 살펴봅시다.

```
<?
$var = $PHPSESSID;
$data = getenv("REMOTE_ADDR");
if($$var == $data) {
?>
축하합니다! 4-3단계를 해결하셨습니다.<br>
4-3단계 해결코드는 ########입니다.<br>
<a href='4-4.php'>4-4단계로</a>
<? } else { ?>
여기 쓸 말도 생각이 안나네-_-
<? } ?>
```

var 변수에 $PHPSESSID 라는 값이 들어갑니다. 이건 php의 환경변수로 세션을 나타냅니다. 그리고 data 변수에는 자신의 아이피를 넣습니다. Getenv는 환경변수 값을 얻어오는 함수로 getenv("REMOTE_ADDR")은 $remote_addr과 실행결과가 똑같습니다.
If문으로 var과 data의 값이 똑같은지 검사를 하는데, var은 가변변수 입니다. 즉, var 변수에 있는 값이 변수가 됩니다. 그러니까 php 세션값이 변수가 되겠네요. 모두 종합해보면 Php 세션값이 변수명이고 값은 자신의 아이피가 들어가야 이 문제를 통과할 수 있습니다.

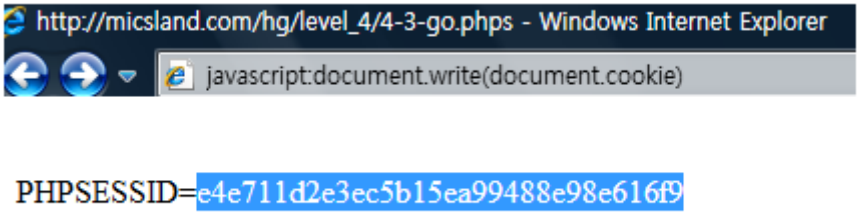
세션값 = 자신의 아이피

그럼 세션아이디를 보는 방법에 대해 알아봅시다. 주소창에 다음과 같이 입력하면 쿠키내용을 볼 수 있는데, 여기에 세션이 들어갑니다.



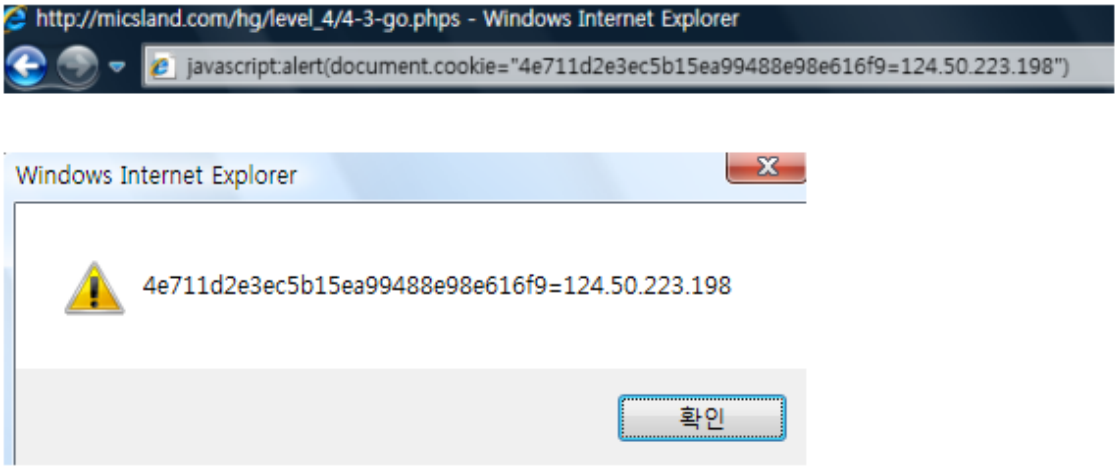
[그림4-3-1. php 세션 확인]

쿠키가 클라이언트에 저장되면 세션은 서버측에 저장됩니다. 당연히 쉽게 수정가능한 쿠키보다 세션이 더 안전하지만 많은 세션을 사용하면 그만큼 서버에 무리가 간다는 단점이 있습니다.
어쨌든 위와 같은 방법으로 세션을 확인할 수 있습니다. 이 세션값을 복사하기 위해 주소창에 아래와 같이 입력합니다.



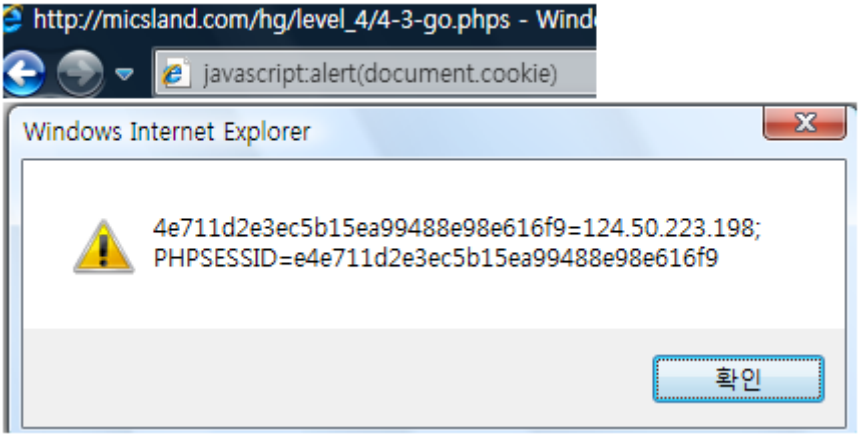
[그림4-3-2. php세션을 복사]

document.write로 화면에 글씨를 쓸 수 있습니다. 세션내용을 화면에 표시함으로써 간단하게 복사할 수 있습니다. 그럼 이 세션값을 복사하여 변수로 만들고,거기에 값을 넣어보겠습니다. 간단하게 GET method를 사용할수도 있지만 이번에는 쿠키를 사용해보겠습니다.



[그림4-3-3. 쿠키 생성]

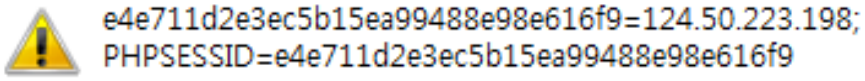
쿠키는 쿠키명=값 으로 이루어져 있습니다. 위에 그림처럼 세션값을 쿠키명으로 하고, 제 아이피를 쿠키값으로 지정해 주었습니다. 그럼 제대로 입력되었나 확인해봅시다.



