Netsky.p Mass Mailer Worm Analysis

GIAC Certified
Incident Handler

Practical Assignment
Version 4.0

Dwayne Edwards
Submitted January 9, 2005

Table of Contents

Table of Figures

Abstract

By understanding in detail one variant of the class of malicious software known mass mailing worms it becomes easier to understand them all. This paper covers the Netsky.p mass mailer of 2004. This is by many accounts the largest worm of 2004. Defending against and handling Netsky.p will not guarantee success against all mass mailers. However it should make the incident handler more adept with them. The baseline knowledge can be extended across close and distant variants to good effect.

Statement of Purpose

This paper covers the Netsky.p worm. This particular worm is of interest for a number of reasons. It typifies the whole class of mass-mailing worms, uses file sharing programs for propagation, and takes advantage of a vulnerability in Microsoft Internet Explorer. This vulnerability obviates the social engineering aspect of the mass mailer. Understanding Netsky.p allows the reader insight into the whole class of email worms such as Sobig, Mydoom, Bagle and their variants. Netsky.p uses the classic mass mailer technique of installing an SMTP server on the desktop. In addition it also uses the simple propagation technique of copying itself into likely file shares and peer-to-peer file sharing points. For users of Microsoft Internet Explorer 5.1 and 5.5 the vulnerability described in MS01-020 propagates the worm without user interaction.

Netsky.p has been reported as the largest worm of 2004[1]. This after many similar worms caused widespread damage. Repeated reporting of problems and warnings to not open unknown attachments seemed to help little. The mass mailers continue to have their adherents in the hacker community. Many of the worm variants are running into the double letter designations. Netsky.p seemed to gain popularity from the use of references to Harry Potter and Britney Spears among others[2]. Because of this Netsky.p deserves a closer analysis.

The lab environment includes an IBM ThinkPad running Windows 2000 as a host system to VMware workstation. This allows for conservation of hardware, ease of containment to the lab network (the network is one PC not connected to any other machine) and a quick reversion to known clean states without the time consuming re-install of operating systems. The target machines running virtually will be Windows XP Service Pack One, and Windows 2000 Professional Service Pack . The attack test will be propagated via both the email and file share method to check success in both area and to examine mitigation factors. Also included in the lab is a server running an SMTP software and DNS resolution software.

This paper describes in detail, the actions of the worm in its propagation, the files, registry settings and other evidence left on computer. It also covers the vulnerability used as part of the propagation methodology. A variety of standard tools available for free on the Internet and included in Windows will track the progress and actions of the worm as it infects a computer. These will include Ethereal, regedit and others. The paper also covers the methodology an incident handler can use to find and mitigate problems associated with this mass mailer and others of its type.

The Exploit

Exploit Name

Common Name - The name of the exploit/worm is Netsky.p. In general it spreads through social engineering although it does exploit a vulnerability in the Microsoft Internet Explorer code that displays emails.

Common Vulnerabilities and Exposures (CVE) - CVE-2001-0154 - [3]
HTML e-mail feature in Internet Explorer 5.5 and earlier allows attackers to execute attachments by setting an unusual MIME type for the attachment, which Internet Explorer does not process correctly.
This from CVE Version: 20040901
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0154

OVAL ID - OVAL141 - More information from Mitre using the Open Vulnerability Assessment Language (OVAL) can be found at[4]:
http://oval.mitre.org/oval/definitions/pseudo/OVAL141.html

The OVAL web site includes many of the vulnerabilities in the CVE. Reading through the entries gives detailed information as to registry entries and computer settings that make a computer vulnerable to exploits. OVAL is a security industry effort to build a common language and database to describe vulnerabilities. More can be found at http://oval.mitre.org/index.html

Bugtraq Entry - Bugtraq Archives – [5]
http://www.securityfocus.com/archive/1/172665

Microsoft Security Bulletin - MS01-020 -[6]
Incorrect MIME Header Can Cause IE to Execute E-mail Attachment
http://www.microsoft.com/technet/security/bulletin/MS01-020.mspx

Microsoft Knowledge Base Number – 290108 -[7]
http://support.microsoft.com/default.aspx?scid=kb;en-us;290108

CERT Advisory Number CA-2001-06 -[8]
http://www.cert.org/advisories/CA-2001-06.html

Variants - Netsky.p is the 16th variant of the Netsky mass mailing worm virus (Netsky a to af and sometimes called Buchon). The close relatives all share the common root name as is industry practice. Mass mailers such as Sobig, Mydoom, Bagle, Buchon and others share some of the same traits, which will be explained elsewhere in this document.

Operating Systems

Operating System Versions affected by the MIME Header Vulnerability[9]:

- Windows 95
- Windows 98
- Windows Me
- Windows NT
- Windows 2000
- Windows Server 2003
- Windows XP

Virtually all Windows Operating Systems service pack levels are vulnerable unless they include the upgrade to Internet Explorer 6.0 or above. The real concern is the version of Microsoft Internet Explorer that is used. These vulnerable versions are[9]:

- Microsoft Internet Explorer 5.01
- Microsoft Internet Explorer 5.01 SP1
- Microsoft Internet Explorer 5.5
- Microsoft Internet Explorer 5.5 SP1
- Microsoft Internet Explorer 6.0 (in some upgrade circumstances)

There are hotfixes available to the SP1, but it is easier, if possible, to move to SP2 for a number of security rollups. Newer versions of Internet Explorer are not vulnerable, as long as the user selects full or typical install from the menu when upgrading (on 95, 98, 98SE or ME)[7]. Microsoft does not test unsupported versions on their software, so it is unclear whether earlier versions are vulnerable or not. As a reminder, patching this vulnerability does not stop the social engineering part of the Netsky.p, only the automatic execution through an email previewer.

Protocols, Services and Applications

The Netsky.p worm mainly uses the email infrastructure of the Internet to spread. The major protocols, services and applications are: SMTP, POP3, MIME, and email clients such as Outlook and Outlook Express. Netsky.p also uses file sharing systems such as the native Windows file share, ftp, http and 3rd party systems like Morpheus and BearShare.

The most important protocol to understand with regards to Netsky.p is Simple Mail Transfer Protocol (SMTP). SMTP works at layer 4 of the Open Systems Interconnect (OSI) Model. A major key to the propagation of the Netsky.p worm is the silent install of a small, fast SMTP server on the target system. We will look at the overall attack and order of infection in the next section.

SMTP

SMTP is the protocol that is responsible for transmitting email from a client machine to an email server. It is described in RFC 2821[10] which supersedes RFC 821[11]. In the SMTP world there are clients and servers. Please refer to figure 1 for the following description. In general, client is what we normally think of, a person's workstation, PC etc. This is the workstation PEBBLES. The SMTP Server BARNEY acts as a server for the PC, however, if it working as a relay (the mail needs to be sent on to other systems for final delivery) then BARNEY is a client to server FRED. Each SMTP server may have different roles depending on the conversation it is having. You may also hear the terms "sender" and "receiver" used, this may add clarity if you are talking about the server role in any given transfer.

At a high level, the email transfer path follows that depicted in figure 1. Client PEBBLES wants to send an email to client BAMBAM. She composes it in Microsoft Outlook and clicks send. Outlook opens an outgoing connection to server BARNEY on port TCP 25. The TCP protocol is beyond the scope of this document.

The server BARNEY receives the email message and concludes it is bound for a user in another domain. BARNEY does a DNS lookup and finds the address of the server in charge of delivering email to that domain. In this instance it is server FRED. BARNEY forwards to email to FRED and FRED forwards the email on the POP3 server WILMA. WILMA stores the email until the client retrieves it. BAMBAM logs in and retrieve email via POP3 over TCP port 110.

In the figure, each service (SMTP, POP3) has its own physical device. In large scale email networks this is the case. In smaller setups the services may run on one physical server. Either way the protocols remain separate, SMTP outbound from a client until it is stored in a POP3 server for inbound delivery.[12]

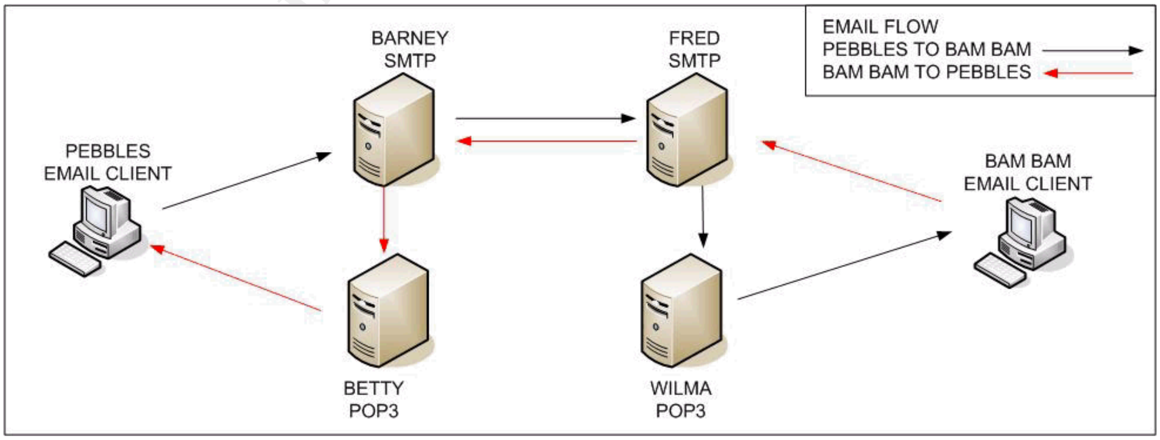


Figure 1 – Email System High Level View

Figure 2 is an Ethereal packet capture of a normal SMTP session sending a short email from PEBBLES to BAMBAM. For readability and space considerations the beginning ARP and TCP setup and the ending TCP FIN packets have been removed. The client to the server packets are highlighted in red, source is .44 destination .10. Server to client highlighted in blue. The IP address has also been truncated to allow for more room on the Information field.

```
SRC   Dest    Prot      Info

Removed TCP syn-syn ack packets
4   .10  .44     SMTP     Response: 220 dcelab.com SurgeSMTP (Version 2.1c7-7)
5   .44  .10     SMTP     Command: HELO pebbles
6   .10  .44     SMTP     Response: 250 dcelab.com. Hello pebbles (172.16.1.44)
7   .44  .10     SMTP     Command: MAIL FROM: <pebbles@dcelab.com>
8   .10  .44     SMTP     Response: 250 Command MAIL OK
9   .44  .10     SMTP     Command: RCPT TO: <bambam@dcelab.com>
10  .10  .44     SMTP     Response: 250 local recipient ok
11  .44  .10     SMTP     Command: DATA
12  .10  .44     SMTP     Response: 354 Command DATA Start mail input; end with
<CRLF>.<CRLF>
13  .44  .10     SMTP     Message Body
14  .10  .44     TCP     smtp > 1035 [ACK] Seq=209 Ack=1290 Win=64246 Len=0
15  .44  .10     SMTP     EOM:
16  .10  .44     SMTP     Response: 250 message sent ok
17  .44  .10     SMTP     Command: QUIT
18  .10  .44     SMTP     Response: 221 Command QUIT, disconnecting
Removed TCP FIN
```

Figure 2 – SMTP Protocol

As you can see SMTP commands are 4 characters in length and some have arguments. There are a few commands in addition to the ones above, but are not used often by the client. For additional information RFC 2821[11] is a great source of information.

For the communication above, the client (WIN2K running Outlook Express 5) connect via port 25 and establishes a TCP connection. The first SMTP communication seen is on line 4, the server identifying itself and waiting for a command.

Line 5 contains the first command from the client, a HELO command. This identifies the client to the server and starts the session. In addition the HELO command, the newer (RFC 2821)[11] command EHLO identifies the client and signifies it can process a greater range of service requests. The answer to an EHLO includes the SMTP extensions the server can understand and process. Since the simpler HELO was sent, in line 6 the server acknowledges the HELO.

With the formalities taken care of the client sends the first part of a mail, a MAIL FROM: command. This is relatively obviously the sender of the email on line 7. If you put in an invalid from address most SMTP email servers will take it. This is part of the problem as to how these mass mailers are successful.

The answer to the MAIL FROM: command is MAIL OK acknowledgement (line 8). Answers from SMTP servers are usually either a 200 series OK type message or a 500 series error type message. Again, refer to the RFC[11] for further details.

Next, the client tells the server the address of the recipient with the RCPT TO: command (line 9). This is checked by the server if the recipient is in its domain. If mistyped, or otherwise sending to a wrong address their server responds:

> 550 No such user (brambam) –ERR brambam@dcelab.com not found

If the domain is something else than local, the SMTP server forwards it on for delivery and the server in charge of that domain sends the error message back. A bad user notification eventually finds it way back to the sender.

Since the user is local and the address is OK, line 10 gives responds with a good acknowledgement (line 10).

Line 11 sends the DATA command, telling the SMTP server the body of the message is next.

The SMTP server responds positively in line 12 and tells how to signify the end of the message. This is a period (.) on a line by itself. You also see one of the other replies, a 300 series, from the server. This is a positive intermediate reply per the RFC[11].

The Message Body (line 13) is just that, if a large message is sent, it is broken up across many TCP packets. The standard for text based email is RFC 822[13] and includes the email sender and recipient, the subject, and some text. As mentioned above it ends with a period on a line all by itself. For today's email programs we use the MIME extensions, which will be discussed later in this paper. Appendix A shows this series of packets in detail.

Line 14, you will notice is a TCP ACK packet, again, large messages may have many of these.

The EOM: is the end of message on line 15, a period by itself. Then the server ACKs (16) the packet. The client then issues a quit on line 17 and the server disconnects the session on line 18. The deleted TCP packets are the FIN to the session at the TCP protocol level[12].

The above capture was done using Ethereal to capture the email traffic between Microsoft Outlook Express email client and the SurgeSMTP email server in the lab. You can also explore the commands and responses by telnetting into an SMTP server on port 25. As seen in the above exercise, it would not be hard to build a simple SMTP "server" for use as a mass mailer worm propagation method. This is exactly what the writer of the Netsky.p worm and other variants did.

SMTP, like most successful standards is simple and easy to implement. It powers virtually all the email on the Internet today through a simple set of commands. The whole class of mass mailer worms works because most SMTP servers will accept an incoming connection from any other server. Understanding how SMTP works is important to better understand how Netsky.p and its variants use this protocol to spread.

