



# 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 permitted 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>



## SMB Shares and Worms – A Parasitic Relationship?

*An analysis of the W32/Deborm.worm.q*

Practical Assignment  
Version 3 (revised July 24, 2003)

GIAC Certified Incident Handler (GCIH)

Ken Ramsay

Date March 29, 2004

## Table of Contents

Summary .....	6
Introduction .....	6
Statement of Purpose .....	7
The Exploit .....	9
Name .....	9
W32/Deborm.worm.q .....	9
The following CVE candidates are applicable .....	9
Operating Systems Affected .....	10
Systems Not Affected .....	10
Protocols/Services/Applications .....	10
Address Resolution Protocol .....	10
TCP Connection Setup .....	12
Server Message Block\ CIFS \ NetBIOS session service .....	13
Variants .....	21
Example differences with the W32/Deborm-R .....	22
W32/Deborm-Q – according to Sophos .....	23
Alias .....	23
Description .....	24
Signatures of the Attack .....	25
Network .....	25
Host Signatures .....	30
Host Modifications .....	32
Windows Event Logs .....	35
The Platforms/Environments .....	36
The Victim .....	36
The Source .....	36
The Source Network .....	36
The Target Network .....	37
Reconnaissance .....	39
Scanning .....	40
Exploiting the System .....	43
Exploit Packets 260 to 262 .....	43
Exploit Packets 263 to 265 .....	44
Exploit Packets 266 to 268 .....	44
Exploit Packets 269 to 270 .....	44
Exploit Packets 271 to 272 .....	44
Exploit Packets 273 to 274 .....	44
Exploit Packets 275 to 292 - Failure .....	44
Exploit Packets 273 to 281 – Successful Worm transfer ! .....	46
Payload .....	46
Keeping Access .....	48
Covering Tracks .....	49
The Incident Handling Process .....	50

Incident background .....	50
Preparation.....	51
Identification .....	55
Containment .....	57
Eradication .....	61
Recovery .....	64
Lessons Learnt.....	64
Technical.....	65
Social .....	66
Extras .....	66
Conclusion .....	67

## Table of Figures

Figure 1 Simple ARP request/response followed by http request .....	11
Figure 2 Simple TCP connection setup.....	13
Figure 3 SMB is protocol independent .....	14
Figure 4 SMB Protocol Negotiation Request to Server .....	15
Figure 5 SMB Protocol Dialect Choice- LANMAN2.1 .....	15
Figure 6 Connection to system hidden share.....	16
Figure 7 Null Session request to \\10.1.1.201\IPC\$.....	17
Figure 8 Null session established.....	18
Figure 9 Connection request from Administrator to \\10.1.1.201\C.....	19
Figure 10 Successful connection to \\10.1.1.201\C .....	20
Figure 11 File creation attempt .....	20
Figure 12 Path not found. ....	20
Figure 13 SMB file create success.....	21
Figure 14 Top ten attacked ports March 2004 .....	24
Figure 15 Incremental ARP Scan.....	25
Figure 16 SMB Connection attempt - Owner .....	26
Figure 17 SMB file transfer attempt to three locations .....	28
Figure 18 Trigger/Filter for pattern ~2.exe .....	28
Figure 19 NetBIOS name service request SON.ATH.CX.....	29
Figure 20 NetBIOS name service request SON.GLINED.US.....	29
Figure 21 Connection attempt NAs33Attack3r.....	30
Figure 22 Task Manager showing CPU at 100% .....	30
Figure 23 Worm process list .....	31
Figure 24 Worm host propagation.....	31
Figure 25 WhatChanged for Windows sample screen .....	32
Figure 26 Explore .exe 12,832 bytes .....	33
Figure 27 Registry modification.....	33
Figure 28 SVCHOST.exe 17,440 bytes .....	33
Figure 29 Registry modification SVCHOST.exe.....	33
Figure 30 Security Event Log.....	34
Figure 31 Auditing login failure.....	34
Figure 32 Source network schematic.....	37
Figure 33 GIAC regional office demo network .....	38
Figure 34 Zone Alarm Zone setup .....	39
Figure 35 Abnormal ARP behavior .....	41
Figure 36 Continuing ARP scan.....	41
Figure 37 Non sequential scanning.....	42
Figure 38 ARP response (packet 259).....	43
Figure 39 TCP Connection and tear down.....	43
Figure 40 LanMan 2.1 SMB dialect negotiation .....	44
Figure 41 Null Session.....	44

Figure 42 SMB connection to C share .....	45
Figure 43 Successful file creation and transfer. ....	46
Figure 44 TCPView capture of SVCHOST.exe .....	48
Figure 45 netstat -an.....	50
Figure 46 GIAC Global Security Website.....	51
Figure 47 GIAC Internet policy.....	52
Figure 48 Regional Security Team Contact Details .....	53
Figure 49 CPU 100% Utilization.....	57
Figure 50 ~2.exe rogue process .....	57
Figure 51 Ghost Image Raw configuration.....	58
Figure 52 Incremental ARP requests .....	59
Figure 53 Sysinternals Autoruns .....	60
Figure 54 Cloned worm files .....	62
Figure 55 File Print share removal .....	63
Figure 56 Zone Alarm settings.....	63
Figure 57 Determining Shared folders .....	67

## Summary

This document will examine the business and technical ramifications of a variant of the Deform worm. This incident actually happened in a real business environment and more or less had the business repercussions detailed. The fact that this was not a very powerful worm is a lesson to all that the Internet is, as Tom Cruise said in Top Gun, a "target rich environment". No matter how simple the exploit, there are billions of targets out there to choose from. This paper will help the reader understand the worm lifecycle and how to defend against the various strategies they use to move around networks and invade host machines. This worm studied is simple but the lessons learned can be applied to the more complex worms that are appearing today.

## Introduction

Microsoft has been criticized by information security specialists around the globe for not doing enough to secure their default software configurations. The same people also criticize them as being slow to react when informed of new software vulnerabilities. Occasionally, patches are released long after they were informed of the problem. The recent ASN.1 vulnerability patch<sup>1</sup> was released about 8 months after the problem was first found<sup>2 3</sup>.

Microsoft's domination of the market has been achieved by making the software setup as simple as possible for the average user. Their philosophy has been to enable as many functions as possible so the average consumer will be up and running as quickly and painlessly as possible. By default, Windows machines that have the networking component installed, will try their best to let every other computer on the network know that they are up and running. This makes the sharing of file systems and printers very simple to achieve. Windows XP even takes this one step further by running a service called Universal Plug and Play (UPnP). This is another set of complex protocols to enable peer to peer networking that is enabled by default and in almost never used! The service has been shown to be vulnerable to a number of exploits.<sup>4 5</sup>

So, by default, Windows will very kindly share, in many cases, the contents of their complete hard drive with anyone who knows or can figure out the relevant password. Further still, if that computer is on a network connected to the Internet then the whole world may possibly be able to connect to any advertised

---

<sup>1</sup> <http://www.microsoft.com/technet/security/bulletin/MS04-007.msp>

<sup>2</sup> Press release on time lag to release patch <http://www.eeye.com/html/Press/PR20040210.html>

<sup>3</sup> Security issues still awaiting a patch <http://www.eeye.com/html/Research/Upcoming/index.html>

<sup>4</sup> Steve Gibson discussion on the subject and removal tool <http://www.grc.com/unpnp/unpnp.htm>

<sup>5</sup> Bruce Schneier's CryptoGram Discussion of UPnP (Jan 2002) <http://www.schneier.com/crypto-gram-0201.html#1>

resources. Once connected, files can be uploaded or downloaded almost as simply as sitting in front of the machine.

Even today, there is a popular misconception by naïve computer users that no one could possibly be interested in their machine(s). They are under the impression that they will not be targeted among the multi millions of computers on the Internet. Unfortunately for them – and indeed all of us – the global reach of the Internet, the automation of the attack and the almost promiscuous nature of Windows, has led to every escalating computer compromises. Now that 24/7 Internet connection is commonplace in some areas, the conditions could not be much better for the propagation of malware.

**“Malware** (for **“malicious software”**) is programming or files that are developed for the purpose of doing harm. Thus, malware includes computer viruses, worms, and Trojan horses.”<sup>6</sup>

## Statement of Purpose

This paper has been written to examine how a worm makes use of standard network configurations to propagate. The worm chosen, W32/Deborm.worm.q, is not that well known but was captured after invading a real network. This paper will not examine code but will examine the effects of the code in order to understand what is happening at the data transmission level. Reverse engineering malicious code is a time consuming and laborious process. Examining **all the effects** of the code is almost as valid an approach to understanding the operation the code – the tricky part is in capturing all the symptoms. This approach is analogous to a doctor treating an infection. An examination of the patient’s symptoms will result in recognition of the infection and a course of treatment will be prescribed. Of course, it should be borne in mind that sometimes a slightly mutated strain will resist the treatment!

So, this paper will look at what network traffic the worm creates and how it infests a host in the first place. The techniques used to further propagate once inside a host will also be examined. All the changes that occur within the infected machine such as file system additions and registry changes will be analyzed.

In particular, the paper will look at the way worms use the Microsoft implementation of Server Message Block protocol (SMB) to setup connections with the target and propagate the infection.

The approach taken in the paper is to use a worm-infected machine that actually caused real business disruption. This is not one of the more common worms but techniques used and the lessons learned are generic and can be used to defend against future attacks of this type.

The lab environment was very simple and consisted of only three machines – the infected machine, the sacrificial victim and a Windows machine loaded with a

---

<sup>6</sup> Malware Definition from <http://searchsecurity.techtarget.com/>



network analyzer.<sup>7</sup> This clean setup facilitated immediate detection of abnormal activity.

The only software added to the victim was a commercial application<sup>8</sup>, which was used to base line the file system and registry so that any changes would be immediately apparent. Without this base line knowledge it is extremely difficult – not to mention error prone -to find changes to a system.

The infected machine was added to the lab network last and powered on so the attack could commence. The first network capture revealed that the attack exploit had failed. This was not really expected but the analysis also revealed what the attacker was trying to achieve and hence, why it failed. So, armed with this knowledge, the sacrificial machine was made vulnerable to the exploit so that the next stage could be monitored. This time the infected machine succeeded with the exploit – the worm had propagated. Only this time it was captured and inside an observation cage!

The next stage was to determine what effect the worm's payload would have on the host. The worm was therefore deliberately launched. This was done and the worm immediately produced the same network traffic as seen earlier but also this time altered a file system and registry that had been base-lined.

Finally, the network traffic and the comparison of the file-system changes against the baseline, allowed identification of the worm and therefore access to further available research.

This experimental process is, by design, long winded to reveal how some worms use standard network protocols and poor configuration to survive and breed. This is a pretty lame worm and simple to defend against but remember evolution of malware mimics real life – these things are only going to get more creative and so cause more damage. The more we can learn about the whole process the better prepared we will be for the next generation. The rest of the paper will examine the exploit in detail and then go through the Incident Handling Process.

---

<sup>7</sup> Analyzer used : WildPackets Etherpeek [http://www.wildpackets.com/products/etherpeek\\_nx](http://www.wildpackets.com/products/etherpeek_nx)

<sup>8</sup> WhatChanged? For Windows V3.0 Prism Microsystems, Inc. <http://www.whatChanged.com>

# The Exploit

## *Name*

### **W32/Deborm.worm.q**

#### **Alias**

Worm.Win32.Deborm, W32.Deborm.Worm,  
Win32/Deborm.Q.Worm, TROJ\_DROPPERFL.A (Trend),  
W32.HLLW.Deborms.C (Symantec), Worm.Win32.Deborm.q (Kaspersky)

**Infection Length:** 56,329 bytes

This worm exploits a single Windows share that has no password using the native Server Message Block (SMB) protocol. It is a member of a family of worms for which there is no specific CVE number. However, there are a number of relevant CVE candidates that are generically applicable.

### **The following CVE candidates are applicable**

[CAN-1999-0504](#)   [CAN-1999-0505](#)   [CAN-1999-0506](#)   [CAN-1999-0519](#)

[CAN-1999-0520](#)

Cert Advisory about Windows shares with Null or poor passwords

<http://www.cert.org/advisories/CA-2003-08.html>

Advisory from Internet Security Systems

<http://xforce.iss.net/xforce/xfdb/19> SMB share writable by Everyone

F-Secure have a good generic write up on the Deborm family of worms at

<http://www.f-secure.com/v-descs/deborm.shtml>

Pest Patrol also describe more of the family

[http://pestpatrol.com/pestinfo/w/worm\\_win32\\_deborm.asp](http://pestpatrol.com/pestinfo/w/worm_win32_deborm.asp)

A more specific write up is on the Network Associates site

[http://vil.nai.com/vil/content/v\\_100234.htm](http://vil.nai.com/vil/content/v_100234.htm) but it is not quite 100% accurate. The startup hook mentions "NAV Live Update" and this was not correct for the worm studied. This directory is not created in the worm studied. This indicates that the worm examined may be a minor variant, which is not that unusual as once the exploit code is in the wild, it is often "tweaked". This is the main way that worms evolve. (Note. All other aspects of the description are accurate apart from the startup hook description. These will be discussed in depth later).

Computer Associates also adds some more detail about the files that are dropped as part of the payload.

<http://www3.ca.com/threatinfo/virusinfo/virus.aspx?id=14636>

## **Operating Systems Affected**

Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows Me

This worm uses Windows SMB file shares to propagate and so all versions of Windows from Windows 95 onwards are vulnerable.

Service pack load and patch status are also irrelevant for the same reason.

## **Systems Not Affected**

Windows 3.x, Macintosh, OS/2, UNIX, Linux

(Note. The attack does not apply to UNIX SMB implementations such as SAMBA for Linux.)

## **Protocols/Services/Applications**

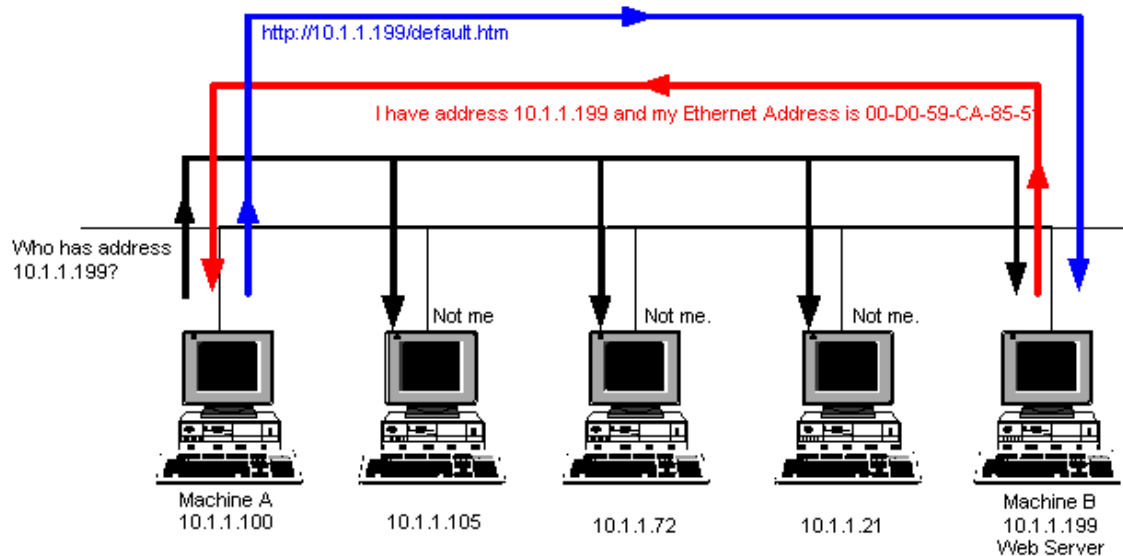
This exploit uses the SMB, ARP & TCP Protocol. A basic knowledge of these protocols is necessary to understand the exploit. However the complete protocols will not be covered here only the parts relevant to the exploit.

### **Address Resolution Protocol**

Later in the paper, we shall see that the first part of the exploit is that the worm has to find out which machines are switched on before it can proceed with the next stage of an attack.

Before a machine can communicate at the logical IP level it needs to be able to talk at the hardware interface level. Every network card has a unique 48 bit number assigned to it by the manufacturer. This is known as the Ethernet or MAC Address. This number is (supposedly) globally unique and (normally) cannot be changed. Computers connected to an Ethernet network need to know the MAC address of the other card before they can communicate.

When a network card sees an Ethernet frame with its own MAC address the device driver captures the frame. It then passes it to upper layers for further processing. That is all it has to do - a device driver has no concept of an IP address, it only works at the wire level. Therefore, in a purely IP network, a computer has two addressing strategies. The conundrum is how can two machines ever talk at the IP level when one initially has no idea of another's Ethernet address? The solution is the aptly named Address Resolution Protocol (ARP). This primarily performs a mapping of IP to Ethernet address.



**Figure 1 Simple ARP request/response followed by http request**

To understand how this is accomplished see Figure 1. Machine A has just been switched on and has an IP address of 10.1.1.100. It wants to get a web page from a machine with an address of 10.1.1.199. To do this it needs to know the MAC address first. So, machine A sends out a broadcast ARP to all machines requesting that the machine that has IP address 10.1.1.199 respond with its MAC address. The nature of a broadcast frame is that all machines on the network examine the packet to see if it can answer the query. So the device driver of EVERY machine on the network will capture and pass the query to a higher level. Only machine B will recognize that it has IP address 10.1.1.199 and so responds back to A with its Ethernet address. A will now be able to communicate directly with B.

A point to note here is that all computers on the network will use a small amount of computer resource to examine the packet. A flood of broadcast packets is an undesirable condition for this very reason – it takes up a lot of distributed resource.

Another common example of ARP is when an IP address is changed on a running system. For example, after making a dynamic change in the Windows Network Properties dialogue box a broadcast ARP is sent to find out if anyone else has already been assigned that address. If it sees a reply an error message will be displayed in a dialogue box saying that it has detected a duplicate IP address.

## ***TCP Connection Setup***

The initial part of the exploit requires a TCP connection to be setup. After finding out the MAC address of the victim, the attacker probes a little deeper to find out if the machine is listening on TCP port 139 – NetBIOS session services. The method this particular exploit uses, and there a few others it could have chosen, is to perform a TCP connection.

TCP is a connection orientated protocol that attaches a sequenced number to each TCP segment to ensure that information is not lost during a connection session. The TCP client also states which port it wants to connect to at setup time and so the server must have a service listening on that port to proceed with the setup.

Very simplistically, it does this by both machines stating at the outset what sequence number (ISN) they will initially attach to any transmitted data. This number is incremented for each TCP segment sent and an acknowledgment sent back to indicate a successful receipt. After transmitting a timer is started and a reply is required before a time-out. In this way the sender knows whether to resend a segment or not. TCP is a lot more complicated than this but no more knowledge is really required to understand the exploit.

A tear down process is used to close a connection so that the resources are freed up for other uses. In some versions of TCP this property was exploited to perform a denial of service attack. Multiple connections were made and never broken. In some TCP implementations the timeout was four minutes and resources would be held for this period. Connections would be made from one or more clients until eventually there would be no more left i.e. a denial of service.