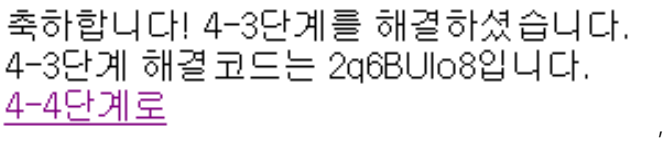
[그림4-3-4. 생성된 쿠키 확인]

제대로 입력되었습니다. 음.. 문제 풀이를 쓰는 사이에 세션값이 변경됬네요.
어쨌든 이런 방법으로 쿠키를 보거나 수정할 수 있습니다.

그럼 문제 페이지로 이동하여 위 방법으로 쿠키를 추가하고 내용을 입력해봅시다.



[그림4-3-5. 세션값을 변수로 하고 내 아이피를 변수값으로 입력한 장면]



[그림4-3-6. 해결코드]

결론

세션값을 변수로 만들고,거기에 자신의 아이피를 넣어야 풀 수 있는 문제였습니다.

micsland.com 워게임 레벨4-4 풀이

개요

이 문제의 원래 의도는 http referer 헤더값을 변경하는 것 같지만 문제 제작자의 실수로 헤더값을 변경하지 않아도 풀 수 있습니다.

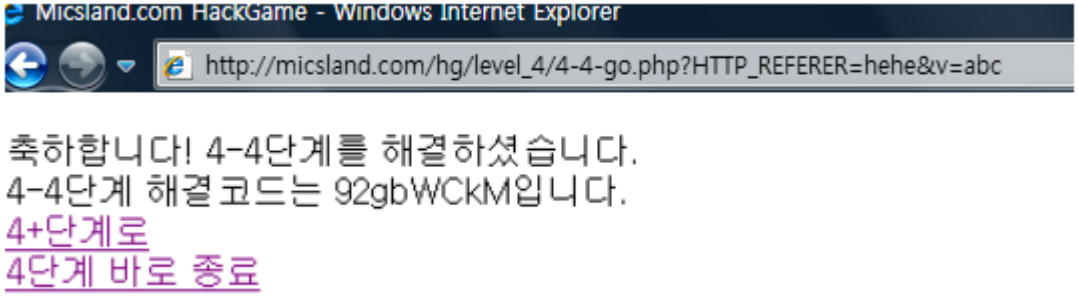
내용

문제 코드를 살펴봅시다.

```
<?
if(!$HTTP_REFERER) {
    echo("이건 안돼요;");
} elseif($_GET[v] == "abc") {
?>
축하합니다! 4-4단계를 해결하셨습니다.<br>
4-4단계 해결코드는 ########입니다.<br>
<a href='4-p.php'>4+단계로</a><br>
<a href='4-final.php'>4단계 바로 종료</a>
<? } else { ?>
이것만 깨면 +단계!
<? } ?>
```

이 문제는 제작자가 의도적으로 이렇게 만든건지 실수한건지는 모르겠지만 php 환경변수에 관한 문제가 아닙니다. $HTTP_REFERER 가 http 헤더의 referer 필드값을 말하는 것처럼 보이지만 referer를 사용할려면 $_SERVER['HTTP_REFERER'] 와 같이 사용해야됩니다.
어쨌든 문제를 살펴봅시다.
$HTTP_REFERER 변수가 있는지 없는지 검사하는데, 앞에 ! 가 있습니다. 이건 논리연산자 Not을 의미하는 것으로 HTTP_REFERER 변수가 "없으면" 이 됩니다.
만약에 HTTP_REFERER 변수가 없으면 조건이 참이 되고 이건 안돼요를 출력합니다.
안돼요 라는 메시지를 안보기 위해서는 if문을 거짓으로 만들어야 되기 때문에,
HTTP_REFERER 변수를 만들어줘야 됩니다. 그럼 elseif 문으로 넘어가게 됩니다.

Elseif 문에서는 GET으로 전송된 v라는 변수 값이 abc면 참이 됩니다.
종합해보면 HTTP_REFERER 변수를 만들어주고, GET으로 전송되는 v 변수를 만들고 값을 abc로 해주면 됩니다.



[그림4-4-1. 해결코드]

http_referer 변수에 임의의 값 hehe를 넣어주었고 v에는 abc를 넣어서 조건을 만족시켰습니다.

결론

간단하게 변수 두 개를 추가시켜주면 풀 수 있는 문제였습니다.
문제 제작자의 실수가 있었나봅니다.

micsland.com 워게임 레벨4-plus 풀이

개요

각 레벨은 마지막에 plus 문제가 존재합니다. 이 문제는 그 레벨에서 나왔던 문제들의 종합으로 그 레벨에서 난이도가 가장 높은 문제입니다.

레벨4의 plus문제는 지금까지 문제에서 볼 수 없었던 함수들이 나옵니다. 이 함수들에 대해 간단하게라도 이해를 하고 있어야 문제를 풀 수 있습니다.

내용

레벨4는 환경변수의 이해도를 묻는 레벨이었습니다. Plus 문제의 소스를 살펴봅시다.

```
<?
$data1 = "a".substr($PHPSESSID,0,3);
$data2 = "a".strlen($HTTP_HOST);
$data3 = strrchr($PHP_SELF,"/");

if($_GET[$data1]) $data1 = "trap";
elseif($_GET[$data2] == $data3) $data1 = "123";

if(!eregi("www.nzeo.com",$HTTP_REFERER)) {
    echo("어허~이게 아닙니다.");
} elseif($data1 == "123" and $_GET[data4] == getenv("REQUEST_METHOD")) {
?>
축하합니다! 4+단계를 해결하셨습니다.<br>
4+단계 해결코드는 ########입니다.<br>
<a href='4-final.php'>4단계 끝!</a>

<? } else { ?>

자아~차근차근 생각해보세요!

<? } ?>
```

이 문제는 지금까지 풀었던 문제에는 없던 한 가지 속임수가 있으므로 프로그램 코드를 매우 자세히 살펴봐야됩니다.

문제 코드를 하나하나 살펴봅시다.

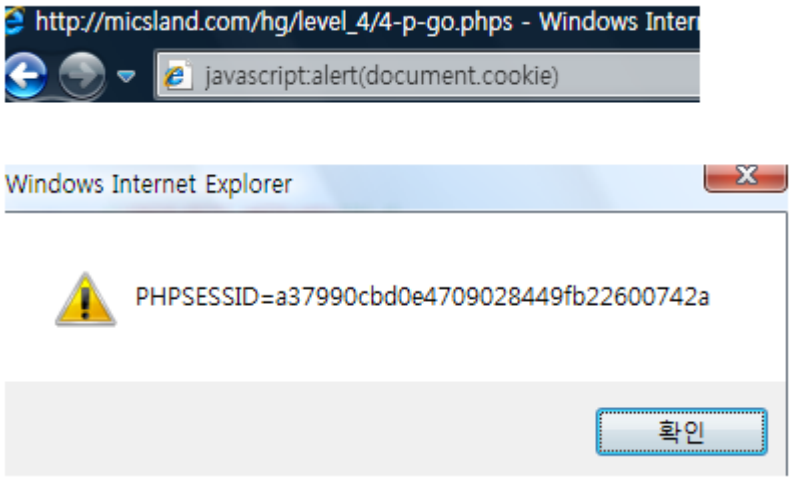
```
$data1 = "a".substr($PHPSESSID,0,3);
$data2 = "a".strlen($HTTP_HOST);
$data3 = strrchr($PHP_SELF,"/");
```

substr은 문자열을 잘라내는 함수입니다.

```
<? $a=substr("test",0,3); echo($a); ?>
```

이것을 실행한 결과 tes가 출력되었습니다. 0부터 3까지 뽑아오고,나머지는 삭제하는 함수라는걸 알 수 있습니다. 여기서는 문자 대신 php 세션을 짤라냅니다.

