

WINDOWS DIGITAL FORENSICS TOOLKIT:
An Analysis of Digital Forensics Tools

Prepared for
Dr. Lorie Liebrock
CS 489: SFS Professional Development
New Mexico Institute of Mining and Technology
Socorro, New Mexico

By
Cynthia Veitch
November 3, 2006

Windows Digital Forensics Toolkit: An Analysis of Digital Forensics Tools

1. Executive Summary

With the increasing population of computer users and the quick advancement of technology, it is necessary that digital forensic investigators be capable of performing analysis on a multitude of systems. The Windows operating system provides a special challenge because of its proprietary nature. Many companies, such as Sysinternals, devote their efforts to building free tools that can be used on Windows systems for performance optimization. Many of these tools can also be used to collect evidence during a digital forensic investigation. This report reviews several of the Sysinternals utilities, in addition to *Registry Mechanic* and *Snort*. These tools are judged based on their digital forensic purpose, any gaps in their capabilities, and their comparison to similarly performing tools.

The burden of a digital forensic investigator is to “confirm unlawful, unacceptable, or unauthorized behavior” (Mandia & Prosis, 292). To this end, the investigator must be able to confidently collect volatile data during an initial response, including system configuration, user profiles, process attributes, event logs, and port states:

- The following tools reviewed in this paper are able to meet the needs of live data collection: *Autoruns*, *Filemon*, *Process Explorer*, the *PsTools* suite, and *RootkitRevealer*.
- Several of the reviewed tools are useful for digital forensics, but not live data collection, including *DiskView*, *Junction*, *SDelete*, *Streams*, and *Registry Mechanic*.
- Four of the tools reviewed are either unnecessary or inappropriate for a Windows digital forensics toolkit:
 - *Contig* could possibly destroy evidence during defragmentation.
 - *DiskExt* provides a minimal amount of information, which is provided in more detail by tools such as *PsInfo* and *Diskpart*.
 - *Diskmon* is unnecessary in a toolkit which employs *Filemon*, a detail-rich tool for collecting file system and hard disk activity.

- *Snort*, while extremely valuable as an intrusion detection system, serves little purpose in the collection or examination of digital evidence.

2. Windows Digital Forensics Toolkit

A Windows digital forensics toolkit should consist of enough tools to allow an investigator to make a decision regarding the extent of damage to or evidence available from a compromised system. To make this decision a digital forensic investigator must perform live data collection to retrieve any valuable volatile data that will be lost upon system shutdown. Once a drive is imaged and shutdown, the forensic duplicate can be taken back to a lab, mounted into a Linux virtual system, and examined using the trusted Sleuthkit tools (or any other digital forensics tool suite). With this in mind, my analysis will concentrate on those tools necessary for live data collection from a Windows system.

It is important that all tools used for the digital forensic investigation are trusted commands executed from a live CD, DVD, or other storage medium. The basic executables needed to operate a command shell are

- *cmd* (command prompt),
- *cd* (change directory),
- *dir* (list directory),
- *date* (current system date),
- *time* (current system time),
- *more* (displays output one screen at a time),
- *doskey* (command history for current shell), and
- *exit* (exits command shell).

In addition to basic commands, the investigator will need tools to collect information regarding system configuration, user profiles and privileges, processes running or scheduled and the files they access, event logs, port states and connections, actions from remote systems, and system security status. Finally, the investigator will need a tool for verifying the integrity of the collected evidence.

This report will analyze the following tools for usage, digital forensics purpose, and suitability in a Windows digital forensic toolkit:

- Sysinternals utilities: *Autoruns*, *Contig*, *DiskExt*, *Diskmon*, *DiskView*, *Filemon*, *Junction*, *Process Explorer*, *PsTools* suite, *RootkitRevealer*, *SDelete*, and *Streams*
- *Registry Mechanic*
- *Snort*: Network Intrusion Detection System

2.1. Sysinternals

Sysinternals, recently acquired by Winternals, is a company that offers free utilities for use on Windows operating systems. The tools, written and maintained by Mark Russinovich and Bryce Cogswell, are useful for many tasks ranging from system optimization to networking. The utilities in the following sections are being considered for their digital forensics purpose.

