STEVEN ALEXANDER

finding malware on compromised Windows machines

Steven is a programmer for Merced College. He manages the college's intrusion detection system.

■ *alexander.steven@sbcglobal.net*

This article discusses possible responses to suspicious activity on Windows machines. It surveys several free tools that are useful in documenting the current state of a system and in detecting and analyzing suspicious processes or open network ports.

Sometimes it is obvious that a machine has been compromised (e.g., when a Web site is defaced). Other times, suspicious behavior is detected that warrants further investigation. This activity could be the result of a break-in, a virus or worm, spyware, or something more benign. If the suspicious behavior is the result of a break-in, law enforcement may need to be contacted (depending on your organization's policy). For this reason, it is important that volatile information be saved and that every step you take is documented. The tools discussed in this article can be used to document the current state of a system that has been compromised or to investigate a system that is behaving suspiciously.

I was recently called upon to investigate two Windows machines that had been exhibiting suspicious behavior. The first thing I did was to create a CD with several tools I thought would be useful in analyzing the system and documenting the state of the system. As it turns out, the machines in question had not been compromised by a human intruder but by two different worms; these two systems were not properly patched.

Starting Out

The first thing I do when investigating any system (Windows, UNIX, or otherwise) is glean lists of the users who are logged on, running processes, and network connections. System administrators should use these tools on freshly installed systems and on production systems to determine what a normal system looks like. If you don't know what should be on your system, it is very difficult to figure out what shouldn't be on your system.

After gathering basic information, I check the Windows registry and the Windows startup folders to see what is starting up with the system. Sometimes it is easy to determine what each of the programs that are scheduled to start automatically actually is. Unfortunately, a lot of legitimate software vendors like to do asinine things such as stick an executable with a weird name into C:\windows or C:\windows\system32. This can make it difficult to determine whether a program is legitimate. The best thing to do is to Google for the filename and see what turns up.

The Reg tool, described below, can be used to dump the appropriate keys from the registry. Any undesired entries can be removed using Regedit or, on Windows XP, Msconfig. I strongly suggest using Msconfig on XP systems, since it enables you to uncheck an entry but, if you wish, restore the entry later. If you are using Regedit, back up the registry before deleting anything. Also, rather than deleting any programs referenced by these entries, move and/or rename them to avoid losing something you need. Administrators should try to become familiar with the software that should be starting automatically on their servers and workstations. Again, it's hard to determine what is anomalous if you don't know what normal is.

My approach when investigating a system is to look for anything that does not belong. This includes spyware, worms, back doors, etc. A centrally managed antivirus program is a good way to detect a lot of malware as soon as it enters the system, but it won't detect everything. Netcat, for instance, is not malware and won't be detected by an antivirus program, but it can be used to bind cmd.exe to a port for use as a back door.

This article is limited to discussing the tools and procedures that can help determine whether a system has been compromised. Responding to a compromise is an even larger issue (one I hope to cover in another article). Still, a response policy—vetted by upper management—must be in place before an incident occurs. Some important points that must be decided include who is responsible for the technical response, how evidence will be handled, who decides whether to contact law enforcement, and whether it matters if the intrusion occurred from within or from outside your organization.

It is also helpful to find out from your local law enforcement agency what they want you to do in the event of a break-in. At what point do they want to be contacted? How should you proceed? How should you preserve possible evidence? Whom, in law enforcement, should you contact? If you simply call your local police department after a break-in, the response will probably be from a uniformed officer who has no training in these matters. Often, you will want to contact the detective or group that handles computer evidence or computer crimes.

See references [1–4] for useful reading on incident response and forensics.

In the following sections, I give brief descriptions of several tools, all freely available, that I've found useful. With the exception of Microsoft's Reg tool, they are available from SysInternals [5].