제 세션을 확인해봅시다.



[그림4-5-1. 세션확인]

앞에서부터 3글자를 뽑아내므로 a37 이 된다는 것을 알 수 있습니다.
그럼 다음것을 살펴봅시다.

```
$data2 = "a".strlen($HTTP_HOST);
```

strlen은 글자의 길이를 뽑아내는 것입니다. HTTP_HOST는 문제를 푸는 호스트가 되므로 micsland.com 의 길이를 넣어주면 됩니다. 12글자네요. 그리고 앞에 a가 있으므로 합치면 a12가 됩니다.

```
$data3 = strrchr($PHP_SELF,"/");
```

마지막으로 strrchr는 문자열에서 문자가 마지막으로 나오는 부분을 찾는 함수입니다.
$PHP_SELF는 현재 실행되고 있는 php 자신의 위치를 나타내는 함수입니다.
앞에 /가 있으므로 마지막 /가 있는 곳부터 시작하면 됩니다.

/4-p-go.php

모두 종합해보면

$data1=a37
$data2=a12
$data3=/4-p-go.php

이렇게 됩니다. 이제 문제의 소스를 본격적으로 살펴봅시다.

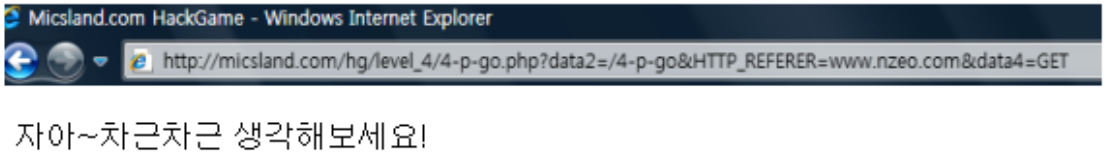
```
if($_GET[$data1]) $data1 = "trap";
elseif($_GET[$data2] == $data3) $data1 = "123";

if(!eregi("www.nzeo.com",$HTTP_REFERER)) {
    echo("어허~이게 아닙니다.");
} elseif($data1 == "123" and $_GET[data4] == getenv("REQUEST_METHOD")) {
?>
```

GET data1이 있으면 data1 변수에 trap을 넣습니다.
Get data2가 data3과 같으면 data1에 123을 넣습니다.

HTTP_REFERER에 www.nzeo.com이 없으면 이게 아닙니다를 출력합니다.
Data1이 123이고 data4가 REQUEST_METHOD면 해결코드를 출력합니다.

마지막 if문에 data1=123 이 있으므로 첫번째 if문은 함정이 됩니다.

그럼 두 번째 if문을 보고
data2=/4-p-go.php
http_referer=www.nzeo.com
data4=get
이 값들을 넣어서 전송해봅시다.



[그림4-5-2. 문제풀이에 실패한 화면]

하지만 결과는 자아~차근차근 생각해보세요! 입니다.

왜 그럴까요? 문제 소스를 다시 살펴봅시다.

```
if($_GET[$data1]) $data1 = "trap";
elseif($_GET[$data2] == $data3) $data1 = "123";

if(!eregi("www.nzeo.com",$HTTP_REFERER)) {
    echo("어허~이게 아닙니다.");
} elseif($data1 == "123" and $_GET[data4] == getenv("REQUEST_METHOD")) {
?>
```

이 문제는 지금까지와 다른 방법을 사용합니다. 첫 번째 if문을 자세히 살펴봅시다.
$_GET[$data1] 이라고 되어있습니다 $_GET[data1] 이라고 되어있었으면 클라이언트가 data1 변수를 만들 수 있지만, 여기에 $ 가 붙어있기 때문에 비교하는 대상이 클라이언트가 만든 변수가 아니라 php코드에서 생성한 변수가 됩니다.

이 점을 감안하여 다시 코드를 살펴봅시다.

```
if($_GET[$data1]) $data1 = "trap";
elseif($_GET[$data2] == $data3) $data1 = "123";

if(!eregi("www.nzeo.com",$HTTP_REFERER)) {
    echo("어허~이게 아닙니다.");
} elseif($data1 == "123" and $_GET[data4] == getenv("REQUEST_METHOD")) {
?>
```

data1변수가 있으면 data1은 trap이 됩니다.
data2 변수의 값이 data3과 같으면 data1을 123으로 만듦니다. 그럼 문제풀이 맨 처음에 살펴봤던 변수들을 다시 살펴봅시다.

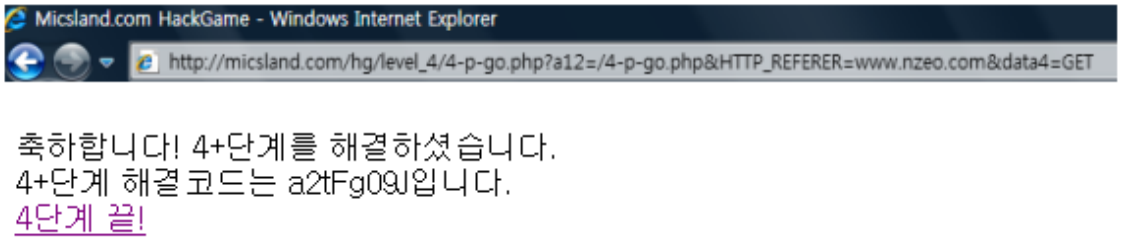
$data1=a37
$data2=a12
$data3=/4-p-go.php

data2의 값은 a12 입니다. 즉 GET으로 전송해야 되는 것은 data2가 아니라 a12 입니다.
a12=/4-p-go.php 를 하면 $data2=$data3 이라는 조건을 만족하게 됩니다.
그러면 $data1이 1234이 됩니다.