2.1.1. *Autoruns*

Autoruns displays scheduled tasks and all programs that are configured to start automatically during a system bootup or user login, including programs from the Startup folder, toolbars, and scheduled tasks. The user opens the utility and a system scan is performed to populate the tables. The entries in the *Autoruns* graphical interface are displayed in the order they are processed by the Windows system. Unfortunately, reviewing the process entries does require a familiarity with the normal scheduled processes of the system. To aid in the investigation of questionable processes, Sysinternals has included a Google function in *Autoruns*. By right-clicking on any entry and selecting “Google,” the user will be directed to an open browser window with the results of a Google Search for the executable’s designation. If the user selects “Jump” instead, an *Explorer* window will open for the file system directory location of the questioned executable. Scheduled processes can be disabled or deleted from within *Autoruns* if the user deems them to be malicious or simply unnecessary.

For general purpose use, *Autoruns* can be used to optimize the performance of a system by allowing the user to disable or delete auto-starting processes. However, this utility is also helpful during digital forensics. Any task that an intruder scheduled to start on bootup or login can be located using *Autoruns*, assuming it is a long-lived process instead of a single occurrence. Although *Autoruns* is a very comprehensive startup monitor, it may not be infallible. By choosing to “Hide Signed Microsoft Entries,” the user can concentrate on third-party auto-starting executables, significantly narrowing the scope of the investigation. During a digital

forensic investigation, it would be inadvisable to ignore Microsoft entries on the basis that an intruder may be able to forge the Microsoft known-good status or alter the contents of a trusted Microsoft executable. As always, a digital forensic investigation should consider all possible evidence.

The *Autoruns* utility does not have any obvious gaps; it provides the user with full control over startup processes and far exceeds most other startup monitors. The Windows operating system does come bundled with a System Configuration Utility (*MSConfig*) that provides control over startup processes. However, it requires a more advanced knowledge of the system, does not provide the in-depth information that *Autoruns* includes, and does not allow the user to permanently delete any startup process. There is also a variety of startup monitors available—freeware and commercial versions—but some warn the user only when an application tries to schedule a new startup process, rather than allowing the in-depth control that *Autoruns* provides. *Startup Inspector for Windows* (<http://www.windowsstartup.com/startupinspector.php>) does provide some of the control of *Autoruns*, but it only lists processes from the Registry and Startup folders; *Autoruns* lists processes from fourteen separate locations. Overall, *Autoruns* is the most comprehensive free startup monitor currently available and a useful tool in a Windows digital forensics toolkit.

2.1.2. Contig

Contig is a defragmenter used on single files. Run from the command line, *Contig* can be used to create a new contiguous file of specified length or to defragment existing files. The five available options for the *Contig* utility are: verbose, quiet, analyze, recurse, and new. By activating the analyze flag, the user can specify a file for quick fragmentation analysis. The most useful option available for the *Contig* utility is the recurse flag. Rather than designating a specific file, the user can choose to analyze or defragment an entire hard disk drive or just a file system directory; *Contig* also works on external and removable media. Although the Windows operating system is bundled with a utility to defragment hard disk drives, it does not allow the in-depth single-file control that *Contig* provides.

For general purpose use, *Contig* can be used to optimize system performance by making contiguous those files that are frequently accessed or are vital functions. Unfortunately, *Contig* is not currently useful for digital forensics. Although it is easier to extract contiguous files, it

would be unwise to defragment files prior to extraction from a suspect drive. The process involved in defragmentation would reposition the file on the storage medium causing destruction of possible evidence. A credible digital forensic investigator must be able to show that minimal changes or damages occurred to the storage medium during an investigation.

In order to become a useful digital forensics tool, the *Contig* utility would need to be able to perform additional tasks. If after running in analysis mode the *Contig* summary provided cluster or inode information for the fragments of the analyzed file, the user could then extract the file in a forensically sound manner based on that information. Used in conjunction with *DiskView* (see Section 2.1.5.), the names of fragmented files provided by *Contig* can be used to discover the cluster location of each file fragment, allowing for quick extraction through file recovery methods. However, no freely available defragmenting tools are designed to provide the additional meta-data information necessary to be useful for digital forensics. Because inode and cluster location can be discovered using other utilities, it is not necessary to include *Contig* as part of a Windows digital forensics toolkit.