This connection setup and tear down process is accomplished by the use of a number of control bits (flags) contained in the TCP segment header. There are six bits but the flags we need to know are

SYN is a request to synchronize sequence numbers  
ACK acknowledges previous transmissions  
FIN states that there will be no more data

Analyzing a connection is slightly complicated by the fact that this is a two way (duplex) connection so multiple flags and two sets of sequence numbers may be contained in each header.

A simplified view of the connection setup process is shown in Figure 2 below.

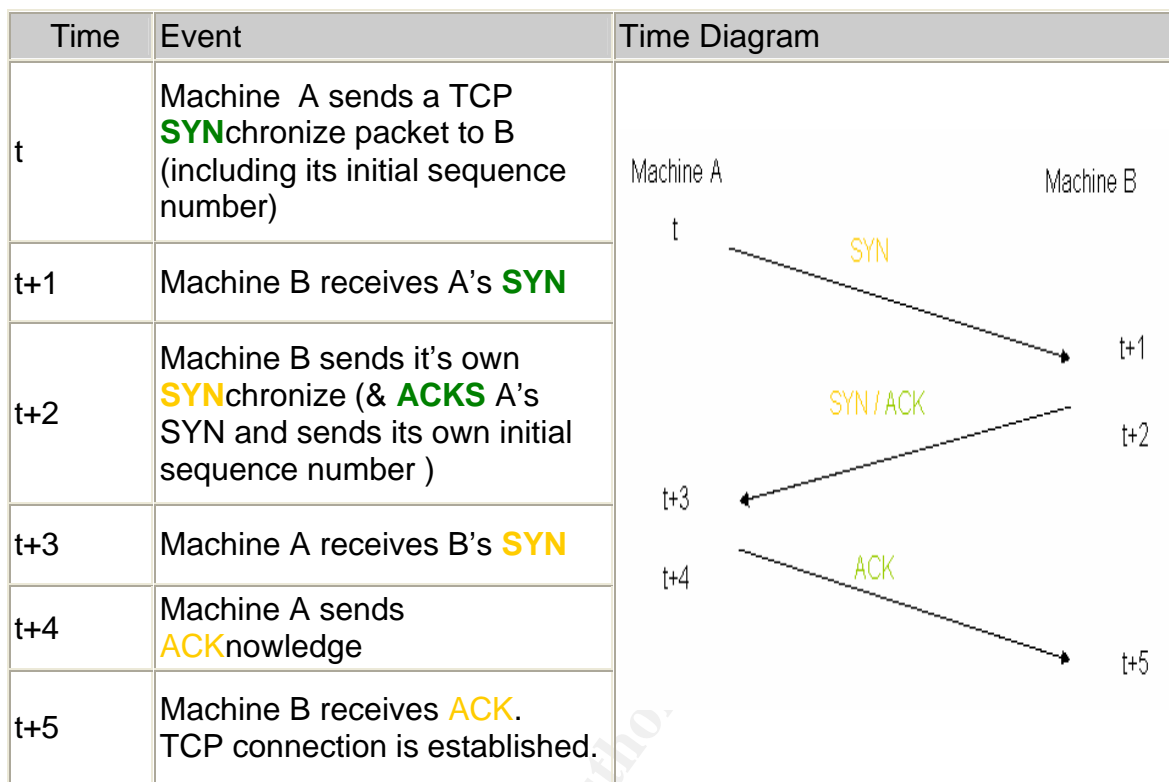


Figure 2 Simple TCP connection setup<sup>9</sup>

## Server Message Block\ CIFS \ NetBIOS session service<sup>10</sup>

[RFC 1001](#) defines a NetBIOS session as:

"A session is a reliable message exchange, conducted between a pair of NetBIOS applications. Sessions are full-duplex, sequenced, and reliable."

A NetBIOS session on port 139 therefore has many similarities in function to a TCP connection. The study of our worm requires no further knowledge of NetBIOS.

The SMB protocol has been around since the eighties and was first developed by IBM. Microsoft and Intel developed it further and in 1996 Microsoft renamed it as a mainly marketing play to Common Internet File System (CIFS). Although quite powerful, it is starting to show its age. SMB could be described as the heart of all Microsoft networking.

The normal function of SMB/CIFS is to allow users access to networked resources such as file shares and printers. It has been around a long while and

<sup>9</sup> Template from : <http://www.inetdaemon.com/tutorials/internet/tcp/connections.html>

<sup>10</sup> Information sourced from [Implementing CIFS - The Common Internet FileSystem](#) Christopher R. Hertel ; Prentice-Hall 2003

has had many updates along the way. To manage these entire different dialects one on the first things done when attempting an SMB connection is to negotiate which dialect to talk. Other computers also use SMB to create browse lists etc.

OSI					TCP/IP
Application	SMB				Application
Presentation					
Session	NetBIOS	NetBEUI	NetBIOS	NetBIOS	
Transport	IPS		DECnet	TCP&UDP	TCP/UDP
Network			IP	IP	
Link	802.2 802.3,802.5	802.2 802.3,802.5	Ethernet V2	Ethernet V2	Ethernet or Others
Physical					

**Figure 3 SMB is protocol independent<sup>11</sup>**

SMB is an application/presentation layer protocol that was designed to be transport independent (Figure 3). It commonly uses NetBIOS over TCP/IP to establish and maintain point-to-point, connection-oriented sessions over TCP port 139. This is the default configuration for Windows NT. Any new NT machine connected to a network will automatically listen for SMB connections on this port.

Windows 2000 and Windows XP added another SMB listening service on port 445 in addition to port 139. This removed the need to use NetBIOS (but this is STILL used for file and print sharing by default). This addition allowed SMB to sit directly on top of TCP and eliminate the need for NetBIOS.

Add Internet connectivity to a network and SMB shares can be found from anywhere on the Internet.

Once the TCP connection is made to Port 139 (NetBIOS over TCP) or port 445 (no NetBIOS) the SMB dialogue can commence. As SMB has been around such a long time a number of manufacturers have modified the protocol. Because of this there are various implementations called dialects. Therefore to at the outset of a connection the two machines must first negotiate which the dialect they both understand. This, the rest of SMB relevant to the exploit will be explained using cut down network traces captured by Etherpeek. Note that Figure 4 and subsequent SMB packets have had SMB Flags and some other irrelevant fields, such as checksums, removed for clarity. The full exploit traffic is included in Appendix 1.

Figure 4 shows the client, in the initial request, making a SMB negotiate dialogue request telling the server what SMB dialects it understands i.e. all dialects. The

<sup>11</sup> Diagram Courtesy of ASL GROUP [http://www.smb-analyser.co.uk/content/multiple\\_protocols.htm](http://www.smb-analyser.co.uk/content/multiple_protocols.htm)

purpose of the message is identified by the assigned command code, which is 114 in this case.

SMB - Server Message Block

Protocol ID: SMB  
Command Code: 114 *Negotiate Protocol*  
Tree ID (TID): 0x0000  
Process ID (PID): 0xCAFE  
User ID (UID): 0x0000  
Multiplex ID (MID): 0x0000

SMB Negotiate Protocol

Transaction Type: 0 *Request*

Dialect #1: PC NETWORK PROGRAM 1.0.  
Dialect #2: XENIX CORE.  
Dialect #3: MICROSOFT NETWORKS 1.03.  
Dialect #4: LANMAN1.0.  
Dialect #5: Windows for Workgroups 3.1a.  
Dialect #6: LM1.2X002.  
Dialect #7: LANMAN2.1.  
Dialect #8: NT LM 0.12.

**Figure 4 SMB Protocol Negotiation Request to Server**

SMB - Server Message Block

Protocol ID: SMB  
Command Code: 114 *Negotiate Protocol*  
Tree ID (TID): 0x0000  
Process ID (PID): 0xCAFE  
User ID (UID): 0x0000  
Multiplex ID (MID): 0x0000

SMB Negotiate Protocol

Transaction Type: 1 *Response*  
Word Count: 17  
Index: 7  
OEM Domain Name: WORKGROUP.SACRIFICIAL1.

**Figure 5 SMB Protocol Dialect Choice- LANMAN2.1**

The server tells the client which protocol to use by setting the index field (Figure 5) The SMB protocol chosen here is LANMAN2.1, which is Dialect or Index 7.



After the protocol negotiation a null session is requested. A null session, is sometimes referred to as the “Holy Grail” of Windows hacking.<sup>12</sup> Simplistically, this feature exists to let other computers know what resources are available. Unfortunately it divulges a bit too much information that can be used to set up an attack. This can be done from a command line (Figure 6Figure 1). Note the successful completion to with the blank user /U:””.

This subject is covered in other GIAC practical assignments<sup>13 14</sup> and as it is not really the main area exploited in this attack, and will not be examined any further.

For a Logon session setup (which must include the user’s logon credentials) the command code is 115. Note that in a null session there is no Account name and the password is blank. Note also that the Tree ID (TID) and User ID (UID) are blank. The TID is a 16 bit number that is allocated by the server to identify the resource that a particular packet is referring to (The term tree refers to the **directory tree** as a shared resource is normally a directory) The UID identifies the client for that SMB session and is also issued by the server. Thus the receipt of a TID and UID indicate a successful SMB session connection. In order to complete the connection the server needs to supply these as they are used for all future communication for that session. Figure 7 shows the null session request which contains no Account Name or password and is targeted to the hidden IPC\$ system share.

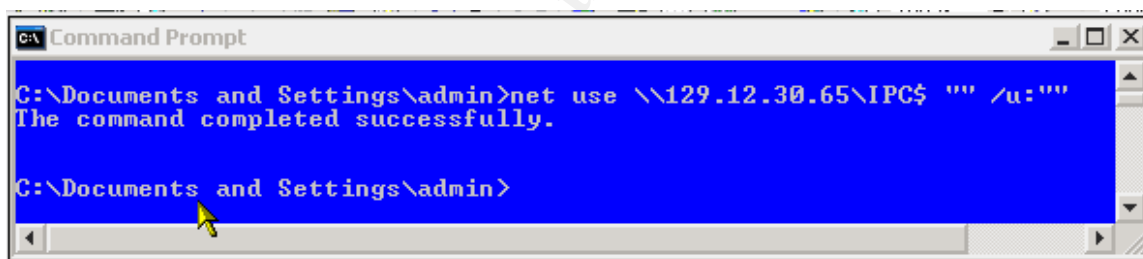


Figure 6 Connection to system hidden share

<sup>12</sup> Hacking Exposed; Stuart McClure, Joel Scambray, George Kurtz McGraw-Hill Osborne Media; (Third Edition 2003) <http://www.hackingexposed.com/>

<sup>13</sup> More info on Null Sessions from past GIAC practicals:

Michael S. Kriss “Weak Passwords + Null Session = Windows 2000 Exploit”

[http://www.giac.org/practical/Michael\\_Kriss\\_GCIH.doc](http://www.giac.org/practical/Michael_Kriss_GCIH.doc)

NULL Sessions In NT/2000 [www.sans.org/rr/papers/67/286.pdf](http://www.sans.org/rr/papers/67/286.pdf) Joe Finamore

<sup>14</sup> Lloyd Conner provides good information on attacking Port 139 (GIAC practical)

[http://www.giac.org/practical/Lloyd\\_Conner\\_GCIH.doc](http://www.giac.org/practical/Lloyd_Conner_GCIH.doc)

#### SMB - Server Message Block

Protocol ID: SMB  
**Command Code:** 115 *Session Set Up And X (Including User Logon)*  
Error Code Class: 0x00 *Success*  
Reserved: 0x00  
Error Code: 0 *Success*  
**Tree ID (TID):** 0x0000  
Process ID (PID): 0xCAFE  
**User ID (UID):** 0x0000  
Multiplex ID (MID): 0x0000

#### SMB Session Set Up & X (Including User Logon)

Transaction Type: 0 *Request*  
Word Count: 13  
Secondary command: 0x75 *Tree Connect And X*  
AndX reserved (MBZ): 0x00  
AndX offset: 132  
Session key: 0x00000000  
Case insensitive pw length (ansi): 1  
Case sensitive pw length (unicode): 0  
Reserved (MBZ): 0x00000000  
Case insensitive pw (ansi): 0x00  
Case sensitive pw (unicode):

#### **Account Name:**

Primary Domain Name:  
Native OS1: Windows NT 1381

#### SMB Tree Connect and X

Transaction Type: 0 *Request*  
Word Count: 4  
Secondary Command: 0xFF *No More Commands*  
Reserved (MBZ): 0x00  
Offset To Command: 0  
Flags: 0x0000  
Password Length: 1  
Byte Count: 43  
Net-Name Password: 0x00  
File Pathname: \\10.1.1.201\IPC\$

Figure 7 Null Session request to \\10.1.1.201\IPC\$

#### SMB - Server Message Block

Protocol ID: SMB  
Command Code: 115 *Session Set Up And X (Including User Logon)*  
Error Code Class: 0x00 *Success*  
Reserved: 0x00  
Error Code: 0 *Success*  
**Tree ID (TID):** 0x0800  
Process ID (PID): 0xCAFE  
**User ID (UID):** 0x0800  
Multiplex ID (MID): 0x0000

#### SMB Session Set Up & X (Including User Logon)

Transaction Type: 1 *Response*  
Word Count: 3  
Secondary command: 0x75 *Tree Connect And X*

```

Reserved:          0x00
AndX offset:       135
Action:            0x0000
Byte Count:        94
Native OS:         .....
Native Lan Man:    .....
Primary Domain Name: .....
Remaining SMB Data: ..IPC... 03 FF 00 96 00 01 00 06 00 49 50 43 00 00 00

```

**Figure 8 Null session established**

Here in Figure 8 we see that the null session has been granted. TID 0x0800 and User ID 0x0800 have been allocated.

The exploit was pre-programmed to establish a null session and then only to look for a certain network share. Once again a SMB command code of 115 is used to signify that this is a logon session request message. Let's look at this step now.

In this message the user is Administrator and no password was sent. Note that as the password authentication method uses the NTLMv2 challenge response mechanism it is not possible to sniff the password.

It can also be seen that at this point there is no TID or UID and the request is for a share called "C". A windows share is a logical reference to a real drive location and the share "C" may or may not be a reference to C:\.

At this point there are three possible replies.

- User authentication failure
- Authentication success but invalid share name
- Success

Figure 10 details a successful connection. Note the TID and UID are **0x801**. These will be used from now on to identify the session.

The last piece of SMB to be understood is file transfer. At this point a SMB session has been established and TID & UID received. To transfer a file the full path must however be known. At this point the attacker has received share level authentication but has yet to actually access the C share. The reason this is important is because the worm needs to know the path to a startup directory to infect the victim on reboot. Windows has not been consistent in creating startup paths in the various versions since Windows 95. As this attack is completely automated the worm needs to try to drop its payload file on three different windows paths. This will be examined in later when the actual exploit is run through. For now let's just look at a failure message and successful SMB file create message.

Figure 11 shows an attempt to create a file called `worm.exe` on a subdirectory `\WINNT\Profiles\AllUsers\StartMenu\Programs\Startup\` of the C share. This uses the SMB command code 162.

#### SMB - Server Message Block

Protocol ID: SMB  
**Command Code:** 115 *Session Set Up And X (Including User Logon)*  
Error Code Class: 0x00 *Success*  
Reserved: 0x00  
Error Code: 0 *Success*  
Tree ID (TID): 0x0000  
Process ID (PID): 0xCAFE  
User ID (UID): 0x0000  
Multiplex ID (MID): 0x0010

#### SMB Session Set Up & X (Including User Logon)

Transaction Type: 0 *Request*  
Word Count: 13  
Secondary command: 0x75 *Tree Connect And X*  
AndX reserved (MBZ): 0x00  
AndX offset: 230  
Session key: 0x00000000  
Case insensitive pw length (ansi): 24  
Case sensitive pw length (unicode): 24  
Case insensitive pw(ansi): 0x8DDC6652E6BF7E86B599495D2E839D01D2E064BE56904E32  
Case sensitive pw (unicode): 0x0C40AC39D5461A2DD23279A8F02CC0C0279CD3C4BB52210  
**Account Name:** Administrator  
Primary Domain Name: SACRAFICIAL2  
Native OS1: Windows NT 1381  
Native Lan Man:

#### SMB Tree Connect and X

Transaction Type: 0 *Request*  
Secondary Command: 0xFF *No More Commands*  
Password Length: 1  
Net-Name Password: 0x00  
**File Pathname:** \\10.1.1.201\C

Figure 9 Connection request from Administrator to \\10.1.1.201\C

#### SMB - Server Message Block

Protocol ID: SMB  
Command Code: 115 *Session Set Up And X (Including User Logon)*  
Error Code Class: 0x00 *Success*  
Reserved: 0x00  
Error Code: 0 *Success*  
**Tree ID (TID):** 0x0801  
Process ID (PID): 0xCAFE  
**User ID (UID):** 0x0801  
Multiplex ID (MID): 0x0010

#### SMB Session Set Up & X (Including User Logon)

Transaction Type: 1 *Response*  
Word Count: 3  
Secondary command: 0x75 *Tree Connect And X*  
Reserved: 0x00  
AndX offset: 135  
Action: 0x0000  
Byte Count: 94  
Native OS: 0.....  
Native Lan Man: .....  
Primary Domain Name: .....

**Figure 10 Successful connection to [\\10.1.1.201\C](#)**

SMB - Server Message Block

Protocol ID: SMB  
Command Code: 162 *CreateAndx*  
Tree ID (TID): 0x0801  
Process ID (PID): 0xF940  
User ID (UID): 0x0801  
Multiplex ID (MID): 0x0040

SMB Create AndX

Transaction Type: 0 *Request*  
File Name: N\WINNT\Profiles\AllUsers\Start  
Menu\Programs\Startup\worm.exe

**Figure 11 File creation attempt**

SMB - Server Message Block

Protocol ID: SMB  
Command Code: 162 *CreateAndx*  
NT Status: 0xC000003A *STATUS\_OBJECT\_PATH\_NOT\_FOUND*  
Tree ID (TID): 0x0801  
Process ID (PID): 0xF940  
User ID (UID): 0x0801  
Multiplex ID (MID): 0x0040

SMB Create AndX

Transaction Type: 1 *Response*  
Word Count: 0  
Byte Count: 0

**Figure 12 Path not found.**

SMB - Server Message Block

Protocol ID: SMB  
Command Code: 162 *CreateAndx*  
Error Code Class: 0x00 *Success*  
Reserved: 0x00  
Error Code: 0 *Success*  
Tree ID (TID): 0x0801  
Process ID (PID): 0xE020  
User ID (UID): 0x0801  
Multiplex ID (MID): 0x00D0

SMB Create AndX

Transaction Type: 1 *Response*  
Word Count: 34  
Secondary Command: 0xFF *No More Commands*  
Reserved (MBZ): 00  
Offset To Command: 0x0067  
Oplock Level: 0x02

```

.....0. Level II Oplock Not Granted
.....1 Exclusive Oplock Granted

FID: 0x4000
Creation Action: 0x00000002
Creation Time: 0x40B56828FFFC301
Last Access Time: 0x0070556E52FFC301
Last Write Time: 0x00B1A228FFFC301
Last Change Time: 0x00F06F2CD2E7A801
File Attributes: 0x00000020
File Type: 0000
DeviceState: 0x0000
Byte Count: 0

```

**Figure 13 SMB file create success**

If this exploit was attempted on a NT machine where this path is not available the following message would be returned (Figure 12)

If this was tried on a Windows 2000 machine where that path is actually available the successful response would be as in Figure 13.

