POC – Hacker's dream Script #1
US-ASCII 방식의 악성 스크립트 분석

HACKING GROUP "OVERTIME"

woos55< wooshack55@gmail.com>2008.10.27

1 . US_ASCII로 인코딩 된 스크립트 디코딩 하기.

[2]사이트 참조 ASCIIExploit.exe d index.html << decode.html

decode.html 로 디코드 된 스크립트를 분석하다 보면 다음과 같은 의심가는 부분을 만날 수 있다. unescape() 안의 값을 디코딩 해보자.

```
<SCRIPT LANGUAGE="JavaScript" id="mineGame3">
// ***************************************************************************
// This is the script for the security of the game.
// ***************************************************************************

document.write(  unescape('%3C%73%63%72%69%70%74%3E%66%75%6E%63%74%69
%6F%6E%20%64%46%28%73%29%7B%76%61%72%20%73%31%3D%75%6E%65%73%
63%61%70%65%28%73%2E%73%75%62%73%74%72%28%30%2C%73%2E%6C%65%6
E%67%74%68%2D%31%29%29%3B%20%76%61%72%20%74%3D%27%27%3B%66%6F
%72%28%69%3D%30%3B%69%3C%73%31%2E%6C%65%6E%67%74%68%3B%69%2B
%2B%29%74%2B%3D%53%74%72%69%6E%67%2E%66%72%6F%6D%43%68%61%72%
43%6F%64%65%28%73%31%2E%63%68%61%72%43%6F%64%65%41%74%28%69%29
%2D%73%2E%73%75%62%73%74%72%28%73%2E%6C%65%6E%67%74%68%2D%31%
2C%31%29%29%3B%64%6F%63%75%6D%65%6E%74%2E%77%72%69%74%65%28%7
5%6E%65%73%63%61%70%65%28%74%29%29%3B%7D%3C%2F%73%63%72%69%70
%74%3E'));

dF('%2A8Hxhwnuy%2A75qfslzflj%2A8lOf%7BfXhwnuy%2A8Jkzshynts%2A75q66666q%
2A7%3Dq6666qq%2A7%3E%2A%3CGyw%7E%2A%3CGqq666NN%2A7%3Dqq66qq66q%
                                                .
                                                .
                                                .

2A77%2A7%3E%2A8H%2A7Kxhwnuy%2A8J5');

</SCRIPT>
```

2. unescape 이용해서 디코딩 하려고 하는 ()안을 값이 무엇인지 확인해보자.

디코딩 해보면 dF함수를 정의한 부분이란 것을 알 수 있다. 그럼, dF()함수를 돌릴 수 있다. 결과를 unescape되지 않은 순수 t값을 찍어서 보자.

```
function dF(s){
var s1=unescape(s.substr(0,s.length-1));
var t='';
for(i=0;i<s1.length;i++)
t+=String.fromCharCode(s1.charCodeAt(i)-s.substr(s.length-1,1));
document.write(t);
//document.write(unescape(t));
}
```

그럼, 다음과 같이 인코딩 된 스크립트를 얻을 수 있다.

%3Cscript%20language%3DJavaScript%3Efunction%20l111111%28l1111ll%29%7Btry%7Bll111II%28ll11l111%29%3Bfor%28var%20IIIIIII%20%3D%200%3B%20IIIIIII%20%3C%20l111ll.length%3B%20IIIIIII%2B%2B%29%20%7B%20IIIIIII%20%2B%3D%20ll1111l.charCodeAt%28IIIIIII%29%20%7DIIIIIII%20%3D%20IIIIIII%20%25%20200000%3Bvar%20llllllll%20%3D%20new%20Array%3B%20llllllll%20%3D%20l1111ll.split%28%22%2C%22%29%3Bvar%20IIIIIIl%20%3D%20%22%22%3B%20for%28var%20IIIIIII%20%3D%200%3B%20IIIIII%20%3C%20llllllll.length%3B%20IIIIIII%2B%2B%29%20%7B%20IIIIIIl%20%2B%3D%20String.fromCharCode%28%28%28llllllll%5BIIIIIII%5D%29-IIIIIII%29%20%5E%20llllllll.charCodeAt%28IIIIIII%25llllllll.length%29%29%3B%7Dvar%20IIIIIIl%3DIIIIIIl.length%2CIIIIIII%2CIIIIIII%2CIIIIIII%2CIIIIIII%3D%28512%2A2%29%2CIIIIIII%3D0%2CIIIIIII%3D0%2CIIIIIII%3D0%3Bfor%28llIIIII%3D%20Math.ceil%28IIIIIII%2FIIIIIII%29%3BIIIIIII%3E0%3BIIIIIII--%29%7BIIIIIII%3D%27%27%3B%20for%28IIIIIII%3DMath.min%28IIIIIII%2CIIIIIII%29%3BIIIIII%3E0%3B%20%20IIIIIII--%2CIIIIIII--%29%7BIIIIIII%7C%3D%28IIIIIII%5B%20IIIIIII.charCodeAt%28IIIIIII%2B%2B%29-48%5D%29%3C%3CIIIIIII%3Bif%28IIIIIII%29%7BIIIIIII%2B%3DString.fromCharCode%28209%5EIIIIIII%26255%29%3BIIIIIII%3E%3E%3D8%3BIIIIIII-%3D2%7Delse%7BIIIIIII%3D6%7D%3B%7D%20ll111II%28IIIIIII%29%20%7D%20%7D%20catch%28error%29%20%7B%7D%7D%3C%2Fscript%3E%3Cscript%20language%3DJavaScript%3Evar%20IIIIIII%3DArray%2863%2C12%2C6%2C53%2C0%2C48%2C43%2C54%