2.1.3. *DiskExt*

DiskExt is a disk extent dumper. Run from the command line, *DiskExt* provides information about the partitions existing on a hard disk drive, including byte offset and length. The utility by default will examine only hard disk drives, but it can also be directed to analyze external or removable media. Because *DiskExt* is a very simple utility without any options, its only purpose for general use or digital forensics use is as a volume-mapping tool. While this information is necessary for a digital forensic investigation, *DiskExt* would be more useful if it included information such as file system structure or the amount of allocated versus unallocated space in the partition; this information is available through Sysinternals' *PsInfo* utility (see Section 2.1.9). The Windows operating system is bundled with the *Diskpart* command line utility that will provide the same information as *DiskExt*. To acquire file system structure from a Windows XP command line, the user should employ the *FSUtil* utility; earlier versions of the Windows operating system include the *FDisk* utility instead. Another option for obtaining NTFS file system information is the free *NTFSInfo* utility from Sysinternals. Although a Windows digital forensics toolkit should include a utility to map disk volumes and partitions (in addition to a tool for obtaining file system information), there are many tools available that will provide

information more valuable than that provided by *DiskExt* and any would be acceptable for a Windows digital forensics toolkit.

2.1.4. *Diskmon*

Diskmon captures all hard disk activities and presents them to the user in a graphical log interface. Upon opening, *Diskmon* begins monitoring all hard disk drives for read and write activity; however, there is no option for monitoring removable media. The information provided for each event includes initiation time, duration, responsible device, disk sector (or cluster) written to or read from, and length of the read/write activity. The utility can be minimized as a disk light in the system tray for continuous silent monitoring.

For general and digital forensics use, *Diskmon* allows for better understanding of hard disk activity. However, for in-depth digital forensics, *Diskmon* should be used in conjunction with other utilities. Because this utility provides only the disk sector written to or read from, a digital forensic investigator would need to use a tool such as *DiskView* (see Section 2.1.5.) to locate the sector and find the associated file being accessed. In addition, to understand which applications are performing the read/write activity, an investigator would be wiser to use a utility such as *Filemon* (see Section 2.1.6.) which provides information on both the application performing the activity and the file being accessed. With this additional information, an investigator can begin to understand the actions of suspicious applications.

There are other system monitoring utilities available, including *System Monitor* bundled with the Windows operating system, that are designed to provide statistics on disk performance, rather than simply reporting read/write activities as *Diskmon* does; any would be more useful for general purpose system maintenance and optimization. Also, utilities such as *Filemon* are more useful for a digital forensics purpose; therefore, *Diskmon* is not a necessary tool for a Windows digital forensics toolkit.

2.1.5. *DiskView*

DiskView displays a graphical map of a disk; the user can specify a hard disk drive or any external storage medium. The user can choose between a bird's-eye-view of the disk and an in-depth view of individual clusters; by zooming in on the disk map, the user can view individual clusters on the disk. Information for individual clusters is provided, including the status of the

cluster (allocated, unallocated, fragment, system, etc.) and the file associated with the cluster. The user can select a cluster and click “Next” to jump among all fragments of the file associated with that cluster. By double-clicking on a cluster, the user can view a cluster’s properties, including its number and position in the file cluster, associated fragments and their cluster number, and the path for the file associated with that cluster. *DiskView* also allows the user to select and locate on the disk a specific file from the file system directory.

For general purpose use, *DiskView* is useful for understanding the file fragmentation of a disk. If the user finds a highly fragmented file, they can use the *Contig* utility (see Section 2.1.2.) to defragment that particular file, allowing for performance optimization. *DiskView* can also be very useful for a digital forensic investigation. The investigator can view every area on the disk whether allocated or unallocated. Because *DiskView* provides the cluster number and length of each file fragment, the digital forensic investigator can quickly extract files using file recovery methods.