다음 if문을 보면 $HTTP_REFERER는 HTTP 헤더의 referer값이 아닌,그냥 GET방식으로 전송되는 변수를 의미합니다. HTTP_REFERER=www.nzeo.com 을 get방식으로 전송해줍니다.

그리고 마지막 if문에서 data1=123은 이미 만들어져 있고, data4를 REQUEST_METHOD와 맞춰줘야됩니다. 이건 전송방식을 뜻하는 것으로 GET방식을 사용할것이기 때문에 GET을 입력해주면 됩니다.

그럼 이것들을 종합하여 입력해봅시다.



[그림4-5-3. 해결코드]

결론

$_GET 을 사용할 때, 변수명을 비교대상으로 할 수 있다는 것을 알아야 되는 문제였습니다.

micsland.com 워게임 레벨5-1 풀이

개요

레벨5는 쿠키와 관련된 문제들이 있습니다. 쿠키를 추가,삭제,수정하는것을 능숙하게 할 수 있어야됩니다.
5-1문제는 쿠키를 다룰 수 있는지 간단하게 테스트하는 문제로, 쿠키를 추가시켜주면 풀 수 있는 문제입니다.

내용

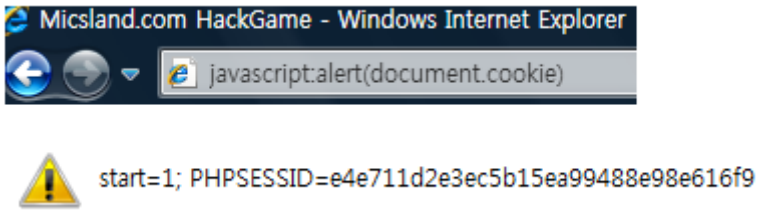
문제 코드를 살펴봅시다.

```
<?
if(!$_GET[easy]) $easy = "1";
setcookie("start",$easy,time() + 30);
if(!$_COOKIE[start]) {
    echo("쿠키 적용을 위해 새로고침을 해주세요.");
} elseif($_COOKIE[start] == "funny") {
?>
축하합니다! 5-1단계를 해결하셨습니다.<br>
5-1단계 해결코드는 ########입니다.<br>
<a href='5-2.php'>5-2단계로</a>
<?
} else {
    echo("어라..이게 아닌데? 혹시 새로고침하면 될까나?");
}
?>
```

GET으로 전송된 easy 변수가 없으면 easy변수를 만들고 1이라는 값을 넣습니다.
그리고 start라는 이름의 쿠키를 넣고,거기에 변수 easy의 값 1을 넣습니다. 이 쿠키의 유효기간은 30초 입니다.

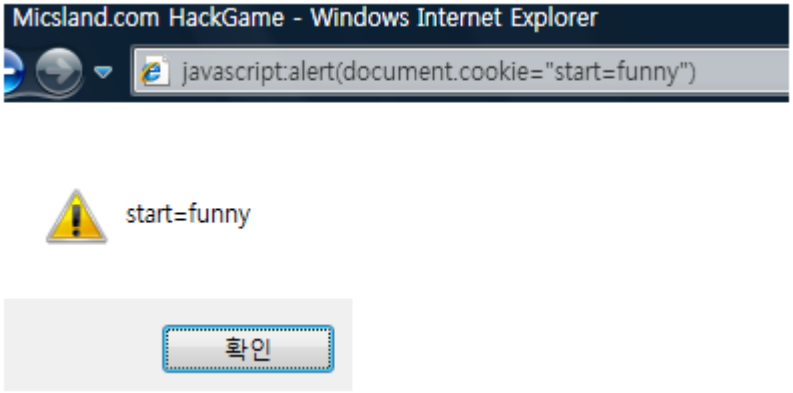
Start라는 쿠키가 사라지면 쿠키 재생성을 위해 새로고침을 하라는 메시지를 보여주고, Start라는 쿠키의 값이 funny로 변경되면 해결코드를 보여줍니다.

즉 start라는 쿠키의 값을 funny로 바꾸는 문제입니다.



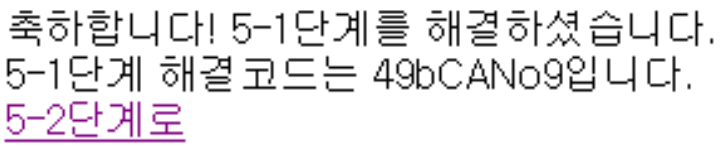
[그림5-1-1. 생성된 쿠키 확인]

먼저 쿠키가 제대로 생성되었는지 확인해보았습니다. start라는 쿠키가 생성되었네요. 그럼 이제 이 쿠키의 내용을 funny로 바꿔봅시다.



[그림5-1-2. 생성된 쿠키의 값 수정]

새로고침을 하면 해결코드를 얻을 수 있습니다.



[그림5-1-3. 해결코드]

결론

쿠키는 클라이언트에 저장되어서 서버에 요청할 때 같이 전송됩니다. 서버에서는 이 쿠키를 가지고 클라이언트의 로그인 정보 등을 비교할때가 있는데,이 쿠키에 지나치게 의존하면 그 서버의 보안은 무용지물이 되버어립니다. 취약점을 가진 홈페이지에 로그인을 했을 때, id=guest 라는 쿠키를 지정해준다면 id 쿠키값을 admin으로 변경하여 admin으로 로그인 된 것처럼 속일 수 있습니다. 그리고 쿠키에 지나치게 많은 정보(주민번호,패스워드 등)를 아무런 암호화 없이,또는 간단하게 인코딩 된 값을 그대로 저장한다면 XSS에 노출되었을 때, 큰 피해를 입게됩니다. 5-1 문제에서 본 것처럼 쿠키는 얼마든지 삭제하거나 추가,수정할 수 있기 때문에 클라이언트가 보내는 쿠키값을 믿어서는 안됩니다.

micsland.com 워게임 레벨5-2 풀이

개요

1번문제와 비슷하게 쿠키를 추가시켜줘야되는데, 쿠키에 입력되어야할 값이 php환경변수와 관련되어있습니다. Php 환경변수의 역할에 대해 알아야 되는 문제입니다.