This is really enough information to recognize the success and failure of this part of the exploit. The rest of the SMB messages manage the byte transfer of the file using more SMB messages such as `NT_CREATE_ANDX` and `WRITE_ANDX`. For further research, an excellent in depth discussion of SMB is <http://www.ubiqx.org/cifs> .

## Variants

According to the Network Associates' description on May 16<sup>th</sup>, 2003, this family of worms was rapidly expanding and had, at that time, 39 known variants.

"W32/Deborm.worm is a file share propagating worm targeting Microsoft Windows NT, W2K and XP machines. There are many many versions of this share propagating worm. This description is merely meant as a guide."<sup>15</sup>

The description was taken from W32/Deborm.worm.gen, which is very vague, and the description is very generic. This is probably a characteristic of an easily modified worm. Some of the variants according to PestPatrol, Inc<sup>16</sup> are listed below

- Worm.Win32.Deborm.aa
- Worm.Win32.Deborm.ac
- Worm.Win32.Deborm.c
- Worm.Win32.Deborm.g
- Worm.Win32.Deborm.j
- Worm.Win32.Deborm.k

<sup>15</sup> [http://vil.nai.com/vil/content/v\\_100143.htm](http://vil.nai.com/vil/content/v_100143.htm)

<sup>16</sup> [http://www.pestpatrol.com/pestinfo/w/worm\\_win32\\_deborm.asp](http://www.pestpatrol.com/pestinfo/w/worm_win32_deborm.asp)

- Worm.Win32.Deborm.l
- Worm.Win32.Deborm.n
- Worm.Win32.Deborm.p
- Worm.Win32.Deborm.u
- Worm.Win32.Deborm.w
- Worm.Win32.Deborm.x
- W32.HLLW.Deborms
- W32.HLLW.Deborms.B

Some of the differences in the variants were that some of the worms had the ability to try a number of different passwords on targets, launch remote processes or indeed do almost anything that SMB allowed. The earlier variants used different accounts and user names for access. Administrator, wwwadmin, database were among account names tried. As well as a blank password, other simple passwords tried were user, admin, password and test.

According to NA one variant is also known to execute without a reboot by calling the NetScheduleJobAdd function. If successful, this would vastly increase the propagation speed of the worm. This property was not exhibited by the examined worm nor could more information be found about a worm that exhibited this property.

#### **Example differences with the W32/Deborm-R**

<http://www.sophos.com/virusinfo/analyses/w32debormr.html>

This variant also targets vulnerable SMB shares but attempts to install a total of three files and adds the following different registry entry which contains the name of the worm file so that it is run each time Windows is started:

HKLM\Software\Microsoft\Windows\CurrentVersion\Run\NAV Live Update.

The use of the NAV Live Update key is possibly an attempt to hide among legitimate anti-virus software. Whereas the worm being studied has a payload of two files, W32/Deborm-R drops one additional file. The first two are the same as used in the studied worm – they are just renamed. The last Trojan is not part of the observed payload with the Deborm-Q variant.

- [Troj/Litmus-203](#)<sup>17</sup> (17440 bytes) backdoor IRC Trojan that allows others to take control of the PC. Among the many uses could be a distributed, denial of service attack. This is where a large number of machines are “taken over” and are used to target possibly just one machine and flood it with traffic, and in effect putting it out of commission for a period of time. This method is used in order to ensure enough traffic generation and also to conceal the identity of the attacker.<sup>18</sup> This Trojan has the name SVCHOST.exe in the exploit studied.

<sup>17</sup> <http://www3.ca.com/threatinfo/virusinfo/virus.aspx?id=11874>

<sup>18</sup> Steve Gibson describes this method in great detail <http://www.grc.com/dos/drdo.htm>

- [Troj/Sdbot-Fam](#)<sup>19</sup> (12832 bytes) Another family of backdoor IRC Trojans. Like many bots this bot has the ability to perform a number of different actions on the host's machine such as downloading and executing files. It can also even connect to a remote site and update itself to possibly avoid detection. This is named as "Explorer.exe" in the studied exploit.
- [Troj/KillAV-Q](#)<sup>20</sup> (17,410 bytes) A nasty Trojan that attempts to disable AV software such as Norton and also to kill firewall software such as Zone Alarm and Black Ice. There are nearly two hundred applications or services targeted (see Computer Associates site for list).

### **W32/Deborm-Q – according to Sophos**<sup>21</sup>

#### **Alias**

Win32/Nebiwo.C, W32.HLLW.Nebiwo

This worm has the same alpha designation – "Q" – as the worm being studied yet the description is not quite as complete as the Network Associates (NA) deborm.worm.q<sup>22</sup>. Although they both target and propagate using the same methodology, there are slight differences in the payload.

Sophos describes the dropping of [Troj/Litmus-203](#) and [Troj/Sdbot-Fam](#) (as in the "R variant above). There is no mention of the names of the payload "Explorer.exe" or "SVCHOST.exe" or of the worm making any NetBIOS queries.

The description on the NA site is the closest to what was observed as a result of the infection on site. This is the ONLY site that accurately describes the names of the dropped files and the NetBIOS name query that is a definite signature of this worm. There appears to be no other site that identifies this defining characteristic.

This highlights the somewhat haphazard approach to malware naming conventions. When a new piece of malware is discovered, there is often a frantic rush to reverse engineer it to find out how to produce a signature. Often this is part of a marketing exercise and the competitive nature of the business, coupled with ever evolving malware, can make for very confusing overlapping nomenclatures. It appears, in this case, that Sophos is less exacting in its description of the worm than NA.

Today, one year after the event, Port 139 is still regularly among the top 10 targeted ports on the Internet (Figure 14). The data below is sourced from the

<sup>19</sup> <http://www3.ca.com/threatinfo/virusinfo/virus.aspx?id=12411>

<sup>20</sup> <http://www3.ca.com/threatinfo/virusinfo/virus.aspx?id=29927>

<sup>21</sup> <http://www.sophos.com/virusinfo/analyses/w32debormq.html>

<sup>22</sup> [http://vil.nai.com/vil/content/v\\_100234.htm](http://vil.nai.com/vil/content/v_100234.htm)



[Internet Storm Center](#) , which collates information from IDS, & firewall logs throughout the globe.



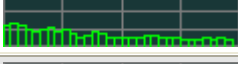

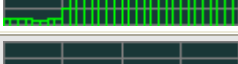

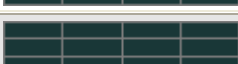
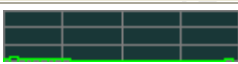
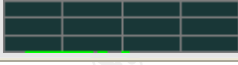

Service Name	Port Number	30 day history	Explanation
mydoom	<a href="#">3127</a>		W32/MyDoom, W32.Novarg.A backdoor
microsoft-ds	<a href="#">445</a>		Win2k+ Server Message Block
epmap	<a href="#">135</a>		DCE endpoint resolution
ms-sql-m	<a href="#">1434</a>		Microsoft-SQL-Monitor
www	<a href="#">80</a>		World Wide Web HTTP
netbios-ns	<a href="#">137</a>		NETBIOS Name Service
ms-sql-s	<a href="#">1433</a>		Microsoft-SQL-Server
socks	<a href="#">1080</a>		Proxy Server
netbios-ssn	<a href="#">139</a>		NETBIOS Session Service
squid-http	<a href="#">3128</a>		Proxy Server

Figure 14 Top ten attacked ports March 2004<sup>23</sup>

## Description

Complete subnets are incrementally ARP probed and, upon a response, a TCP connection attempt is made to Port 139, the NetBIOS session service.

If a successful connection is made a null SMB connection is then established.

Next the worm looks to see if a network share with the name "C" exists.

If it does, an attempt is made to authenticate using the Administrator, Guest and Owner user IDs with no password.

If a successful connection is made then the worm attempts a transfer of a file called "~2.exe" to the following shares

C:\Documents and Settings All Users\Start Menu\Programs\Startup  
C:\WINDOWS\Start Menu\Programs\Startup\

<sup>23</sup> <http://isc.incidents.org/top10.html> Note that Port 445 (SMB over TCP/IP) is second!

C:\WINNT\Profiles\All\Users\Start Menu\Programs\Startup\

Three directories are tried as at least one of these Windows directories are used in each Windows version ME,NT,2000,XP.

Once transferred to a new host, to one of the Windows Startup hooks mentioned, it is activated by a reboot which then extracts, drops and runs some more files – Explorer .exe (note the space) & SVCHOST.exe. The registry is also modified to run these two programs at boot up.

So, unlike a buffer overflow attack, this exploit uses the standard Windows file sharing protocol, SMB to perform a file transfer across networks and hosts. The only feature that is “exploited” is a lack of password protection on network shares and an easily guessable share name.

## Signatures of the Attack

### Network

This is a very “noisy” and therefore simple to detect attack. There is absolutely no attempt at stealth. There are many signatures that an Intrusion Detection System (IDS) could use to detect the network stage of the attack.

A rapid, incremental subnet ARP scan is the most obvious network signature. While a broadcast ARP is normal an incremental scan is not. There would be no normal condition that could cause this type of traffic.(Figure 15 shows a summary of a portion of the traffic)

Packet	Source Physical	Destination	Delta Time	Summary
1	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF		10.0.0.2 = ?
2	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000456	10.0.0.3 = ?
3	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000534	10.0.0.4 = ?
4	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000540	10.0.0.5 = ?
5	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000602	10.0.0.6 = ?
6	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000536	10.0.0.7 = ?
7	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000527	10.0.0.8 = ?
8	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000548	10.0.0.9 = ?
9	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000530	10.0.0.10 = ?
10	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000629	10.0.0.11 = ?

Figure 15 Incremental ARP Scan

The SMB attempt to connect using the Administrator, Guest or Owner may, depending on policy be an indication of attack. If for example a company never created a “Owner” account there should never be an attempt to remotely setup a SMB connection with that account. (see Figure 16)

#### SMB - Server Message Block

Protocol ID: SMB  
Command Code: 115 *Session Set Up And X (Including User Logon)*  
Error Code Class: 0x00 *Success*  
Reserved: 0x00  
Error Code: 0 *Success*  
SMB Flags: %00011000  
SMB Flags2: %1000000000000011  
Reserved:  
..H..)+.p... 00 00 48 06 E2 29 2B B2 70 BA 00 00  
Tree ID (TID): 0x0000  
Process ID (PID): 0xCAFE  
User ID (UID): 0x0000  
Multiplex ID (MID): 0x0080

#### SMB Session Set Up & X (Including User Logon)

Transaction Type: 0 *Request*  
Word Count: 13  
Secondary command: 0x75 *Tree Connect And X*  
AndX reserved (MBZ): 0x00  
AndX offset: 214  
Max buffer size: 4356  
Max multiplex count: 10  
VC number: 1  
Session key: 0x00000000  
Case insensitive pw length (ansi):24  
Case sensitive pw length (unicode):24  
Case insensitive pw  
(ansi):0x8DDC6652E6BF7E86B599495D2E839D01D2E064BE56904E32  
Case sensitive pw  
(unicode):0x0C40AC39D5461A2DD23279EA8F02CC0C0279CD3C4BB52210  
Account Name: Owner  
Primary Domain Name: SACRIFICIAL2  
Native OS1: Windows NT 1381  
Native Lan Man:  
Extra Bytes:  
W.i.n.d.o.w.s. . 57 00 69 00 6E 00 64 00 6F 00 77 00 73 00 20 00  
N.T. .4...0..... 4E 00 54 00 20 00 34 00 2E 00 30 00 00 00 00 00

#### SMB Tree Connect and X

Transaction Type: 0 *Request*  
Word Count: 4  
Secondary Command: 0xFF *No More Commands*  
Reserved (MBZ): 0x00  
Offset To Command: 0  
Flags: 0x0000  
Password Length: 1  
Byte Count: 37  
Net-Name Password: 0x00  
File Pathname: \\10.1.1.201\C  
Service Name: ?????

**Figure 16 SMB Connection attempt - Owner**

## SMB - Server Message Block

Protocol ID: SMB  
Command Code: 162 *CreateAndX*  
Error Code Class: 0x00 *Success*  
Reserved: 0x00  
Error Code: 0 *Success*  
SMB Flags: %00011000  
SMB Flags2: %1000000000000011  
Tree ID (TID): 0x0802  
Process ID (PID): 0xE020  
User ID (UID): 0x0802  
Multiplex ID (MID): 0x00B0

### SMB Create AndX

Transaction Type: 0 *Request*  
Word Count: 24  
Secondary Command: 0xFF *No More Commands*  
Reserved (MBZ): 00  
Offset to Command: 0000  
Reserved (MBZ): 00  
Name Length: 136  
Flags: 0x00000006

..... 0... *Not a Directory*  
..... .1.. *Request a Batch Oplock*  
..... .1. *Request an Oplock*

Root Directory FID: 0x00000000  
Desired Access: 0x00030196

0..... *No Generic Read*  
..0..... *No Generic Write*  
..0..... *No Generic Execute*  
...0..... *No Generic All*  
....0.... *Maximum Not Allowed*  
.....0.. *No Access to System Security*  
.....0..... *No Write Owner*  
.....0.. *No Write DAC*  
.....1. *Read Control*  
.....1 *Delete Access*  
.....1.....1 *Write Attributes Access*  
.....1..... *Read Attributes Access*  
.....1.... *Write Extended Attributes Access*  
.....0... *No Read Extended Attributes Access*  
.....1. *Append Data Access*  
.....1. *Write Data Access*  
.....0 *No Read Data / List Directory Access*

Allocation size: 0  
File Attributes: 0x00000020

..... 0..... *Not Encrypted*  
..... 0..... *May be indexed*  
..... 0..... *Not Offline*  
..... 0.... *Not Compressed*  
..... 0.. *No Reparse Point*  
..... 0. *Not Sparse*  
..... 0 *Not Temporary*  
..... 0..... *Not Normal*  
..... 0..... *Not a device*  
..... 1..... *Archive*  
..... 0.... *Not a directory*  
..... 0... *Not a Volume ID*  
..... 0.. *Not a System file*  
..... 0. *Not a Hidden file*  
..... 0 *Not Read Only*

Share Access: 0x00000000  
..... 0.. *No Share Delete*  
..... 0. *No Share Write*  
..... 0 *Share Prevention*

Create Disposition: 5 *Overwrite If - Action if file does/does not exist*  
Create Options: 0x00000044  
..... 0..... *Do Not Delete On Close*

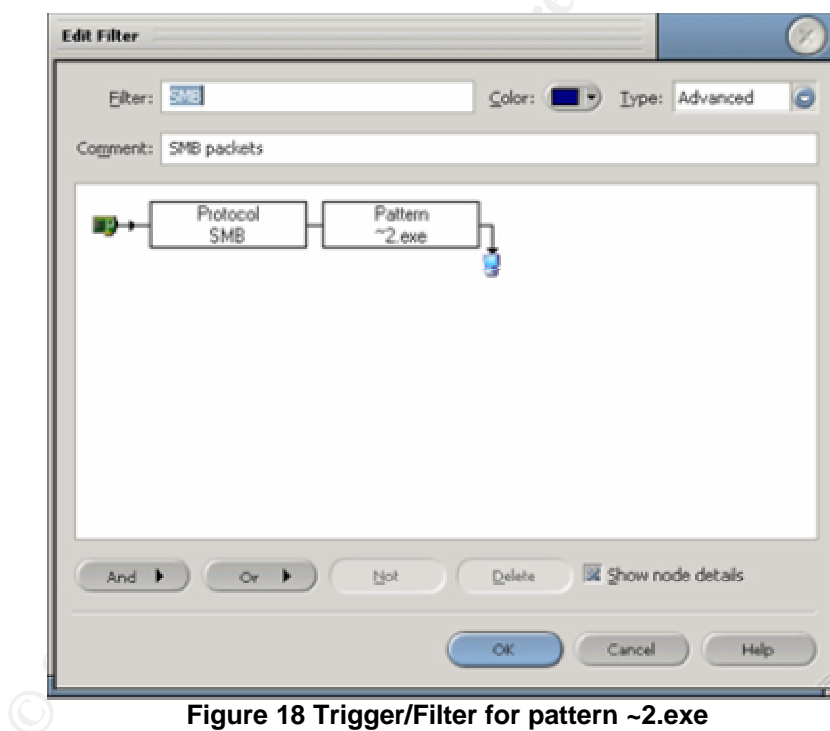
```

.....0..... No Random Access
.....0..... Long File Names
.....0..... No Extended Attributes
.....1..... Non-Directory File
.....0..... Non-Sync I/O Non-Alert
.....0..... Non-Sync I/O Alert
.....1..... File is accessed sequentially
.....0..... Write Through buffer does not
need to be deleted
.....0 File is not a directory

Level:                2  Impersonation
Security Flags:       03
.....1. Effective Only
.....1 Context Tracking

Byte Count:          139
File Name:           N\Documents and Settings\All Users\Start Menu\Programs\Startup\~2.exe
{File Name:          N\WINDOWS\Start Menu\Programs\Startup\~2.exe      }
{File Name:          N\WINNT\Profiles\All Users\Start Menu\Programs\Startup\~2.exe  }
```

**Figure 17 SMB file transfer attempt to three locations**



**Figure 18 Trigger/Filter for pattern ~2.exe**

#### NetBIOS Name Service - Network Basic Input/Output System

Identification: 0x8095  
DNS Flags: 0x0110  
Questions: 1  
Answers: 0  
Authority: 0  
Additional: 0

#### Question

Domain Name: SON.ATH.CX <00> *Workstation*  
Type: 32 *NetBIOS General Name Service*  
Class: 1 *Internet*

**Figure 19 NetBIOS name service request SON.ATH.CX**

#### NetBIOS Name Service - Network Basic Input/Output System

Identification: 0x8097  
DNS Flags: 0x0110  
Questions: 1  
Answers: 0  
Authority: 0  
Additional: 0

#### Question

Domain Name: SON.GLINED.US <00> *Workstation*  
Type: 32 *NetBIOS General Name Service*  
Class: 1 *Internet*

**Figure 20 NetBIOS name service request SON.GLINED.US**

The attempt to transfer the file “~2.exe” is an absolutely clear indication of this exploit. Figure 17 shows relevant traffic for all three locations attempted. The SMB CreateAndx message contains a lot of information about the creation of the file. For example the “Hidden” attribute here is set to 0 so the file will be easily found visible in the chosen transfer directory. If this was changed to a 1 this would make the exploit a bit more difficult to detect. This SMB frame is shown in its entirety so the reader can see the many options available.