DiskView, unfortunately, provides properties for only allocated clusters, as unallocated clusters are simply displayed as blank; addition of an unallocated cluster analysis function would greatly increase the value of *DiskView* for digital forensic investigators. It would also be beneficial to include a “Go to Cluster” function that would allow an investigator to quickly see the properties of a specific cluster number. There are other free graphical disk map utilities available, such as the Zero Assumption *Disk Space Visualizer* utility (<http://www.z-a-recovery.com/tools-visualizer.htm>) and the SourceForge *Graphical Disk Map* utility (<http://sourceforge.net/projects/gdmap/>); however, no other tool appears to provide the in-depth cluster information available with *DiskView*. Due to the detailed view available, *DiskView* is a valuable tool to have in a Windows digital forensics toolkit.

2.1.6. Filemon

Filemon monitors all file system activity and displays the actions in a graphical log interface. At start time, *Filemon* begins logging all open, read, and write requests and any successes or errors; logs are created for all hard disk drives or the user can specify a particular disk or removable medium. Information presented for each log event includes initiation time and duration, process name, type of request, path of file accessed, activity result, and other information such as the disk sector accessed and the length of read/write activity. The user can choose to filter or highlight

processes based on the process name or path of the file accessed. By double-clicking on an event, the user is directed to an *Explorer* window for the file system directory location of the file accessed by the process. If a user right-clicks on an event, he can view the properties for the process or file path, or he can add the process or file path to the log filter for inclusion or exclusion.

For general purpose use, *Filemon* provides information on the inner-workings of Windows and third-party applications. Although the utility is user-intensive, it aids in understanding how files are used and in troubleshooting system configurations. For digital forensics use, *Filemon* is extremely helpful in understanding the actions of malicious applications. After filtering out (excluding or highlighting) known-good processes, the digital forensic investigator can run a malicious process in a virtual environment and view all file calls and accesses performed by the application. This information could aid the investigator in locating all the files affected by a malicious application and in creating rules to prevent future intrusion by similar applications.

The only obvious feature that could be added to the *Filemon* utility is a method for identifying known-good processes. At this time, it requires a user deeply familiar with the day-to-day function of the Windows operating system—or a user willing to study *Filemon* logs long enough to learn each event's function. There are very few free file system activity monitoring utilities available, and even the commercial versions do not provide the wealth of information available through *Filemon*. This utility is a welcome addition to a Windows digital forensics toolkit.

2.1.7. Junction

Junction, run from the command line, allows the user to create and delete symbolic links and search for reparse points. The search function can be set to recursively search all subdirectories. For general purpose use, establishing symbolic links may aid a user in file system organization. However, the identification of reparse points may be especially useful for a digital forensic investigator. Malicious applications may employ symbolic links to mask the purpose of the application or to mislead a user into deleting system files. Standard utilities bundled with the Microsoft operating system include *Mountvol* used to create volume mount points (similar to symbolic links, but only apply to NTFS storage volumes) and *FSUtil* used to create hard links

between files (functionality not present in *Junction*). A search for symbolic links by the *Junction* utility will identify mount points, but not hard links, as reparse points. A version of this utility that enabled volume mounting and identified linked files would be even more useful in a digital forensic investigation. However, even in its current state, *Junction* is a useful tool in a Windows digital forensics toolkit.

2.1.8. Process Explorer

Process Explorer is a process management utility similar to the Windows *Task Manager*. The main advantage that *Process Explorer* has over any other free process manager is its inclusion of a sub-window displaying information about opened or loaded handles, dynamic link libraries (DLL), and memory-mapped files. At run time, the *Process Explorer* graphical interface is populated with entries for currently active processes. The user can choose to display columns with information such as the process' command line, window title, and start time. By right-clicking on a process or DLL, the user has access to an internet search function similar to that of the *Autoruns* utility (see Section 2.1.1). A right-click on a handle supplies the user with the option of closing the handle. The user can always choose to kill a process if it is malicious or unnecessary.

For general purpose use, the *Process Explorer* can be used to replace the Windows *Task Manager* for application and process management. A digital forensic investigator can use *Process Explorer* similarly to the *Autoruns* and *Filemon* utilities (see Section 2.1.6). The interface can be set to transparent, allowing the investigator to follow process events happening in the background as an application is executing. There are no other free process management systems that appear to have the diverse capabilities of *Process Explorer*. A utility that fully incorporated all features of the separate *Autoruns*, *Filemon*, and *Process Explorer* tools would be invaluable to a Windows digital forensic investigation; any Windows digital forensics toolkit should include all three utilities for the collection of all possible information related to running processes.