내용

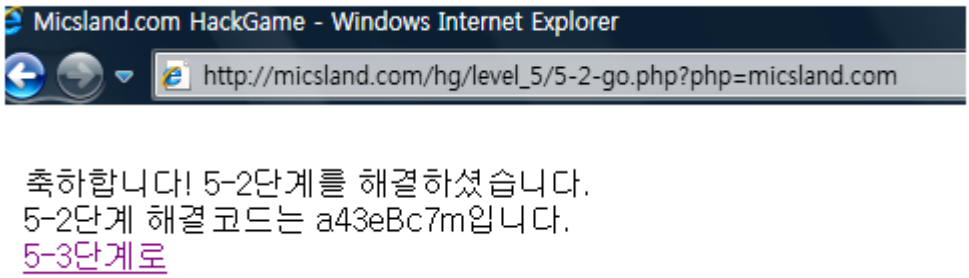
문제 코드를 살펴봅시다.

```
<?
setcookie("php",$php,time() + 30);
setcookie("cookie1",$HTTP_HOST,time() + 30);
if(!$_COOKIE[cookie1]) {
    echo("쿠키 적용을 위해 새로고침을 한번만 해주세요.");
} elseif($_COOKIE[cookie1] == $php) {
?>
축하합니다! 5-2단계를 해결하셨습니다.<br>
5-2단계 해결코드는 ########입니다.<br>
<a href='5-3.php'>5-3단계로</a>
<?
} else {
    echo("꽝;ㅁ;/ 새로고침이라도 해보시길.");
}
?>
```

Php라는 이름의 쿠키를 만들고,거기에는 php라는 이름의 변수의 값을 넣습니다.
Cookie1이라는 이름의 쿠키를 생성,쿠키값은 환경변수 HTTP_HOST 입니다.

처음 php쿠키에 들어가는 $php 변수는 문제 소스에서 지정되지 않았습니다. 즉 이 php변수는 사용자가 직접 만들어야됩니다. 그냥 $php 라고 했으므로 get,post,cookie 어느 방식을 사용해도 상관없습니다.
그리고 두 번째 cookie1 쿠키에는 환경변수 $HTTP_HOST가 들어가는데, 여기에서는 문제서버의 HOST micsland.com이 들어갑니다.

Elseif로 cookie1의 값이 php변수와 같은지 확인하므로, php 변수에 micsland.com를 입력해 주면 됩니다.



[그림5-2-1. 해결코드]

get 방식으로 php 변수를 만들고 micsland.com을 입력해주면

php변수에는 $php 변수의 값이 들어가는데 get에 php=micsland.com을 입력해줬으므로 php변수에는 micsland.com이 들어갑니다.

Cookie1에는 http_host가 들어가는데,이 값은 micsland.com 입니다.
결국 cookie1와 php 변수의 값이 같아지므로 조건을 만족하게 되고 해결코드를 볼 수 있게 됩니다.

결론

쿠키의 값을 환경변수로 수정하는 문제였습니다.

micsland.com 워게임 레벨5-3 풀이

개요

쿠키에 들어가는 값에 대해 약간 생각해봐야되는 문제입니다.
if문을 자세히 보세요.

내용

문제 코드를 살펴봅시다.

```
<?
if(!$_GET[high]) $high = "and";
if($_GET[low] == "and") $low = "error";
setcookie("game",$high,time() + 30);
if(!$_COOKIE[game]) {
    echo("쿠키 적용을 위해 새로고침을 한번만 해주세요.");
} elseif($_COOKIE[game] == $low) {
?>
축하합니다! 5-3단계를 해결하셨습니다.<br>
5-3단계 해결코드는 ########입니다.<br>
<a href='5-p.php'>5+단계로</a><br>
<a href='5-final.php'>5단계 바로 종료</a>
<?
} else {
    echo("혹시 될지도 몰라...한번 새로고침이라도..");
}
?>
```

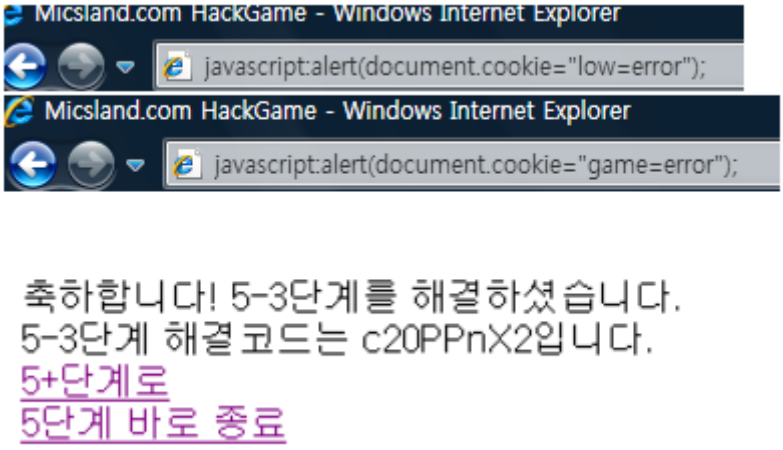
GET방식으로 전송된 high라는 이름의 쿠키가 없으면 high변수를 만들고 and라는 값을 넣습니다.

GET방식으로 전송된 low라는 이름의 쿠키의 값이 and면 low변수를 만들고 error라는 값을 넣습니다.

Game이라는 이름의 쿠키를 만들고 high 변수의 값을 넣습니다.
Game쿠키와 low변수 값이 같으면 해결코드를 출력합니다.
음..high변수는 **없을 때** and라는 값이 들어가므로 특별히 high 변수를 만들어줄 필요는 없어보입니다.
Get method로 low 변수를 만들어주고 값을 and로 하면 low변수의 값이 error로 변경됩니다.
그리고 game이라는 쿠키에는 high 변수의 값 and가 들어갑니다.
Game쿠키와 low변수 값을 똑같이 만들어주는게 목표이므로 low변수를 game쿠키의 값 and로 수정해줘야 되지만, low변수를 and로 하면 error로 강제로 변경되어 버립니다.

즉 game=and, low=and → game=and, low=error 가 되어 조건을 만족시킬 수 없게됩니다.
그러면 다른 방법을 써서 game쿠키가 low변수를 따라가게 해봅시다.

Game쿠키와 low변수의 값을 모두 error로 변경해봅시다.



[그림5-3-1. game,low쿠키값을 error로 변경하여 해결코드를 얻는 화면]

문제를 통과했습니다.

결론

쿠키값을 강제로 변경시키는 것을 우회하는 문제였습니다.

micsland.com 워게임 레벨5-plus 풀이

개요

각 레벨은 마지막에 plus 문제가 존재합니다. 이 문제는 그 레벨에서 나왔던 문제들의 종합으로 그 레벨에서 난이도가 가장 높은 문제입니다.
레벨5의 plus 문제는 생성되는 쿠키의 값을 모르는 상태에서 풀어야 됩니다.