MIME

Although there is earlier work, the basic standard for sending text email messages is RFC822[13]. This was written in 1982 and contained much good work. However it had line length and message length limitations and could only send 7 bit ASCII text. The MIME extensions build on top of RFC822 and allow a much more flexible set of messages and attachments. MIME allows for character sets other than ASCII, unlimited length in line and message, use of fonts and multiple attachments, including application specific binary files. Figure 3 depicts the salient points of a typical multi-part MIME email.

```
1      From: "PEBBLES FLINTSTONE" <pebbles@dcelab.com>\r\n
2      To: <bambam@dcelab.com>\r\n
3      Subject: Attachment to show MIME headers\r\n
4      Date: Sat, 1 Jan 2005 17:46:56 -0500\r\n
5      MIME-Version: 1.0\r\n
6      Content-Type: multipart/mixed;\r\n
7      \tboundary="----=_NextPart_000_0009_01C4F029.E2AE5DE0"\r\n
8      X-Priority: 3\r\n
9      X-MSMail-Priority: Normal\r\n
10     X-Mailer: Microsoft Outlook Express 5.00.2919.6700\r\n
11     X-MimeOLE: Produced By Microsoft MimeOLE V5.00.2919.6700\r\n
12     This is a multi-part message in MIME format.\r\n
13     \r\n
14     ------=_NextPart_000_0009_01C4F029.E2AE5DE0\r\n
15     Content-Type: multipart/alternative;\r\n
16     \tboundary="----=_NextPart_001_000A_01C4F029.E2AE5DE0"\r\n
17     \r\n
18     ------=_NextPart_001_000A_01C4F029.E2AE5DE0\r\n
19     Content-Type: text/plain;\r\n
20     \tcharset="iso-8859-1"\r\n
21     Content-Transfer-Encoding: quoted-printable\r\n
22     \r\n
23     This sentence is the text body\r\n
24     \r\n
25     ------=_NextPart_001_000A_01C4F029.E2AE5DE0\r\n
26     Content-Type: text/html;\r\n
27     \tcharset="iso-8859-1"\r\n
28     Content-Transfer-Encoding: quoted-printable\r\n
29     \r\n
30     <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">\r\n
31     <HTML><HEAD>\r\n
32     <META content=3D"text/html; charset=3Diso-8859-1" =\r\n
33     http-equiv=3DContent-Type>\r\n
34     <META content=3D"MSHTML 5.00.2920.0" name=3DGENERATOR>\r\n
35     <STYLE></STYLE>\r\n
36     </HEAD>\r\n
37     <BODY bgColor=3D#ffffff>\r\n
38     <DIV><FONT face=3DArial size=3D2>This sentence is the text=20\r\n
39     body</FONT></DIV></BODY></HTML>\r\n
40     \r\n
41     ------=_NextPart_001_000A_01C4F029.E2AE5DE0--\r\n
42     \r\n
43     ------=_NextPart_000_0009_01C4F029.E2AE5DE0\r\n
44     Content-Type: image/jpeg;\r\n
45     \tname="Sample.jpg"\r\n
46     Content-Transfer-Encoding: base64\r\n
47     Content-Disposition: attachment;\r\n
48     \tfilename="Sample.jpg"\r\n
49     \r\n
50     /9j/4AAQSkZJRgABAgAAZABkAAD//gASQWRvYmUgSW1hZ2…… \r\n
51     AAD/7gAOQWRvYmUAZMAAAAAB/9sAhAAKBwcHBwcKBwc……..\r\n
      (MANY LINES OF THE ZIP FILE ENCODED AS ON 50,51)
181    ------=_NextPart_000_0009_01C4F029.E2AE5DE0\r\n
```

Figure 3 - MIME Format

For a balance between completeness and brevity included is a fair portion of the header as it passed through Ethereal, however, deleted are many of the lines after 54 as they are the actual encoded form of the sample. jpg included in the Windows 2000 under MyDocuments.

On lines 1 through 4 are some self explanatory RFC822 fields. The MIME field starts on line 5 with the statement of version. This identifies the email as a MIME email, as of this writing there is only one version of MIME, 1.0[14].

On line 6 is the field Content-Type, the real workhorse of the MIME definition. This is the instruction to the email reader program of how to handle the content that follows this section. The MIME authors expect that the seven types of content defined in the standard should cover most everything. They believe that no additional type need be added without changing the MIME standard. They do, however give themselves an out with the inclusion of “non-standard types”.

Each content type may have multiple subtypes. The idea behind the type/subtype model is the **type** specifies generally what the content is, and the **subtype** specifies the format precisely. In this manner, the email program may see it has some default way to handle Audio types even though it may not precisely know what to do with an .avi file. The seven content types are[15]:

- Application – A catch all type that fits in no other category, usually a binary type for the email reader itself. You may see Octet Stream, Postscript or others.
- Audio – Self explanatory, could be RealAudio, .mpg etc.
- Image – Again, subtype may be .jpeg, .gif etc.
- Message – Show an encapsulated message, may be from a forward, or a pointer to an external place or handler.
- Multipart – Used to show there are multiple different types of data within the email, used frequently with attachments. Also needs a boundary parameter explained later
- Text – default type, may see plain or richtext as a subtype
- Video – May see mpeg or avi as subtypes
- X-TypeName – A private or non-standard type

So, line 6 tells us that we have a multipart email data set.

Line 7, the boundary definition, is a unique string of data that is not used in the email, to signal the delineation of the different data types. You will see the same boundary string used throughout the email.

We next see a number of pieces of information that Microsoft Outlook puts in the pre-amble of the email. The real meat doesn’t start until we see the first boundary string. Line 15 is the boundary string. Next is the Content-Type multipart/alternative. Outlook Express puts the message in both text and html formats unless you change the defaults. The multipart/alternative tells the email reader program that these sections are the same content formatted differently. Typically an email reader will have a built in hierarchy of what data type is “best”. For example if the email is both .RTF and plaintext it will show the data in .RTF.

Line 20 is the boundary string again and starts the plaintext. Then we see another boundary at 27 and the html content type on line 28. If you look, you will see that line 25 and line 40 contain the brief content “This sentence is the text”.

Line 45 starts the last data section, as you see the type is image/jpeg. The attachment name is there as is the encoding type. The encoding type tells the email reader how to change the ASCII strings back to a .jpeg file. In this case the lines 52-54 show a typical encode, many lines were deleted. Then line 56 shows the end of the data sections.

MIME, like SMTP, is relatively simple to understand and implement. With a handful of commands and definitions it has added a high value to the utility of email programs. It is the MIME handler in Outlook and Outlook Express that passes control off the Internet Explorer. Internet Explorer is used to display HTML encoded emails in those applications. The flaw in Internet Explorer allowed the Netsky.p executable to run without the need for the user to click an attachment. The social engineering aspect was present also, so the two pronged approach helped Netsky.p spread. Understanding MIME is also basic to understanding the vulnerability taken advantage of by Netsky.p

POP3

Post Office Protocol 3 (POP3) is the other side of the email coin from SMTP. SMTP taken from the perspective of the user, is the outgoing email protocol. It stores emails only long enough to send them on to the next hop toward the recipient. POP3 is the incoming protocol where email is stored until retrieved. POP3 is described in detail in RFC1939[16]. A short POP3 session is depicted in figure 4. The client to the server packets are highlighted in red, source is .44 destination .10. Server to client highlighted in blue. Again IP addresses are truncated to allow for more room on the Information field

```
SRC    Dest      Prot      Info

Removed TCP syn-syn ack packets
4   .10   .44        POP     Response: +OK POP3 dcelab.com (Version 2.1c7-7)
http://surgemail.com
5   .44   .10        POP     Request: USER pebbles
6   .10   .44        POP     Response: +OK pebbles nice to hear from you –password required
7   .44   .10        POP     Request: PASS password
8   .10   .44        POP     Response: +OK pebbles has 1 mail messages
9   .44   .10        POP     Request: STAT
10  .10   .44        POP     Response: +OK  2 3172
11  .44   .10        POP     Request: LIST
12  .10   .44        POP     Response: +OK 2 (3172)
13  .10   .44        POP     Response: 1  1580
14  .10   .44        POP     Response: 2  1592
15  .10   .44        POP     Continuation
16  .10   .44        POP     Continuation
17  .44   .10        TCP     1028 > pop3 [ACK] Seq=39 Ack=183 Win=64058
18  .44   .10        POP     Request: RETR 1
19  .10   .44        POP     Response: +OK message follows
20  .10   .44        POP     Continuation
-------Many deleted-------------------------
42  .10   .44        POP     Continuation
46  .44   .10        POP     Request: DELE 2
47  .10   .44        POP     Response: +OK message deleted
48  .44   .10        POP     Request: QUIT
49  .10   .44        POP     Response: +OK closing connection
Removed TCP fin packets
```

Figure 4 – POP3 Protocol

This figure is similar in construct to the SMTP example. After the TCP three way handshake on line 4 the server responds with the +OK message and identifies itself. POP3 has a simpler construct for showing correct and incorrect commands than SMTP. You either receive an +OK or a –ERR as acknowledgement for a command.

In line 5 the client identifies itself with the USER command, the server responds on 6 with an +OK. Unfortunately this is a response that means the user exists. POP3 was written in a friendlier era on the Internet. Regardless, it's your email name and pretty easy to find out.

Next, on line 7 the client must authenticate using the PASS command, and yes, it is sent in the clear. Since this is a lab it was left it in the trace. The server then responds with the +OK on line 8 as it received the correct password. It also responds with how many email messages are there for pebbles. This is not necessary per the RFC and could be any greeting the writer of the server cooks up, like **HI!** Or **server ready** or **robots will destroy the planet**. The RFC just states positive response, one line greeting.

The system is now in the transaction state. There is no indication in responses, but the POP3 server is now ready to respond to requests for mail. Next, on line 9 the client sends a STAT command. The response to this is the drop list on line 10. The drop list shows how many messages the user has and their total aggregate size. The RFC highly discourages any other information on this line.

Since the answer to STAT was non zero, the client now does a LIST (line 11). The server lists the total number and total size of the messages, then each individual email and it's size (lines 12-14). The next interesting line is 18 where the client RETR command retrieves message one. Many lines are now cut out of the trace this is the sending of the messages to the client. The client then deletes the messages off the server and sends the QUIT command to signal the end of the conversation.

DNS

DNS is the lookup method that the Internet uses to pair hard to remember IP addresses with easier to remember host.domain names. This is so that people don't need to type in things like 172.16.1.10, when they want to hit a web site like barney.dcelab.com. DNS name resolution works through a hierarchy of servers and distributed administration that, much like HTML has whole books written on them. In the name of brevity, this paper will cover the local interaction between a client and the DNS server. For the purpose of this discussion there are 2 important records in a DNS database. One is an A record and looks like this in its simplest form[17].

```
barney.dcelab.com  172.16.1.10
```

As you can see it matches a name to an address, there are some other details but this is all that is required. The other entry is an MX or Mail eXchanger record. It looks like this[17]

```
MX     10      barney.dcelab.com
```

This signifies that the Mail eXchanger, in other words SMTP server, for dcelab.com is barney.dcelab.com. The 10 is a way to set precedence for companies that run more than one incoming email server. Note also the reference back to a name as opposed to an IP address. So for every MX record you should have an A record also. RFC1034 covers DNS, as do many good web sites and books.

HTML

Hyper Text Markup Language, or its result, is familiar to any user of the Internet. The flaw in Internet Explorer has to do with the way it treats MIME types that are inside HTML pages. Since Outlook and Outlook Express use Internet Explorer to display web pages the flaw in Internet Explorer 5.01 and 5.5 helped Netsky.p spread. HTML is outside the scope of this document, many good books and articles can help the reader understand HTML code. You may want to start with RFC1866[18]. The real keys to the Netsky.p are the email transport methods and the use of MIME.

Description of the Attack

Three vulnerabilities exist which make the Netsky.p worm successful. On an architectural level, the way the SMTP system works[12], and in particular, how it is configured, is a large vulnerability. On a user level, social engineering, in other words human weakness, is the vulnerability. On the individual PC technical level, there is a vulnerability in the Internet Explorer 5.01 and 5.5.

As explained earlier, SMTP servers can take the role of both server (receiver) and client (sender). Since there is little or no verification checking turned on for most SMTP servers, rogue SMTP servers can send email as a client (sender) to a regular SMTP server. A variety of schemes have been suggested, such as:

- SMTP Authentication in RFC 2554 some of this work is based on
- SMTP Service Extensions in RFC 1869
- Realtime Blocking List - rblsmtpd - http://cr.yp.to/ucspi-tcp/rblsmtpd.html although this is more of a blocker designed for spam than an authenticator
- User/Sender Based systems like POP-before-SMTP

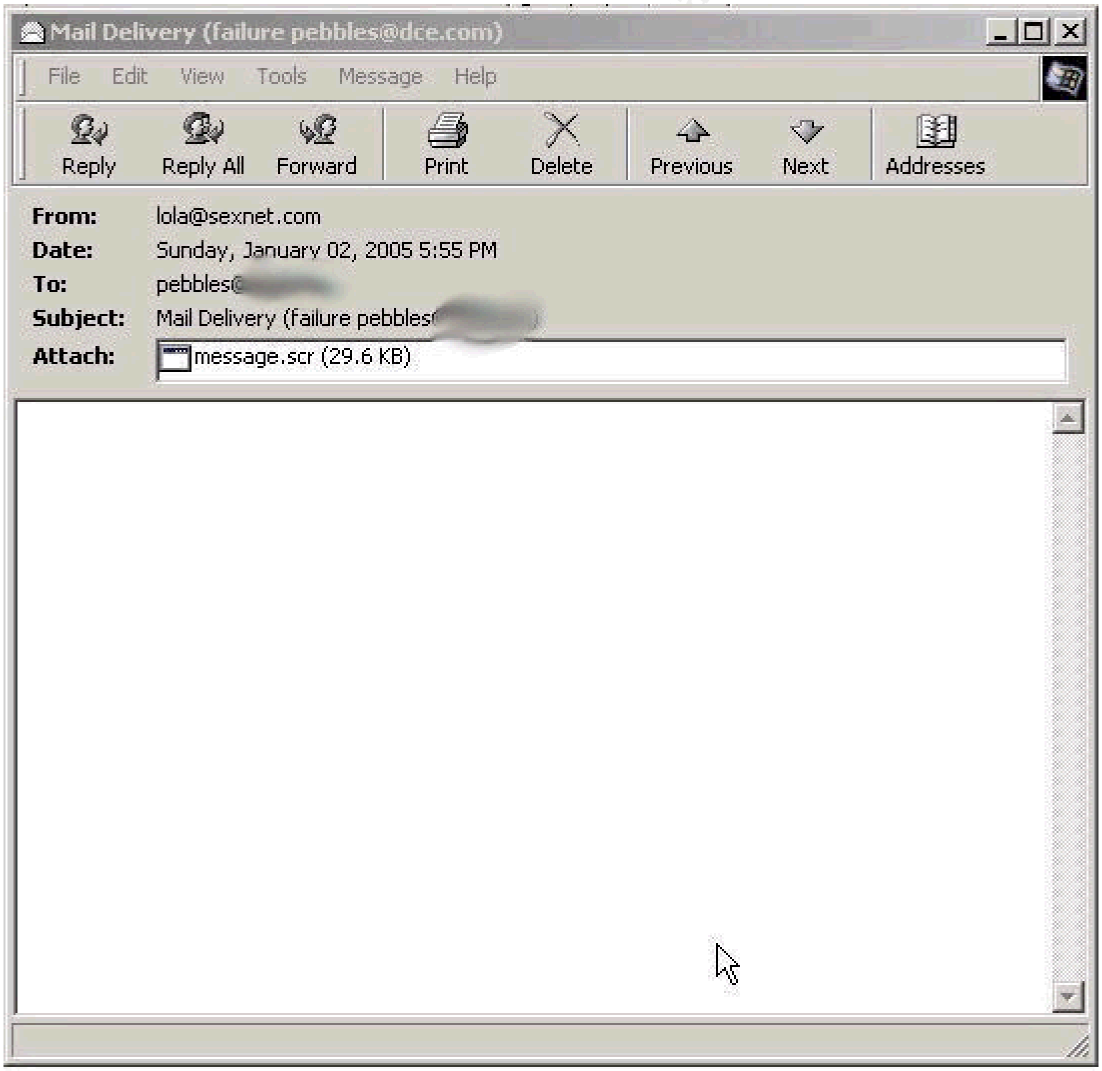
Although most SMTP servers in use today support authentication it would appear that few, if any use it. As usual security measures add a layer of complexity/administration onto the shoulders of staff at enterprises and ISP's alike. However, this would certainly drastically reduce problems of this type. Even a more even handed use of reverse DNS lookups, where we look to finding a system with MX records show some level of "authentication" not in evidence today.

