HackTM 2020 - PLOP (488pt)

FEBRUARY 5, 2020
REVERSING

PLOP (488pt)

`Reversing`

Description: Author: trupples

I've been playing around with mathy obfuscation, see if you can break this one!

P.S. there are multiple "flags" the binary would say are correct, but only one of them matches the flag format.

Files:

- `PLOP`

Solution:

This binary does some tricky things with signal handlers to obfuscate the control flow. To run it in gdb we need to do something like:

```
handle SIGSEGV noprint nostop pass
handle SIGALRM noprint nostop pass
```

We eventually find that `fini_3` sets up a `SIGSEGV` handler that acts as an unpacker. Essentially, this handler is triggered with information about the faulting instruction and it will decode part of the subsequent code and return execution.

In `fini_2` we hit a faulting instruction and trigger the handler. This happens several more times recursively and during each call we check `8` bytes of our input. The first unpacked segment looks like:

```
0x7ffff7ff7000:    mov    rax,QWORD PTR ds:0x1337000
0x7ffff7ff7008:    rol    rax,0xe
0x7ffff7ff700c:    movabs rdx,0xdc3126bd558bb7a5
0x7ffff7ff7016:    xor    rax,rdx
0x7ffff7ff7019:    mov    r8,QWORD PTR ds:0x1337080
0x7ffff7ff7021:    cmp    r8,0x0
0x7ffff7ff7025:    jne    0x7ffff7ff7031
0x7ffff7ff7027:    mov    QWORD PTR ds:0x1337080,rax
0x7ffff7ff702f:    jmp    0x7ffff7ff7051
0x7ffff7ff7031:    cmp    rax,QWORD PTR ds:0x1337080
0x7ffff7ff7039:    mov    bx,WORD PTR ds:0x1337064
```

```
0x7ffff7ff7041:    mov    ax,0x1
0x7ffff7ff7045:    cmovne bx,ax
0x7ffff7ff7049:    mov    WORD PTR ds:0x1337064,bx
```

We can encode all of these constraints into the following, *beautiful* z3 script and get the flag:

```
from z3 import *
import binascii

s = Solver()

a = BitVec('a', 64)
b = BitVec('b', 64)
c = BitVec('c', 64)
d = BitVec('d', 64)
e = BitVec('e', 64)
f = BitVec('f', 64)
g = BitVec('g', 64)
i = BitVec('i', 64)

t = RotateLeft(a, 0xe) ^ 0xdc3126bd558bb7a5
s.add(b == RotateRight(t ^ 0x76085304e4b4ccd5, 0x28))
h = RotateLeft(b, 0x28) ^ 0x76085304e4b4ccd5
s.add(RotateLeft(c, 0x3e) ^ 0x1cb8213f560270a0 == h)
s.add(RotateLeft(d, 2) ^ 0x4ef5a9b4344c0672 == h)
s.add(e == RotateRight(h ^ 0xe28a714820758df7, 0x2d))
h = RotateLeft(e, 0x2d) ^ 0xe28a714820758df7
s.add(RotateLeft(f, 0x27) ^ 0xa0d78b57bae31402 == h)
v = 0x4474f2ed7223940
v = ((v << 0x35) | (v >> (64-0x35))) & 0xffffffffffffffff
s.add(RotateRight(v ^ g, 0x35) == h)
```

```
s.add(RotateRight(h^0xb18ceeb56b236b4b, 0x19) == i)
h = RotateLeft(i, 0x19) ^ 0xb18ceeb56b236b4b

s.add(Extract(7,0,a) == ord('H'))
s.add(Extract(15,8,a) == ord('a'))
s.add(Extract(23,16,a) == ord('c'))
s.add(Extract(31,24,a) == ord('k'))
s.add(Extract(39,32,a) == ord('T'))
s.add(Extract(47,40,a) == ord('M'))
s.add(Extract(55,48,a) == ord('{'))
s.add(Extract(63,56,a) == ord('P'))

def pp(t):
    return binascii.unhexlify(hex(t)[2:].zfill(16))[::-1]

s.check()
m = s.model()

print(''.join([pp(m[x].as_long()) for x in [a,b,c,d,e,f,g,i]]))
```

`HackTM{PolynomialLookupOrientedProgramming_sounds_kinda_shit_xd}`

hgarrereyn

ALSO ON **HGAR**

2 years ago • 3 comments

CTF writeups

8 months ago • 1 comment

CTF writeups

a year ago • 2 comments

CTF writeups

Read more

MidnightSun CTF 2021 - twi-light (428pt / 4 solves)	Apr 10 2021
Google CTF - Exceptional (363pt / 10 solves)	Aug 23 2020
Google CTF - Registers Matter (347pt / 12 solves)	Aug 23 2020
Samsung CTF - Legitimate (240pt)	Aug 18 2020
PoseidonCTF 2020 - VM & VM Reloaded (1000+1000pt)	Aug 9 2020
ASIS CTF 2020 - sshateau (285pt)	Jul 5 2020
0CTF/TCTF 2020 - sham (733pt)	Jun 29 2020
redpwnCTF 2020 - JavaIsEZ2 (497pt)	Jun 25 2020
redpwnCTF 2020 - r1sc (487pt)	Jun 25 2020
PlaidCTF 2020 - That's a lot of fish (400pt)	Apr 19 2020