내용

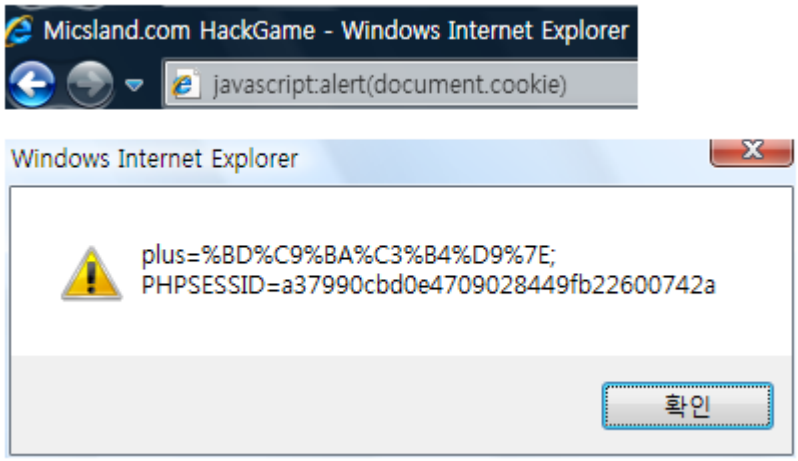
레벨5는 쿠키에 관한 문제였습니다.
Plus문제의 소스를 살펴봅시다.

```
<?
$hidden = "########";
setcookie("plus",$hidden,time() + 30);

if(!$_COOKIE[plus]) {
    echo("쿠키 적용을 위해 새로고침을 한번만 해주세요.");
} elseif($_COOKIE[plus] == $aha) {
?>

축하합니다! 5+단계를 해결하셨습니다.<br>
5+단계 해결코드는 wM49ZNfo입니다.<br>
<a href='5-final.php'>5단계 끝!</a>

<?
} else {
    echo("마지막으로 새로고침을 눌러볼까?");
}
?>
```

```
$hidden = "########";
setcookie("plus",$hidden,time() + 30);

if(!$_COOKIE[plus]) {
    echo("쿠키 적용을 위해 새로고침을 한번만 해주세요.");
} elseif($_COOKIE[plus] == $aha) {
?>
```

hidden 변수에 무언가 값을 넣고 plus 쿠키를 생성, 이곳에 hidden의 값을 넣습니다.

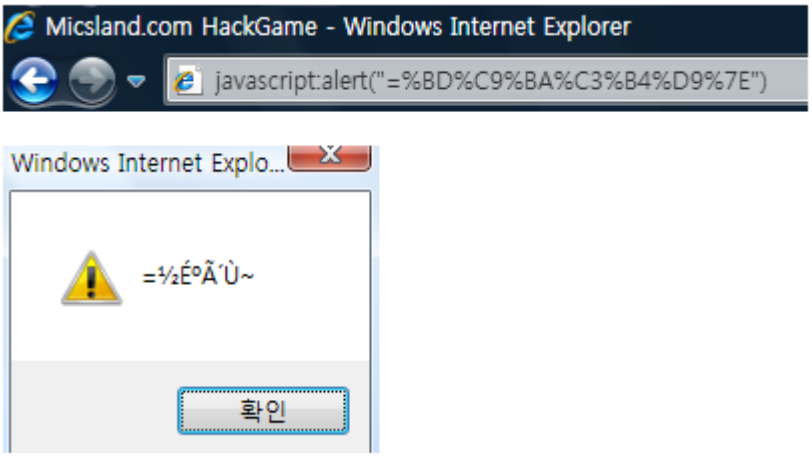
Plus쿠키의 값이 aha변수와 같으면 해결코드를 출력합니다. Hidden변수의 값은 모르지만 지금까지 배워온 간단한 방법으로 확인할 수 있습니다. 그럼 페이지에 접속하여 확인해봅시다.



[그림5-4-1. 생성된 쿠키 확인]

plus 쿠키에 %BD%C9%BA%C3%B4%D9%7E 라는 무언가 인코딩된 값이 들어가있습니다.

뭐라고 쓰여있는건지 확인해봅시다.



[그림5-4-2. 디코딩된 정답이 깨져서 나오는 화면]

음..이번에도 글씨가 깨져서 나옵니다. Php의 urldecode 함수를 사용합시다.

```
<?
$a=urldecode("=%BD%C9%BA%C3%B4%D9%7E");
echo($a);
?>
```

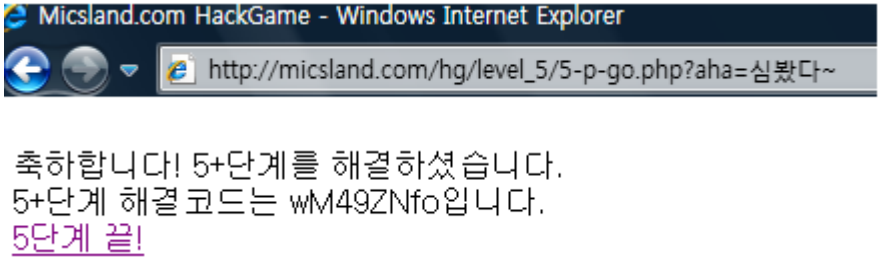
=심봤다~

[그림5-4-3. 제대로 디코딩된 정답]

=심봤다~ 가 출력되었습니다. 소스를 다시 보니 제가 실수로 plus= 에서 =까지 복사해버렸네요. 정답은 심봤다~ 입니다.

```
elseif($_COOKIE[plus] == $aha) {
```

이 plus쿠키의 값과 변수 aha의 값을 똑같이 만들어줘야 되므로 aha 변수를 만든 후, 심봤다~ 라는 값을 넣어주겠습니다.



[그림5-4-4. 해결코드]

해결코드가 출력되었습니다.

결론

url 디코딩에 관한 문제였습니다. 단순하게 url에 인코딩된 문자를 입력했을 경우 글씨가 깨져서 나오므로 php의 urldecode함수를 사용했습니다.

micsland.com 워게임 레벨6-1 풀이

개요

레벨5에서는 쿠키를 단순하게 추가하면 통과할 수 있는 문제들이었지만 레벨6은 쿠키를 수정,삭제해야 풀 수 있는 문제들로 구성되어 있습니다.