2.1.9. PsTools Suite

The *PsTools* suite includes tools used to manage processes on local and remote systems. They are all run from the command line of the local system and do not require the tools to be installed on any remote computers. The following tools are included in the suite:

- *PsExec*: Allows the user to remotely execute processes with a fully interactive console. Replaces telnet programs.
- *PsFile*: Lists files that are opened remotely (similar to Windows *net file* command) and allows those files to be closed.
- *PsGetSid*: Queries a computer's SID. Provides SID for user accounts and allows translation to a representative name.
- *PsInfo*: Provides information about local or remote systems, including uptime, kernel and operating system version, registered owner, system root, and processor type. Includes options for displaying disk volume information and installed hotfixes and applications with their version number.
- *PsKill*: Terminates specified processes (by process ID or name) on local and remote systems.
- *PsList*: Displays information for all processes currently running on system. Command line utility for *Process Explorer* functionality (see Section 2.1.8).
- *PsLoggedOn*: Displays users logged on locally (similar to Windows *net session* command) or remotely and the date and time of logon. Able to search network neighborhood for specified user.
- *PsLogList*: Dumps contents of specified event logs from local or remote computers.
- *PsPasswd*: Allows password changes to accounts on local or remote computers. Can be used to perform efficient change to administrator password across the network.
- *PsService*: Displays services, both running and stopped, on the system. Provides service name, display name, and explanation of the service, among other information.
- *PsShutdown*: Allows user to shutdown or lock local or remote system, cancel shutdown, or logoff users.
- *PsSuspend*: Suspends specified processes (by process ID or name) on local and remote systems.

For general purpose use, the *PsTools* suite can be used to administer local and remote systems. However, many of these tools also have a digital forensics purpose. In *Incident Response & Computer Forensics*, Kevin Mandia and Chris Prosis suggest including

PsLoggedOn, *PsList*, *PsLogList*, *PsInfo*, *PsFile*, and *PsService* utilities in a Windows digital forensics toolkit (97-98). *PsLoggedOn* can be used to determine users logged on to a compromised system, potentially narrowing the list of possible attackers. *PsList* can be used to record all of the processes running on a compromised system before it is powered off. *PsLogList* can be used to capture the history of events that occurred on a compromised system. However, even the *PsGetSid* can be useful to a digital forensic investigation; according to Mandia and Prosis, “SIDs can be the digital fingerprints that prove that a remote system was used to log on to a machine and access a domain” (328). Although there are other utilities available that enable actions similar to the *PsTools* suite, few freely available tools can be administered remotely. Because all the *PsTools* can be administered on remote systems, the investigator is able to acquire digital evidence remotely. The *PsTools* suite is a welcome addition to a Windows digital forensics toolkit.

2.1.10. RootkitRevealer

RootkitRevealer is a rootkit detection utility that successfully detects persistent, memory-based, user-mode, and kernel-mode rootkits—programs that hide the presence of malicious programs. The utility is able to detect all the persistent rootkits currently published by www.rootkit.com, but does not search for rootkits that do not attempt to hide files or registry keys. The current version of *RootkitRevealer* is available only as a graphical interface because some malicious programmers have begun targeting its executable name; to prevent this attack, *RootkitRevealer* runs as a Windows service from a randomly named copy of itself. It is best to perform the scan with all other applications closed and the system idle to minimize discrepancies in the Windows API, directory index, and Master File Table (MFT). When performing a scan, the user can choose to “Hide NTFS Metadata Files” or to “Scan Registry.” Remote scans can be performed using the *PsExec* utility (see Section 2.1.10). *RootkitRevealer* only identifies potential rootkits and does not attempt to delete them.