Simply making a trigger pattern “~2.exe” on a network analyzer would capture the exploit in progress. This would capture both successful and unsuccessful exploit attempts. Obviously, a pattern for an Intrusion Detection System such as SNORT could also be simply created.

Figure 18 shows detail of trigger start setup using Etherpeek. It is obvious from the intuitive nature of the dialogue box that this is an easy filter to setup. It should be stated that this ease of use is prevalent throughout the whole application.

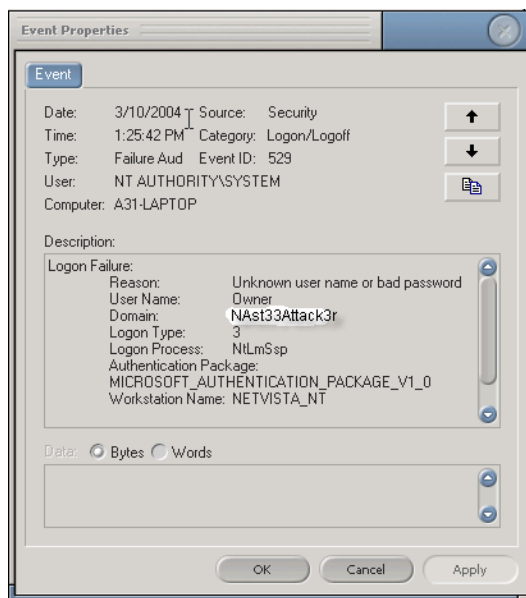
There are two NetBIOS name service requests made (Figure 20) (Note. Network Associates also details a NetBIOS name request to LTR1.SOCKETPIMPS.NET but this was not apparent on the worm analyzed. This is further information to think that this is another minor variant).

## Host Signatures

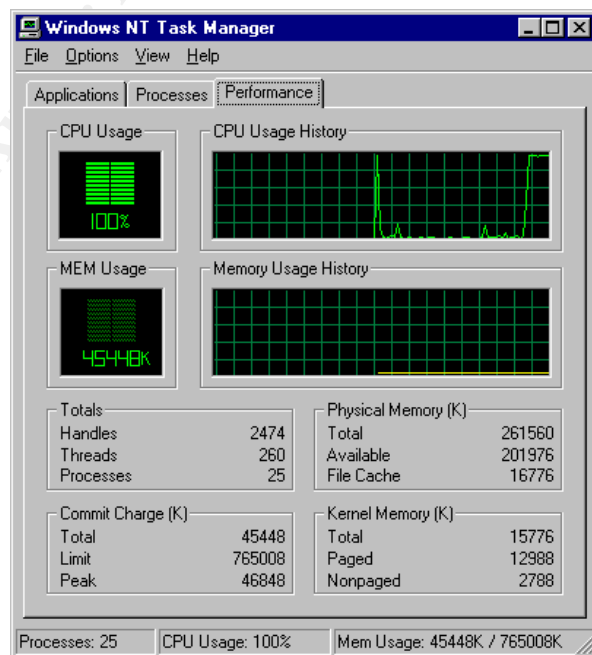
When this worm sees that the target is listening on Port 139 it tries to connect to the network C share with the user names Administrator, Owner and Guest using no password. If this fails on the host due to either password protection, account not available or account disabled the event will be captured in the security event log. (Figure 21)

If the victim has been exploited but a reboot has not yet occurred then the only other host signature will be the presence of the “~2.exe” file (56,320 bytes) in a startup folder (location dependent on version of windows). Nothing else will be seen at this point so the worm could lie dormant if the machine is never switched off. If, for example a print server was infected then this may take a long time – if ever to become apparent.

After the victim has been rebooted the worm will activate and drop its payload. One of the most obvious effects to the user at this time is the slowness of the machine. The Task Manager will show the CPU pegged at 100% (Figure 22)



**Figure 21 Connection attempt  
NAs33Attack3r**



**Figure 22 Task Manager showing CPU at  
100%**

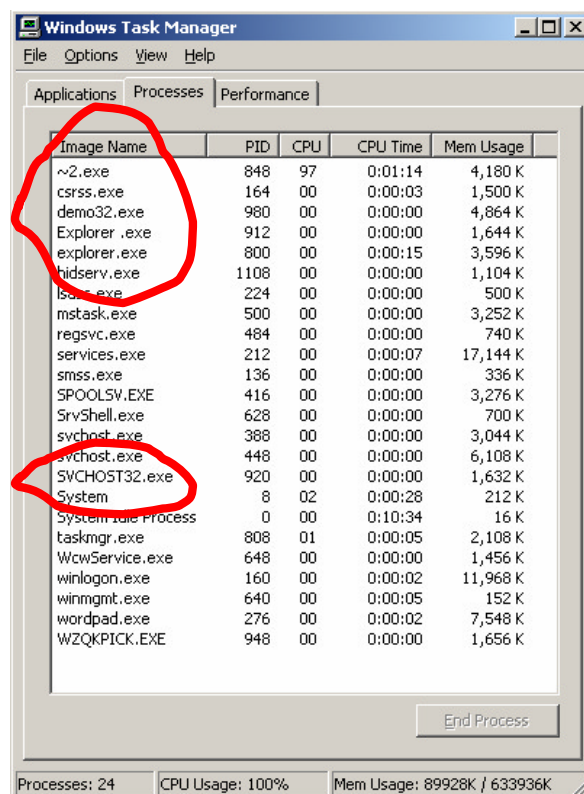


Figure 23 Worm process list

The process consuming 99% of the CPU can be seen under the processes tab. (Figure 23) “~2.exe” is the actual worm. (Note. Subsequent reboots to try and clear the problem will spawn multiple instances or child processes of the worm of the form “~x.exe” where x will increment each time the machine is rebooted). These “~x.exe” files are also copied into the %TEMP% directory and will be another host signature.(Figure 24)

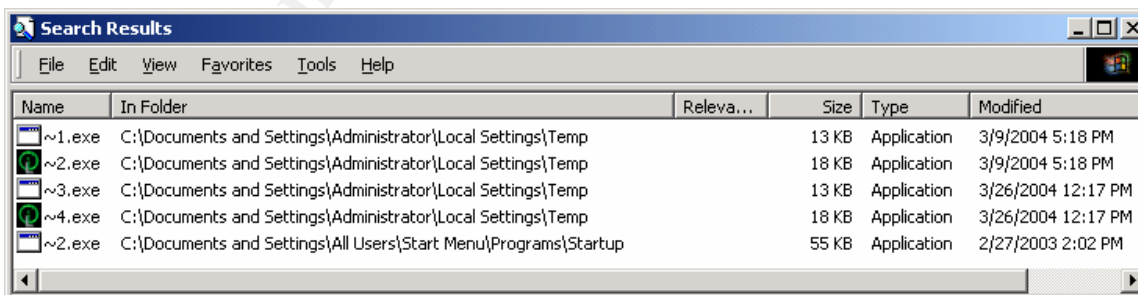


Figure 24 Worm host propagation

Two other files will be visible in the Task Manager (Figure 23) Explorer.exe (note the space) SVCHOST32.exe (note the upper case – the real svchost.exe will be in lower case and does not have the number 32 appended).



## Host Modifications

Determining the areas of compromise of the system is a difficult task as Windows has many places that can be used to either hide files or run files on boot up. This varies also from version to version. Searching all these locations is both laborious and error prone. The best way is to have a record or take a snapshot of the system in a known good configuration. This can then be used to for comparison in the future. The most well known application for achieving this aim is [Tripwire](#) but I took the opportunity to try a product called “WhatChanged for Windows” by [Prism Microsystems Inc.](#). The title is pretty self explanatory. It takes a system snapshot of the file system and registry and can be used to compare the system at a later date. All the registry and file additions were detailed in a easy to use graphical interface. An example screenshot is shown below (Figure 25) – the changes are shown in green.

The file additions and registry modification are obtained by simply drilling down in the GUI. (Figure 26, Figure 27, Figure 28, Figure 29)

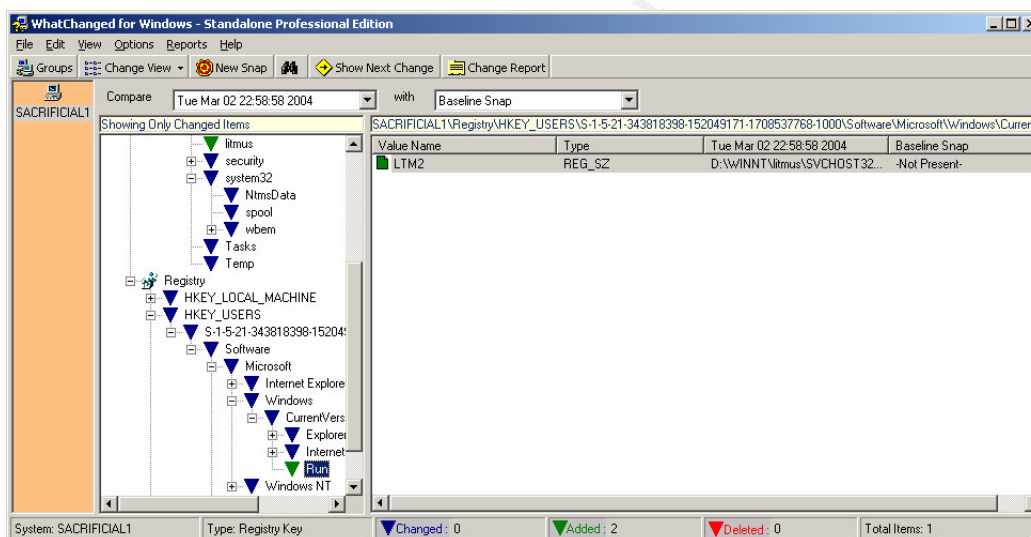


Figure 25 WhatChanged for Windows sample screen

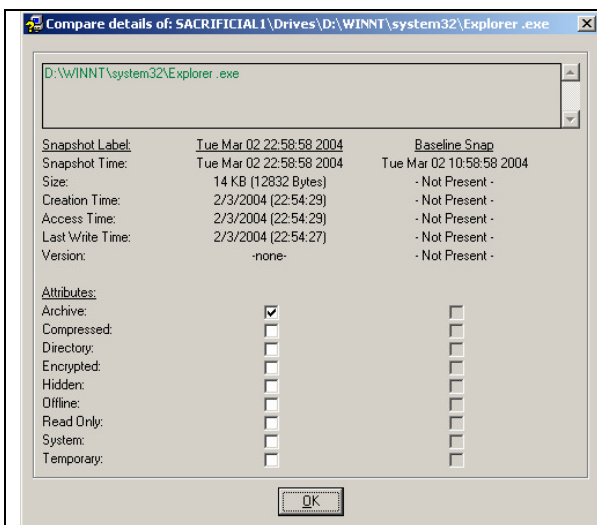


Figure 26 Explore .exe 12,832 bytes

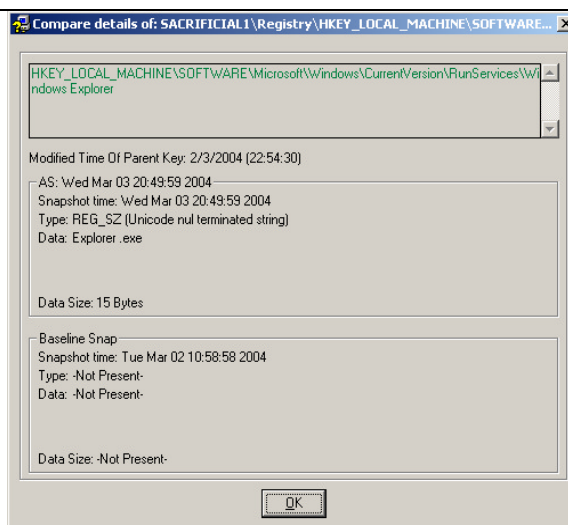


Figure 27 Registry modification Explorer .exe

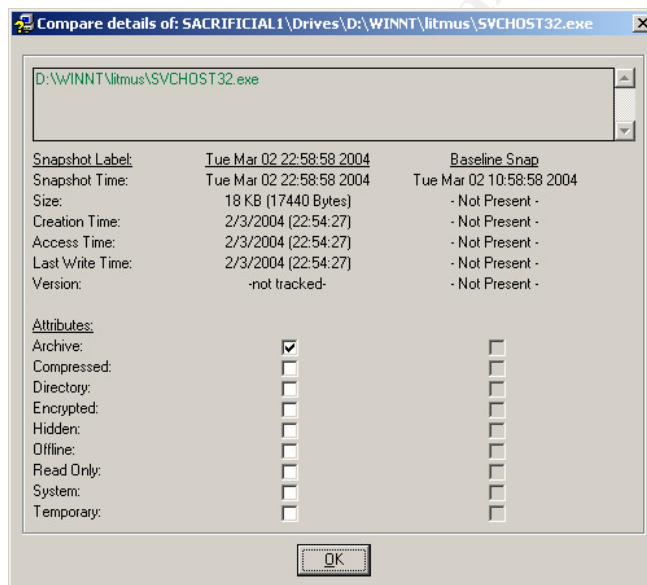


Figure 28 SVCHOST.exe 17,440 bytes

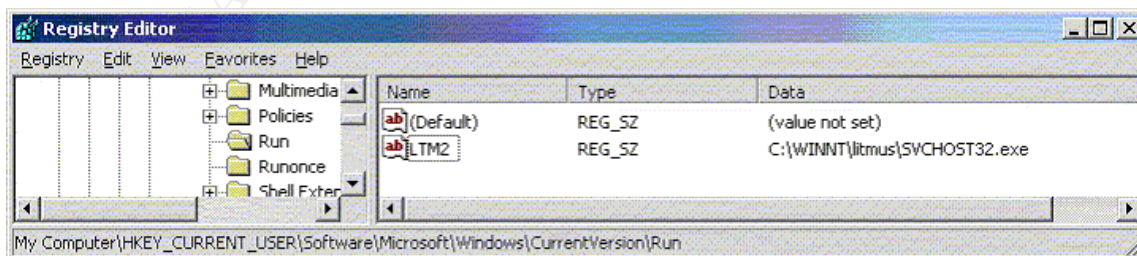


Figure 29 Registry modification SVCHOST.exe

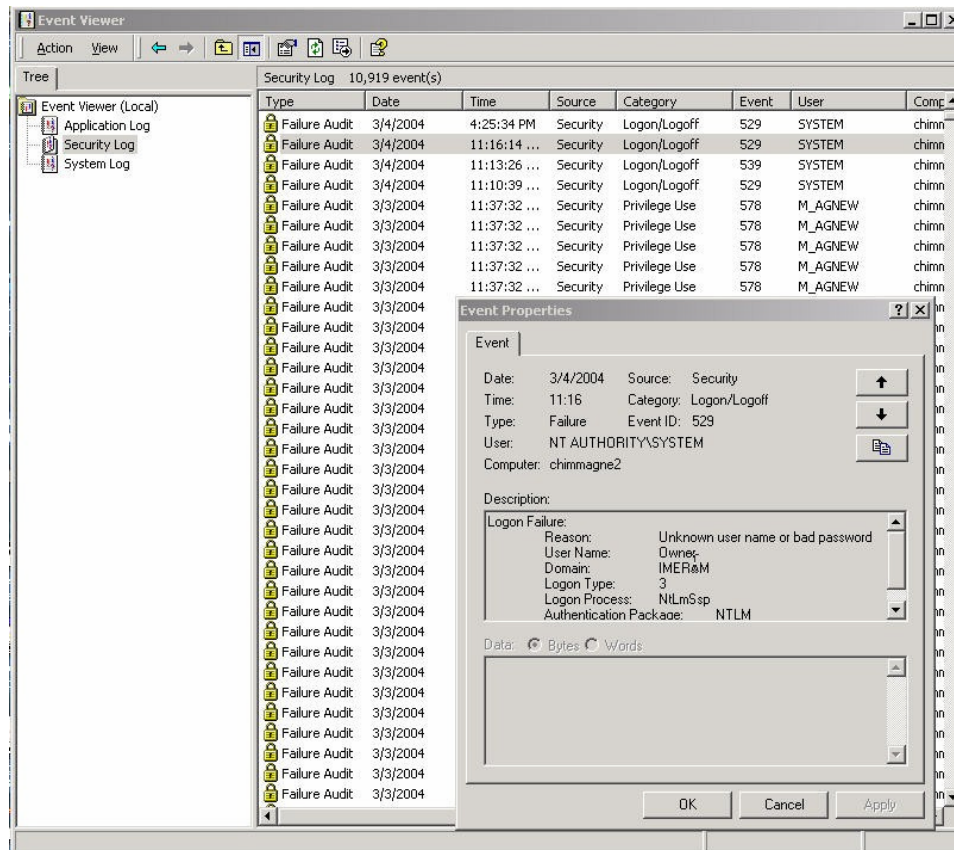


Figure 30 Security Event Log



Figure 31 Auditing login failure

## Windows Event Logs

The Windows Security log in the event viewer can monitor all successful and all unsuccessful attempts to mount to SMB shares. (Figure 30).

However, for some reason, once again the Windows default does not help security – this feature needs to be turned on as it is off by default. This should be a matter of right clicking on the Security log but unfortunately it is not that easy.

The auditing features are turned on by going to

- Administrative tools,
  - Local Security Policy,
    - Local Security Settings,
      - Audit Policy ,
        - Audit Logon Events
          - check Logon Failure.

That was easy, wasn't it! (Figure 31). Since Windows 2000 Microsoft has shipped with the possibility of very good system auditing. However, by default, it is all turned off. In my opinion, this stance is indefensible in a modern computer environment. Perhaps Windows XP SP2 will address this issue. Until then, it is urged that all readers of this document review their auditing policies.

## **The Platforms/Environments**

### ***The Victim***

The victims were the four machines in the sales demonstration network (Figure 33). These machines had initially been all ghosted from an image created on April 2002. The build had been fully patched at installation time and consisted of

- Windows 2000 SP2
- Office 2000 SP2
- GIAC proprietary application software

No AV software or host firewall was installed.

### ***The Source***

John's laptop was traced to be the source. This had been given to him when he joined the company in mid 2002 and consisted of ;

- Windows 2000 (with full local admin rights)
- Office 2000 SP2
- Lotus Notes 5
- GIAC proprietary application software
- Trend Micro AV (updated monthly)
- No firewall

### ***The Source Network***

Like many families, John had a DSL Internet connection at home. This was used for both home and business use. (Figure 32)

When he had gone home the previous evening to work on his presentation he had connected to his home network for Internet access. He needed to use his PPTP (Microsoft's Point to Point Tunneling Protocol) VPN connection to get some marketing information from the sales application database to include in his presentation. As he disliked using the small laptop screen and keyboard at home, he had duplicated his PPTP setup from his work laptop to his home machine. This was possible because authentication was not tied to his laptop but to his RSA SecurID card and PIN number. This then allowed him to use his home desktop to connect to the corporate LAN.