내용

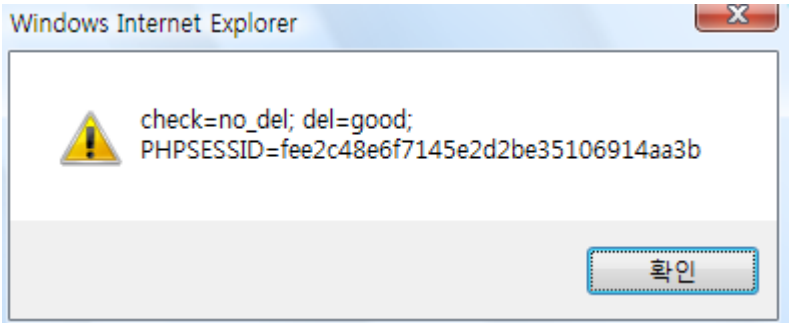
문제 코드를 살펴봅시다.

```
<?
setcookie("check","no_del",time() + 180);
setcookie("del","good",time() + 180);
if(!$_COOKIE[check]) {
    echo("쿠키 적용을 위해 새로고침을 해주세요.");
} elseif(!$_COOKIE[del]) {
?>
축하합니다! 6-1단계를 해결하셨습니다.<br>
6-1단계 해결코드는 ########입니다.<br>
<a href='6-2.php'>6-2단계로</a>
<?
} else {
    echo("쿠키 조작 첫번째 시도!");
}
?>
```

Check라는 쿠키를 만들고 거기에 no_del 이라는 값을 넣습니다. 유효기간은 3분이구요.
그리고 del 이라는 쿠키를 생성, good 이라는 값을 넣습니다.
Elseif 문으로 del 쿠키가 **없다면** 해결코드를 보여주게 됩니다.

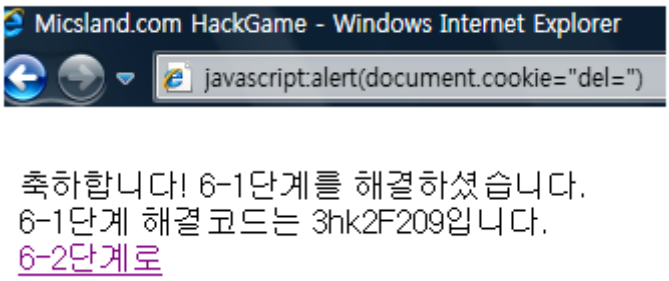
이 문제는 생성된 del 쿠키를 삭제하는 문제입니다.

먼저 문제 페이지에 접속하고 생성된 쿠키를 확인해봅시다.



[그림6-1-1. 생성된 쿠키 확인]

check,del 쿠키를 확인할 수 있습니다. 목표는 del 쿠키를 삭제하는 것이므로 삭제해보겠습니다. 삭제하는 방법은 간단하게 del 쿠키를 공백으로 만들어주면 됩니다.



[그림6-1-2. 생성된 쿠키 제거]

del 쿠키의 값이 공백이 되고, 새로고침을 누르면 해결코드를 얻을 수 있습니다.

결론

생성된 쿠키를 삭제하는 문제였습니다.

micsland.com 워게임 레벨6-2 풀이

개요

생성된 쿠키를 수정하는 문제입니다. 쿠키를 수정하는 방법에 대해 알아봅시다.

내용

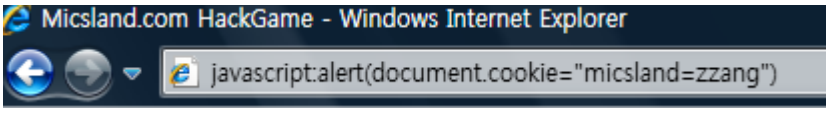
문제 코드를 살펴봅시다.

```
<?
setcookie("micsland","com",time() + 180);
if(!$_COOKIE[micsland]) {
	echo("쿠키 적용을 위해 새로고침을 해주세요.");
} elseif($_COOKIE[micsland] == "zzang") {
?>

축하합니다! 6-2단계를 해결하셨습니다.<br>
6-2단계 해결코드는 ########입니다.<br>
<a href='6-3.php'>6-3단계로</a>

<?
} else {
	echo("이번에는 삭제가 아니죠?");
}
?>
```

Micsland라는 쿠키를 만들고 거기에 com이라는 값을 넣습니다.
만약에 micsland 쿠키의 값이 zzang이라면 해결코드를 출력합니다.

간단한 쿠키조작 문제입니다. 주소창에 아래와 같이 입력하여 쿠키를 수정할 수 있습니다.

쿠키는 변수명=값 으로 이루어져 있다는 것을 잊지마세요.



[그림6-2-1. 쿠키값 변경]

새로고침을 누르면 해결코드를 얻을 수 있습니다.

축하합니다! 6-2단계를 해결하셨습니다.
6-2단계 해결코드는 ReQc2nBC입니다.
6-3단계로

[그림6-2-2. 해결코드]

결론

이렇게 쿠키는 클라이언트에 의해 얼마든지 삭제,수정될 수 있기 때문에 쿠키를 이용한 인증은 전혀 안전하지 않습니다.

micsland.com 워게임 레벨6-3 풀이

개요

쿠키에 대한 마지막 문제입니다.
이 문제에는 약간의 오류가 있습니다.

내용

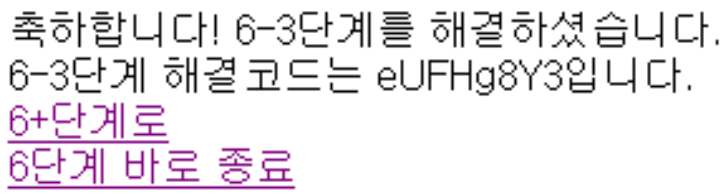
문제 코드를 살펴봅시다.

```
<?
setcookie("micsland","com",time() + 180);
if($_COOKIE[master] == $_GET[success]) {
?>

축하합니다! 6-3단계를 해결하셨습니다.<br>
6-3단계 해결코드는 ########입니다.<br>
<a href='6-p.php'>6+단계로</a><br>
<a href='6-final.php'>6단계 바로 종료</a>

<?
} else {
    echo("이것만 깨면 +입니다~");
}
?>
```

Micsland 쿠키를 생성, com 이라는 값을 넣습니다.
그리고 master라는 쿠키와 get method로 전송된 success 쿠키의 값이 같으면
해결코드를 보여줍니다.

Master라는 이름의 쿠키와 GET METHOD로 전송되는 success 변수를 직접 생성해줘야 되는데, 처음에는 존재하지 않으니 둘 다 값이 공백으로 되어있습니다. 둘 다 공백이니까 if문의 같으면 조건이 참이 되고 어이없게 문제페이지에 접속하자마자 해결코드가 출력됩니다.

문제에 약간의 오류가 있어서 간단하게 풀 수 있었지만 정상적인 문제였다면 master쿠키값을 1로 해주고 문제 url에 success=1을 추가해서 풀 수 있습니다.



[그림6-3-1. 해결코드]

결론

제작자의 의도는 쿠키를 생성하는 것이었지만 잘못된 if문으로 인해 시작하자마자 정답을 알 수 있는 문제였습니다. 만약에 이 문제를 제대로 수정할려면 if문에 두 변수의 값이 공백이 아니면 이라는 조건을 추가시켜줘야 됩니다.

micsland.com 워게임 레벨6-plus 풀이

개요

각 레벨은 마지막에 plus 문제가 존재합니다. 이 문제는 그 레벨에서 나왔던 문제들의 종합으로 그 레벨에서 난이도가 가장 높은 문제입니다.

