

Detecting and Removing Trojans and Malicious Code from Win2K

H. Carvey 2002-09-18

Introduction

The amount of malicious code directed at Windows systems seems to be increasing on a continual curve [1]. [IRC bots](#), [backdoor Trojans](#) and [worms](#) abound. It seems that few Windows systems, particularly Win2K, are immune from infection, regardless of how diligent the user or administrator may be. Many posters to public lists continue to report Code Red and Nimda scans, as well as port scans for popular Trojan applications, on an almost weekly basis.

The flip side of this is that users and administrators are also reporting that their systems have been infected or "hacked", without having solid evidence to support their assumptions. Many times, the reported activity may be, in reality, normal activity of an application on the system.

The purpose of this article is to recommend steps that an administrator can use to determine whether or not a Win2K system has been infected with malicious code or "malware" and, if so, to remove it. This article will specifically address network backdoor Trojans and IRC bots, but the information delivered in this article should assist the reader in a variety of situations.

While it may be easy to recover from some compromises and infections, administrators should keep in mind that the most effective way of recovering from an Administrator-level compromise (referred to as "root compromise" in the Linux world) is to simply reinstall the system and install the appropriate patches. However, a "root cause" investigation should still be conducted to determine the infection vector, or the route used by the attacker, so that the issue may be addressed.

First Steps

As the saying goes, an ounce of prevention is worth a pound of cure, particularly when it comes to computer security incidents. By taking steps to prepare for incidents of all types, administrators can prevent or severely hamper the installation and activation of all sorts of malware.

The first step to preventing malware from being installed on a Win2K system is to understand how the malware gets onto the system in the first place, and what it does once it is activated.

The point at which malware infiltrates a system is referred to as the "infection vector".

Unfortunately, there seems to be an almost unending supply of infection vectors, ranging from age-old misconfiguration issues (i.e., open network shares) to more recent issues such as [Malware combining multiple IE vulnerabilities](#)). Then, of course, there are e-mail attachments and users downloading infected files. Due to the sheer enormity of possibilities, the infection vectors will not be addressed specifically in this article. (However, Paul Schmehl's SecurityFocus article [Infection Vectors: Past, Present, and Future](#) offers a good discussion.)

Once a backdoor Trojan is on the system, for the most part, it usually instigates some sort of action, usually one that allows the attacker to manipulate or control the target system. At a bare minimum, the Trojan or bot needs to be activated, therefore activating will be a process the system. For a process to be running, in most cases, a file (or several files) will need to be copied to the victim system. For the attacker to control the Trojan (and hence, the victim system) a network port will need to be opened. In the case of Trojans, a port will be opened and netstat.exe will report a state of "LISTENING". Most IRC bots will open a client port and connect to a remote IRC server, most generally on port 6667.

Finally, most Trojans and bots will include a mechanism for establishing persistence, so that they will be active across reboots, and logins by different users. Reviews of [anti-virus Web sites](#) clearly demonstrate how this is done. For the most part, Trojans and bots create an entry in a Registry key so that they are restarted whenever the system restarts or the user logs in. For example, [Backdoor.Latinus.B](#) creates an entry in the ubiquitous "Run" key, as does the "russiantopz" IRC bot:

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run
```

[Backdoor.OptixPro.10](#) modifies the Registry key that defines the standard Windows execution of ".exe" files:

```
HKEY_CLASSES_ROOT\exefile\shell\open\command
```

Another persistence mechanism includes creating a shortcut in a Startup directory.

The prevention mechanisms available to administrators are quite simple [2]. While setting up systems from a minimalist perspective, administrators can set access control lists (ACLs) on directories such as `c:\winnt\system32` to prevent users from writing to (i.e., creating or

modifying files) them. The same applies to certain Registry keys, particularly those mentioned above. If the infrastructure and corporate environment allows administrators to restrict a user's ability to install software, then setting the appropriate ACLs will prevent or severely hamper the installation of Trojans and bots.

Tracking Down the Trojan

Many times, the administrator will suspect the presence of a Trojan or bot due to some unusual behavior on the infected system. This may be seen as a slight change in how the system responds, as unusual traffic captured via an intrusion detection system (IDS), or via filtering at a firewall. Other times, the initial discovery may come from an outside source, as the infected system may be used to scan or attack other systems, and the victim of that attack puts forth the effort to contact the administrator.

However the malware is discovered, the follow-up steps taken by the administrator are critical. Public lists abound with posts by administrators who port scan systems to determine open ports, or discover "something unusual" in the Registry of the victim system and simply delete the entry. Both of these actions are, at best, ineffective incident response measures, as they really don't provide the administrator with usable forensic information. Open ports obtained from a port scan usually lead to the administrator searching for executables or services that "officially" use those ports. However, new Trojans are being developed all the time, and while they generally have default ports that they bind to in order to listen for connections, this port assignment is usually configurable by the user.

A more effective means of determining if a Trojan or bot is installed on a system is to gather very granular process information from the system. There is a small set of five tools that can be run to collect this information. Specifically, these tools are:

- [handle.exe](#)
- [listdlls.exe](#)
- [pslist.exe](#)
- [fport.exe](#)
- [netstat.exe](#)

The first three tools are available from [SysInternals](#), fport.exe is available from [FoundStone](#), and netstat.exe is a tool native to Win2K systems.

(Author's Note: There has been a good deal of discussion about rootkits and Trojaned binaries on Win2K systems. While this is something that happens on Linux systems quite a bit, similar activity has not been reported to a similar extent on Win2K systems. This is not to say that such things don't exist. [Greg Hoggund's NTRootKit](#) is an example of such a thing for NT and 2K systems. However, as of the publication date of this article, there have been no reports of extensive use of the rootkit. If there is a concern about Trojaned binaries, specifically netstat.exe, then copy the executable from a "clean" system when creating an incident response CD.)

The listed tools, along with a batch file that launches them, can be copied to a diskette in the following manner:

```
@echo off
@echo Running listdlls...
listdlls > a:\listdlls.log
@echo Running handle...
handle > a:\handle.log
@echo Running pslist...
pslist > a:\pslist.log
@echo Running fport...
fport > a:\fport.log
@echo Running netstat...
netstat -a > a:\netstat.log
@echo Done.
```

The batch file runs each of the tools in turn, redirecting the output of the files to log files on the diskette. The administrator can run the batch file, or have a [first responder](#) run it. Other methods of obtaining the information include using [psexec.exe](#) from SysInternals to run the tools remotely via an Administrator connection to the victim system.

Running the preceding batch file (or executing the commands by hand) will result in five log files being created. Next, the investigator will need to parse through these files, correlating data on a per-process basis, and then analyze that data. The available information will include such things as the process name and identifier (PID), the path to the executable image for the process, the security context that the process is running in, the command line used to launch the process, modules and handles used by the process, and so on. This is a lot of information to sort through, the analysis of which is prone to errors. A much faster way of correlating the data

into a single, easy to read view is to use a tool like [procdmp.pl](#) (which the author created). This script parses through the five files and correlates information about each process. An example HTML output file is available [here](#).

Notice in the example HTML output file how the information for each process is listed in a tabular format. Each process is listed with the PID, command line, context, open filehandles, and open ports, if any exist. The third process should be of particular interest, with a PID of 1148. At first glance, a process called "inetinfo" might seem to be relatively harmless. In fact, this process is most often associated with the IIS Web server. However, a look at the command line will show that this particular process is probably a copy of [Netcat](#), running on port 8080. The "open ports" section of the output not only shows that port 8080 is open in "LISTENING" mode, but also that there is a remote system connected to the system.

As a point of interest, another interesting process listed is PID 776. This process has the command line shown with a background color of red, indicating that the process was launched from an [NTFS alternate data stream](#). This may be another indicator of malicious code, because alternate data streams are not normally used.

There is also a standalone executable version of [procdmp.pl](#) that provides a GUI for selecting the files to be parsed. This utility is available from the same Web site as [procdmp.pl](#).

These five utilities will provide a comprehensive snapshot of the running processes on the system. Correlating the data using tools such as [procdmp.pl](#) makes the data easier to read, and is less prone to misinterpretation. By using these tools, investigators can quickly retrieve and analyze data, and make accurate decisions regarding follow-up steps in their investigations.

Removing the Trojan

After collecting and analyzing process information, the investigator may realize that a Trojan or IRC bot has been installed on the system. There may be an unusual process running. The path or command line (see above) for the process may be suspicious, or an unusually named process may not only have a port open (found via [fport.exe](#)), but also have a connection to a remote system (determined via [netstat.exe](#)). In one recent discussion on a public list, the IRC bot in question used the name "taskmgr.exe" to "hide" the mIRC client. In the case of the "russiantopz" IRC bot, the mIRC client was renamed to "statistics.exe". A search of Internet sources showed that the name was associated with a valid database product. While Internet

searches may be useful in determining whether a process is "normal" (or "valid") or not, they are not always conclusive. One place to search for information on Win2K-specific processes is the [Microsoft Advanced Search](#).

From the process information, the investigator will have path and file name information he can use to track down the files in question. The process information will also show how to open file handles and modules (DLLs) used by the process. The executable image can be copied to a safe location for analysis, and then deleted from the "victim" system. The Registry should be searched for footprints, which can then be removed. If the process information shows that the process was running in an elevated user context (i.e., Administrator, Domain Admin, etc), then additional steps will need to be taken to determine whether files or data were modified or copied. That said, in most cases an Administrator- or System-level compromise will result in a complete re-installation of the operating system and application files from clean media. However, prior to doing so, care should be taken to determine how the system was compromised in the first place. Otherwise, any efforts to return the system to normal operation will be wasted.

If the administrator of the system had enabled Process Tracking via the EventLog, and the events in question had not been overwritten, the investigator would have an additional source of information regarding the suspect process(es). For example, analysis of the "russiantopz" mIRC bot showed that while the mIRC client was running, the process to hide the client window from view on the desktop had been launched and then terminated. Enabling auditing and logging mechanisms in the operating system and applications can provide a wealth of information to the administrator and investigator alike. These mechanisms, when used in conjunction with other mechanisms (i.e., strong passwords, installed patches, adequate ACLs, etc.) can not only serve to prevent installations of Trojans and bots, but will also generate "noise" when there has been an attempt to do so. This information can then be used to determine the likely avenue of attack, or infection vector.

Conclusion

The necessary steps to locate and remove a Trojan or IRC bot from an infected system are relatively simple and effective. With a couple of utilities, and some knowledge of Win2K systems, the malware can be located, removed from the "victim" system for later analysis, and the system restored to use relatively quickly. Instances in which an Administrator-level compromise has been determined (or is strongly suspected), appropriate steps should be taken.

However, by adequately configuring Win2K systems, and keeping a modest incident response toolkit available, incident handlers can respond quickly and effectively.

References

[1] [2001 Malicious Code Trends, 2002 Predictions](#) whitepaper, [iDefense](#)

[2] [Network Trojans: What you really need to know](#), H. Carvey

[Privacy Statement](#)

Copyright 2006, SecurityFocus