To make the transfer of files easier from desktop to laptop, he had set up a Windows share of his C drive on both machines. This share was never removed when he finished. Both machines had no password for administrator.

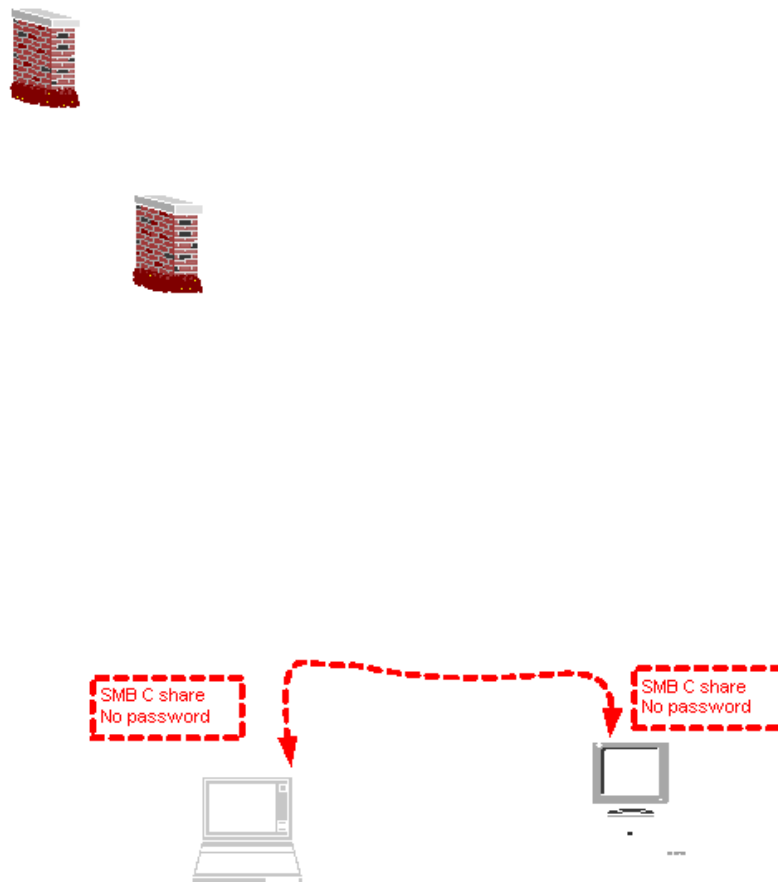


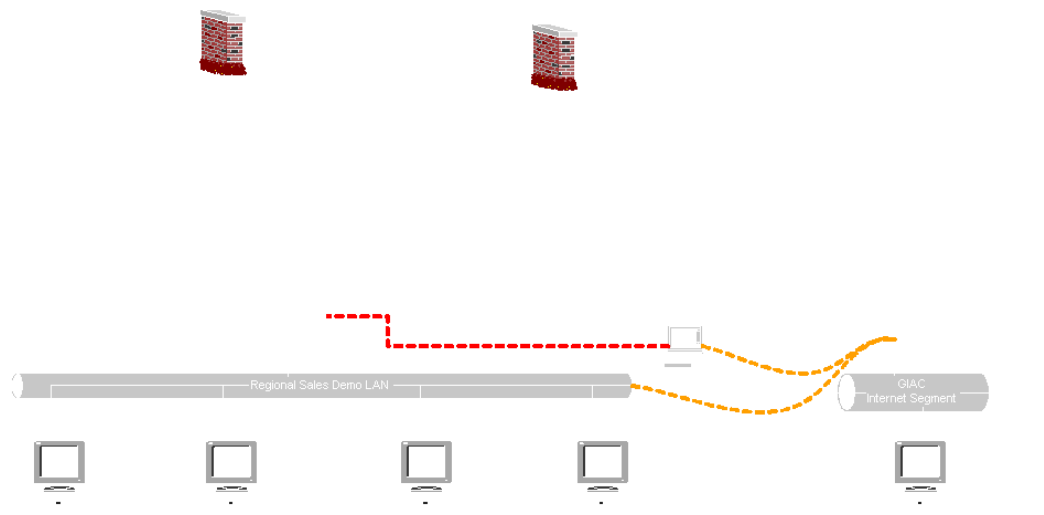
Figure 32 Source network schematic

## ***The Target Network***

GIAC Enterprises Inc. is a global company with many regional offices. Some of these regional offices have their own sales network to facilitate client demonstrations of new software. The network that was attacked was a very simple system that sourced proprietary data from the GIAC global network (Figure 33). All the “demo” machines were all standard Intel P3 machines running Windows 2000 SR2, Office 2000 SP2 and some GIAC proprietary software.

They had been fully patched at installation time (one year previously Spring 2002). The “demo” network could also be temporarily connected directly to the

Internet information if required. There was sometimes a requirement to download files from the Internet and this was very slow over the corporate Internet proxy connection. It was a simple matter to plug into the Internet Hub when necessary and have almost a full T1 connection available.



**Figure 33 GIAC regional office demo network**

Initially, when the network was built, it had been attempted to lock down the rights to the system so that only the technical department could make changes. This was soon changed for operational reasons. The rate of change of new software releases had started to accelerate and platforms were needed for new release training purposes. On a couple of occasions an engineer was not available to immediately load new software. The sales director exerted pressure on the technical department and so the demonstration network was “opened up” to allow the sales staff to make additions/changes when they needed.

- As the machines were used by many different sales staff the administrator account was used as default with a blank password
- At some time after installation, the C drives of all the machines were shared to allow easy file sharing between machines.

The physically separate Internet access machine had been loaded with Zone Alarm Pro<sup>24</sup> that was auto-configured at installation time. (Figure 34). Zone Alarm can be minutely fine tuned but this had never been done on this machine. The machine was also running AV software by Trend Micro. The automatic Windows Update feature had also been enabled to help protect this machine by keeping the patches up to date.

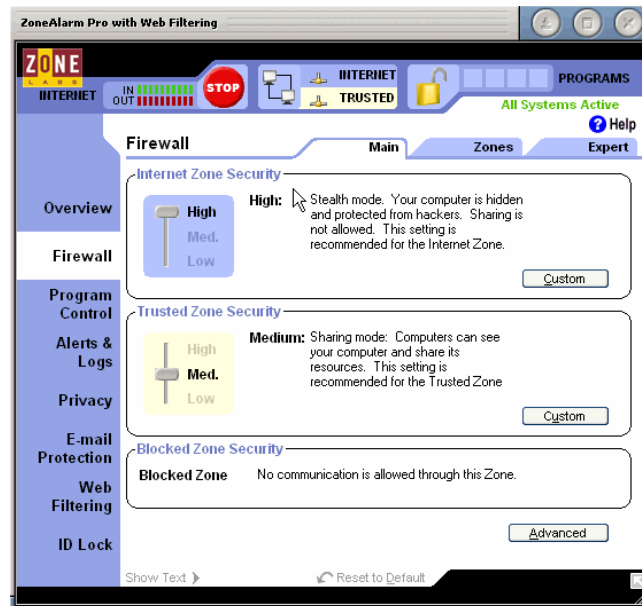


Figure 34 Zone Alarm Zone setup

## Reconnaissance

The reconnaissance phase is to prepare for an attack by gathering as much information as possible. This is similar to bank robbers watching a bank over a period of time to gather information on the valuables. They do not just break in to the bank at a random time - they wait for the most favorable time to get the most money for the least risk.

There is normally however, little or no reconnaissance associated with worm exploits in general. As soon as they are released they normally just go ahead and attack. The only piece of reconnaissance that sometimes occurs is when a worm looks to see if the target has already been infected. The virus writer in this case does not want to waste time or draw more attention to the spread of the worm. (This was touched earlier on when propagation methods were discussed)

However, there is no reconnaissance associated with the Deborm worm.

<sup>24</sup> [http://www.zonelabs.com/store/content/company/products/zap/trial/zap4x\\_trial.jsp?lid=pdb\\_zaptrial](http://www.zonelabs.com/store/content/company/products/zap/trial/zap4x_trial.jsp?lid=pdb_zaptrial)



## Scanning

In order to find out what I was dealing with I decided to that the best approach would be to remove and isolate an infected machine on its own network. I wanted a clean controlled environment where the only traffic would be suspicious traffic. Next I built a new sacrificial Windows 2000 machine using the saved ghosted image and attached it to the new “infected” segment but left it powered down.

As a network worm was suspected, I also attached a network analyzer, Etherpeek, to the segment to capture the traffic, if any, between the two machines. Etherpeek is one of many network analyzers on the market and was used as it was easily available to me on a laptop. It was hoped that the analysis of the traffic would reveal the nature of the exploit. It was extremely important that the laptop did not get infected too. The analyzer was protected by a host based firewall – also ZoneAlarm Pro and was fully up to date with patches. The Windows operating system was also fairly hardened i.e. only the bare required services were running, unused accounts were disabled and strong passwords were employed. A final precaution was taken “just in case” and I ghosted the laptop so if I had missed something and got infected, I could easily recover. In dealing with the unknown it is best to be very careful! I now had three machines connected by a simple hub.

Worms, by their very nature, spread from computer to computer exploiting some weakness in either the operating system or an application. This sometimes allows them to spread so quickly that a patch may yet to be released. This was the case with the Slammer worm<sup>25</sup> (also known as SQL Slammer Worm [ISS], DDOS.SQLP1434.A [Trend], W32/SQLSlammer [McAfee], Slammer [F-Secure], Sapphire [eEye], W32/SQLSlam-A [Sophos]) which spread at a rate the had never been seen before. There is a consensus of opinion that this almost elegant exploit – it was all contained in one single 376 byte UDP packet– infected the majority of the vulnerable systems within FIFTEEN minutes!<sup>26</sup>

Incidentally, up to this point in time, a Google search was not very helpful in trying to find out more about this infection as it kept treating the “~” character as a single wildcard - even in advanced mode! My only clue at this point was that strange process “~2.exe” and the CPU utilization. In some respects, this investigation was very interesting as it was done using basic network analysis. Often in the case of an attack, a quick search on Google will reveal the “fix”. This may simply be to load the current AV definitions which will disinfect the machine and, if necessary, load a patch. As this information was not available to me, I had to dig deep to find out what was happening.

---

<sup>25</sup> <http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2002-0649>

<sup>26</sup> <http://www.securityfocus.com/infocus/1752> A Comparison Study of Three Worm Families and their Propagation Methods ; Dec 10, 2003

The infected demo machine had an IP address of 10.0.0.10/8 and was powered up. At this point only the laptop was on the network and it had an address of 10.0.99.10. As can be seen from the captured Etherpeek screen shot below abnormal activity was immediately seen (Figure 35).

Packet	Source Physical	Destination	Delta Time	Summary
1	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF		10.0.0.2 = ?
2	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000456	10.0.0.3 = ?
3	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000534	10.0.0.4 = ?
4	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000540	10.0.0.5 = ?
5	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000602	10.0.0.6 = ?
6	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000536	10.0.0.7 = ?
7	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000527	10.0.0.8 = ?
8	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000548	10.0.0.9 = ?
9	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000530	10.0.0.10 = ?
10	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000629	10.0.0.11 = ?

Figure 35 Abnormal ARP behavior

Now that we understand very basically what ARP does, it is pretty obvious that the sequential incrementing of addresses in the Summary column is indicative of a machine that is trying to find out any machine on the network. Normal ARP activity is generally only a small percentage of all network traffic; here it is almost 100%. Also note the speed of the scan indicated by the delta time between packets – about half a millisecond. Also, note that the scan has commenced in the Subnet range of the network 10.0.0.0. The scan did not start at address 1.1.1.1, instead it started scanning its own subnet looking for close by computers. ( Network address & subnet changes were done to confirm this). Figure 36 shows that the scan continues to increment as there was no machine to respond.

Note also that the first address is not 10.0.0.1 but 10.0.0.2 . Note also from Figure 36 that host IDs .241-.254 are missed out.

418	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000582	10.0.0.235...
419	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000580	10.0.0.236...
420	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000579	10.0.0.237...
421	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000579	10.0.0.238...
422	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000580	10.0.0.239...
423	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000586	10.0.0.240...
424	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000589	10.0.1.1 = ?
425	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000573	10.0.1.2 = ?
426	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000582	10.0.1.3 = ?

Figure 36 Continuing ARP scan

Are these a mistake or an example of poor programming? My feeling is that it is a simple programming mistake as there were more indications of sloppy programming. The ARP requests were made in blocks of one hundred. After cycling through the first 100 IP addresses, it would retry a subset (approximately 80) of the addresses already tried, in a randomized order, before recommencing the requests of the next one hundred addresses (Figure 37).

479	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000594	10.0.1.56 = ?
480	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000590	10.0.1.57 = ?
481	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000596	10.0.1.58 = ?
482	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000591	10.0.1.59 = ?
483	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000585	10.0.1.60 = ?
484	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000593	10.0.1.61 = ?
485	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	02.765756	10.0.0.203 = ?
486	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000018	10.0.0.219 = ?
487	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000004	10.0.0.215 = ?
488	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000004	10.0.0.207 = ?
489	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000010	10.0.0.211 = ?
490	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000004	10.0.0.223 = ?
491	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000004	10.0.0.204 = ?
492	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000003	10.0.0.208 = ?
493	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000007	10.0.0.220 = ?
494	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000004	10.0.0.216 = ?

Figure 37 Non sequential scanning

This would appear to be unintentional as I can think of no reason to rescan ports that already have not replied – once again I suspected poor programming. Remember that scan speed is crucial to the speed of propagation of the worm. This is proof that not all worm and virus writers are as clever as the media sometimes makes them out to be! Another point worth mentioning at this point is that worms by are not very stealthy – they generate a lot of network traffic and so can be detected very easily by an alert sys admin or Intrusion Detection system

The choice of scanning technique is an important part of the malicious code makeup as it normally does not want to waste time and so employs what is know as a *spread algorithm* or *target acquisition function*<sup>27</sup> to maximize the speed with which it can spread. There are a number of variations in the techniques used and this subject is covered in depth in the Worm FAQ. One interesting discussion on this topic is the theoretical maximum speed that a worm can spread to the whole Internet. Nicholas C Weaver coined the term the “Warhol worm” which theoretically would spread to 99% of the Internet in less than 15 minutes<sup>28</sup>! Techniques such as pre-scanning allow the attacker to identify vulnerable systems before releasing the new worm. In this way it is possible to initially directly target systems and infect may systems immediately on release. Due to the exponential nature of worm infection this would vastly increase the rate of global saturation.

<sup>27</sup> <http://www.networm.org/faq> The Worm FAQ ; Silicon Defense

<sup>28</sup> Nicholas C Weaver “Warhol Worms: The Potential for Very Fast Internet Plagues  
<http://www.cs.berkeley.edu/~nweaver/warhol.html>

However, it was definitely apparent that some piece of malware was indeed seeking a target. It was time to find out what it wanted to do if someone replied. It was time to connect the newly ghosted machine and find out what happened next.

## Exploiting the System

Now the exploit starts to be revealed – let's examine the information captured more closely. Figure 38 details the ghosted victim responding to the broadcast ARP packet. The attacker (infected machine) is now aware of the MAC address /IP address combination of the victim

### ARP - Address Resolution Protocol

```
Hardware:      1 Ethernet (10Mb)
Protocol:      0x0800 IP
Hardware Addr Length: 6
Protocol Addr Length: 4
Operation:     2 ARP Response
Sender Hardware Addr: 00:10:5A:F6:6F:C6 Ghosted Victim
Sender Internet Addr: 10.1.1.201
Target Hardware Addr: 00:C0:4F:59:F6:D6 Infected Machine
Target Internet Addr: 10.1.1.10
```

Figure 38 ARP response (packet 259)

This time as soon as the ARP response was made to the query, a TCP connection was made, tore down and made again on Port 139 (Figure 39).

258	Infected Machine	FF:FF:FF:FF:FF:FF		Ghosted Victim = ?
259	Ghosted Victim	Infected Machine		Ghosted Victim = Ghosted Victim
260	Infected Machine	Ghosted Victim	IP-139	Src= 2472,Dst= 139,...S,S=2255000057,L= 0,A= 0,W= 8192
261	Ghosted Victim	Infected Machine	IP-2472	Src= 139,Dst= 2472,.A..S,S=1873308402,L= 0,A=2255000058,W=17520
262	Infected Machine	Ghosted Victim	IP-139	Src= 2472,Dst= 139,.A....S=2255000058,L= 0,A=1873308403,W= 8760
263	Infected Machine	Ghosted Victim	IP-139	Src= 2472,Dst= 139,.A...F,S=2255000058,L= 0,A=1873308403,W= 8760
264	Ghosted Victim	Infected Machine	IP-2472	Src= 139,Dst= 2472,.A...F,S=1873308403,L= 0,A=2255000059,W=17520
265	Infected Machine	Ghosted Victim	IP-139	Src= 2472,Dst= 139,.A....S=2255000059,L= 0,A=1873308404,W= 8760
266	Infected Machine	Ghosted Victim	IP-139	Src= 2473,Dst= 139,...S,S=2255036380,L= 0,A= 0,W= 8192
267	Ghosted Victim	Infected Machine	IP-2473	Src= 139,Dst= 2473,.A..S,S=1873373257,L= 0,A=2255036381,W=17520
268	Infected Machine	Ghosted Victim	IP-139	Src= 2473,Dst= 139,.A....S=2255036381,L= 0,A=1873373258,W= 8760
269	Infected Machine	Ghosted Victim	IP-139	Src= 2473,Dst= 139,.AP...S=2255036381,L= 72,A=1873373258,W= 8760
270	Ghosted Victim	Infected Machine	IP-2473	Src= 139,Dst= 2473,.AP...S=1873373258,L= 4,A=2255036453,W=17448

Figure 39 TCP Connection and tear down

### Exploit Packets 260 to 262

These packets show that the attacker completes a TCP connection to confirm that the victim is listening on port 139 for inbound connections. Having learnt the Ethernet address in packet 259, a TCP connection request (S bit set displayed in the summary column of Figure 39 as S (SYN) is sent (packet 260). Included in this packet is the sequence number that it will use to send data and the TCP port it wants to connect to -port 139. As discussed in the protocols section this port is associated with the NetBIOS session service and Server Message Block (SMB) Protocol.

Packet 261 is telling the attacker that the port is open and also sends its sequence number. S + A (ACK)

Packet 262 confirms the receipt of the sequence number (ACK) and completes the connection process known as the Three Way Handshake. (There is a bit more information exchanged but it is irrelevant to our analysis)

#### Exploit Packets 263 to 265

Inexplicably, the attacker now tears down the connection gracefully. (Indicated by the F (FIN) being set and acknowledged). These packets are strange as the attacker only now goes back to re-establish the TCP connection. This would appear to be an unnecessary overhead and is probably just another piece of sloppy programming. Remember speed is normally important to propagation and six packets have just been wasted.

However, the attacker has now confirmed that the victim will let it talk to the SMB port. (SMB will be discussed shortly).

#### Exploit Packets 266 to 268

Now the attacker establishes a new connection to Port 139 from Port 2473 to carry out the next part of the attack

#### Exploit Packets 269 to 270

NetBIOS length extension OFF negotiation.