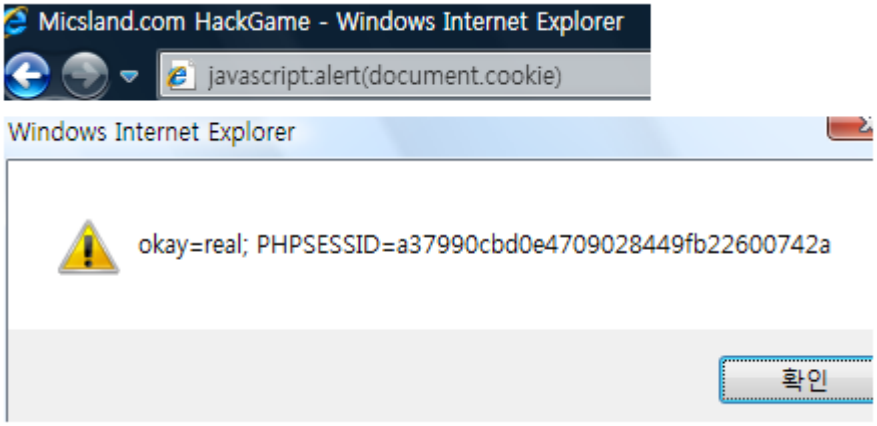
레벨6의 plus문제는 단순히 쿠키값만 조작하는 것이 아닌, 전송 방식을 수정해야 풀 수 있는 문제입니다. 웹프록시 툴로 전송방식을 조작하는 방법에 대해 알아봅시다.

내용

레벨6은 쿠키조작에 관한 레벨이었습니다. 마지막 plus 문제를 살펴봅시다.

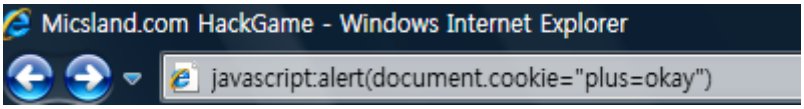
```
<?
setcookie("########","########",time() + 180);
if(!$_COOKIE[########]) {
    echo("쿠키 적용을 위해 새로고침을 해주세요.");
} elseif(($_COOKIE[plus] == $_COOKIE[########]) and $other and getenv("REQUEST_METHOD") != "GET") {
?>

축하합니다! 6+단계를 해결하셨습니다.<br>
6+단계 해결코드는 ########입니다.<br>
<a href='6-final.php'>6단계 끝!</a>

<?
} else {
    echo("마지막 단계! 힘을 내세요~");
}
?>
```

어떤 쿠키를 생성하고, plus 쿠키의 값이 이 어떤 쿠키의 값과 같으면서 $other 변수가 존재하고, METHOD가 GET이 아니면 문제를 통과합니다. 일단 이 생성되는 쿠키가 어떤건지 문제페이지에 접속하여 확인해봅시다.



[그림6-4-1. 생성된 쿠키 확인]

쿠키는 okay=real 이었습니다. 그럼 plus 쿠키를 생성하고 그 값을 real로 해줍시다.



[그림6-4-2. 쿠키생성]

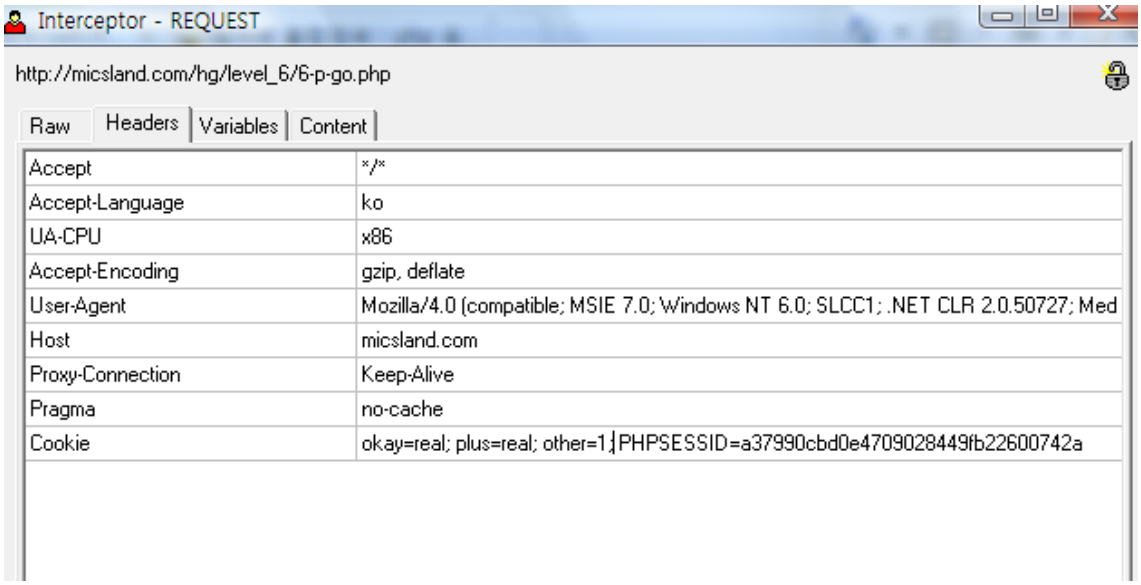
그리고 other 쿠키를 만들어서 아무 값이나 넣습니다.

other=1

그러면 plus=okay, other 두 개의 조건을 만족하게 됩니다. 마지막 비교문은 전송방식이 GET이 아니면 이므로 POST로 이것을 전송합시다.

웹프록시를 이용하여 전송방식을 GET이 아닌 POST로 변경시켜보겠습니다.

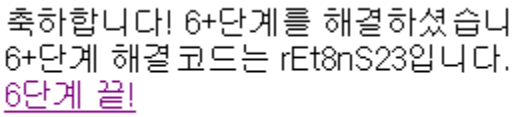
.



[그림6-4-3. 전송될 쿠키들을 확인하는 장면]

[그림6-4-4. 전송방식을 GET에서 POST로 수정하는 장면]

okay의 값은 real, plus의 값도 real 이므로 둘 다 일치하게 됩니다.
그리고 other 변수의 값도 존재합니다. 또, 전송방식은 POST이므로 모든 조건을 만족하게 됩니다.



[그림6-4-5. 해결코드]

결론

쿠키를 생성하고 전송방식을 변경하는 문제였습니다. 전송방식을 변경할때에는 웹 프록시를 사용했는데, 웹프록시는 웹해킹을 할 때 매우 유용하므로 어느정도 사용할 수 있어야됩니다.