

Branch: master ▾


CTFs-Writeups / picoCTF-2018 / Reversing / 06-quackme-200 / solution.md

Find file

Copy path

 ZackNoyes Fixed typo

faf48ca 26 days ago

2 contributors 

136 lines (92 sloc) 3.85 KB

Problem

Can you deal with the Duck Web? Get us the flag from this [program](#). You can also find the program in `/problems/quackme_3_9a15a74731538ce2076cd6590cf9e6ca`.

Hints:

Objdump or something similar is probably a good place to start.

Solution:

First lets download the file and try to execute it

```
wget https://2018shell1.picoctf.com/static/1d21f78fd2b82ebff2ad54a8b09081c8/main
chmod +x ./main
./main
```

```
You have now entered the Duck Web, and you're in for a honkin' good time.
Can you figure out my trick?
<INPUT>
That's all folks.
```

Lets try to understand whats going on there, we can use [IDA](#) for disassembly (or even objdump/gdb)

```

; Attributes: bp-based frame fuzzy-sp

; int __cdecl main(int argc, const char **argv, const char **envp)
public main
main proc near

var_4= dword ptr -4
argc= dword ptr 8
argv= dword ptr 0Ch
envp= dword ptr 10h

lea     ecx, [esp+4]
and     esp, 0FFFFFF0h
push    dword ptr [ecx-4]
push    ebp
mov     ebp, esp
push    ecx
sub     esp, 4
mov     eax, ds:stdout@@GLIBC_2_0
push    0
push    2
push    0
push    eax
call    _setvbuf
add     esp, 10h
sub     esp, 0Ch
push    offset aYouHaveNowEnte ; "You have now entered the Duck Web, and "...
call    _puts
add     esp, 10h
call    do_magic
sub     esp, 0Ch
push    offset aThatSAIIFolks ; "That's all folks."
call    _puts
add     esp, 10h
mov     eax, 0
mov     ecx, [ebp+var_4]
leave
lea     esp, [ecx-4]
retn
main endp

```

And do_magic() :

```

; Attributes: bp-based frame

public do_magic
do_magic proc near

var_1D= byte ptr -1Dh
var_1C= dword ptr -1Ch
var_18= dword ptr -18h
var_14= dword ptr -14h
var_10= dword ptr -10h
var_C= dword ptr -0Ch

push    ebp
mov     ebp, esp
sub     esp, 28h
call    read_input
mov     [ebp+var_14], eax
sub     esp, 0Ch
push    [ebp+var_14]
call    _strlen
add     esp, 10h
mov     [ebp+var_10], eax
mov     eax, [ebp+var_10]
add     eax, 1
sub     esp, 0Ch
push    eax
call    _malloc
add     esp, 10h
mov     [ebp+var_C], eax
cmp     [ebp+var_C], 0
jnz     short loc_8048696

```

```

sub     esp, 0Ch
push    offset aMallocReturned ; "malloc() returned NULL. Out of Memory\n"
call    _puts
add     esp, 10h
sub     esp, 0Ch
push    0FFFFFFFh
call    _exit

```

```

loc_8048696:
mov     eax, [ebp+var_10]
add     eax, 1
sub     esp, 4
push    eax
push    0
push    [ebp+var_C]
call    _memset
add     esp, 10h
mov     [ebp+var_1C], 0
mov     [ebp+var_18], 0
jmp     short loc_804870B

```

```

loc_804870B:
mov     eax, [ebp+var_18]
cmp     eax, [ebp+var_10]
jnl     short loc_80486BD

```

```

loc_80486BD:
mov     eax, [ebp+var_18]
add     eax, 8048858h
movzx   ecx, byte ptr [eax]
mov     edx, [ebp+var_18]
mov     eax, [ebp+var_14]
add     eax, edx
movzx   eax, byte ptr [eax]
xor     eax, ecx
mov     [ebp+var_1D], al
mov     edx, greetingMessage
mov     eax, [ebp+var_18]
add     eax, edx
movzx   eax, byte ptr [eax]
cmp     al, [ebp+var_1D]
jnz     short loc_80486EF

```

```

add     [ebp+var_1C], 1

```

Method 2 - Brute-Forcing (like a retard)

We can see that each successful comparison increased var_1C by one.

This is done in the instruction `add [ebp+var_1C], 1` which is at address 0x80486eb in the binary.

We can set a breakpoint there, brute-force one byte, and count hits (each time increase the amount of deired hits).

Using this ugly code:

```
#!/usr/bin/env python

import gdb
import string

class MyBreakpoint(gdb.Breakpoint):
    def stop (self):
        global count

        count += 1

        return False

gdb.execute('file ./main')

# Suppress output
gdb.execute('set logging file /dev/null')
gdb.execute('set logging redirect on')
gdb.execute('set logging off')
gdb.execute('set print inferior-events off')

bp = MyBreakpoint("*0x80486EB")

count = 0
flag = 'pico'

while flag[-1] != '}':
    for c in string.ascii_lowercase + string.ascii_uppercase + string.digits + '!@#%&*(){}_':
        count = 0

        gdb.execute('run <<(echo "{}") > /dev/null'.format(flag + c), to_string=True)
        if count > len(flag):
            flag = flag + c

            print ('Partial flag: {}'.format(flag))

            break

gdb.execute('quit')
print ('Flag: {}'.format(flag))
```

Note: I'm sure there is a proper way doing this with pwntools, pls tell me if you know how

Method 3 - Brute-Forcing (like a boss)

Working with [angr](#).

Installation

```
sudo apt-get install python3-dev libffi-dev build-essential virtualenvwrapper
pip install angr
```

Solution

TODO

Flag: picoCTF{qu4ckm3_7ed36e4b}