The second vulnerability human nature, the social engineering aspect of both the email and the file sharing system exploit part of Netsky.p. This is exploitable for a variety of reasons. One is basic trust, most people trust others even when they have no basis for that trust nor may they even know the other person. This is especially true of someone at the other end of an email system. Next is the millions of computers out on the Internet that have been sold as home appliances as opposed to business tools. This is not going to change, but it is important to the business community to understand the existence and impact these folks have. User education on the subject of computer security continues to fail, even after the many lessons from these mass mailers. Windows XP

Service Pack 2 may be a step in the right direction, as are the growing number of ISP's doing traffic filtering and virus checks on email.

The last vulnerability is the software application vulnerability "Incorrect MIME Header Can Cause Internet Explorer to Run E-mail Attachment", in Microsoft Internet Explorer 5.01 and 5.5[6]. Other versions may also be vulnerable, but Microsoft does not test older unsupported versions. It is also possible to have newer versions of IE vulnerable if the following occurs. A user upgrades from a vulnerable 5.x version to 6.x and does not choose "typical" or "full install" from the IE 6.x install wizard[7]. Presumably the vulnerable .dll does not get overwritten.

This vulnerability, allows an attacker to run arbitrary code on the target machine. No user interaction is required by the user other than opening or previewing an email. In Outlook and Outlook Express HTML emails are rendered by Internet Explorer automatically in the preview pane.



From: lola@sexnet.com
Date: Sunday, January 02, 2005 5:55 PM
To: pebbles@[redacted]
Subject: Mail Delivery (failure pebbles[redacted]
Attach: message.scr (29.6 KB)

Figure 5 – Infected email

Figure 5 shows an email generated by the Netsky.p worm. The following section shows the user interface, some selected screen shots, and Ethereal captures to illustrate the exploit infecting a Windows 2000 virtual machine running Outlook Express 5 and Internet Explorer 5.5.

```
1   Content-Type: multipart/alternative;\r\n
3    ------=_NextPart_001_001C_01C0CA80.6B015D10\r\n
4    Content-Type: text/plain;\r\n
5    \tcharset="iso-8859-1"\r\n
6    Content-Transfer-Encoding: quoted-printable\r\n
7    ------=_NextPart_001_001C_01C0CA80.6B015D10\r\n
8    Content-Type: text/html;\r\n
9    <BODY bgColor=3D#ffffff>If the message will not displayed
10    ------=_NextPart_00_001C_01C0CA80.6B015D10--\r\n
11   Content-Type: audio/x-wav;\r\n
12   \tname="message.scr"\r\n
13   Content-Transfer-Encoding: base64\r\n
14  Content-ID:<031401Mfdab4$3f3dL780$73387018@57W81fa70Re>\r\n
15  TVqQAAMAAAAEAAAA//8AA
```

Figure 6 – Netsky.p in MIME Format

Figure 6 shows part of the raw email as captured by Ethereal. Some lines have been removed for clarity. The full text of the capture can be found in Appendix A. From the earlier dissection of MIME we can see that this is a multipart/alternative MIME message (line 1). It consists of an empty text/plain content section (line 4), a text/html content section, in lines 8, 9 and some removed text. Although the Netsky.p variant tends to be smoother in messages and the other social engineering aspects, notice the English language constructs. Many email worms are written by people whose primary language is not English and it shows up in the body of the message.

Last there is the attachment exploit section. In this instance a type audio/x-wav content type with the attachment message.scr. In Windows a .scr file is an executable. The incorrect (mismatch) MIME type sets off the vulnerability and causes the message.scr file to execute. Microsoft does not tell us which MIME content types are vulnerable. In testing, only x-wav and octet streams showed up. This should not be taken as a comprehensive list, but it appears that Netsky.p uses these 2 types. Disassembly of the worm would be necessary to be sure. To make this harder the worm writer has encrypted the .exe and the .dll that are placed on the target computer.

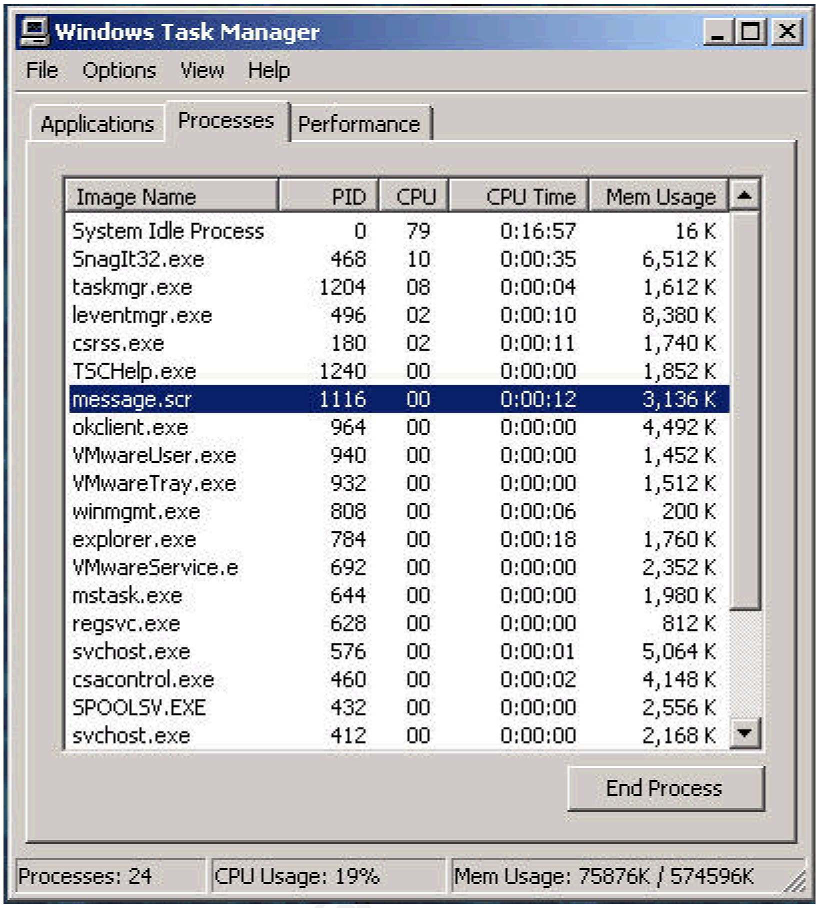


Image Name	PID	CPU	CPU Time	Mem Usage
System Idle Process	0	79	0:16:57	16 K
SnagIt32.exe	468	10	0:00:35	6,512 K
taskmgr.exe	1204	08	0:00:04	1,612 K
leventmgr.exe	496	02	0:00:10	8,380 K
csrss.exe	180	02	0:00:11	1,740 K
TSCHelp.exe	1240	00	0:00:00	1,852 K
message.scr	1116	00	0:00:12	3,136 K
okclient.exe	964	00	0:00:00	4,492 K
VMwareUser.exe	940	00	0:00:00	1,452 K
VMwareTray.exe	932	00	0:00:00	1,512 K
winmgmt.exe	808	00	0:00:06	200 K
explorer.exe	784	00	0:00:18	1,760 K
VMwareService.e	692	00	0:00:00	2,352 K
mstask.exe	644	00	0:00:00	1,980 K
regsvc.exe	628	00	0:00:00	812 K
svchost.exe	576	00	0:00:01	5,064 K
csacontrol.exe	460	00	0:00:02	4,148 K
SPOOLSV.EXE	432	00	0:00:00	2,556 K
svchost.exe	412	00	0:00:00	2,168 K

Processes: 24   CPU Usage: 19%   Mem Usage: 75876K / 574596K

Figure 7 – Message.scr in Task Manager

Figure 7 showing message.scr as a process running in the Windows Task Manager. This will change depending on the name of the attachment or shared file executed. This screen shot is immediately after infection. After a reboot the SMTP worm process will always be FVProtect.exe.

At this point the target machine is compromised and starts looking for propagation paths. The next section looks in detail at the effects of the worm on the target machine and the various methods to spread. All efforts have been made to go in strict sequence of events.

When Netsky.p is run, regardless of its infection vector, it performs the following[9]:

Creates the mutex **'D'r'o'p'p'e'd'S'k'y'N'e't'** a mutex is a mutual exclusion object. There is a class/object in the Microsoft Foundation class library to support mutex[19]. The use for a mutex is to provide for good control of resources. If there is a resource such that only one process or thread should access it at a time, such as a file or port, then a mutex should be created to show it is in use. The Netsky.p uses this mutex to insure only one instance of the worm is running.

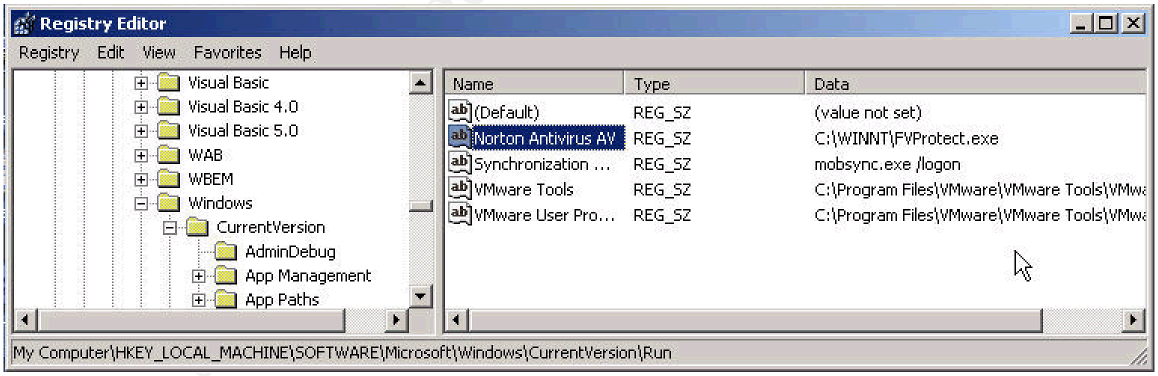
Using the %Windir% environmental variable copies itself to %Windir%\FVProtect.exe This is usually WINNT or WINDOWS depending on your operating system, or if you installed your Windows files somewhere else. From now on it will just be called the Windows directory[9].

Next it creates the file userconfig9x.dll in the Windows directory, which creates its own mutex **_-oO]xX|-S-k-y-N-e-t-|Xx[Oo-_** . This is the program that does the heavy lifting of SMTP engine, shared folder search and propagate and other nasty stuff[9].

Next, again in the Windows directory it creates[9:]

- ZIP1.TMP (40,882 bytes) – a zipped MIME encoded copy of the worm, contains the file document.txt <many spaces>.exe
- ZIP2.TMP (40,894 bytes) – like zip1, but data.rtf <many spaces>.scr
- ZIP3.TMP (40,886 bytes) – like zip 1 & 2, details.txt <spaces>.pif
- ZIPPED.TMP (29,834 bytes) - ZIP format copy of the worm
- BASE64.TMP (40,520 bytes) – MIME encoded copy of worm

Next Netsky.p changes the registry such that when you reboot it starts up again. The value "Norton Antivirus AV"="%Windir%\FVProtect.exe" for the registry key HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run .



Name	Type	Data
(Default)	REG_SZ	(value not set)
Norton Antivirus AV	REG_SZ	C:\WINNT\FVProtect.exe
Synchronization ...	REG_SZ	mobsync.exe /logon
VMware Tools	REG_SZ	C:\Program Files\VMware\VMware Tools\VMwa
VMware User Pro...	REG_SZ	C:\Program Files\VMware\VMware Tools\VMwa

My Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

Figure 8 – FVProtect.exe in the Registry

While in the registry the battle between the Netsky and Bagle, and other virus writers show up. The following registry keys are associated with Bagle, Nachi and MyDoom among others and are deleted:

From the registry key[9]:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

- Explorer
- system.
- msgsvr32
- winupd.exe
- direct.exe
- jijbl
- service
- Sentry

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run Services

- system
- Video

HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

- Explorer
- au.exe
- direct.exe
- d3dupdate.exe
- OLE
- gouday.exe
- rate.exe
- Taskmon
- Windows Services Host
- sysmon.exe
- srate.exe
- ssate.exe
- winupd.exe

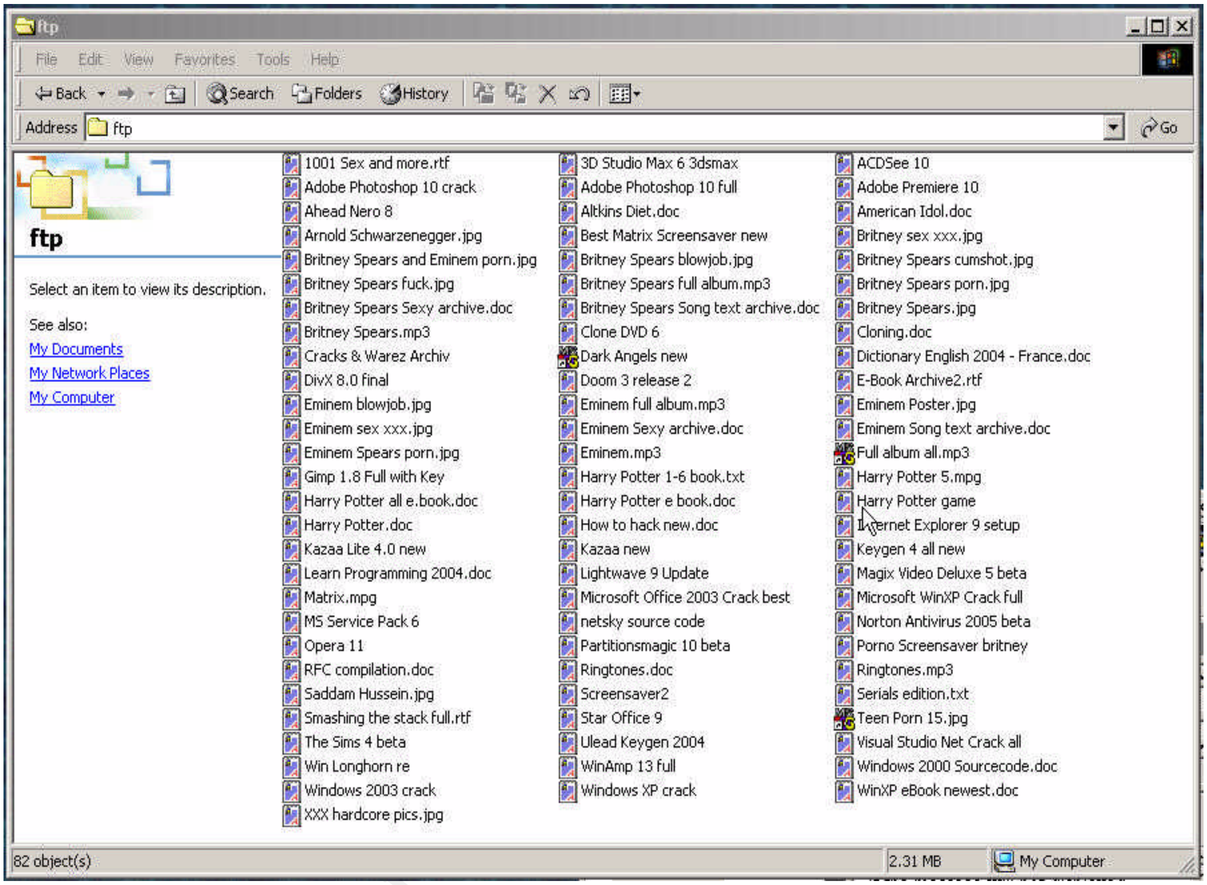
And the following subkeys

- HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\
- Explorer\PINF
- HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WksPatch
- HKEY_CLASSES_ROOT\CLSID\CLSID\
- {E6FB5E20-DE35-11CF-9C87-00AA005127ED}\InProcServer32

Next the worm looks for directories that are share points, the following directories are either common share names, common server share points or default share directories for various peer to peer file transfer programs[9].