#### Exploit Packets 271 to 272

Here the client (infected machine) initiates the SMB dialogue by telling the victim the dialects it understands and the victim responds that it will use LanMan 2.1. (Figure 40)

271	Infected Machine	Ghosted Victim	IP-139	C Verify dialect Count=8
272	Ghosted Victim	Infected Machine	IP-2473	R Verify dialect Status=OK Dialect=7

Figure 40 LanMan 2.1 SMB dialect negotiation

#### Exploit Packets 273 to 274

Having established the dialect a null session is set up to the hidden share IPC\$ (Figure 41)

273	Infected Machine	Ghosted Victim	IP-139	C Session setup and X Tree connect and X Path=\\10.1.1.201\IPC\$
274	Ghosted Victim	Infected Machine	IP-2473	R Session setup and X Status=OK Primary Domain=00000000 Tree com

Figure 41 Null Session

#### Exploit Packets 275 to 292 - Failure

Here an attempt is made to establish a connection to a system share [\\10.1.1.201\C](#) . (Figure 42) In this capture the victim machine has the following attributes:

- It is sharing C (although this is NOT the C:\ but refers to C:\temp for experimental purposes.
- has a null Administrator password
- Has the guest account disabled
- Does not possess a Owner account

Examining these packets reveals that the worm tries three common user names – Administrator, Guest and Owner. All of these are tried with no password and fail. Next the share name was modified to map to C:\ and this time the results were different.

274	Ghosted Victim	Infected Machine	IP-2473	R Session setup and X Status=OK Primary Domain=00000000 Tree connect and X St..
275	Infected Machine	Ghosted Victim	IP-139	C Session setup and X Tree connect and X Path=\\10.1.1.201\C Service=?????
276	Ghosted Victim	Infected Machine	IP-2473	R Session setup and X Status=OK Primary Domain=00000000 Tree connect and X St..
277	Infected Machine	Ghosted Victim	IP-139	C NT Create and X Name=\Documents and Settings\All Users\Start Menu\Programs\S..
278	Ghosted Victim	Infected Machine	IP-2473	R NT Create and X Status=Object path not found
279	Infected Machine	Ghosted Victim	IP-139	C NT Create and X Name=\WINDOWS\Start Menu\Programs\Startup\~2.exe
280	Ghosted Victim	Infected Machine	IP-2473	R NT Create and X Status=Object path not found
281	Infected Machine	Ghosted Victim	IP-139	C NT Create and X Name=\WINNT\Profiles\All Users\Start Menu\Programs\Startup\~..
282	Ghosted Victim	Infected Machine	IP-2473	R NT Create and X Status=Object path not found
283	Infected Machine	Ghosted Victim	IP-139	C End connection TID=0x0801
284	Ghosted Victim	Infected Machine	IP-2473	R End connection Status=OK
285	Infected Machine	Ghosted Victim	IP-139	C Logoff and X
286	Ghosted Victim	Infected Machine	IP-2473	R Logoff and X Status=OK
287	Infected Machine	Ghosted Victim	IP-139	C Session setup and X Tree connect and X Path=\\10.1.1.201\C Service=?????
288	Ghosted Victim	Infected Machine	IP-2473	R Session setup and X Status=Account disabled
289	Infected Machine	Ghosted Victim	IP-139	C Session setup and X Tree connect and X Path=\\10.1.1.201\C Service=?????
290	Ghosted Victim	Infected Machine	IP-2473	R Session setup and X Status=Logon failure
291	Infected Machine	Ghosted Victim	IP-139	C Session setup and X Tree connect and X Path=\\10.1.1.201\C Service=?????
292	Ghosted Victim	Infected Machine	IP-2473	R Session setup and X Status=Logon failure

Figure 42 SMB connection to C share



### Exploit Packets 273 to 281 – Successful Worm transfer !

273	Infected Machine	Ghosted Victim	IP-139	C NT Create and X Name=\WINNT\Profiles\All Users\Start Menu\Programs\Startup\~
274	Ghosted Victim	Infected Machine	IP-2473	R NT Create and X Status=OK Handle=0x4000
275	Infected Machine	Ghosted Victim	IP-139	C Transaction 2 Get FS Info TID=0x0802
276	Ghosted Victim	Infected Machine	IP-2473	R Transaction 2 Status=OK
277	Infected Machine	Ghosted Victim	IP-139	C Write byte block Handle=0x4000 Offset=56320 Bytes=0
278	Ghosted Victim	Infected Machine	IP-2473	R Write byte block Status=OK
279	Infected Machine	Ghosted Victim	IP-139	C Write block raw
280	Infected Machine	Ghosted Victim	IP-139	Src= 2473,Dst= 139,.A...,S=2255041087,L= 1460,A=1873374489,W= 7529
281	Infected Machine	Ghosted Victim	IP-139	Src= 2473,Dst= 139,.AP...,S=2255042547,L= 1440,A=1873374489,W= 7529
{				
337	Ghosted Victim	Infected Machine	IP-2473	Src= 139,Dst= 2473,.A...,S=1873374530,L= 0,A=2255096019,W=17520
338	Infected Machine	Ghosted Victim	IP-139	C Transaction 2 Set File Info Handle=0x4000
339	Ghosted Victim	Infected Machine	IP-2473	R Transaction 2 Status=OK
340	Infected Machine	Ghosted Victim	IP-139	C Close file Handle=0x4000 Bytes=0
341	Ghosted Victim	Infected Machine	IP-2473	R Close file Status=OK
346	Infected Machine	Ghosted Victim	IP-139	C End connection TID=0x0802
347	Ghosted Victim	Infected Machine	IP-2473	R End connection Status=OK
348	Infected Machine	Ghosted Victim	IP-139	C Logoff and X
349	Ghosted Victim	Infected Machine	IP-2473	R Logoff and X Status=OK

Figure 43 Successful file creation and transfer.

A successful attempt to the SMB CreateAndX (for file transfer) request is indicated by a response containing a TID and UID as detailed in the SMB protocol section (Figure 10 and packet 274 in Figure 43). Once the Tree ID has been obtained (0x802) a couple of more SMB packets are exchanged to set up file transfer parameters (packets 275-279) and the file transfer begins (packet 280). After successful transfer the SMB connection is closed ( packets 346-349). Now the worm is sitting in the startup directory of the victim but nothing will happen until a reboot allows the worm to drop its payload.

### Payload

This has been discussed in the host signatures section. Here is the MacAfee<sup>29</sup> write up

*"The worm will drop (and execute) the following Trojans on the victim machine:*

- [IRC-Sdbot](#) : %SysDir%\EXPLORER .EXE (12,832 bytes)
- [BackDoor-JZ](#) : C:\WINNT\LITMUS\SVCHOST32.EXE (17,440 bytes)

<sup>29</sup> [http://au.mcafee.com/virusInfo/default.asp?id=description&virus\\_k=100234#characteristics](http://au.mcafee.com/virusInfo/default.asp?id=description&virus_k=100234#characteristics)

*These Trojans are detected by McAfee products using the specified engine/DATs (or greater).*

*The following registry run keys are also created:*

- *HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\  
Run "LTM2" = C:\WINNT\litmus\SVCHOST32.exe*
- *HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\  
Run "Windows Explorer" = Explorer .exe*
- *HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\  
RunServices "Windows Explorer" = Explorer .exe*

*The [IRC-Sdbot] trojan<sup>30</sup> connects to an IRC channel and accepts commands from there. The commands are related to performing denial of service attacks and downloading and running files on the victim's computer.*

*Many different versions of this backdoor trojan are detected as BackDoor-JZ. The following description is fairly general, although port numbers, exact filenames and Registry key names typically vary between versions.*

*This, UPX packed, Trojan [BackDoor-JZ] opens TCP/IP port 30005 on a victim's machine. An attacker can then open, execute and delete files on the user's local system. They can also shutdown windows, and send out pings.*

*The trojan also copies itself to the Windows directory as traywnd.exe and adds the following Registry key value to allow the program to load at startup:*

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion_  
\Run "Taskschd" = %WINDIR%\traywnd
```

*Other versions of this Trojan copy themselves to a directory named 'Litmus' in the Windows directory (with varying filenames), hooking the Registry in a similar manner to above.*

*The source of this backdoor program is available among the hackers and there are many variants available (so some variants are frequently detected by other AV programs under different names).*

*A couple of points to mention here:*

- *The Explorer .exe, IRCbot, was observed to make the NetBIOS name queries described earlier (Figure 19 & Figure 20).*
- *No network traffic was observed as a result of the SVCHOST.exe, BackDoor-JZ. Therefore, in order to see if what was happening at the TCP level, TCPView<sup>31</sup> freeware from SYSInternals was used. The Task Manager and netstat each reveal different halves of the story – TCPview fills in the gap and maps the processes to the ports.*

<sup>30</sup> [http://vil.nai.com/vil/content/v\\_99410.htm#VirusChar](http://vil.nai.com/vil/content/v_99410.htm#VirusChar)

<sup>31</sup> <http://www.sysinternals.com/ntw2k/source/tcpview.shtml>



Process	Protocol	Local Address	Remote Address	State
System:8	UDP	0.0.0.0:445	0.0.0.0	
System:8	UDP	10.1.1.202:137	0.0.0.0	
System:8	UDP	10.1.1.202:138	0.0.0.0	
svchost.exe:384	UDP	0.0.0.0:135	0.0.0.0	
services.exe:212	UDP	0.0.0.0:1026	0.0.0.0	
lsass.exe:224	UDP	10.1.1.202:500	0.0.0.0	
System:8	TCP	0.0.0.0:445	0.0.0.0	LISTENING
System:8	TCP	0.0.0.0:1027	0.0.0.0	LISTENING
System:8	TCP	10.1.1.202:139	0.0.0.0	LISTENING
svchost.exe:384	TCP	0.0.0.0:135	0.0.0.0	LISTENING
mstask.exe:520	TCP	0.0.0.0:1025	0.0.0.0	LISTENING
SVCHOST32.exe:816	TCP	0.0.0.0:113	0.0.0.0	LISTENING

Figure 44 TCPView capture of SVCHOST.exe

TCPView clearly showed that the [BackDoor-JZ](#) (SVCHOST.exe) was intermittently listening on TCP port 113, not port 3005 as it details in the MacAfee site. Ironically, port 113 is a well known port for authentication services (normally unused). Once again, this is another indication that this is a variant with no accurate write up. The service would listen for about 10seconds and then sleep for one minute. No connections were ever attempted, as this machine was not connected to the Internet.

## Keeping Access

This is where the bank robber metaphor does not work. In real life a bank robber will not carry out this phase of the attack –unless a hostage situation developed!

At this point the worm has dropped its payload of the two Trojans into

- %SYSTEM%\Litmus\SVCHOST32.exe
- %SYSTEM%\system32\Explorer.exe

It has also successfully modified the registry to start both of these at boot time

- HKLM\Software\Microsoft\Windows\CurrentVersion\RunServices\Windows Explorer\Explore .exe
- HkeyCurrentUser\Software\Microsoft\Windows\CurrentVersion\Run\LTM2\%SYSTEM%\system32\SVCHOST32.exe

The work done so far to achieve the exploit is typically followed by an attempt to keep future access to the exploited machine. There are two main ways our worm does this task.

- The registry modifications above will launch the dropped executables SVCHOST32.exe and Explorer .exe on reboot. Note there are many other

different ways that the various Windows versions use to launch programs at startup. It is very difficult to manually find the startup method and there are many programs that can assist in locating them. Microsoft's own MSCONFIG.exe can be used but is not shipped with all versions. It can be downloaded [here](#). More information on startups can be found [here](#).

- TCPView (Figure 44) clearly showed the [BackDoor-JZ](#) Trojan intermittently launching and listening on Port 113. This Trojan is known to connect to an IRC server and so accept commands. These could be used to further reconfigure the machine in order to continue to keep access even after the original infection was cleared.

The terms "backdoor" and "trojan" need to be further clarified, as they are actually two different terms. A Trojan is a piece of malware that is disguised to look like some useful piece of software. Hence "Explorer .exe" is a Trojan. A backdoor is a piece of software that facilitates the circumvention of normal computer security controls. The two can be mixed together as in this case, to become a Trojan Horse Backdoor i.e. a useful looking process that allows backdoor access.

### ***Covering Tracks***

Typically, during this phase, attackers try to

- remove any traces of what they did by modifying UNIX log files or Windows Event Viewer
- hide files (change file attributes or prepend a period to the file in UNIX)

As mentioned earlier, this is a worm that makes a minimal attempt to cover its tracks. The use of the names Explore .exe and SVCHOST32.exe were designed to hide them in the process list as on first glance they look like normal Windows processes. A person familiar with the normal process list would see this anomaly immediately. However, for the most part, many would probably miss this clue.

The entry for the Explore .exe in the registry was also designed to hide among expected startups by using the string name "Windows Explorer".

The entry for SVCHOST32.exe was a bit less stealthy in that the string value name chosen was "LTM2".

Lastly the intermittent nature of the launching of the Backdoor Trojan was designed to hide from a curious sys admin who might run a cursory "netstat -an" to check on listening ports (Figure 45).