2C42%2C47%2C0%2C0%2C0%2C0%2C0%2C0%2C59%2C13%2C25%2C30%2C50%2C60%2C1%2C18%2C61%2C8%2C16%2C56%2C20%2C49%2C51%2C21%2C45%2C28%2C22%2C31%2C35%2C5%2C14%2C23%2C27%2C37%2C2%2C0%2C0%2C0%2C0%2C55%2C0%2C11%2C33%2C9%2C19%2C40%2C41%2C15%2C62%2C3%2C39%2C10%2C32%2C17%2C44%2C36%2C24%2C7%2C52%2C26%2C29%2C58%2C57%2C46%2C4%2C34%2C38%29%3Bl111111l%28%2271496%2C71517%2C71488%2C71525%2C71484%2C71487%2C71467%2C71528%2C71541%2C71503%2C71531%2C71511%2C71530%2C71530%2C71546%2C71482%2C71515%2C71499%2C71531%2C71511%2C71442%2C71540%2C71525%2C71497%2C71516%2C71474%2C71507%2C71464%2C71441%2C71449%2C71525%2C71526%2C71542%2C71540%2C71546%2C71474%2C71443%2C71543%2C71477%2C71532%2C71506%2C71448%2C71448%2C71492%2C71546%2C71542%2C71486%2C71471%2C71455%2C71522%2C71455%2C71475%2C71565%2C71477%2C71459%2C71467%2C71464%2C71473%2C71488%2C71478%2C71473%2C71475%2C71452%2C71453%2C71548%2C71527%2C71524%2C71558%2C71554%2C71531%2C71450%2C71553%2C71502%2C71495%2C71450%2C71480%2C71445%2C71506%2C71558%2C71553%2C71481%2C71484%2C71509%2C71525%2C71563%2C71503%2C71448%2C71463%2C71474%2C71442%2C71454%2C71528%2C71473%2C71488%2C71454%2C71510%2C71484%2C71501%2C71450%2C71526%2C71489%2C71455%2C71458%2C71551%2C71549%2C71551%2C71537%2C71507%2C71562%2C71455%2C71450%2C71485%2C71567%2C71470%2C71553%2C71501%2C71554%2C71459%2C71565%2C71464%2C71443%2C71481%2C71562%2C71447%2C71537%2C71526%2C71486%2C71466%2C71481%2C71484%2C71498%2C71467%2C71455%2C71489%2C71523%2C71533%2C71446%2C71487%2C71553%2C71488%2C71443%2C71501%2C71485%2C71553%2C71562%2C71476%2C71513%2C71519%2C71445%2C71492%2C71458%2C71453%2C71480%2C71448%2C71451%2C71453%2C71464%2C71533%2C71512%2C71479%2C71460%2C71455%2C71488%2C71479%2C71567%2C71497%2C71512%2C71518%2C71554%2C71452%2C71554%2C71448%2C71449%2C71557%2C71561%2C71453%2C71447%2C71494%2C71447%2C71510%2C71497%2C71462%2C71485%2C71553%2C71494%2C71466%2C71469%2C71510%2C71449%2C71516%2C71527%2C71441%2C71454%2C71526%2C71442%2C71464%2C71506%2C71468%2C71471%2C71524%2C71465%2C71475%2C71477%2C71532%2C71498%2C71475%2C71448%2C71492%2C71561%2C71546%2C71538%2C71501%2C71562%2C71531%2C71531%2C71538%2C71502%2C71526%2C71552%2C71509%2C71513%2C71481%2C71474%2C71508%2C71529%2C71559%2C71550%2C71517%2C71538%2C71565%2C71495%2C71445%2C71484%2C71531%2C71553%2C71442%2C71468%2C71501%2C71568%2C71526%2C71536%2C71513%2C71527%2C71542%2C71552%2C71563%2C71534%2C71466%2C71484%2C71496%2C71461%2C71458%

2C71508%2C71498%2C71462%2C71513%2C71554%2C71448%2C71442%2C71532%2C71498%2C71448%2C71475%2C71568%2C71542%2C71514%22%29%3C%2Fscript%3E

3. 패스워드 스크립트 디코딩하여 패스워드 알아내기.