- bear
- donkey
- download
- ftp
- htdocs
- http
- icq
- kazaa
- lime
- morpheus
- mule
- my shared folder
- shar
- shared files
- upload

Netsky.p then copies itself into that directory using the file names in figure 9. It does not seem to look any further. In the lab, if multiple directories existed one was chosen and the others ignored. If deleted, these files did not reappear after reboot to this or any other folder.



ftp

Select an item to view its description.

See also:
My Documents
My Network Places
My Computer

1001 Sex and more.rtf
3D Studio Max 6 3dsmax
ACDSee 10
Adobe Photoshop 10 crack
Adobe Photoshop 10 full
Adobe Premiere 10
Ahead Nero 8
Altkins Diet.doc
American Idol.doc
Arnold Schwarzenegger.jpg
Best Matrix Screensaver new
Britney sex xxx.jpg
Britney Spears and Eminem porn.jpg
Britney Spears blowjob.jpg
Britney Spears cumshot.jpg
Britney Spears fuck.jpg
Britney Spears full album.mp3
Britney Spears porn.jpg
Britney Spears Sexy archive.doc
Britney Spears Song text archive.doc
Britney Spears.jpg
Britney Spears.mp3
Clone DVD 6
Cloning.doc
Cracks & Warez Archiv
Dark Angels new
Dictionary English 2004 - France.doc
DivX 8.0 final
Doom 3 release 2
E-Book Archive2.rtf
Eminem blowjob.jpg
Eminem full album.mp3
Eminem Poster.jpg
Eminem sex xxx.jpg
Eminem Sexy archive.doc
Eminem Song text archive.doc
Eminem Spears porn.jpg
Eminem.mp3
Full album all.mp3
Gimp 1.8 Full with Key
Harry Potter 1-6 book.txt
Harry Potter 5.mpg
Harry Potter all e.book.doc
Harry Potter e book.doc
Harry Potter game
Harry Potter.doc
How to hack new.doc
Internet Explorer 9 setup
Kazaa Lite 4.0 new
Kazaa new
Keygen 4 all new
Learn Programming 2004.doc
Lightwave 9 Update
Magix Video Deluxe 5 beta
Matrix.mpg
Microsoft Office 2003 Crack best
Microsoft WinXP Crack full
MS Service Pack 6
netsky source code
Norton Antivirus 2005 beta
Opera 11
Partitionsmagic 10 beta
Porno Screensaver britney
RFC compilation.doc
Ringtones.doc
Ringtones.mp3
Saddam Hussein.jpg
Screensaver2
Serials edition.txt
Smashing the stack full.rtf
Star Office 9
Teen Porn 15.jpg
The Sims 4 beta
Ulead Keygen 2004
Visual Studio Net Crack all
Win Longhorn re
WinAmp 13 full
Windows 2000 Sourcecode.doc
Windows 2003 crack
Windows XP crack
WinXP eBook newest.doc
XXX hardcore pics.jpg

82 object(s) 2.31 MB My Computer

Figure 9 – Netsky.p Files in C:\FTP

A searchable, text version of this list is in Appendix B. Netsky.p has successfully set up the other social engineering trick it plays. This list contains a little something for almost anyone. These files are placed in directories that are likely to be shared. If someone came across them they may be tempted to grab a copy for themselves. Once executed, the worm goes back to its two pronged approach regardless of the vector. The Harry Potter book files helped boost Netsky.p in October, long after the worm had been released[2].

Now that the worm is ensconced on the target machine, the real work starts, the mass mailer propagation. The SMTP engine is running and ready to go, first, Netsky.p scans all logical drives that are not CD-ROM's. Any files with the extensions listed in figure here are searched for email addresses[9].

- .adb
- .asp
- .cgi
- .dbx
- .dhtm
- .doc
- .eml
- .htm
- .html
- .jsp
- .msg
- .oft
- .php
- .pl
- .rtf
- .sht
- .shtm
- .tbb
- .txt
- .uin
- .vbs
- .wab
- .wsh
- .xml

Netsky.p uses the harvested addresses as both the to and from in the SMTP MAIL command. It also carries a set of spoofed email addresses to use as the from address. After a file is searched, Netsky.p does not search it again. However this table must be internal as a reboot will cause new emails to be sent to the harvested emails again.

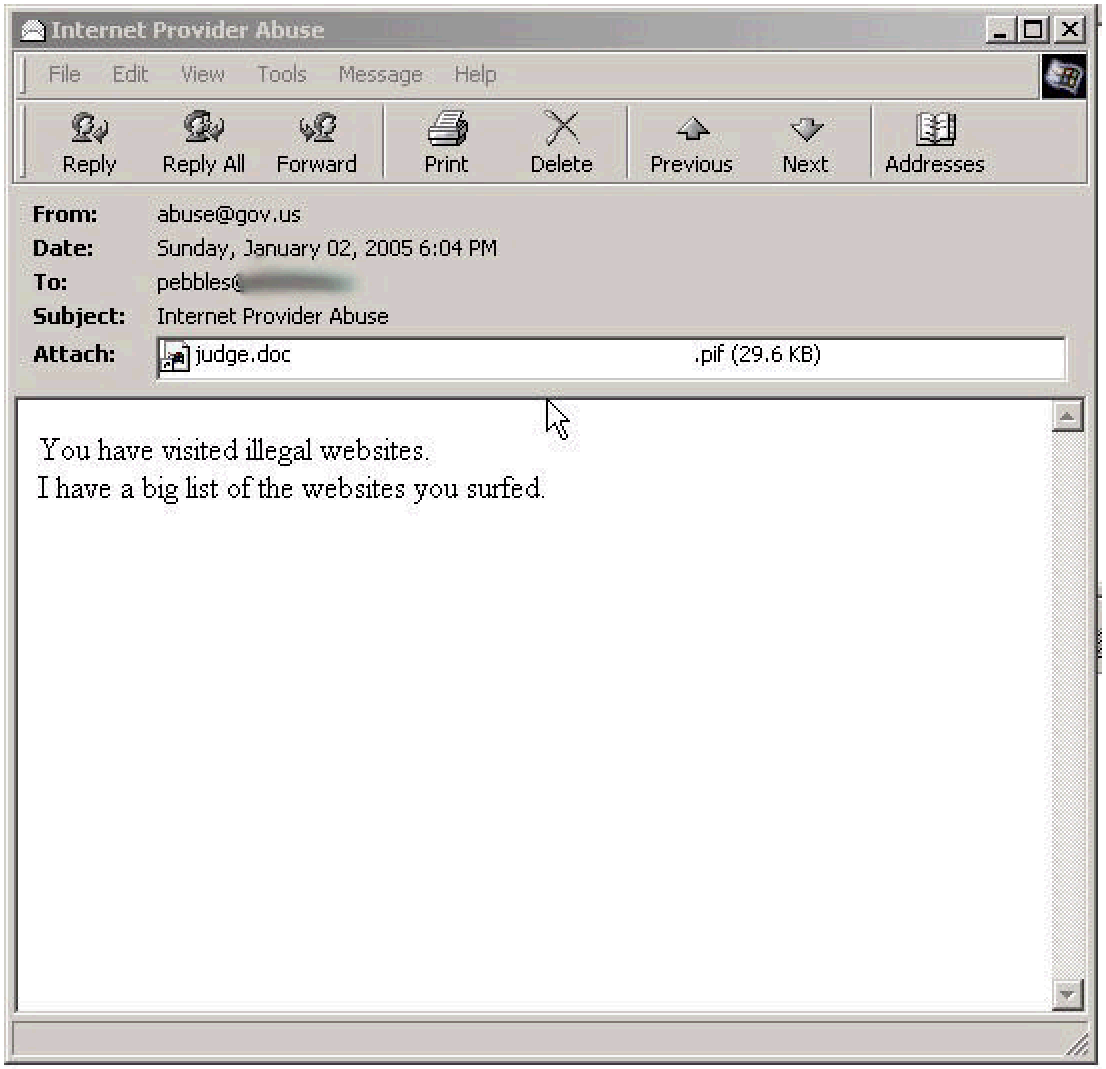
Next Netsky.p must next create an email message with an attachment. The email message itself is built from a variety of different content source phrases. The complete list can be found in Appendix C. In addition to a large set of random subject, body and attachment messages Netsky.p uses approximately 30 template messages with between 2 and 10 different choices for subject body and text. Most of these 30 combinations do a better job at building a convincing email than the more random ones pulled out of a longer list of possible subject/body/address combinations.

The following is one example out of the 30 possible emails[20].

Subject:
Illegal Website
- OR -
Internet Provider Abuse
Body:
- ONE OF THE FOLLOWING -
I noticed that you have visited illegal websites.
See the name in the list!
You have visited illegal websites.
I have a big list of the websites you surfed.

Attachment:
- ONE OF THE FOLLOWING -
list.<ext>
abuselist.<ext>
judge.<ext>
readme.<ext>
details.<ext>

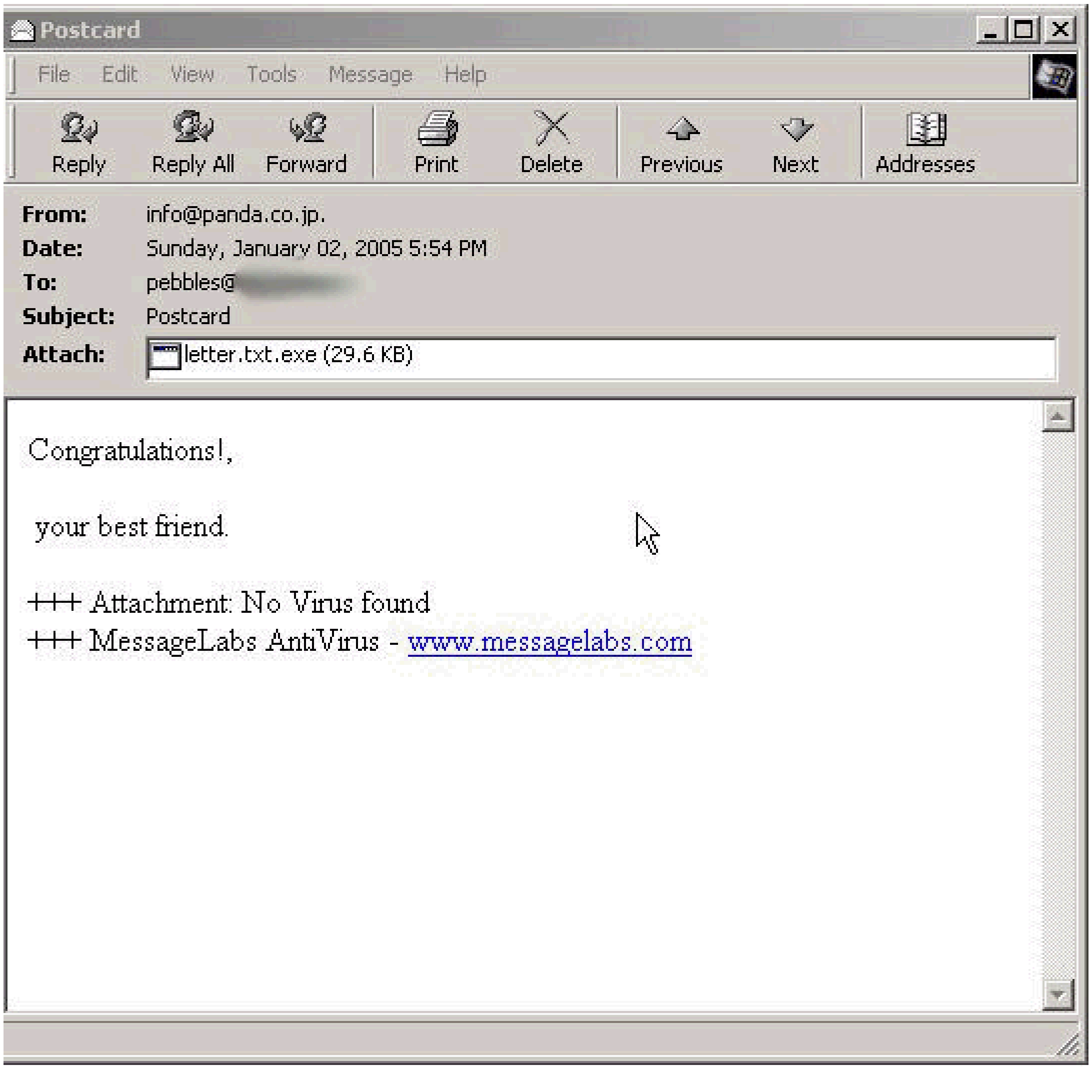
Figure 10 and 11 show these results from the lab:



From: abuse@gov.us
Date: Sunday, January 02, 2005 6:04 PM
To: pebbles@[redacted]
Subject: Internet Provider Abuse
Attach: judge.doc   .pif (29.6 KB)

You have visited illegal websites.
I have a big list of the websites you surfed.

Figure 10 – Example of Infected email



From: info@panda.co.jp.
Date: Sunday, January 02, 2005 5:54 PM
To: pebbles@[redacted]
Subject: Postcard
Attach: letter.txt.exe (29.6 KB)

Congratulations!,

your best friend.