```
C:\WINNT\system32\cmd.exe
H:\>netstat -an

Active Connections

Proto Local Address          Foreign Address         State
TCP   0.0.0.0:25              0.0.0.0:0               LISTENING
TCP   0.0.0.0:110             0.0.0.0:0               LISTENING
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   0.0.0.0:1047            0.0.0.0:0               LISTENING
TCP   0.0.0.0:1049            0.0.0.0:0               LISTENING
TCP   0.0.0.0:1050            0.0.0.0:0               LISTENING
TCP   0.0.0.0:1052            0.0.0.0:0               LISTENING
TCP   0.0.0.0:1053            0.0.0.0:0               LISTENING
TCP   0.0.0.0:1099            0.0.0.0:0               LISTENING
TCP   0.0.0.0:1109            0.0.0.0:0               LISTENING
TCP   0.0.0.0:2166            0.0.0.0:0               LISTENING
TCP   0.0.0.0:2441            0.0.0.0:0               LISTENING
TCP   0.0.0.0:3460            0.0.0.0:0               LISTENING
TCP   0.0.0.0:3465            0.0.0.0:0               LISTENING
TCP   0.0.0.0:3555            0.0.0.0:0               LISTENING
TCP   0.0.0.0:3891            0.0.0.0:0               LISTENING
TCP   0.0.0.0:6164            0.0.0.0:0               LISTENING
TCP   0.0.0.0:6165            0.0.0.0:0               LISTENING
TCP   0.0.0.0:9495            0.0.0.0:0               LISTENING
```

Figure 45 netstat -an

## The Incident Handling Process

### *Incident background*

On Tuesday afternoon of May 19<sup>th</sup>, I was at my desk checking e-mail. John in the sales department phoned to report that something strange was happening to a couple of machines in our demonstration room. John's description of the problem – "my laptop and the demo workstations are running really slow. Rebooting has not helped!" - immediately sounded much more interesting than e-mail! The problem could have been caused by slow server response but as I had finished my e-mail so went to take a look. I have a secondary responsibility as a member of the regional incident response team and so followed our incident handling procedures (detailed later) and went to investigate.

John was in the process of making a presentation to a large group of people from a prospective major client. Our company had just released a new version of our financial trading client software that was viewed as a competition killer. John had been attempting to explain the benefits to the client using a mixture of a PowerPoint presentation and real time demonstration. Due to the problems the client had left and the sales demo had gone badly wrong. It was apparent John was looking for someone to blame.

As soon as I sat in front of one of the workstations I confirmed that the Windows machines and his laptop had a problem. Any task that was attempted such as opening explorer or the start menu took minutes to occur. The actual success meant that Windows was not frozen just extremely slow.

Further questioning revealed that John had worked on his presentation both in the office and on his laptop at home the previous evening. He had arrived early

to add some final customized PowerPoint screenshots of the new software in action and so had connected his laptop to the targeted network.

At this point I had no idea what was happening but suspected some form of worm activity, as a number of machines were symptomatic. Unlike a virus, worms can self-replicate across a network without human intervention. Viruses generally hide inside a file and require someone to launch them – or often as e-mail attachments. The problem was found but it took a while to understand the complete puzzle and to be confident of a solution to the problem before bringing the network back on line. The following process documents the six step incident handling procedure that our company has implemented.

## Preparation

Our company spans the globe and our data is our lifeblood. As such, we have implemented strong security policies and procedures in order to protect both the data and our company. We have invested a lot of effort into making these more accessible for employees. A recent redesign of the Global Security web site allows for quick and easy site navigation (Figure 46). For example, the patch management policy can be found in two clicks as can the acceptable use of the Internet policy (Figure 47).



Figure 46 GIAC Global Security Website

#### APPLICABILITY

This Policy applies globally to all systems within GIAC Enterprises that have connections to the Internet.

This specifically includes:

- Direct Internet connections used by production systems;
- Direct Internet connections used for product development, testing, demonstration and support;
- Any Internet connections that are not via a Corporate Internet Gateway, for example, "fast" Internet access by Editorial;
- Any connections with third-party suppliers, business partners or consultants that are made over a dedicated Internet connection (even where a VPN is used).

The Policy does not apply to systems that connect to the Internet via an approved Corporate Internet Gateway.

If you are unsure whether this policy applies to an Internet connection that you manage or use, please send an email to [secrisk@giac.com](mailto:secrisk@giac.com). Do not assume that it is someone else's problem.

#### RESPONSIBILITY

Responsibility for complying with this policy rests with all individuals who use, manage or own any of the Internet connections described above.

**This policy plays a key role in securing GIAC infrastructure and systems. Failure to meet these responsibilities is therefore a disciplinary offence which may result in dismissal.**

#### Figure 47 GIAC Internet policy

These policies lay down the basis for the procedures that have been put in place in order to safeguard the company data, personnel, office environments etc. One of these policies defines the role of the Information Security Incident Response Team (ISIRT). This is a team that has been set in place and is prepared to handle any information security "incident". This has been defined within our company as

"An incident can be defined as an event that interrupts/hinders normal operating procedure and precipitates some level of crisis.

Incidents, specifically are computer intrusions, denial of service attacks, insider theft of information and any unauthorized or unlawful network based activity that require computer security personnel, system administrators, or computer crime investigators to respond."

The team is comprised of cross-functional members who are physically located in the GIAC head office in New York as that is where our most of our staff and our physical data centers are located. The team is comprised of staff from HR, legal, technical, public affairs among others. However regional security team members throughout the globe have been appointed as a rapid response measure and are kept in the loop during monthly telephone conference meetings.(Figure 48). These are used to quickly determine the severity of a local regional incident and as a trained liaison with the main ISIRT. They are trained to contain the spread, if possible, as they will undoubtedly be the first on the scene. There are only two members of each regional team and one must always be available by cell or bleeper. If a regional incident occurs and only one member is available then a coworker is deputized to provide supervised support. In the

Deborn worm incident the local team simply consisted of myself and one contractor, whose normal main role was local desktop support.

REGIONAL SECURITY CONTACTS	NAME	PHONE NUMBER	MOBILE NUMBER
Asia Pacific (Primary)	<a href="#">LipPing</a>	+ 65 170 1 1 1	+ 65 9762 71 3
Asia Pacific (Secondary)	<a href="#">George</a>	+ 65 667 1 1 74	+ 65 91 1 1 31
Americas (Primary)	<a href="#">Sunny</a>	+ 1 48 1 1 1 89	+ 1 917 96 1 1 2
Americas (Secondary)	<a href="#">Andre</a>	+ 1 61 225 00 1 1	+ 1 771 131 1 1 281
Japan	<a href="#">Toshiy</a>	+ 81 3432 1 1 1	+ 81 90 1 1 1 1 893
Back of Beyond	<a href="#">Sean</a>	+ 1 207 542 1 1 1	+ 1 779 1 1 1 1 1 3
Back of Beyond 2	<a href="#">Ian</a>	+ 1 707 1 1 1 1 84	+ 1 47 1 1 1 1 1 34
Back of Beyond 3	<a href="#">Gary</a>	+ 1 420 1 1 1 1 1 9	+ 1 479 1 1 1 1 1 579

Figure 48 Regional Security Team Contact Details

All members of the regional teams carry all the contact details of the ISIRT members. This is kept up to date through normal e-mail channels and verified by occasional testing. There are no formal qualifications to joining the team, just really a willingness to learn and be involved. My own personal interest in computer security was how I “made the team”.

Incident handling training was given to me when I first took on the role and is updated by formal training such as SANS, e-learning and peer group contact. Email notifications are sent out to the global IRT team almost every day to keep abreast of current threat levels.

A vital part of the incident handling preparation phase is making sure that employees know whom to contact when they first notice an incident. In a large departmentalized company this may be difficult but can be achieved by a number of different mechanisms.

- The web site has already been mentioned and is very common nowadays, however people need to know about it in the first place. If the company has a corporate home page, this can be used to highlight incident issues and so gain greater user “visibility”.
- New hires should be educated about incident handling as part of their induction process.
- E-mail is also another well-established common method – however bear in mind that many people suffer from “email fatigue”.
- Integrate incident handling into an Employee of the month scheme.

Another vital part of the preparation phase is what is termed a “jump bag”. This is actually a suitcase held at each location and is preloaded with almost everything

that experience has shown is required to handle an incident. The thought behind this is that once an incident happens there is a lot of pressure to get control of the situation and get everything back to normal. The jump bag is a tried and tested method for making sure that everything the incident handler will need is available when needed. It is also continually evolving as new ways are found to accomplish tasks.

The jump bag contains the following:

- A selection of batteries
- A tape recorder with spare tapes
- Notepads (with page numbers)
- Pens (pencils are not allowed)
- Copy of the ISIRT call list
- Cell phone with spare batteries and charger
- Incident handling forms
- Copy of Norton Ghost 2003 software to perform backups (cdrom and boot floppies).
- Ghost image files on CDROM
- Standard OS media
- A HUB and Cat 5 cables
- Power leads and Power strip (6 outlet)
- Small computer toolkit
- A couple of 20Gb IDE drives
- 2 x 64MB USB RAM
- 2 boxes of blank floppies
- 100 blank CDs.
- CD containing incident software such as Foundstone / Sysinternals downloads
- USB CD Burner & software
- Linux boot disk for Windows password recovery<sup>32</sup>

A major required component that is not specifically included in the regional jump bag is a powerful laptop loaded with network analyzer software. This is partly a cost issue but mainly it was felt that the regional members already had their own personal laptop with loaded software. As a matter of policy, this laptop would be fully patched and, more importantly, the incident handler would be fully comfortable using it under pressure.

It should be emphasized that the jump bag must only be used in an incident and must always be fully stocked and batteries fully charged. As a matter of procedure it has to be checked monthly.

---

<sup>32</sup> [home.eunet.no/~pnordahl/ntpasswd/bootdisk.html](http://home.eunet.no/~pnordahl/ntpasswd/bootdisk.html)



The incident handling preparation phase also includes taking measures to prevent an incident from happening in the first place. (It should be bourn in mind that the security stance of the whole target network was changed when all the access controls were removed).

In the targeted network in the sales demo room, all the systems were built from a fully patched Ghosted image. They were not running any host firewalls as they relied on the corporate protection of the central IDS and firewalls. They were kept up to date with Windows patches during the period when the network was managed the technical department. These patches were often downloaded from the Internet by temporarily connecting the segment to the Internet segment as the corporate Internet connection, at that time, was often very slow. Also during this time, no AV software was loaded.

Windows 2000 security policies are very granular and can be modified to really limit the tasks a user can perform on the machine. Initially, the machines were all logged on with "User" account privileges, which provided limited machine protection. This laws later changed when the administrator account was instigated.

All these machines were ghosted and backups were held in a physically different location. There were no other technical countermeasures taken to guard this network against an attack.

## ***Identification***

The second phase of the incident handling process is the identification or detection of an incident. The longer it takes to detect an incident the more time it to spread. This is similar to a fire – the quicker you react, the quicker you can contain the fire and limit the damage. However, as in fire control, you have to call in trained professionals for their expertise in handling the problem in the most efficient way.

Identification can occur at different levels depending on what preparatory measures have been taken. The installation of an Intrusion Detection System, for example, may provide an early warning system even before it reaches a host machine. A crucial ally in this identification phase is the actual end user highlighting a suspect issue to the right person - as happened in this case.

### **9 a.m. May 19th**

I received the initial call about the incident from John at 9 a.m. on May 19<sup>th</sup>. Computer incidents can have a real emotional effect on many people. Irreplaceable data may have been lost or a large sale opportunity missed as in our incident. John was severely frustrated and looking for someone to blame. However, remaining calm is the best way to extract as much information about the incident as possible. This is not the time to jump to any conclusions but I was



fairly sure from John's problem description that this was definitely an "incident" that had to be investigated. My first responsibility, as defined in our guidelines, was to confirm the incident. My office is in the same building on a different floor so I grabbed my laptop and jump bag and went to investigate. As I walked to the elevator I called my colleague Ian, a technical contractor, to come and assist in case he was required. Incident handling is not a solitary pursuit. If the incident was confirmed then my next immediate action is to escalate by contacting the main ISIRT team. In this case there may be many tasks to be carried out and it is much more efficient using at least two people.

The first handler's duties include

- To maintain the communications channels with the ISIRT team and local management
- To maintain a complete record of all actions taken.
- Physical security of the evidence – logs, disks, backups, tapes etc
- Supporting the other handler in containing the situation

T

Ian used the forms from the back of the SANS book<sup>33</sup> to maintain a record of our actions. The reason for these strict procedures is that at the outset of an incident there is no way of knowing the severity and it may turn out that the incident may result in criminal proceedings. Therefore the recorded information may have to be used as evidence in court.

The second handler's main task is initially information gathering to verify an incident is under way.

#### **9.15 a.m. May 19th**

The targeted network is physically located in a demonstration room and so the first task was to physically secure the room. As John was the only person in the room I asked him to leave his laptop and go back to his desk as I did not want to risk a possible production network infection. I had already received the problem symptoms from him during our telephone conversation.

#### **9.20 a.m. May 19th**

Sales Demo 1, Sales Demo 2 and John's laptop where extremely slow. Sales Demo 3, Sales Demo 4 and the Internet machine did not seem to have a problem.

The obvious next step was to open the Task Manager on all the slow machines to have a look at the process list and CPU utilization. All the slow machines had a CPU utilization of 100% and displayed at least one process called ~2.exe (Figure 49, Figure 50). Ian also noticed that there appeared to be two explorer.exe processes one of which had a capital E. It was immediately obvious that this was not normal behavior and so the incident was confirmed.

---

<sup>33</sup> Computer Security Incident Handling Version 2.3.1 Stephen Northcutt  
[https://store.sans.org/store\\_item.php?item=62](https://store.sans.org/store_item.php?item=62)

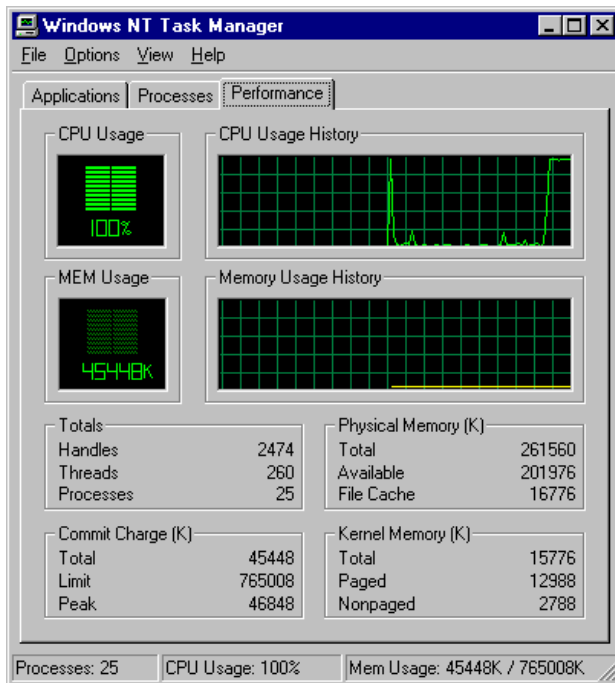


Figure 49 CPU 100% Utilization

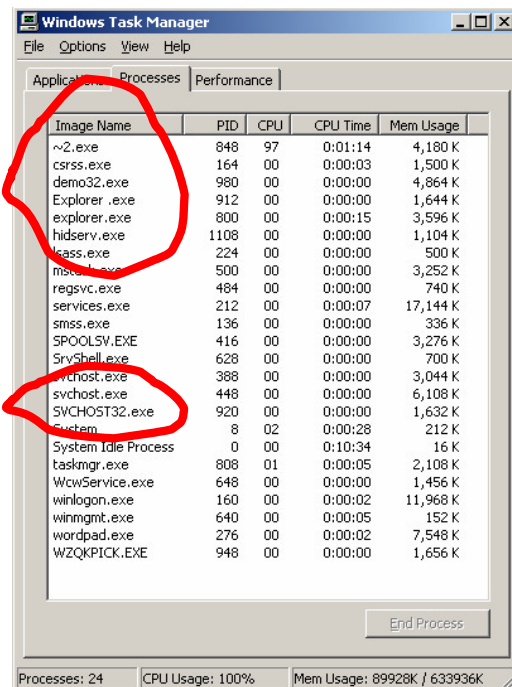


Figure 50 ~2.exe rogue process

Unsurprisingly, Sales Demo 2, Sales Demo 4 and the Internet machine all had normal task lists and normal CPU utilization.

#### 9.25 a.m. May 19th

Ian escalated the confirmed incident to the main ISIRT team, the local sales director and regional technical director.

### Containment

This is the third stage of the process and the objective is to analyze the incident and stop it from getting worse. This can be quite complex depending on the nature of the problem. There are a few issues to consider. Take, for example, a commercial web site that has been defaced. Should it be taken off line and rebuilt immediately? Should just the affected index page be replaced? If the problem results in financial loss then the hard drive contents may be required as evidence in a trial and have to be preserved. Should the police be informed?

These questions and scenarios are all part of the preparation phase and as many of these situations are covered as possible beforehand. This is also another reason the ISIRT team comprises cross-functional departments. The bottom line is that in most cases, business needs will normally have priority over all other considerations. That means that the incident handling process must pay close regard to business needs.

#### 9.35a.m. May 19th

The target network was not part of a production environment so a team decision was made to isolate the network from the corporate WAN to contain any further

spread. This was simply a matter of unplugging the WAN connection from the HUB. Obviously, as discussed earlier, the spread of infection across computing networks can be extremely fast and we may have been “shutting the door after the horse had bolted”. However, as we did not know at this stage what we were dealing with, this was agreed as the correct course of action.

This is the stage in the process where backups of infected hard drives are normally made. If the incident becomes the subject of a criminal investigation, the original disc may need to be used as evidence and so must be kept as pristine as possible. A bit by bit copy of this disc would be made for further forensic copies for later analysis. One of these copies may be used to return the system back to production once the source of infection has been identified and removed. Note that this is not the time to learn how to do this! Creating a forensic image is something that should be practiced beforehand.

Norton Ghost 2003 by default does not do a bit by bit copy, it will do a logical or “native” copy which will just copy all the files and directories. The reasoning behind this is that this is the quickest way to backup a disk and that is what the majority of people use the product for. A forensic copy takes much longer as it requires all the information on a disk not just the files that can be seen including

- Unused disk space
- Slack space at the end of a cluster. Sometimes this contains information from previously deleted files. Also information can be hidden here.
- Bad sectors. It is possible to manually mark sectors as bad and use them to store information
- Unpartitioned space

It is necessary to use the advanced option switch “-IR” (Image Raw - Figure 51) to create an exact bit-by-bit copy. More information on this topic can be found [here](#).

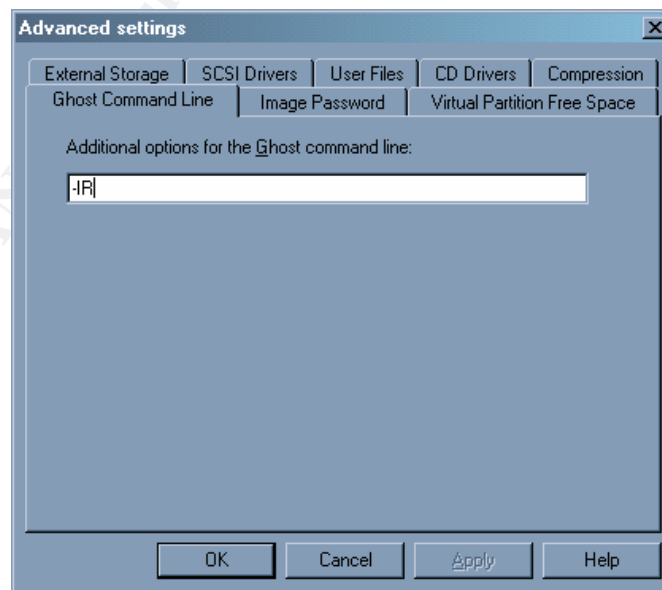