위의 스크립트를 디코딩한 결과이다. 여기서 주목해야 할 부분은 빨간색으로 된 함수와 변수이다. 원문 스크립트 어느 부분을 찾아보아도 이와 관련된 함수는 찾아 볼 수 없다. 그러나 분명 원문 스크립트 어딘가에 있을 것이므로 디코드된 원문을 다시 분석해보자.
(다음 설명에서부터는 이 스크립트를 "패스워드 스크립트"라 명명하겠다.)

```
<script language=JavaScript>function l111111(l11111l){try{ll111II(ll11ll11);for(var IIIIIII = 0; IIIIIII < ll1111l.length; IIIIIII++) { IIIIIII += ll1111l.charCodeAt(IIIIIII) }IIIIIII = IIIIIII % 200000;var llIIIII = new Array; llIIIII = l11111l.split(",");var IIIIIII = ""; for(var IIIIIII = 0; IIIIIII < llIIIII.length; IIIIIII++) { IIIIIII += String.fromCharCode(((llIIIII[IIIIIII])-IIIIIII) ^ llIIIII.charCodeAt(IIIIIII%llIIIII.length));}var IIIIIII=IIIIIII.length,lIIIIII,IIIIII,IIIIII,IIIIII=(512*2),lIIIIII=0,lIIIIII=0,lIIIIII=0;for(lIIIIII=Math.ceil(IIIIIII/IIIIIII);lIIIIII>0;lIIIIII--){IIIIIII='';for(IIIIIII=Math.min(IIIIIII,IIIIII);IIIIIII>0;lIIIIII--,IIIIIII--){lIIIIII|=(IIIIIII[IIIIII.charCodeAt(lIIIIII++)-48])<<lIIIIII;if(lIIIIII){IIIIIII+=String.fromCharCode(209^lIIIIII&255);lIIIIII>>=8;lIIIIII-=2}else{lIIIIII=6};}ll111II(IIIIIII)}}catch(error){}}</script><script language=JavaScript>var IIIIIII=Array(63,12,6,53,0,48,43,54,42,47,0,0,0,0,0,0,59,13,25,30,50,60,1,18,61,8,16,56,20,49,51,21,45,28,22,31,35,5,14,23,27,37,2,0,0,0,0,55,0,11,33,9,19,40,41,15,62,3,39,10,32,17,44,36,24,7,52,26,29,58,57,46,4,34,38);l111111("71496,71517,71488,71525,71484,71487,71467,71528,71541,71503,71531,71511,71530,71530,71546,71482,71515,71499,71531,71511,71442,71540,71525,71497,71516,71474,71507,71464,71441,71449,71525,71526,71542,71540,71546,71474,71443,71543,71477,71532,71506,71448,71448,71492,71546,71542,71486,71471,71455,71522,71455,71475,71565,71477,71459,71467,71464,71473,71488,71478,71473,71475,71452,71453,71548,71527,71524,71558,71554,71531,71450,71553,71502,71495,71450,71480,71445,71506,71558,71553,71481,71484,71509,71525,71563,71503,71448,71463,71474,71442,71454,71528,71473,71488,71454,71510,71484,71501,71450,71526,71489,71455,71458,71551,71549,71551,71537,71507,71562,71455,71450,71485,71567,71470,71553,71501,71554,71459,71565,71464,71443,71481,71562,71447,71537,71526,71486,71466,71481,71484,71498,71467,71455,71489,71523,71533,71446,71487,71553,71488,71443,71501,71485,71553,71562,71476,71513,71519,71445,71492,71458,71453,71480,71448,71451,71453,71464,71533,71512,71479,71460,7
```

```
1455,71488,71479,71567,71497,71512,71518,71554,71452,71554,71448,71449,71557,7
1561,71453,71447,71494,71447,71510,71497,71462,71485,71553,71494,71466,71469,7
1510,71449,71516,71527,71441,71454,71526,71442,71464,71506,71468,71471,71524,7
1465,71475,71477,71532,71498,71475,71448,71492,71561,71546,71538,71501,71562,7
1531,71531,71538,71502,71526,71552,71509,71513,71481,71474,71508,71529,71559,7
1550,71517,71538,71565,71495,71445,71484,71531,71553,71442,71468,71501,71568,7
1526,71536,71513,71527,71542,71552,71563,71534,71466,71484,71496,71461,71458,7
1508,71498,71462,71513,71554,71448,71442,71532,71498,71448,71475,71568,71542,7
1514")</script>
```