+++ Attachment: No Virus found
+++ MessageLabs AntiVirus - www.messagelabs.com

Figure 11 – Example of Infected email

The worm also will not send to certain email addresses or address with certain substrings. This is an attempt to avoid detection for a longer period of time. If the TO: contains any of the following it will not be sent[9] :

- @microsof
- @antivi
- @symantec
- @spam
- @avp
- @f-secur
- @bitdefender
- @norman
- @mcafee
- @kaspersky

- @f-pro
- @norton
- @fbi
- abuse@
- @messagel
- @skynet
- @pandasof
- @freeav
- @sophos
- ntivir
- @viruslis
- noreply@
- spam@
- reports@

To summarize appendix C, Netsky.p tries to look like spoof email from:

- An authority accusing you of bad behavior
- A co-worker or boss sending you a document of some sort
- A friend sending something of use or entertaining

In the lab, the authority emails were all picked from what would appear to be an internal list as though a government agency were accusing you of something or returned email undeliverable messages. Netsky.p also randomly attaches footers to the email making it appear that the email passed through a virus scanner at some point and came out OK. Mass mailers have grown in sophistication in terms of the messages they sent and the Netsky.p author seems to have learned well. This tactic, of course, is designed to make it harder to warn users of specific subject matter in the text to identify the worm. It also obfuscates against writing spam or other email filters in the beginning stages of infection (i.e.
before anti-virus vendors come up with signatures).

Next Netsky.p will find an SMTP server to send the message to. Two methods are used[20]. First Netsky.p tries to find the SMTP server for the domain that the addressee is in. For example, if the MAIL TO: is bob@bobspropane.net then the worm will use the DNSAPI.DLL to look up the MX record for the domain bobspropane.net. Then it will try to connect directly to that SMTP server and send the message. DNSAPI.DLL will only work in Windows 2000, Windows XP and .NET Servers. Older systems do not have this library.

If that method fails, Netsky.p uses IPHLPAPI.DLL library to find the local domain DNS server. This is the DNS server the infected machine points to for its DNS queries. It then queries the mail exchanger that is in the domain for the local host. In other words, regardless of the MAIL FROM: or RCPT: TO the local domain is discovered. Then the SMTP server in Netsky.p sends the email to the MX server for the local domain. This works on all versions of Windows and provides a fallback if port 25 is blocked across an ISP or other transport.

```
No.  SRC  Dest  Prot   Info

15   .44  .10   DNS    Standard query MX sexnet.com
16   .44  .10   DNS    Standard query MX sexnet.com
17   .44  .10   DNS    Standard query MX techsmith.com
18   .44  .10   DNS    Standard query MX techsmith.com
19   .44  .10   DNS    Standard query MX dcelab.com.com
20   .10  .44   DNS    Standard query response MX 10 barney.dcelab.com
...
110  .44  .10   DNS    Standard query MX panda.co.jp
111  .44  .10   DNS    Standard query MX panda.co.jp
112  .44  .10   DNS    Standard query MX cisco.com
113  .44  .10   DNS    Standard query MX cisco.com
114  .44  .10   DNS    Standard query MX dcelab.com.com
115  .10  .44   DNS    Standard query response MX 10 barney.dcelab.com
```

Figure 12 – DNS Lookup Traffic Capture

Figure 12 shows a Windows 2000 virtual machine infected with Netsky.p try to look up remote MX servers. Since this lab only resolves a set number of hosts in the domain dcelab.com the lookup fail. After a few tries it falls back to the 2nd method and successfully queries the DNS server for its own MX server.

Signatures of the Attack

The various anti-virus writers have signatures available for the Netsky.p worm. For obvious reasons they do not make the details of the signature they use public. Many of the IDS vendors and deep packet firewall inspections check for signatures to the Netsky.p worm also. For users of IDS systems and or email filters that you may want to tune, you can look for attachment types ,exe, .scr, .pif and .zip. It can be argued that people have little business emailing attachments of the first three types. The .zip file will unfortunately probably cause many false positives on most systems.

Blocking outgoing port 25 traffic at the firewall, and logging the IP address of clients that try to break that rule is one network based method of finding infected hosts. On an IDS system you may also look at DNS queries for MX records of both foreign and local domains. Properly configured email clients won't look for these. These should only come from SMTP servers.

Described above, in the workings of the worm are a variety of traces that the worm leaves on a computer. For traces of the attack, first, check the registry, run regedit and look for the string "FVProtect.exe". As noted above, this is how Netsky.p persists in the infected computer. You may also check the Windows directory for the various files listed above also. If you fear a file system sharing point is infected look for any of the files in appendix B. Netsky.p copies all the file names listed into a directory it targets as a share. These details are listed above in the text describing the worms' actions. You may also look for the running process FVProtect.exe, this will show up after a reboot of the system. Before a reboot, the process could be named one of many different names of the attachment from the infected email. It would be very hard to detect at this stage.

Stages of the Attack Process

Some aspects of the mass mailer worm do not fit into the classic attack target scenario. Described in the following sections are the ways the attack does and does not fit into these categories

Reconnaissance

The basic architecture and structure of the Netsky.p and mass mailers in general make reconnaissance a different thought process for the incident handler. What traditional reconnaissance there is, is built into the code itself. This is described in the describing the attack section above. This is where the worm gathers email addresses from the various files on the hard drive of an infected computer. The author plays a statistical game here. Not all systems or people need the vulnerability, just a small portion. The second reconnaissance built into the code is shown in figure 12. Again not classical reconnaissance, this is the DNS lookup of SMTP servers to send the worm on to.

Again, the worm author assumes a percentage of systems are vulnerable to the both the social engineering and the technical vulnerability exploited. Windows users running Outlook or Outlook Express make up a huge percentage of the people connected to the Internet. However, some basic reconnaissance and homework could make a mass mailer take off more quickly. By sending out the first wave of infected emails that are crafted carefully to insure response and sending it to a large number of people spread across multiple domains it could take off quicker. A smart method here might be to find domains that contain unsophisticated users from large domains, add in emails you can find from usenet and other public forums of special interest and look to smaller companies that are not likely to have security awareness programs. Examples here include any ISP's running cable modems, Yahoo, Hotmail and others for large amounts of home users email addresses.

Also the peer to peer sharing networks, can be of use. In fact, the first waves of attacks may be sent out via scripts or other injection methods that the worm code doesn't even use.

Scanning

Netsky.p does not perform any network scanning to look for vulnerabilities. It does look for SMTP servers on the internet to connect to and send the malicious email. As discussed above, the SMTP server installed on the target computer looks for specific servers to connect to.

It could be theoretically possible to find machines that are vulnerable to attack due to other virus or worm infections on the internet. Termed zombies or botnets, a scan for common open ports due to previous hijacks may give a leg up for email worms to start their infection. This type of scanning could use any one the common tools built to look for open ports such as NMAP. As this is not done we will leave it for later xxx.

Exploiting The System

Since this was a lab exercise the exploited target was a contrived machine setup. The computer bambam with the email user bambam@dcelab.com was used as the attacking computer. An email with the virus payload was hand built. Since this was a lab simulation extra work at building the social engineering was not done. A random set of the Netsky.p messages was used. The target machine, pebbles with email user pebbles@dcelab.com was setup. A vulnerable version of Microsoft Internet Explorer 5.5 was loaded along with Outlook Express 5.0. Figure 8 shows the screen capture of the email. A file named test.txt was seeded in the directory structure to test the email address harvesting. Netsky.p did use the address in this file eventually. Also pebbles email address book was lightly populated. In addition, preparation was made to test the file sharing part of the exploit. Directories C:\FTP, C:\DONKEY and C:\TEST\KAZAA were created on target pebbles. Task manager was started and checked as to running processes. All user programs were closed.

The crafted email was then sent from bambam to pebbles. The attachment name for the exploit was message.scr. Outlook Express was used to preview this email. No other user interaction occurred and it was immediately obvious that the attack worked from the disc activity on pebbles. The 80 files listed in appendix C were copied to the directory C:\FTP, none of the other directory in the common file share name list were touched. However, the files were also copied to the following directories when the target was Windows XP.

C:\Documents and Settings\All Users\Application Data\Microsoft\Network\Downloader\
C:\WINDOWS\Downloaded Program Files\
C:\WINDOWS\PCHEALTH\UploadLB\
C:\WINDOWS\PCHEALTH\UploadLB\Binaries\
C:\WINDOWS\PCHEALTH\UploadLB\Config\

The other files and registry settings listed in the description of the exploit section were observed. In the setup of target pebbles, none of the registry settings that are commonly deleted existed prior to or after the infection. The task manager window was checked and the process message.scr was running as depicted in figure 7.

Network Diagram

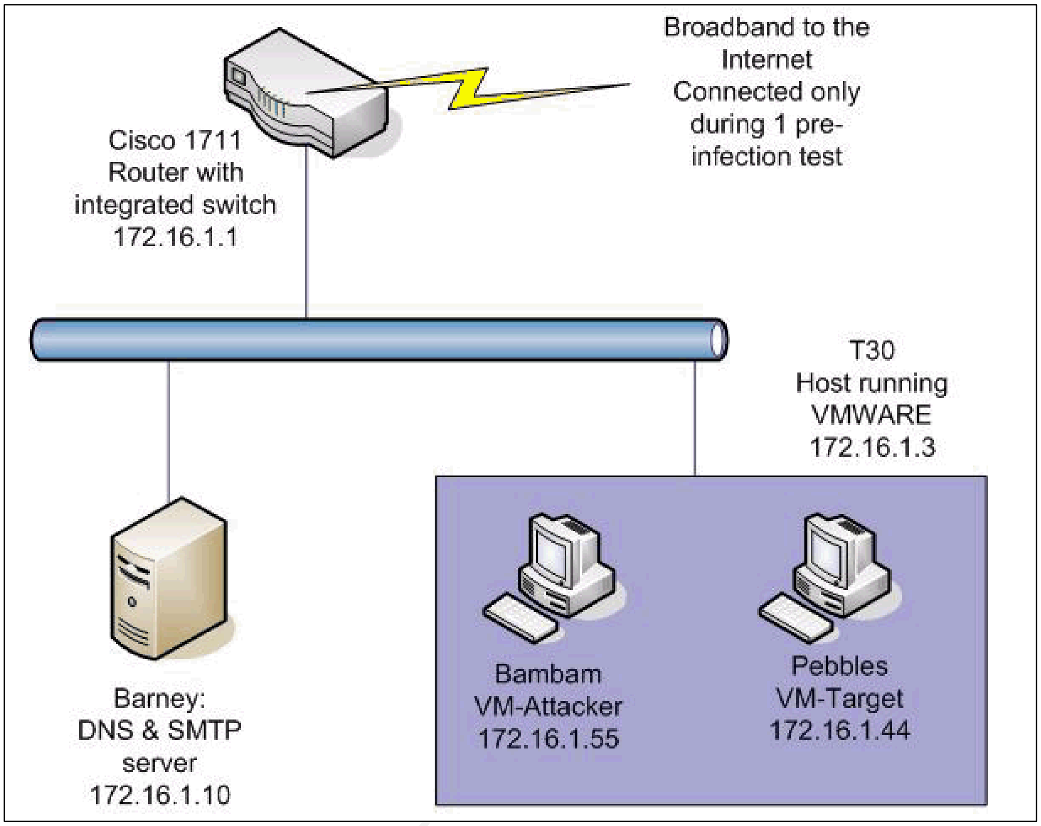

Broadband to the Internet Connected only during 1 pre-infection test
Cisco 1711 Router with integrated switch 172.16.1.1
T30 Host running VMWARE 172.16.1.3
Barney: DNS & SMTP server 172.16.1.10
Bambam VM-Attacker 172.16.1.55
Pebbles VM-Target 172.16.1.44

Figure 13 – Lab Network Diagram

HOST	IP ADDRESS	DESCRIPTION
Router/Switch	172.16.1.1	Gateway to Internet
T30	172.16.1.3	VMWARE host
Barney	172.16.1.10	DNS and SMTP Server
Pebbles	172.16.1.44	VM-Target
Bambam	172.16.1.55	VM-Attacker

Figure 14 – IP Table For Lab

Network Components:

Router: The Cisco Systems router with integrated 4 port switch is a 1711 series router running IOS 12.3(8)T advanced security feature set. This router functionality used in only one phase of the lab. For the better part of the experiment the router served only as a switch to connect the hosts together.

Host Barney: This host is an IBM small server, Intel Pentium class computer with Windows Advanced Server operating system loaded on it. Service Pack 4 is installed. This computer acts as a DNS and SMTP server. One email user was configured on this server. Internet Explorer 6 SP1 and Outlook Express 6 were installed and an email user dce@dcelab.com was configured. The Netsky.p worm did not infect this machine, but the email user did receive infected emails.

Host T30: This host is an IBM ThinkPad T30 with Windows Professional operating system loaded on it. Service Pack 4 is installed One email user was configured on this server. Internet Explorer 6 SP1 and Outlook Express 6 were installed and an email user lme@dcelab.com was configured. The Netsky.p worm did not infect this machine, but the email user did receive infected emails. WMware Workstation 4.5.1 build 7568 is installed and used to run 2 virtual hosts on 1 physical machine. Ethereal is also installed to capture network traffic to and from the attacker, target and the rest of the network

Host Bambam: This host is a virtual machine running Window XP Service Pack 1. After being used as the initial attacker for target pebbles it was also used to test the resilience of version 6.0 Service Pack 1 if Internet Explorer and Outlook Express 6.0. It eventually was infected via the social engineering simulation.

Host Pebbles: This host is a virtual machine running Window 2000 Professional Service Pack 4. A vulnerable version of Microsoft Internet Explorer 5.5 was loaded along with Outlook Express 5.0. Pebbles was the primary target of the attack, most of the information in the Ethereal captures and image grabs came from pebbles.

Keeping Access

Netsky.p retains access, or persists, through the use of the registry key HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
.
This key is set to the value Norton Antivirus AV"="%Windir%\FVProtect.exe. This registry entry is show in figure 8. This key's purpose is to allow for the automatic running of programs on Windows boot. The particular HLEY_LOCAL_MACHINE means to run the program regardless of the user logged in. While this is an effective way of a worm "keeping access" or running through multiple reboots of the computer, it is easy to detect through a text search of the registry. Most AV and other host based security products look for changes in this key also. Figure 15 shows the Windows task manager after a reboot the FVProtect .exe process is shown running.