For general purpose use, the *RootkitRevealer* utility allows the user to discover and investigate possible rootkits infiltrating a system. For digital forensics purposes, this information can be used as evidence for an attack against a compromised system; it can also aid in quicker recovery of an organization’s essential systems. According to Bryce Cogswell and Mark Russinovich of Sysinternals, “It is theoretically possible for a rootkit to hide from

RootkitRevealer... however, this would require a level of sophistication not seen in rootkits to date” (<http://www.sysinternals.com/Utilities/RootkitRevealer.html>). Although *RootkitRevealer* may detect all current versions of rootkits, it would be useful if it provided more information about the scan results, such as a quick Google Search function, similar to the functionality of other Sysinternals utilities. There are not many other rootkit detection utilities freely available but one, the *Sophos AntiRootkit* (<http://www.sophos.com/products/free-tools/sophos-anti-rootkit.html>), claims to delete detected rootkits—functionality not provided by *RootkitRevealer*. However, even without this functionality, *RootkitRevealer* is a valuable tool for a Windows digital forensics toolkit.

2.1.11. *SDelete*

SDelete is a secure delete utility that implements the Department of Defense’s standards for clearing and sanitizing storage media. The utility can be used to delete existing files and securely erase any data in unallocated space. *SDelete* will also overwrite the names of files that are securely deleted, ensuring that not even the file name can indicate what may have existed in the file. The user has the option of deleting a single file, using a wildcard to delete related files, or recursively deleting entire directories. This utility can be used for the general purpose of ensuring that sensitive data cannot be recovered once deleted. The digital forensic use is two-fold: (1) an investigator can use *SDelete* to sanitize a storage medium before digital evidence collection, similar to performing a `/dev/zero` command; and, (2) an investigator can recognize when a secure delete has been performed on a compromised system by understanding how the utility functions. There are many free secure deletion utilities available, but most target only specific areas, such as browser history and temporary files; others securely delete files, but not unallocated files; some sanitize only an entire disk. Because of its functionality, specifically the ability to sanitize unallocated space, and lack of obvious gaps, *SDelete* is a valuable tool for a Windows digital forensics toolkit.

2.1.12. *Streams*

Streams is an alternate data stream identification utility. It will examine specified files and directories for alternate data streams and report the name and sizes of any encountered. The user can then view the alternate data stream using the *more* command. *Streams* also allows for the deletion of alternate data streams, but leaves the host file intact. For general purpose use and

digital forensics use, *Streams* can be used to detect hidden data in existing files. A digital forensic investigator can use this utility to discover if suspects are attempting to hide data (such as proprietary information) in alternate data streams. Because this is a relatively simple utility, there does not seem to be any gaps in its functionality. There is at least one other alternate data stream detection utility freely available; *ADS Spy* (<http://www.richardthelionhearted.com>) uses a graphical interface to accomplish the same tasks as *Streams*. For its functionality and simplicity of use, *Streams* is a valuable tool in a Windows digital forensics toolkit.

2.2. Registry Mechanic

Registry Mechanic is a utility to scan the Windows registry and repair problems caused by invalid registry entries, which may result from uninstalled software, corrupt drivers, and embedded malware. Using the evaluation version of *Registry Mechanic*, the user can perform a free scan of their system, review a report of registry errors found, and repair some items. However, a complete repair requires the user to purchase a fully functional version of the utility (approximately \$30). For the general purpose user, this purchase may be worthwhile for the quick-and-easy system optimization that *Registry Mechanic* provides. A digital forensic investigator can use the information provided by the free scan to identify potential evidence of malicious applications affecting the Windows registry. There are some additional tools that an advanced user can employ to edit or delete registry keys that *Registry Mechanic* has identified as problems. The *regedit* utility is bundled with the Windows operating system and provides a graphical tree interface of registry entries. *RegDelNull*, another free utility from Sysinternals (<http://www.sysinternals.com/Utilities/RegDelNull.html>), allows the user to delete registry entries with embedded-null characters. Because of the cost of *Registry Mechanic*, it is difficult to say that it is a necessary tool for a Windows digital forensics toolkit. However, the absence of comparable free utilities and its full-functionality use for quickly repairing registry damage to an organization's essential systems make it a valuable tool.

2.3. Snort

Snort is a popular packet detection utility that is capable of running in four modes:

- Sniffer: Displays packets from the network.
- Packet Logger: Logs packets to a storage medium.

- Network Intrusion Detection System: Analyzes network traffic, compares packets to user established rules, and performs actions based on the comparison.
- Inline: Reads packets from IP tables, compares them to rules, and performs actions based on the comparison.