Figure 51 Ghost Image Raw configuration

#### 9.40a.m. May 19th

I removed Sales Demo2 to make a backup of the drive. After powering down the machine, the case was opened and one of the spare IDE drives from the jump bag was attached to the machine's IDE cable. The ghost boot floppy was inserted and the machine powered up. The backup was started using the "-IR" switch.

#### 10.00a.m. May 19th

As well as copious notes describing every action, Ian used the digital camera to keep a record of all screen shots. These could be easily zipped and sent to other members of the ISIRT if required.

John called wanting to know what I had found out. It is dangerous to jump to conclusions so I calmly explained that I was still investigating. I also wanted to contain the information about the incident until we had more information.

The main ISIRT team were also liaising with the IS sys admins and security departments that monitor the corporate firewalls. I did not know it at the time, but they had noted no abnormal network activity or unusual log entries.

The status of all the machines on the target network had not apparently changed since the incident commenced and so it was decided to attach the network analyzer to the target network. Note that one other item of preparation had been carried out to safeguard the laptop - regular laptop Ghost backups are done weekly to safeguard any local data.

I manually configured an IP address, connected the laptop and fired up Etherpeek. This was an isolated network segment and network traffic was expected to be minimal. This was not the case as an almost continual stream of ARP requests was being generated from Sales Demo 2(Figure 52)

Packet	Source Physical	Destination	Delta Time	Summary
1	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF		10.0.0.2 = ?
2	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000456	10.0.0.3 = ?
3	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000534	10.0.0.4 = ?
4	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000540	10.0.0.5 = ?
5	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000602	10.0.0.6 = ?
6	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000536	10.0.0.7 = ?
7	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000527	10.0.0.8 = ?
8	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000548	10.0.0.9 = ?
9	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000530	10.0.0.10 = ?
10	00:C0:4F:59:F6:D6	FF:FF:FF:FF:FF:FF	00.000629	10.0.0.11 = ?

Figure 52 Incremental ARP requests

#### 10.15a.m. May 19th

The Etherpeek trace was the first indication that we were probably dealing with worm like activity, which propagates across a network. As soon as this was noted we passed this information to the main ISIRT and this allowed them to monitor different parts of the network for unusual ARP activity. It appeared at that time that would be a good indicator of the spread of the worm. No abnormal activity was reported anywhere on the WAN but close monitoring continued.

### 10.25a.m. May 19th

By now Sales Demo 1 had finished cloning. The cloned drive was removed and Ian put it in one of the labeled evidence bags. This drive would be used only if some problem occurred with the original disk, which would be used for evidence. As a matter of procedure a second clone was then done, as this would be used for analysis and further investigation.

The next step taken was to remove Sales Demo 2 from the network to check if all abnormal traffic stopped. There are some pieces of malware that monitor the network interface and will destroy evidence when removed. To prevent this happening, the small hub from the jump bag was used to keep the link active while maintaining separation.

### 10.35a.m. May 19th

It was time to review the facts and work out the next course of action.

### 10.45a.m. May 19th

The consensus of opinion was to examine known Windows startup locations on Sales Demo 3 to see if anything unusual was noted. This was an educated guess as worms were well known to use the network to propagate and infect a machine on reboot by modifying a startup hook.

There are many ways that Windows uses to launch programs on startup and Sysinternals<sup>34</sup>, who produce a number of Windows freeware utilities, have a program called Autoruns that quickly lets you monitor what will start at boot time.

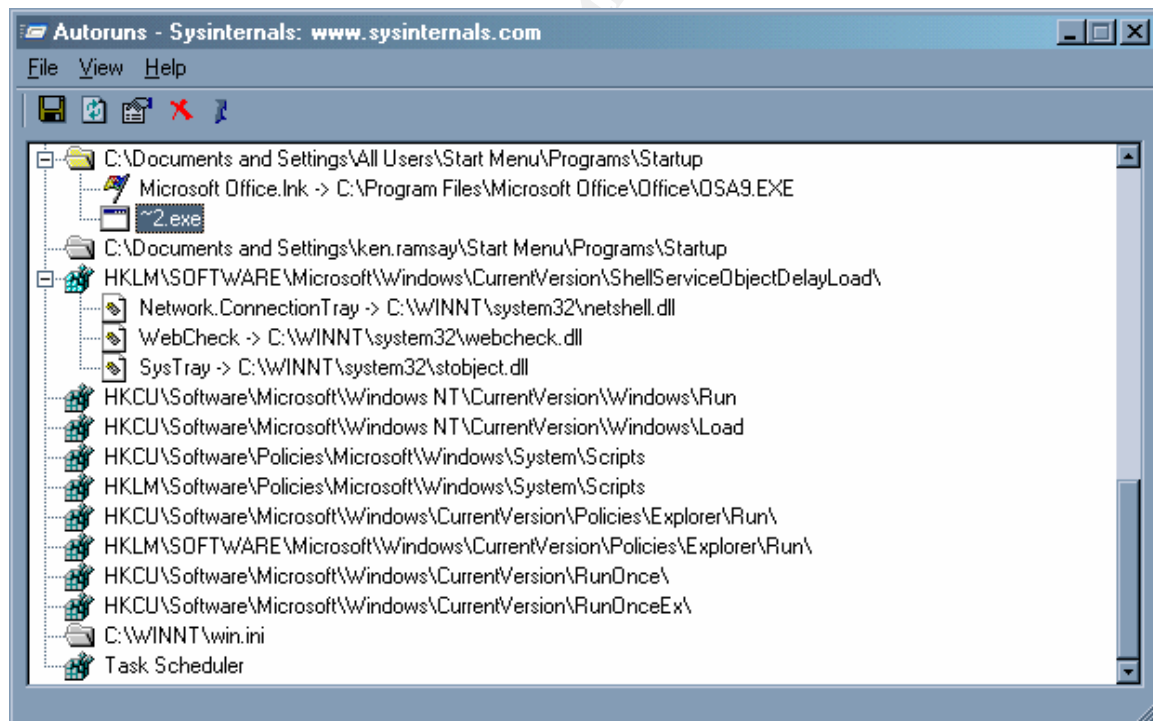


Figure 53 Sysinternals Autoruns

<sup>34</sup> <http://www.sysinternals.com/ntw2k/utilities.shtml>

#### **10.50a.m. May 19th**

After running Autoruns (Figure 53) from the jump bag CD, it was immediately seen that C:\Documents and Settings All Users\Start Menu\Programs\Startup had been modified.

It was obvious that this contained the worm as the name of the file contained was the same as the hog process that was seen on the infected machines --2.exe. At roughly the same time the ISIRT team confirmed with Ian that the worm was indeed recognized as a member of the Deborm family.

Interestingly, we now noticed that we had missed one pretty obvious symptom of the infection - the presence of rogue process SVCHOST32.exe in the task list.

I was informed that there was a data sheet on this worm on the Network Associates site and so I printed this off from the stand-alone Internet machine .We were now armed with a lot more information to analyze the infected machine Sales Demo 2. This would further confirm the identity of the worm.

#### **11.00a.m. May 19<sup>th</sup>**

Armed with the NA document I confirmed all the symptoms displayed information matched the documented information.

### ***Eradication***

This fourth stage in the incident handling can be one of the trickiest. It may be very tempting to just rebuild the compromised system to get back to normal business as quickly as possible. Sometimes this may be the only option as competitive forces might dictate that a prolonged downtime might be totally unacceptable. Sometimes this may be compromised by the lack of a recent backup (or none at all!) Normally the OS rebuild may be straight forward, but perhaps there is no data backup available. These types of issues are all part of the preparation stage and are the reason for scenario planning.

In order to prevent a re-occurrence of the attack it is necessary to determine how the attack happened in the first place. If, for example, a system is re-imaged, there is a strong risk of re-infection. It is therefore necessary to fully understand the vulnerability that was exploited so that a solid defense can be implemented. The more complicated the attack the longer this may take.

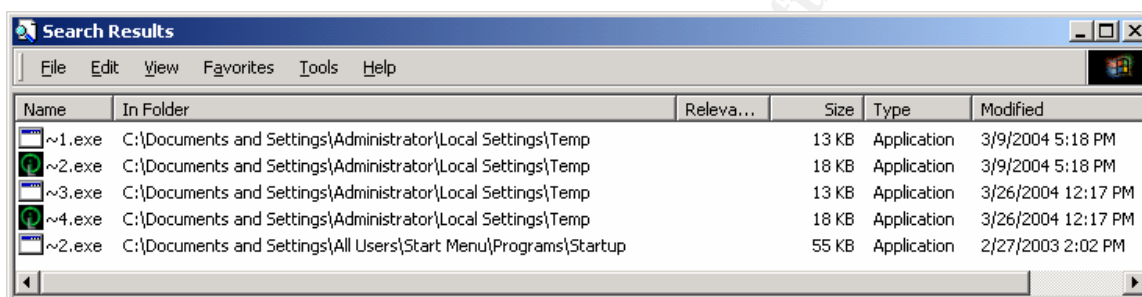
#### **11.05a.m. May 19<sup>th</sup>**

The deborm worm infection process was now reasonably clear. The machines were all confirmed to have a SMB share of the hard disk named "C" and a blank administrator password. We were all in agreement that it was not necessary to re-image all the machines in order to remove the infection. There were three main steps to fix the infection on Sales Demo 1 and 2:

- **Kill the infected processes.** This was achieved by simply bringing up the task manager and right clicking on ~2.exe, Explorer.exe, and SVCHOST.exe processes. Further investigation later showed that repeated reboots would spawn a number of random number of ~\*.exe

processes. These also needed to be killed. (Figure 54) (Note that none of the processes were seen to ever restart automatically after they had been manually killed).

- **Remove the dropped files** from the machine.
  - ~2.exe (and all of the clones) from the startup and temp directory (Figure 54)
  - C:\WINNT\system32\Explorer.exe
  - C:\WINNT\litmus\SVCHOST32.exe (also remove directory)
- **Clean the registry** – delete :
  - HKLM\Software\Microsoft\Windows\CurrentVersion\RunServices\Windows Explorer\Explore.exe
  - HkeyCurrentUser\Software\Microsoft\Windows\CurrentVersion\Run\LTM2\%SYSTEM%\system32\SVCHOST32.exe



**Figure 54 Cloned worm files**

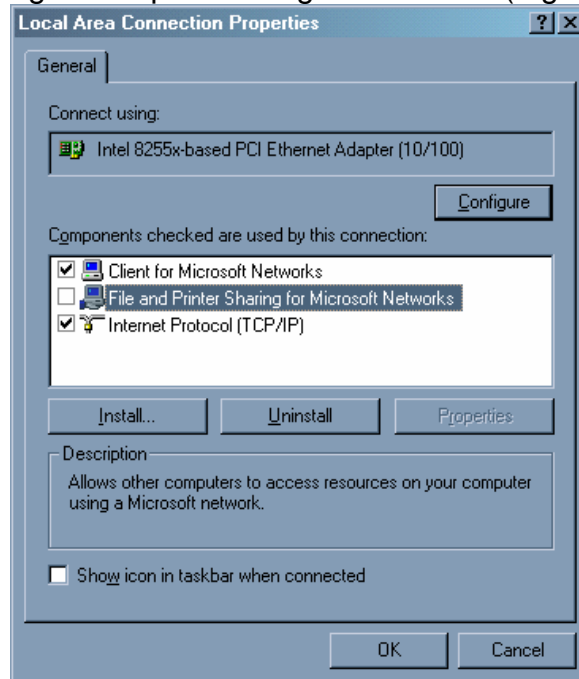
As Sales Demo 3 and 4 had not been rebooted the only requirement was to remove the one file ~2.exe.

**11.20a.m. May 19<sup>th</sup>**

The machines were all disinfected but now it was necessary to fix the vulnerability that the worm used to spread. The attack had highlighted the poor security stance of this network and so before bringing the machines back on line the following changes made to help secure the system for future sales demonstrations

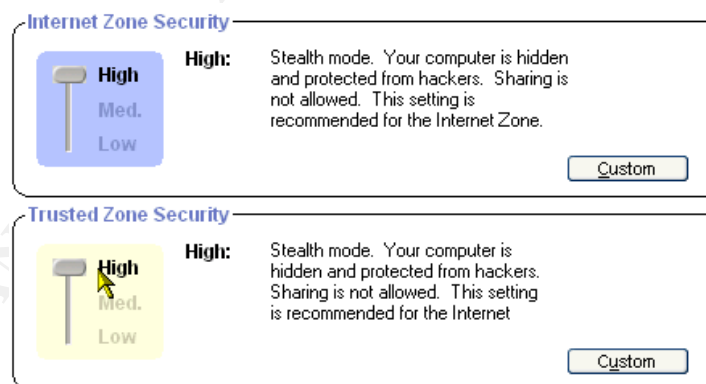


- File and Print sharing removed – no NetBIOS used. Port 139 will therefore no longer accept incoming connections (Figure 55)



**Figure 55 File Print share removal**

- Administrator account renamed and strong password applied
- Zone Alarm Pro installed and default firewall rules amended . By default the Trusted Zone is set to medium which allows SMB shares on ports 139 or 445. This was changed to High and an admin password set to prevent modification of the rule.



**Figure 56 Zone Alarm settings**

- The Internet HUB was removed and bandwidth increased on corporate Internet access.
- All machines were loaded with the corporate AV software – Trend Micro Virus.
- Remote IDS sensor installation on the network to be prioritized for next quarter.

Auditing features were turned on on all the machines by going to



- Administrative tools,
  - Local Security Policy,
    - Local Security Settings,
      - Audit Policy ,
        - Audit Logon Events
          - check Logon Failure.

Disinfecting John's laptop was a different story. John had not had his machine backed up and had added a lot of "custom" applications that were not GIAC certified. It would have been very difficult to and time consuming to try to certify that his machine had no other malware. This turned out to not be a problem as John was dismissed for breaking a number of security policies and never returned to the office. His laptop was recycled.

## ***Recovery***

This part of the process is when a decision is made to put the system back on line. We were confident that the complete system was worm free. However it is always wise to be careful and so testing to prove that the vulnerability is gone is a very important part of the procedure.

All the machines were connected to the Hub and powered up with the WAN connection removed. The network analyzer program was then used to monitor traffic. The idea was to run the worm executable and see what happened. This could not be done on any of the machines as the AV software stopped the process from executing. The AV software on this one machine was therefore turned off temporarily and the worm process launched. The laptop immediately displayed the incremental ARP scan. Each of the machines were seen to respond to the ARP request and the attempt TCP connection attempt to port 139 was observed. In each case the three-way handshake was not completed and the ARP scans re-commenced indicating a failure of the exploit.

The worm cleanup routine was then carried out as detailed previously. After discussions with the main ISIRT team it was decided that the incident was now fully under control and the new security measures described earlier would eliminate any reoccurrence of the problem. The WAN connection was therefore restored and as an added precaution, the network analyzer was left on the segment for the rest of the day.

## ***Lessons Learnt***

There were many lessons learned from this rather simple and lame attack. These were all discussed in depth at our monthly conference call, which went on a bit longer than usual. It has always been our security director's attempt at these

meetings to nurture a supportive atmosphere and to encourage frank and open exchange. As in all incidents there really are two main aspects to discuss

- Technical contributing factors
- The human “weakest link”

## Technical

Most of these have been discussed except for the reason for the direct Internet connection. The Internet connection was installed before Internet policy statements were produced and as it was under a local cost center, it was quietly forgotten about. Staff all enjoyed unrestricted access to the Internet sites that was not available through the corporate proxy. I include myself in this group as a lot of sites containing security information are banned on our corporate web. However this access coupled with the local placement of a simple hub, was now sent to clearly be a possible threat vector to the corporate network. This type of installation bypassed all the corporate safeguards such as IDS and firewalls. The only protection from the outside world depended on the host based firewall configuration. The Internet connection was therefore removed and a clearer process introduced to modify the web filters for business requirements. A internal web site was set up to simplify the procedure for setting up access to web sites for which there was a business requirement.

In some cases direct connection to the Internet was still required and a clear process was setup to monitor this access. Each access was assigned a business owner and it was their responsibility to ensure compliance with proper procedures.

One general lesson that we all agreed on was that the laptop acceptable use policy needed to be more strongly emphasized. Laptops are well known to be a weak link in the security chain and this needed to be sorted. It was proposed to setup a laptop working group to investigate laptop hardening. Also a laptop education road show was also proposed to explain the dangers of hotel/home connections.

Physical security in a shared demonstration room is very difficult, however a major contributing factor in this case was the simple access to a desk-mounted hub. If this had not been available this problem might not have occurred. This hub was removed.

The failure to implement a password policy was also a major contributing factor and even a simple password would have prevented this attack. After discussions with the Sales Director, agreement on a satisfactory password policy was reached.

There is no business requirement for SMB shares to either egress or ingress from the public Internet. Policies were all updated to state mandatory compliance was required. A validation program was instated. At the same time a review of

SMB requirement was scheduled to see if a more secure corporate standard could be implemented. Furthermore, OS hardening in general was discussed to see if a corporate standard for demonstration rooms could be implemented.

## **Social**

This problem would never have happened if the Sales Director had not forced the change of network access control. However, the technical department is there to support the corporate business needs and this is why security policies are written. They are available as clear guidelines for all to access.

John clearly disregarded many Internet access and laptop policies. The reasons behind this were discussed in depth and an outcome of this was that a corporate questionnaire would be produced to test knowledge of policies. It was clear that now the issue would now be a hot topic as it had affected business. It was time to strike while the iron was hot and raise the problem with senior management. It would be proposed that all staff would be directed by e-mail to the Group Security Policy web site and mandated to sign an on-line form. This would indicate that they have read and understood the appropriate policies. A follow up at local level would be necessary to keep the topic in the employee radar. John was well known in the company and so it was thought that this might not be impossible to achieve! It was also proposed that local managers be put through a short appropriate computer security course.

## **Extras**

This was a pretty lame worm but it still got through and did a considerable amount of damage. Only today I was called by a diligent colleague who noticed that his login audit logs grew each day. On further investigation I could see a couple of "cousins" of my old friend was still around and located somewhere in Buenos Aires and Sydney! The logs showed that Administrator, Owner and Guest had all tried and failed to gain access. The logs also contained the machine name, so a quick call and the machines were identified -the owners were no longer in the company! Probably their machines were probably left under a disused desk somewhere quietly waiting for a power outage to come to life. This feature of "hard to kill" was touched on in the text. If the machine is never rebooted then in many cases the payload will not be activated. In the case of a server in a cupboard this may take a while.

Now imagine this simple worm did not look just for a share named C, but also C\$, D\$, share, etc. then it would have been a lot more effective. Then imagine that a file with a simple password list is accessed and sequentially tried. This is not imagination, this already exists in the wild and is also considered old by now. There are many other flavors of the Deborm and the [Mumu.bat](#) worm that work this way. Worms are evolving. Security Strategy for Predictive Systems VP [Ed Skoudis](#) hypothesizes that the worms we have seen so far, like Deborm and

Code Red<sup>35</sup>, are mild in comparison to the coming Superworms. These will build on the worm strategies already seen but will combine the best of breed techniques in an almost intelligent fashion. Nimda was one of the first of this breed and combined a lot of the techniques of earlier worms but rolled them into one attack package. Therefore it is imperative to study and understand these building blocks. An interesting paper on three family types is available [here](#).

Operating system patch levels were not a factor in the Deborn attack, but many worms around today look for OS weaknesses and contain buffer overflow attacks to gain access to the host. It is therefore imperative to keep systems patched to lessen risk exposure to this threat.

How do you quickly check to see if a machine is sharing files ? Right click on My Computer, Manage, Shared Folders, Shares (Figure 57). The screen shot shows the default shares. Clicking on Sessions will also show if anyone is connected.

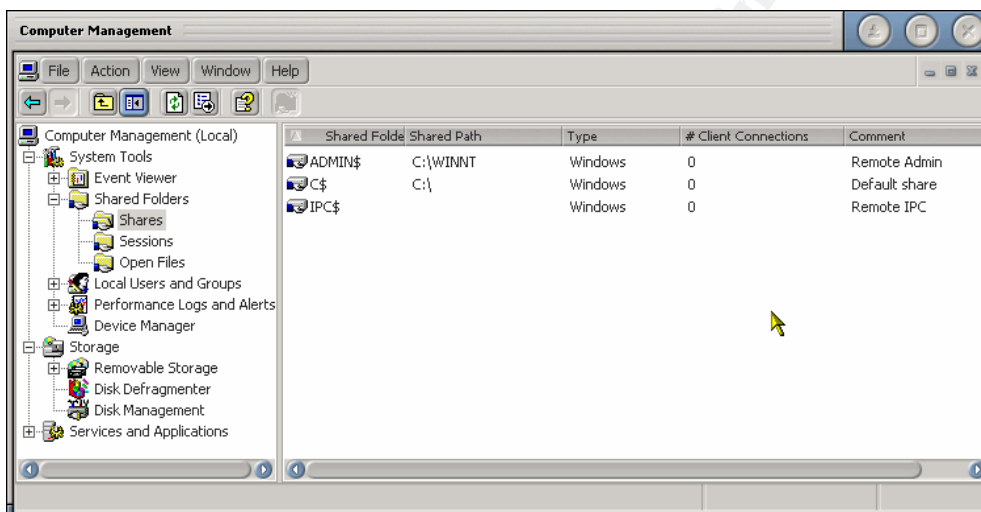


Figure 57 Determining Shared folders

There a number of very readable books out there on the subject that even the novice will be able to follow. These are detailed in the reference section.

## Conclusion

There is a saying that is repeated often during flight training about accidents happening because of a chain of events. Break any link in that chain and you prevent the accident. In many ways this is relevant to each exploit. Malware looks for a certain number of conditions to be fulfilled before the end payload can be dropped. Remove any one of the conditions and that **single** exploit will be stopped. However, to continue the flying metaphor, imagine a situation where a wing breaks off and is magically repaired then seconds later the engine seizes, the tail falls off, the cockpit fills with smoke..... do you get the point?

<sup>35</sup> <http://www.cert.org/advisories/CA-2001-19.html>

# References

## GIAC Practicals

Michael S. Kriss "Weak Passwords + Null Session = Windows 2000 Exploit"  
[http://www.giac.org/practical/Michael\\_Kriss\\_GCIH.doc](http://www.giac.org/practical/Michael_Kriss_GCIH.doc)

Joe Finamore "NULL Sessions In NT/2000"  
[www.sans.org/rr/papers/67/286.pdf](http://www.sans.org/rr/papers/67/286.pdf)

Lloyd Conner "Top Ten Port 139"  
[http://www.giac.org/practical/Lloyd\\_Conner\\_GCIH.doc](http://www.giac.org/practical/Lloyd_Conner_GCIH.doc)

## Other References

Just what is SMB? V1.2 Richard Sharpe 8-Oct-2002  
<http://samba.anu.edu.au/cifs/docs/what-is-smb.html>

Implementing CIFS: The Common Internet File System; Christopher Hertel, Prentice Hall PTR; 1st edition (August 14, 2003)

Malware: Fighting Malicious Code; Ed Skoudis, with Lenny Zeltser, Prentice Hall PTR; 1st edition (November 9, 2003)

Counter Hack: A Step-by-Step Guide to Computer Attacks and Effective Defenses; Ed Skoudis Prentice Hall PTR; 1st edition (July 23, 2001)

TCP/IP Illustrated, Volume 1: The Protocols; W.Richard Stevens, Addison Wesley 1994

Viruses Revealed; C. David Harley, David Harley, Urs E. Gattiker, Robert M. Slade, McGraw-Hill Osborne Media; (September 21, 2001)

Hacking Exposed; Stuart McClure, Joel Scambray, George Kurtz McGraw-Hill Osborne Media; (Third (2003) & Fourth Edition 2004)

Windows XP Professional Security; Chris Weber and Gary Bahadur McGraw-Hill Osborne 2002