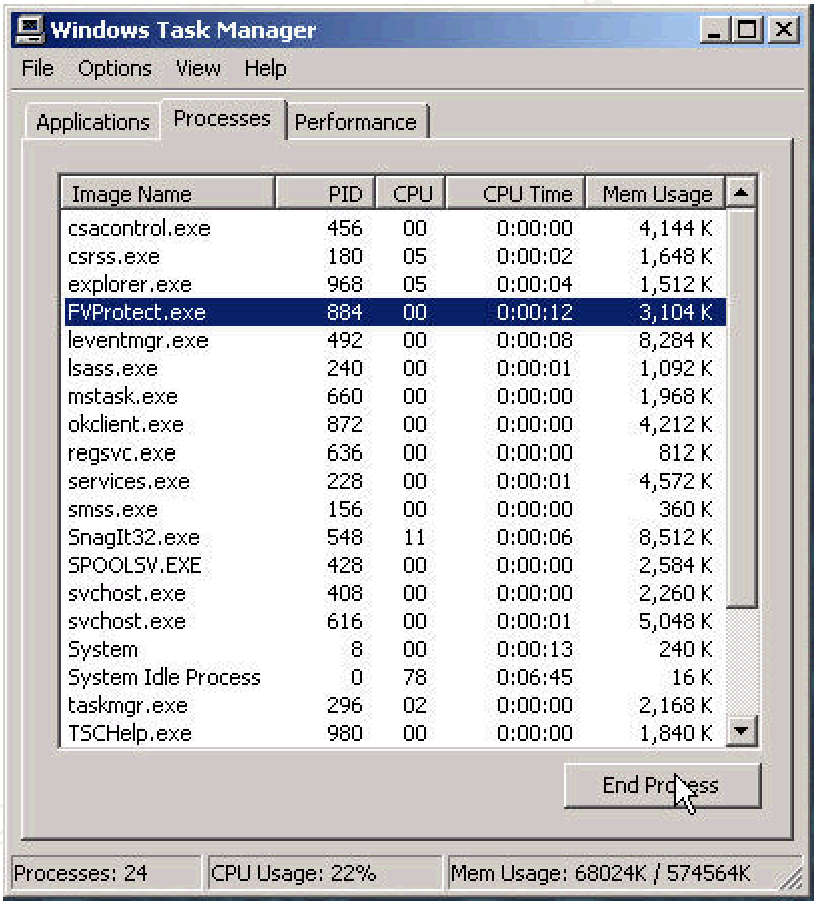


Image Name	PID	CPU	CPU Time	Mem Usage
csacontrol.exe	456	00	0:00:00	4,144 K
csrss.exe	180	05	0:00:02	1,648 K
explorer.exe	968	05	0:00:04	1,512 K
FVProtect.exe	884	00	0:00:12	3,104 K
leventmgr.exe	492	00	0:00:08	8,284 K
lsass.exe	240	00	0:00:01	1,092 K
mstask.exe	660	00	0:00:00	1,968 K
okclient.exe	872	00	0:00:00	4,212 K
regsvc.exe	636	00	0:00:00	812 K
services.exe	228	00	0:00:01	4,572 K
smss.exe	156	00	0:00:00	360 K
SnagIt32.exe	548	11	0:00:06	8,512 K
SPOOLSV.EXE	428	00	0:00:00	2,584 K
svchost.exe	408	00	0:00:00	2,260 K
svchost.exe	616	00	0:00:01	5,048 K
System	8	00	0:00:13	240 K
System Idle Process	0	78	0:06:45	16 K
taskmgr.exe	296	02	0:00:00	2,168 K
TSCHelp.exe	980	00	0:00:00	1,840 K

Processes: 24   CPU Usage: 22%   Mem Usage: 68024K / 574564K

Figure 15 - FVProtect.exe Process in Task Manager

Covering Tracks

Netsky.p does not cover its tracks well after infecting a computer. The files copied, the registry key changes and the use of the SMTP server are all easy to find. The names of the files and the names of the registry keys do not change, nor does the method of zipping or encoding those files. The author does place the files in the Windows directory, but this is a pretty common tactic, and done as much to assure the directory is there as any other. The worm does do a decent job of hiding the form of the social engineering aspect of the email. In other words relatively large pool of subject, body and attachment name make it harder to find from simple string searches on email or TCP streams.

The Incident Handling Process

The following narrative closely follows a real incident at a real company. The names have been changed as well as the vector for attack. The company here is fictional, but represents the real concerns of an actual small, high value, intellectual property firm.

Polymer Controls Inc. is a small niche business providing consulting services to the rubber industry in and around Akron, Ohio. While small, Polymer Controls Inc. (PCI) generates, controls, or owns intellectual property worth millions of dollars in the form of recipes for making a variety of rubber or other polymer compounds. In order to test and certify these recipes they run a small scale manufacturing plant that also serves as a lab for compound development and process tuning.

PCI has only one network administrator, Sandy Johnson, who is tasked with keeping the computers running and doing the important job of running the software that holds the compound recipes and process notes along with all OSHA guidelines for chemicals used. Sandy's father, Bruce runs the company and they both take security pretty seriously as their clients intellectual property also resides on their computer system. Unfortunately neither of them is extremely technical.

Preparation

Approximately 2 years ago PCI contracted with a local firm to help with some basic, standard security practices to follow. After a series of interviews a short to the point security policy was put into place. PCI had, for years put employees through rigorous background searches and made them sign binding contracts for non-disclosure of trade secrets. So when the security policy came out it followed some of PCIs guiding principles in terms of a strict and simple, to the point policy. PCI viewed prevention and preparation as an important security step, they did not mind paying for the services that coincided with this attitude. Among other things, PCI:

- Hosted their small, relatively static web site, at a hosting service company instead of on site. This lowered the amount of incoming traffic and the complexity of DMZ, incoming traffic rules and monitoring. No e-business was done on the web, nor did vendors or customers need intranet access.

- Used the ISP email server for their email, for many of the same reasons sighted for web hosting.

- Purchased a small Cisco PIX firewall to protect the corporate network. The setup allowed outgoing traffic on a few well defined ports such as WEB, EMAIL, FTP etc. No incoming traffic was allowed except as responses to established outgoing connections.

- Laptops were given to highly trusted employees only (family members)

- An acceptable use form was written, discussed at length with new employees and signed by employer and employee. Consequences for infractions of this and the corporate confidentiality policy were clearly communicated and harsh.

- A small unused office was converted to a computer room, only the owner and IT staff had a key. A plan was put in place to change from antiquated coaxial cable carrying Ethernet traffic to unshielded twisted pair cabling. While not viewed as a security problem per se, it was causing reliability issues.

- Anti-virus software was installed on all computers. While mentioned in the recommendations updates were not particularly taken seriously.

No incident handling team was created. The same company that aided PCI in authoring their policy was used for other IT related needs. If there was an incident handling team or methodology it was “call the computer guys”

Identification

On Monday, October 11th 2004, Sandy is approached by Allan M. The head engineer at the firm. Alan tells Sandy of an email he has received from her with some rather harsh language concerning a document he wrote and an attachment. When he tried to open the attachment nothing happened. Sandy promises to come take a look and promptly forgets the conversation. She usually waits until the employees complain about something twice before taking action. It sounds like one of the other engineers is playing tricks on Alan, who is a brilliant chemist but still figuring out email.

The next day, just before 10:00AM Sandy gets a call from a customers purchasing agent who has become friends with Sandy over the last couple of years. “Sandy, you sent me a strange email yesterday and our virus scanning software is barking about it”, says Myra. Myra works at an important customer so Sandy calls the Computer Guys (CG’s) right away.

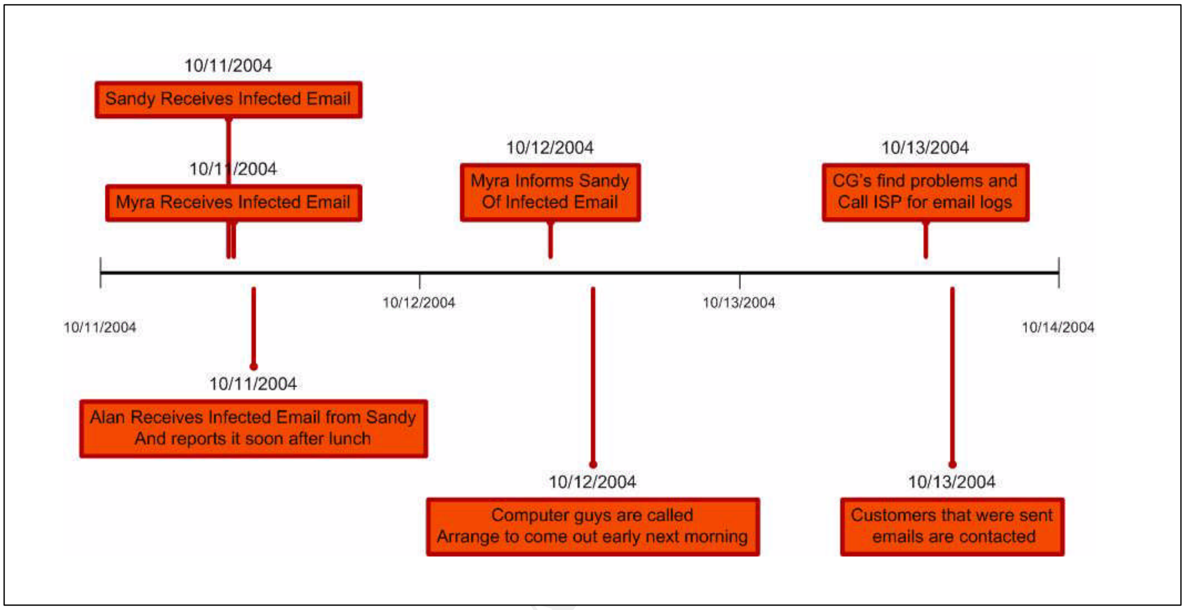

10/11/2004
Sandy Receives Infected Email
10/11/2004
Myra Receives Infected Email
10/12/2004
Myra Informs Sandy Of Infected Email
10/13/2004
CG’s find problems and Call ISP for email logs
10/11/2004
10/12/2004
10/13/2004
10/14/2004
10/11/2004
Alan Receives Infected Email from Sandy And reports it soon after lunch
10/12/2004
Computer guys are called Arrange to come out early next morning
10/13/2004
Customers that were sent emails are contacted

Figure 16 – Incident Timeline

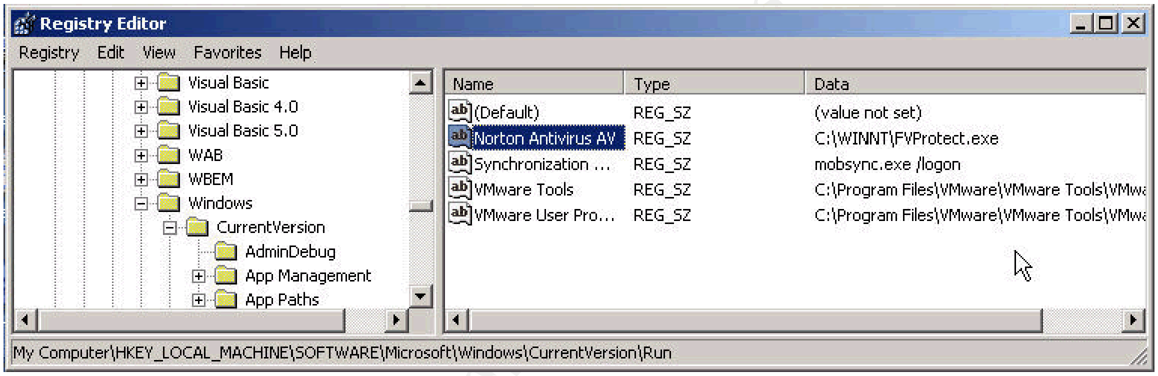
Sandy and Bruce do not have any formal definition of when an incident is in progress. But they do know that in their business when they retain trade secrets for their customers, any idea that they are less than 100% secure is bad for business. So when problems crop up internally it’s not a big deal. Customers getting email viruses from PCI are definitely incidents. Figure 16 shows the timeline for this incident.

The Computer Guys come out early the next day. The CG’s are very responsive to this particular customer as they are trying their hand at application development and this is their first custom application. When they find out “the big problem” is email, they breathe a sigh of relief. Jack looks over Sandy’s machine and can’t find anything in the outbox, Jim goes and checks with Alan.

After a little work to no avail Jim suggests they should try a test restore of the compound recipe to the “hot standby server”. Jack argues it’s a waste of time, but it is the lifeblood of the company and the application they have so much at stake on. The hot standby server is actually the largest user computer in the company, Bruce’s. They disconnect Bruce’s computer from the network and restore last nights backup database to it, the application runs locally on all machines, so they just need to change some data sources.

While Jim is doing this, Jack continues his investigation and begins to suspect a malicious email worm. The CG's use Symantec's Norton Anti-Virus and recommends it to all their customers. Unfortunately, PCI uses Trend Micro, an AV program Jack is not as comfortable with. Jack sees that PCI is woefully behind the times on their anti-virus update. So he works on updating it. After the update, Jack sees that Sandy's machine is infected with Netsky.p. He check's out the Symantec website and downloads the FXNetsky.exe file and starts looking for the clues listed on the Symantec website.

Jack is still a bit leery of automated cleanup tools, so he likes to check out things manually, it makes him feel more certain that the software is doing its job. He also likes to point out to his customers that he knows the inside of the OS. First Jim looks to Sandy's machine and the registry. If the machine is infected there should be an entry in the registry that doesn't belong there.

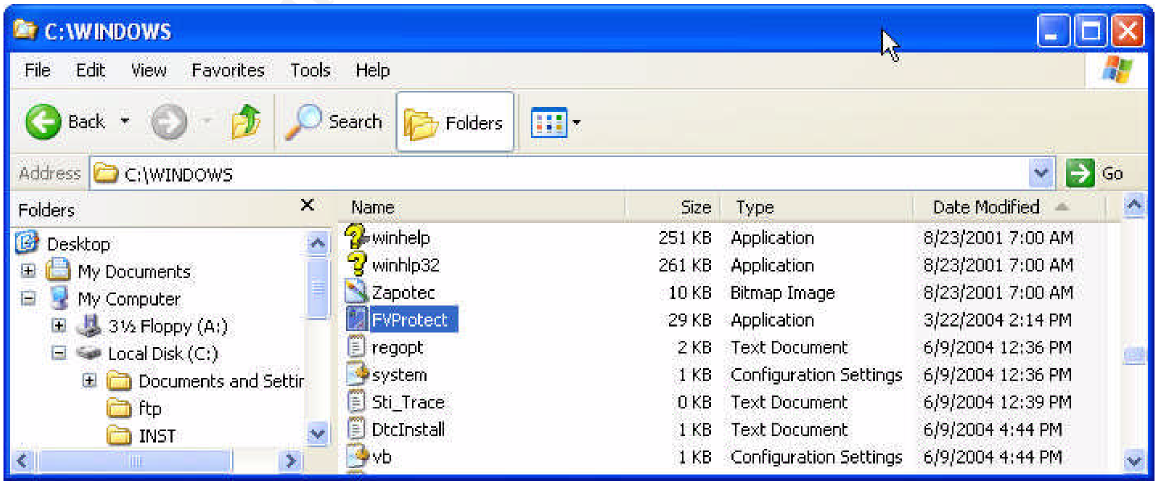


Name	Type	Data
(Default)	REG_SZ	(value not set)
Norton Antivirus AV	REG_SZ	C:\WINNT\FVProtect.exe
Synchronization ...	REG_SZ	mobsync.exe /logon
VMware Tools	REG_SZ	C:\Program Files\VMware\VMware Tools\VMwa
VMware User Pro...	REG_SZ	C:\Program Files\VMware\VMware Tools\VMwa

My Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

Figure 17 – FVProtect.exe in the Registry

Figure 17 shows the FVProtect.exe entry in the registry key as described earlier. This confirms what the AV software says. Next Jack looks to the WINDOWS directory to see if he can find the files listed in the bulletin.