The most common and complex use of *Snort* is as a network intrusion detection system (IDS). For general use, it allows for in-depth control over the types of packets allowed on a network. Due to the volatility of network data, *Snort* cannot be easily used by digital forensic investigators to find a malicious packet that has already been accepted into the system. An exception would be in the case of logged network packets, but due to enormous amounts of traffic, these logs do not often provide much of a history. However, an investigator can run *Snort* in a virtual environment, execute a suspicious process, and monitor for packets. Once a signature for a malicious process has been identified, *Snort* can be used to implement rules that will prevent further packets with that signature from entering the network. Although the use of *Snort* is wide ranging, it is a tool that seems more useful to a system administrator than the typical digital forensic investigator, who can employ tools such as *tcpdump* and *nmap* to monitor network packets produced by a malicious process. The complexity of *Snort* makes it an unnecessary utility for a Windows digital forensics toolkit.

3. Analysis

This analysis will consider the appropriateness of each of the tools discussed and their purpose in a Windows digital forensics toolkit used for live data collection. Additional tools will be suggested where one of the above tools does not meet the digital forensics purpose. The tools in a Windows digital forensics toolkit must be able to collect information regarding system configuration, user profiles and privileges, processes running or scheduled and the files they access, event logs, port state and connections, actions from remote systems, and system security status. Finally, a tool is needed for verifying the integrity of the collected evidence. Following are the results of this Windows digital forensics toolkit analysis:

- System configuration:
 - *PsInfo* provides information about the system build, including disk partitions and installed applications.

- Mandia and Prorise suggest including the Windows *arp* utility to enumerate MAC address of systems that have communicated with the victim system (97).
- The *getmac* Windows utility is also useful for collecting the MAC address of each network device connected to the victim system.
- The Windows *ipconfig* utility provides information about the configuration of all system network connections.
- User profiles and privileges:
 - *PsLoggedOn* collects information on users connected locally and remotely.
 - Sysinternals maintains another utility named *LogonSessions* that will enumerate the active logon sessions for a system (<http://www.sysinternals.com/Utilities/LogonSessions.html>).
 - Mandia and Prorise suggest including the *rasusers* utility from the NT Resource Kit that provides information regarding users with remote-access privileges (97).
- Processes:
 - *PsList* enumerates all running processes.
 - *PsService* enumerates services on system, whether running or stopped.
 - *Process Explorer* provides graphical log interface of all running processes, their command line arguments, and any handles or DLLs they are using. The command line tool for enumerating running process and their DLLs is *ListDLLs*, also available from Sysinternals.
 - *Autoruns* enumerates all processes scheduled to start during system bootup or user login.
 - *Pskill* terminates a running process.
 - *PsSuspend* suspends a running process.
 - *Filemon* monitors all file system activity (including all read/write requests) and displays the actions in a graphical log interface, with in-depth details.
- Event logs:

- *PsLogList* dumps the contents of event logs.

- Ports:
 - Mandia and Prosis suggest including the Windows *netstat* utility to collect information regarding listening ports and their current connections (97).
 - Sysinternals maintains the *TCPView* utility which provides a graphical log interface of TCP/UDP ports and their connections; *tcpvcon* is the command line version that functions similar to *netstat* (<http://www.sysinternals.com/Utilities/TcpView.html>).
- Remote access:
 - *PsExec* allows the investigator to execute tools on a remote system.
 - *PsFile* collects information on files opened remotely.
 - Mandia and Prosis suggest including the NT Resource Kit utility *rmtshare* to view shares available on remote systems (97).
 - The Sysinternals utility *ShareEnum* uses a graphical log interface to display available shares and their security settings (<http://www.sysinternals.com/Utilities/ShareEnum.html>).
- Security status:
 - *PsShutdown* allows the investigator to shutdown or lock a system.
 - *RootkitRevealer* detects rootkits that hide the presence of malicious programs.
 - Mandia and Prosis suggest including the NT Resource Kit utility *auditpol* to collect the current security audit settings (98).
- Evidence integrity:
 - Mandia and Prosis suggest including a tool for conducting hash sums on collected evidence (97). Using the *md5sum* utility will insure that the results can be compared against hashes acquired from the Autopsy toolkit.