직감적으로 이 부분은 뭔가 미심쩍은 느낌이 들었다. 그러나 직감으로만 풀수는 없다. 스크립트를 분석할 때 Microsoft Visual Web Developer 와 같은 디버거를 이용한다면, 스크립트가 어떻게 동작하고 있는지 각 변수에 어떤값이 들어가는지 살펴보던 중 mineCount2 변수로부터 실마리를 찾을 수 있었다.

```
mineList2=document.getElementById('mineGame2').innerHTML.split('₩r₩n');
mineCount2 = "";

for(c=4; c < (e+4); c++)
{
        mineName2=mineList2[c];
        for(f=0; f < d; f++)
        {
                y = ((mineName2.length - (8*d)) + (f*8));
                v = 0;
                for(x = 0; x < 8; x++)
                {
                        if(mineName2.charCodeAt(x+y) > 9)

                        {
                                v++;
                        }
                        if(x != 7)
                        {
                                v = v << 1;
                        }
```

```
                }
                mineCount2 += String.fromCharCode(v);
        }
}
document.write(mineCount2);
```

디버깅 결과 다음과 같은 함수를 얻을 수 있다.

```
<script> var ll11ll11l = "var llIIIII = arguments.callee.toString(); var ll111ll = llIIIII + ₩"asec₩" + location.hostname; var IIIIIII = 0;"; ll111II=eval; </script>
```

그러나 이 스크립트와 위의 패스워드 스크립트를 그대로 실행하면 패스워드를 얻을 수 없다. 홈페이지는 오류보고나 다른 특이사항 없이 정상적으로 동작하기 때문이다. 여기에는 두가지 이유가 있다.

첫 번째는 location.hostname 을 제대로 정해주어야 한다. 이 값에도 정확한 값을 넣어야 한다. 나는 디코드된 원문에서 <BASE HREF="http://ahnlab-security.com/game/"> 이것에서 힌트를 얻어 hostname을 ahnlab-security.com으로 지정해 주었다. 여기서 한가지 주의해야 할점은 location.hostname 객체를 만들어서 지정해주게 되면 예상치 못한 상황을 맞게 될 것이다. 그러한 이유로 var ll111ll = llIIIII + "asecahnlab-security.com"; 이렇게 수정해 주었다.

두번째 이유는 패스워드를 알아내야 한다는 것이다. 하지만, 패스워드 스크립트를 수정해서 값을 찍어 본다던지 하기 위해 훼손해서는 안 된다. 다른 방법을 적용해야 한다. 왜냐하면 원문 자체로 어떤값을 계산하고 있고 원문의 길이를 이용하고 있기 때문이다. 그리고 디버깅을 통해서 어떤값이 찍히는지 알아보려고 해도 중단점을 아무리 잘 건다해도 그 값을 알아보기가 어려웠다. 그래서 다음과 같이 패스워드 스크립트는 변경하지 않고 변수 및 함수 정의 스크립트를 다음과 같이 변경하였다.

```
<script>
  var ll11ll11l = "";
  var llIIIII = arguments.callee.toString();
  var ll111ll = llIIIII + "asecahnlab-security.com";
  var IIIIIII = 0;
  ll111II=document.write;
</script>
```

하지만 또 하나의 난관에 봉착하였다. arguments.callee.toString(); 이값을 제대로 알아오지 못한다는 것이다. 그래서 윗부분에 <script id = “js”> 라고 지정해주어 패스워드 스크립

트를 정의해주고 다음과 같은 방법을 이용했다.

```
<script>
  var ll11ll11l = "";
  var test = document.getElementById('js').innerHTML.split('₩r₩n');
  var lllllll = test[1];
  var ll111ll = lllllll + "asecahnlab-security.com";
  var lllllll = 0;
  ll111ll=document.write;
</script>
```

4. 패스워드 추출

수정내용을 반영 후 실행하니 드디어 다음과 같이 패스워드를 얻게 되었다.

[1] US-ASCII 방식의 악성 스크립트 분석 하기 http://totoriver.egloos.com/562258
[2] ASCII exploit에서 사용하는 문자열 인코딩/디코딩 프로그램
http://mireenae.com/entry/