The Tools

WINDOWS: REG

Reg is a Windows utility that can be used to extract data from the Windows registry. A large number of malware programs add an entry in the registry so that they will be started automatically if the computer reboots. Under most circumstances (if you're not currently installing something), the RunOnce and RunOnceEx keys should be empty. Track down any programs listed under the Run keys and make sure they represent legitimate software.

```
reg query "HKEY_CURRENT_USER\Software\Microsoft\Windows\Currentversion\Run" /s
reg query "HK_LOCAL_MACHINE\Software\Microsoft\Windows\Currentversion\RunOnceEx" /s
reg query "HK_LOCAL_MACHINE\Software\Microsoft\Windows\Currentversion\RunOnce" /s
reg query "HK_LOCAL_MACHINE\Software\Microsoft\Windows\Currentversion\Run" /s
```

SYSINTERNALS: PSLIST

PsList is similar in function to the UNIX ps command. It displays a list of the running processes on the system, including the process ID, priority, and number of threads. With the -m option, PsList also displays memory-usage information, including the working set size. With the -t option, the running processes are displayed in a process tree instead of a flat list.

```
PsList 1.26 - Process Information Lister
Copyright (C) 1999-2004 Mark Russinovich
Sysinternals - www.sysinternals.com

Process information for C37163ALEXANDER:
```

Name	Pid	Pri	Thc	Hnd	Priv	CPU Time	Elapsed Time
Idle	0	0	1	0	0	85:28:28.343	0:00:00.000
System	4	8	80	292	0	0:04:00.593	0:00:00.000
smss	452	11	3	19	164	0:00:00.046	143:29:16.906
csrss	524	13	11	407	1648	0:02:12.312	143:29:12.890
winlogon	548	13	19	581	7728	0:00:09.875	143:29:11.250
services	592	9	16	286	4244	0:00:15.750	143:29:09.187
lsass	604	9	17	377	2468	0:00:06.515	143:29:09.046
svchost	796	8	5	133	1360	0:00:00.109	143:29:06.093

SYSINTERNALS: HANDLE

Handle is a command-line utility that shows the handles open by every process on the system. When used without options, Handle only displays open file handles. When used with the -a option, Handle displays open handles to all objects, including files, registry keys, processes, ports, and semaphores. I suggest dumping all open handles to one file and dumping just open file handles to a second file for convenience. Handle is very useful for figuring out what a program is actually doing.

```
Handle v2.2
Copyright (C) 1997-2004 Mark Russinovich
Sysinternals - www.sysinternals.com
------------------------------------------------------------
smss.exe pid: 452 NT AUTHORITY\SYSTEM
     8: File         C:\WINDOWS
    1c: File         c:\WINDOWS\system32
------------------------------------------------------------
csrss.exe pid  524 NT AUTHORITY\SYSTEM
     c: File         C:\WINDOWS\system32
    38: section        \NLS\NlsSectionUnicode
    40: Section        \NLS\NlsSectionLocale
    44: Section        \NLS\NlsSectionCType
    48: Section        \NLS\NlsSectionSortkey
    4c: Section        \NLS\NlsSectionSortTbls
   2e0: Section        \BaseNameObjects\ShimSharedMemory
   564: File         c:\WINDOWS\system32\ega.cpi
```

SYSINTERNALS: SYSLISTDLLS

ListDLLs is a command-line utility that displays the DLL files loaded by each process running on the system. The utility shows the full path of each DLL. I find that the path information is particularly helpful because it helps me to identify what application or service a process is associated with.

```
ListDLLs v2.25 - DLL lister for Win9x/NT
Copyright (c) 1997-2004 Mark Russinovich
Sysinternals - www.sysinternals.com
------------------------------------------------------------
```

```
System pic: 4
Command line: <no command line>
------------------------------------------------------------------------------

smss.exe pid: 452
Command line: \SystemRoot\System32\smss.exe

Base        Size      Version         Path
0x48580000  0xf000                    \SystemRoot\System32\smss.exe
0x7c900000  0xb0000   5.01.2600.2180  C:\WINDOWS\system32\ntdll.dll
------------------------------------------------------------------------------

winlogon.exe pid: A548
Command line: winlogon.exe

Base        Size      Version         Path
0x01000000  0x80000                   \??\C:\WINDOWS\system32\winlogon.exe
0x7c900000  0xb0000   5.01.2600.2180   c:\WINDOWS\system32\ntdll.dll
0x7c800000  0xf4000   5.01.2600.2180   c:\WINDOWS\system32\kernel32.dll
0x77dd0000  0x9b000   5.01.2600.2180  c:\WINDOWS\system32\ADVAPI32.dll
0x77e70000  0x91000   5.01.2600.2180   c:\WINDOWS\system32\RPCRT4.dll
0x776c0000  0x11000   5.01.2600.2180   c:\WINDOWS\system32\AUTHZ.dll
0x77c10000  0x58000   7.00.2600.2180   c:\WINDOWS\system32\msvcrt.dll
```