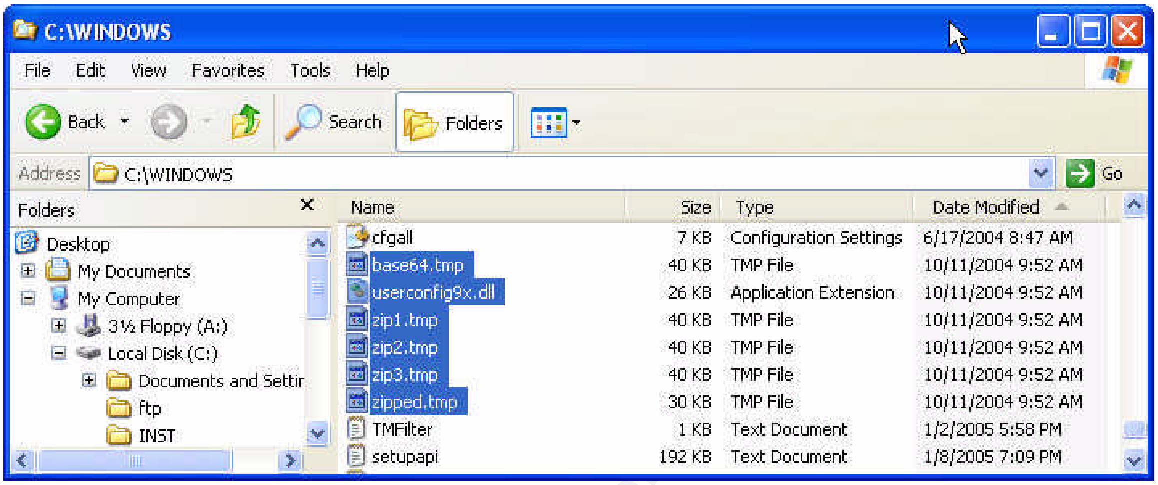


Address: C:\WINDOWS

Name	Size	Type	Date Modified
winhelp	251 KB	Application	8/23/2001 7:00 AM
winhlp32	261 KB	Application	8/23/2001 7:00 AM
Zapotec	10 KB	Bitmap Image	8/23/2001 7:00 AM
FVProtect	29 KB	Application	3/22/2004 2:14 PM
regopt	2 KB	Text Document	6/9/2004 12:36 PM
system	1 KB	Configuration Settings	6/9/2004 12:36 PM
Sti_Trace	0 KB	Text Document	6/9/2004 12:39 PM
DtcInstall	1 KB	Text Document	6/9/2004 4:44 PM
vb	1 KB	Configuration Settings	6/9/2004 4:44 PM

Figure 18 – FVProtect.exe Files and Date

He finds FVProtect.exe and is shocked to see the creation date is 3/22/2004 (figure 18). At first he is amazed they have had the worm a long time. After further search he sees the Netsky.p created files in the WINDOWS directory are more recent.



Name	Size	Type	Date Modified
cfgall	7 KB	Configuration Settings	6/17/2004 8:47 AM
base64.tmp	40 KB	TMP File	10/11/2004 9:52 AM
userconfig9x.dll	26 KB	Application Extension	10/11/2004 9:52 AM
zip1.tmp	40 KB	TMP File	10/11/2004 9:52 AM
zip2.tmp	40 KB	TMP File	10/11/2004 9:52 AM
zip3.tmp	40 KB	TMP File	10/11/2004 9:52 AM
zipped.tmp	30 KB	TMP File	10/11/2004 9:52 AM
TMFilter	1 KB	Text Document	1/2/2005 5:58 PM
setupapi	192 KB	Text Document	1/8/2005 7:09 PM

Figure 19 – Dropped Files and their Dates

The worm must retain its own creation date but it looks like the files it makes have the creation date from Monday, thinks Jack. This should help us on pinpointing the infection time. Figure 19 shows this as almost 10:00AM Monday. Later on, checking Alan's computer will confirm Jack's suspicion. Jack then looks for the process FVprotect.exe in the task manager and it is also there (figure 20).

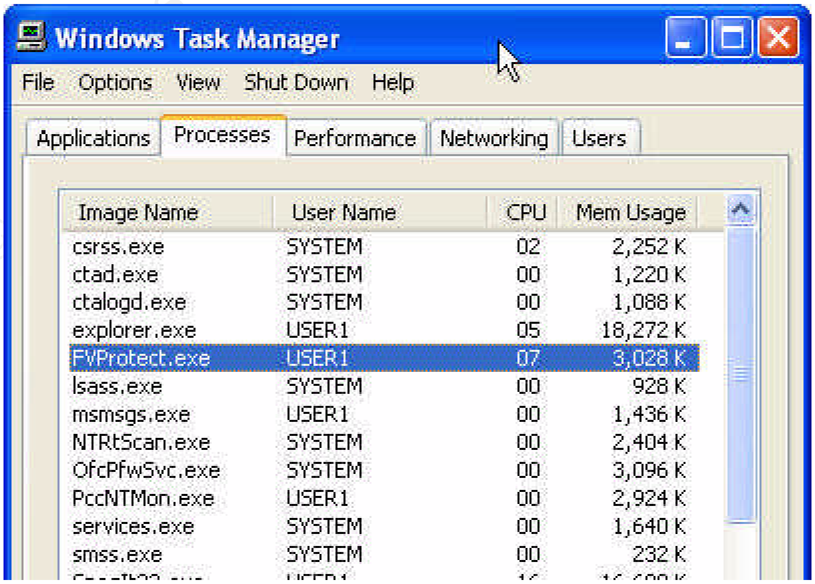

Windows Task Manager
File Options View Shut Down Help
Applications Processes Performance Networking Users

Image Name	User Name	CPU	Mem Usage
csrss.exe	SYSTEM	02	2,252 K
ctad.exe	SYSTEM	00	1,220 K
ctalogd.exe	SYSTEM	00	1,088 K
explorer.exe	USER1	05	18,272 K
FVProtect.exe	USER1	07	3,028 K
lsass.exe	SYSTEM	00	928 K
msmsgs.exe	USER1	00	1,436 K
NTRtScan.exe	SYSTEM	00	2,404 K
OfcPfwSvc.exe	SYSTEM	00	3,096 K
PccNTMon.exe	USER1	00	2,924 K
services.exe	SYSTEM	00	1,640 K
smss.exe	SYSTEM	00	232 K

Figure 20 – FVProtect.exe Process

During this Identification phase Jim and Jack made 2 rather large errors. First, they did nothing to maintain a chain of custody. While in the long run mass mailer worm writers are rarely caught, or have small businesses chasing them to ground, the CG's didn't know this. A good incident handler needs to follow the rules at all times. The GC's should have made a low level copy of the hard drive, taken copious notes in a ledger book with numbered pages and signed and locked away all the evidence together. A written procedure that CG's use at all the times would be advisable If the incident did end up involving the authorities then a chain of custody tag should be signed when handing everything over to them.

Second, in his haste to insure the custom application they wrote was unharmed, Jim may have brought down the companies ability to operate for some time. No one knew if Bruce's computer was infected or with what. Although the tape backup of the data would be difficult to corrupt through a virus the restore of the data to the hot spare was unnecessary and an added risk.

Containment

Once Jack has satisfied himself this is an email worm he has a short meeting with Bruce, Sandy and Jim. While they could change a few rules on the firewall Jack suggests the radical step of disconnecting from the internet. The only real loss is email and that is the compromised application. No other vital business process is harmed. Since this is a small company, the team can also direct the employees on the correct course of action in a quick company meeting. Bruce asks everyone to leave their computer on and not use them for anything except the ERP program from the CG's. Shut all other programs down.

Next Jim and Jack walk around and kill the FVProtect.exe process from every machine they can find it on. They use the simple expedient of CTRL-ALT –DEL and killing the process out of Task Manager by clicking end process as depicted in figure 21. This is done to try and contain any further spreading of the worm in case they missed something. They note each machine that has the running process FVProtect.exe on each machine as they go. They further ask that the users do not reboot their computer unless they tell Jim or Jack.

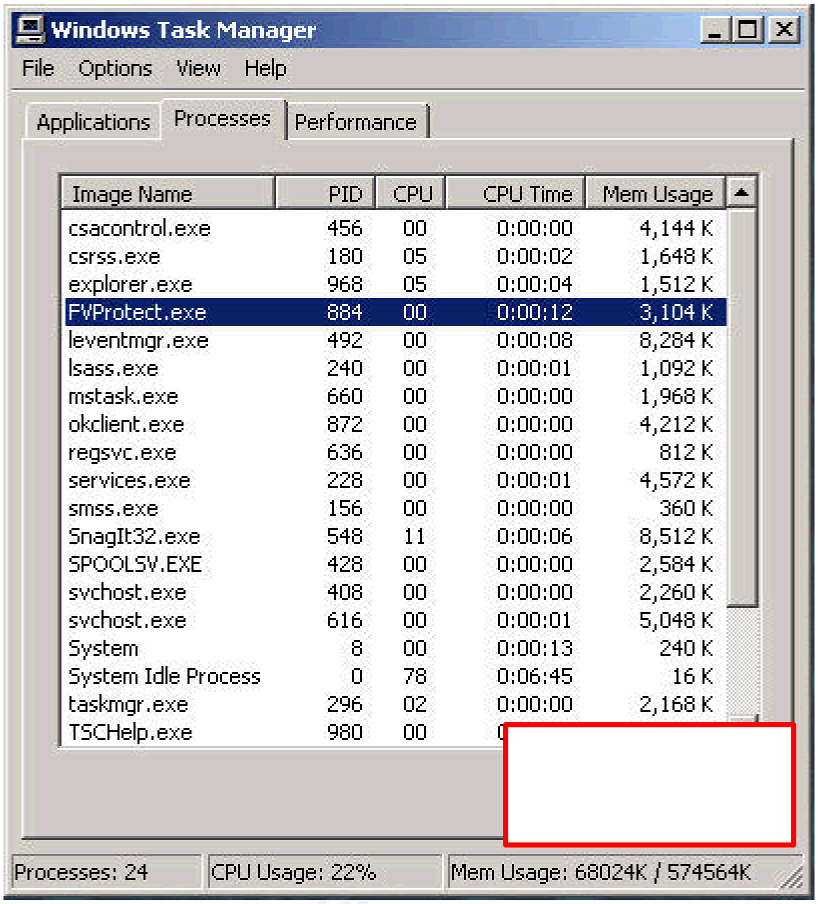


Image Name	PID	CPU	CPU Time	Mem Usage
csacontrol.exe	456	00	0:00:00	4,144 K
csrss.exe	180	05	0:00:02	1,648 K
explorer.exe	968	05	0:00:04	1,512 K
FVProtect.exe	884	00	0:00:12	3,104 K
leventmgr.exe	492	00	0:00:08	8,284 K
lsass.exe	240	00	0:00:01	1,092 K
mstask.exe	660	00	0:00:00	1,968 K
okclient.exe	872	00	0:00:00	4,212 K
regsvc.exe	636	00	0:00:00	812 K
services.exe	228	00	0:00:01	4,572 K
smss.exe	156	00	0:00:00	360 K
SnagIt32.exe	548	11	0:00:06	8,512 K
SPOOLSV.EXE	428	00	0:00:00	2,584 K
svchost.exe	408	00	0:00:00	2,260 K
svchost.exe	616	00	0:00:01	5,048 K
System	8	00	0:00:13	240 K
System Idle Process	0	78	0:06:45	16 K
taskmgr.exe	296	02	0:00:00	2,168 K
TSCHelp.exe	980	00	0	

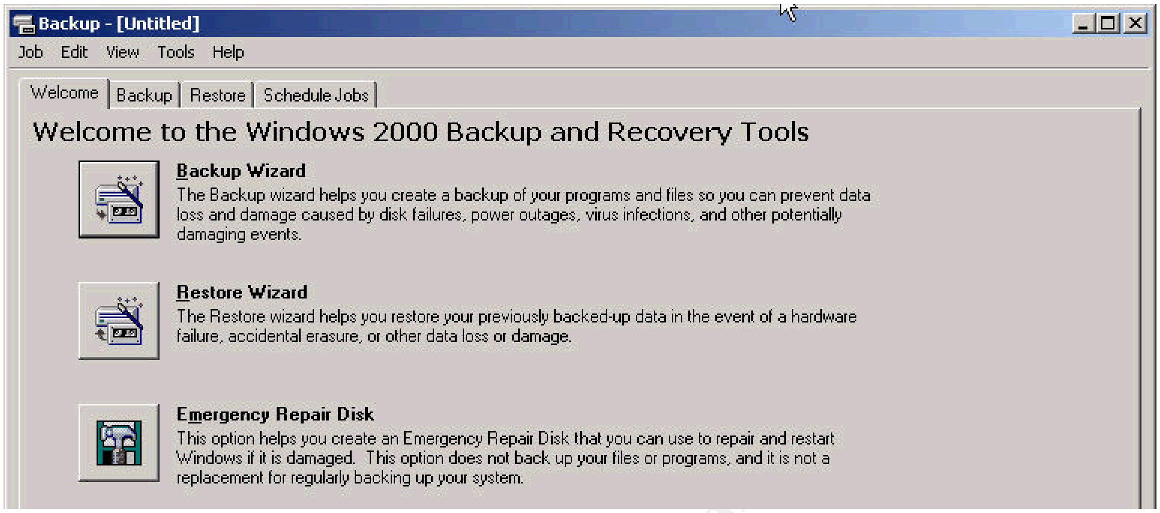
Processes: 24 | CPU Usage: 22% | Mem Usage: 68024K / 574564K

Figure 21 – End Process

The file server that stores the database for PCI's business critical application was found to be uninfected by Netsky.p. However, as a safety measure the ad hoc incident handling team of Bruce, Sandy Jim and Jack decided to do a mid-day backup on the ERP system and then run the eradication software supplied by Symantec to be safe.

PCI does not have huge amounts of data to back up, the data they produce, while high value, is not much more complicated than a cake recipe. A complex formula might be 10 printed pages long if they include the OSHA MSDS safety sheets. PCI therefore uses the ntbackup command scheduled to run every night and backup the entire database and ERP program. This backs up to a CD-ROM burner mounted in the file server and appearing as drive Z:. Every Friday they also copy any business correspondence to the file server and back that directory up also. PCI is somewhat old fashioned in that they generate a number of personal letters and keep a hard copy with each customer file.

For the backup, Jack logged into the file server and at the DOS command line typed ntbackup. The screen in figure 22 appeared.