SYSINTERNALS: TCPVCON

Tcpvcon displays a list of all established TCP connections along with their owning process. When used with the -a option, it will display all connection endpoints (TCP and UDP), established or not.

```
[TCP] C:\Program Files\Netscape\Netscape\Netscp.exe
    PID        2676
    State:     ESTABLISHED
    Local:     c37163alexanders:3283
    Remote:    c37163alexanders:3284
[TCP] C:\Program Files\Netscape\Netscape\Netscp.exe
    PID:       2676
    State:     ESTABLISHED
    Local:     c37163alexanders:3284
    Remote:    c37163alexanders:3283
```

PSLOGLIST

Psloglist displays the contents of the event logs. By default, Psloglist dumps the system log, but it can be used to dump the other logs by running psloglist logname. If used to dump the Directory Service or File Replication Service logs, quote the name on the command line. The program is also capable of dumping records from after a specified date by running it with the -a option, e.g., psloglist security -a 01/01/05.

It is essential to remember that your system needs to be configured to log important events in the first place. The standard events that are logged by Windows simply do not provide you with enough detail about a break-in.

At a minimum, enable auditing for policy change, privilege use, and logon events in the Local Security Policy under Administrative Tools in the Control Panel (this can also be accessed using mmc). You may also wish to audit access to important or confidential data (this can be configured by right-clicking on any file or folder, choosing Properties, and clicking Advanced under the Security tab).

```
[005] Security
    Type:      AUDIT SUCCESS
    Computer:  SEGFAULT
    Time:      12/30/2003 3:34:52 PM   ID:      643
    User:      MCCEDU\alexander.s
```

```
Domain Policy Changed: Password Policy modified
    Domain Name:        SEGFAULT
    Domain ID:          %{S-1-5-21-123456789-123456789-123456789}
    Caller User Name:   alexander.s
    Caller Domain:      MCCEDU
    Caller Logon ID:    (0x0,0xC6EF)
    Privileges:         -
```

Conclusion

Your best tool is wetware. The more familiar you are with the normal processes and services running on your systems, the easier it will be to detect anything out of place. Also, the Event Logs are of little use unless you enable additional auditing.

Some intrusions are hard to detect. If your firewall logs or IDS indicates that there may be a problem with a machine and your initial investigation turns up nothing, you may wish to crank up the logging for a spell and see what turns up.

I do not recommend blindly searching the file system unless you are willing to image the drives on the system (or, less preferably, make a tape backup) before searching, since you might inadvertently modify the file access times. I know of one utility that is supposed to be able to search for files accessed within a given time range without updating the attributes, but that utility fails to find many files.

Not searching the file system has the drawback that you may miss the signs of a break-in. Nevertheless, I would rather increase logging and wait things out. If the attacker is discreet, you may miss whatever he has left behind anyway.

Sometimes it is known that a system has been compromised, but it is not known whether the attack is the work of self-propagating malware or a human intruder. If you don't find any evidence of a virus or worm that accounts for previously observed events, it may be appropriate to treat it as a human attack until proven otherwise. At this point, you should take the affected system offline and image the drives. If you do find malware on the system, make sure that particular malware accounts for any observed events and is not independent of or a cover for a human attack.

POSTSCRIPT

Since this article was written, SysInternals has released a new tool, Rootkit Revealer. It looks promising: check it out.

REFERENCES

[1] Jamie Morris, "Forensics on the Windows Platform, Part One," http://www.securityfocus.com/infocus/1661.

[2] Jamie Morris, "Forensics on the Windows Platform, Part Two," http://www.securityfocus.com/infocus/1665.

[3] H. Carvey, "Win2K First Responder's Guide," http://www.securityfocus.com/infocus/1624.

[4] Chris Prosise, Kevin Mandia, and Matt Pepe, "Incident Response and Computer Forensics," McGraw-Hill Osborne Media, 2003.

[5] Mark Russinovich and Bryce Cogswell, SysInternals, http://www.sysinternals.com.