Welcome to the Windows 2000 Backup and Recovery Tools

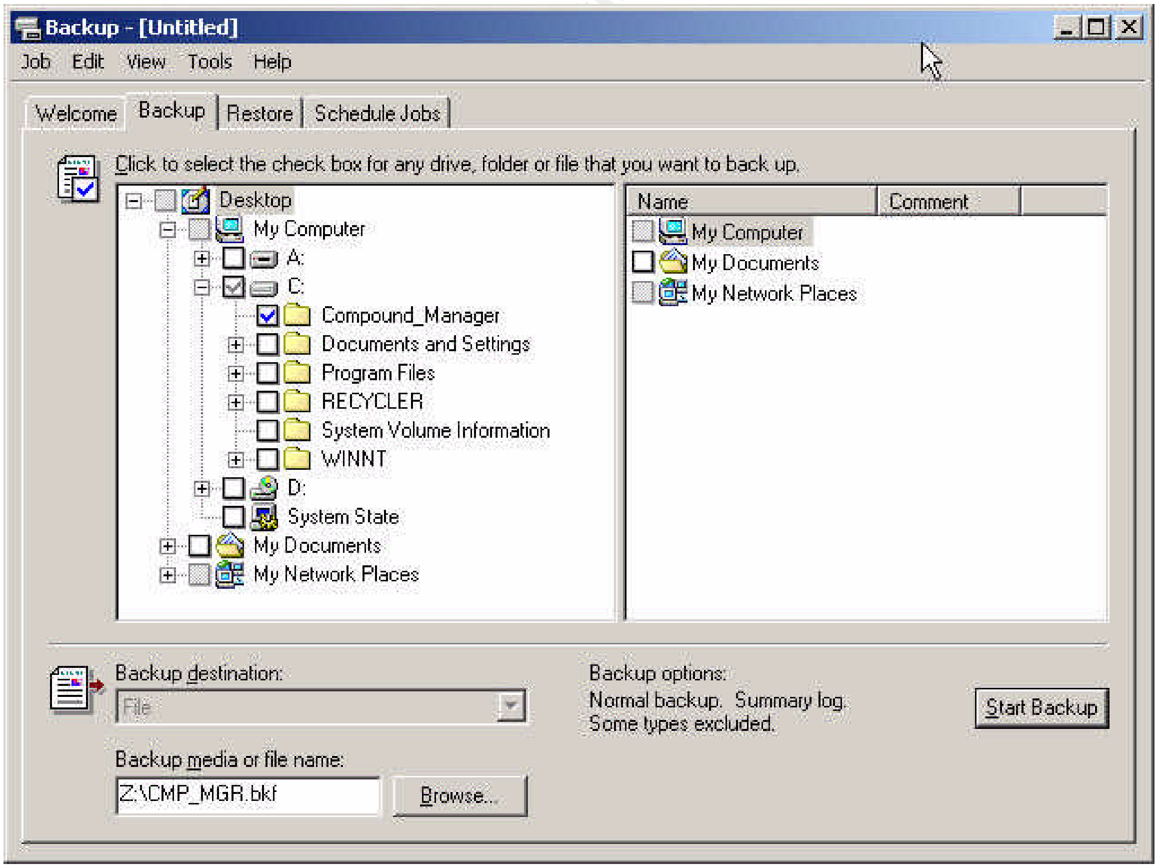
Backup Wizard
The Backup wizard helps you create a backup of your programs and files so you can prevent data loss and damage caused by disk failures, power outages, virus infections, and other potentially damaging events.

Restore Wizard
The Restore wizard helps you restore your previously backed-up data in the event of a hardware failure, accidental erasure, or other data loss or damage.

Emergency Repair Disk
This option helps you create an Emergency Repair Disk that you can use to repair and restart Windows if it is damaged. This option does not back up your files or programs, and it is not a replacement for regularly backing up your system.

Figure 22 – Ntbackup Opening Screen

Jack clicks the backup tab to check which files he wants backed up.



Click to select the check box for any drive, folder or file that you want to back up.

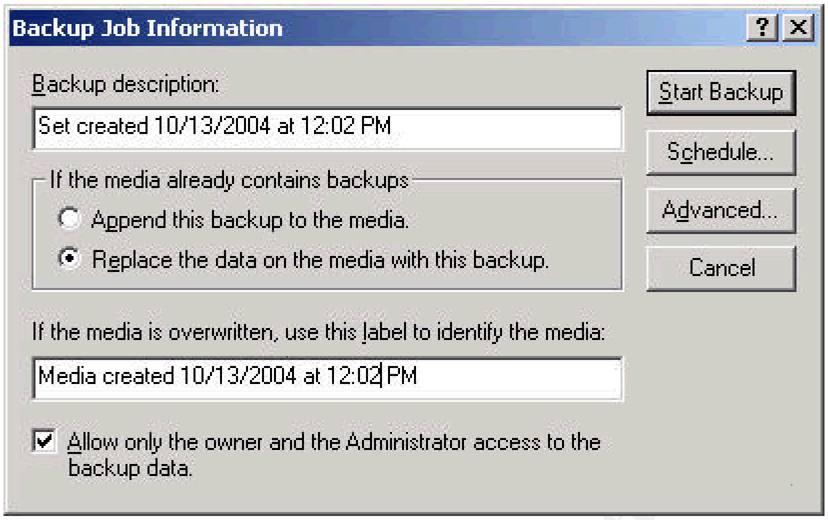
- ☑ C: → ☑ Compound_Manager

Backup destination: File

Backup media or file name: Z:\CMP_MGR.bkf

Backup options:
Normal backup. Summary log.
Some types excluded.

Figure 23 – Choosing backup files



Backup Job Information

Backup description:

Set created 10/13/2004 at 12:02 PM

If the media already contains backups

○ Append this backup to the media.

● Replace the data on the media with this backup.

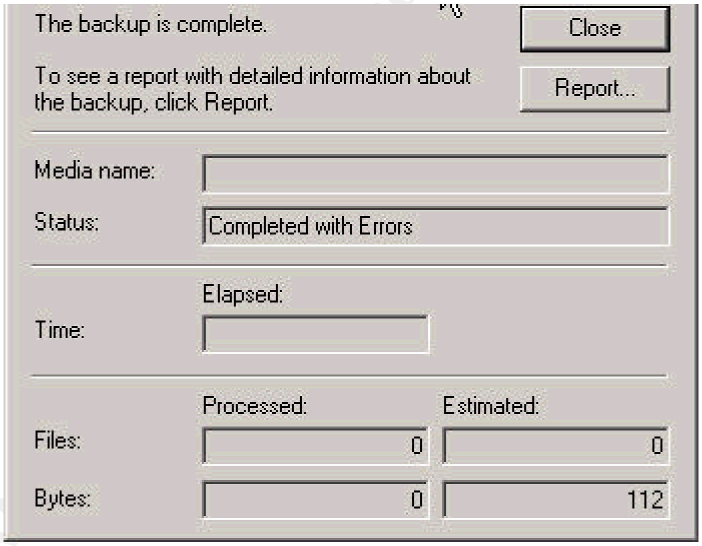
If the media is overwritten, use this label to identify the media:

Media created 10/13/2004 at 12:02 PM

☑ Allow only the owner and the Administrator access to the backup data.

Figure 24 – Setting up the job

The header information is added to the backup job and the job starts.



The backup is complete.

To see a report with detailed information about the backup, click Report.

Media name:

Status: Completed with Errors

Time: Elapsed:

	Processed:	Estimated:
Files:	0	0
Bytes:	0	112

Figure 25 – Backup Errors

The backup had errors, (figure 25) after some checking the team finds a supervisor on the shop floor did not know he should have closed the Compound Manager application. They start through the process and it succeeds the second time.

3. Click the System Restore tab.
4. Check the "Turn off System Restore" or "Turn off System Restore on all drives" check box as shown in figure 27:

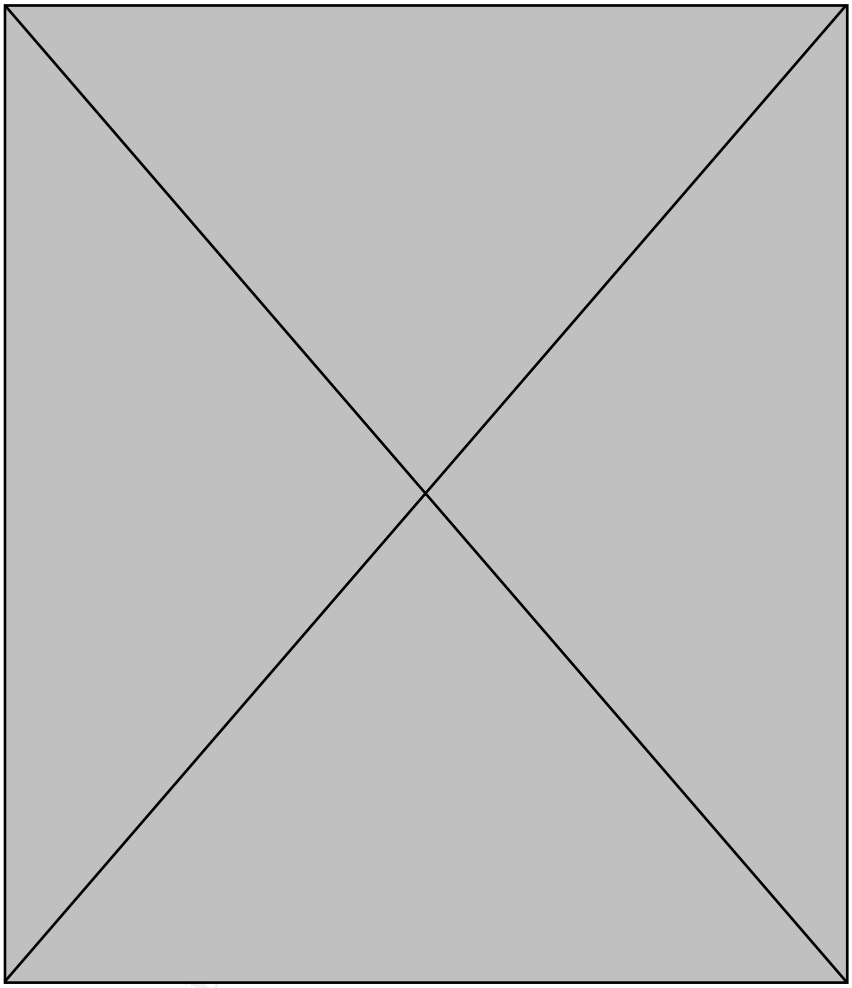
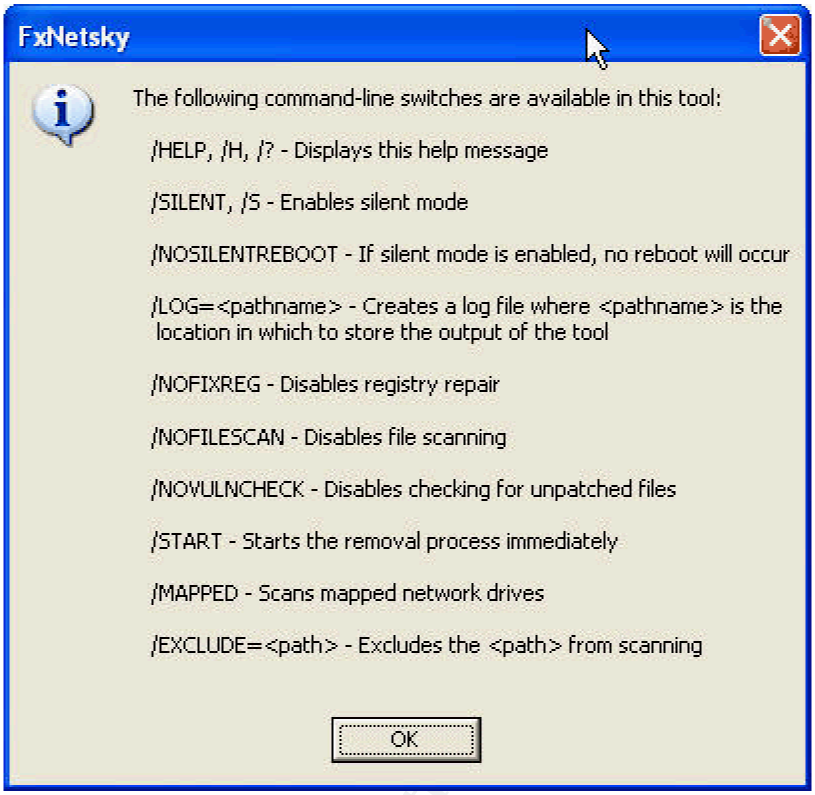


Figure 27 – How to turn off System Restore

It is also highly recommended that you close down all other applications during the disinfection. The cleaner is downloaded in a .zip file format. Jack chose to unzip it to an empty directory called C:\INST. The name of the executable for the cleaner is FxNetsky.exe. Typing in **FxNetsky /?** On the command line pops up the following help screen in figure 28.



The following command-line switches are available in this tool:

/HELP, /H, /? - Displays this help message

/SILENT, /S - Enables silent mode

/NOSILENTREBOOT - If silent mode is enabled, no reboot will occur

/LOG=<pathname> - Creates a log file where <pathname> is the location in which to store the output of the tool

/NOFIXREG - Disables registry repair

/NOFILESCAN - Disables file scanning

/NOVULNCHECK - Disables checking for unpatched files

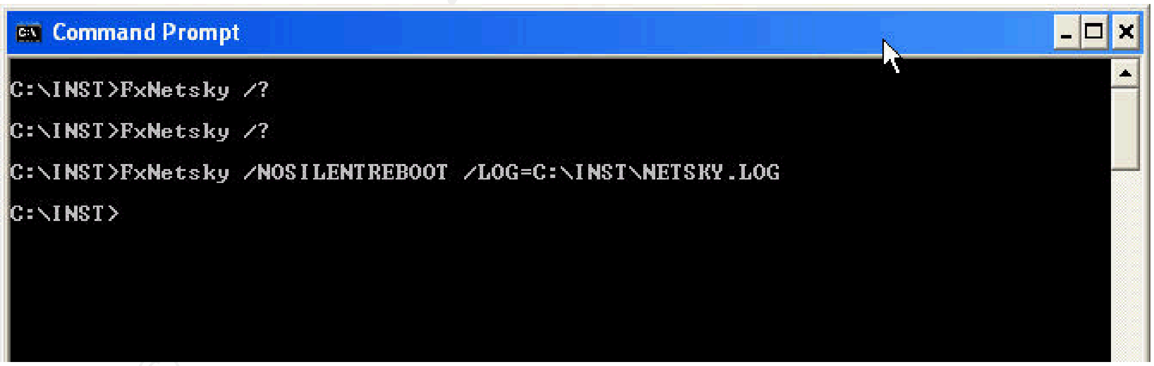
/START - Starts the removal process immediately

/MAPPED - Scans mapped network drives

/EXCLUDE=<path> - Excludes the <path> from scanning

Figure 28 – FxNetsky Help Screen

The following screen capture shows the command line switches Jack chose.



```
C:\INST>FxNetsky /?

C:\INST>FxNetsky /?

C:\INST>FxNetsky /NOSILENTREBOOT /LOG=C:\INST\NETSKY.LOG

C:\INST>
```

Figure 29