The following reviewed tools, while not useful during live data collection, can be used for digital forensic investigations:

- *DiskView* provides a graphical map of a storage medium, including cluster numbers and file fragment lengths. This tool may aid in the extraction of fragmented files.

- *Junction* enumerates reparse points and provides a method for deleting them.
- *SDelete* sanitizes storage media in preparation for evidence storage.
- *Streams* identifies alternate data streams in files.
- *Registry Mechanic* identifies invalid registry entries.

The following are reviewed tools that are not necessary in a Windows digital forensics toolkit for live data collection:

- *Contig* is inappropriate for digital forensics because evidence should be extracted in the form in which it is found. Defragmenting a file or disk before collection could possibly destroy critical evidence.
- *DiskExt* does not provide as much detailed information as a tool such as *PsInfo* or *Diskpart* (bundled with Windows).
- *Diskmon*, while not inappropriate, is unnecessary in a toolkit that contains *Filemon*, which provides more detailed information about hard disk activities.
- *Snort*, as an intrusion detection system, is not necessary for digital forensics, but is useful for network security to prevent compromise to systems.

References

- Caswell, B., & Hewlett, J. (2006). *Snort Users Manual*. Retrieved October 28, 2006, from http://www.snort.org/docs/snort_htmanuals/htmanual_260/.
- Cogswell, B. (September 27, 2005). *DiskView*. Retrieved October 28, 2006, from <http://www.sysinternals.com/Utilities/DiskView.html>.
- Cogswell, B., & Russinovich, M. (February 2, 2006). *RootkitRevealer*. Retrieved October 28, 2006, from <http://www.sysinternals.com/Utilities/RootkitRevealer.html>.
- Prosis, C., & Mandia, K. (2003). Live data collection from Windows systems. In *Incident response & computer forensics, second edition*. (pp. 95-124). New York: McGraw-Hill.
- . (2003). Investigating Windows systems. In *Incident response & computer forensics, second edition*. (pp. 291-333). New York: McGraw-Hill.
- Registry Mechanic*. <http://www.pctools.com/registry-mechanic/>. Retrieved October 28, 2006.
- Russinovich, M. (March 27, 2006). *Contig*. Retrieved October 28, 2006, from <http://www.sysinternals.com/Utilities/Contig.html>.
- . (August 27, 2001). *DiskExt*. Retrieved October 28, 2006, from <http://www.sysinternals.com/Utilities/DiskExt.html>.
- . (December 11, 2003). *Diskmon*. Retrieved October 28, 2006, from <http://www.sysinternals.com/Utilities/Diskmon.html>.
- . (August 16, 2005). *Junction*. Retrieved October 28, 2006, from <http://www.sysinternals.com/Utilities/Junction.html>.
- . (July 10, 2006). *Process Explorer*. Retrieved October 28, 2006, from <http://www.sysinternals.com/Utilities/ProcessExplorer.html>.
- . (July 10, 2006). *PsExec*. Retrieved October 28, 2006, from <http://www.sysinternals.com/Utilities/PsExec.html>.
- . (December 30, 2005). *PsFile*. Retrieved October 28, 2006, from <http://www.sysinternals.com/Utilities/PsFile.html>.
- . (August 2, 2005). *PsGetSid*. Retrieved October 28, 2006, from <http://www.sysinternals.com/Utilities/PsGetSid.html>.
- . (September 1, 2005). *PsInfo*. Retrieved October 28, 2006, from <http://www.sysinternals.com/Utilities/PsInfo.html>.
- . (August 2, 2005). *PsKill*. Retrieved October 28, 2006, from <http://www.sysinternals.com/Utilities/PsKill.html>.
- . (August 9, 2004). *PsList*. Retrieved October 28, 2006, from <http://www.sysinternals.com/Utilities/PsList.html>.
- . (March 27, 2006). *PsLoggedOn*. Retrieved October 28, 2006, from <http://www.sysinternals.com/Utilities/PsLoggedOn.html>.

- . (April 10, 2006). *PsLogList*. Retrieved October 28, 2006, from <http://www.sysinternals.com/Utilities/PsLogList.html>.
 - . (May 19, 2004). *PsPasswd*. Retrieved October 28, 2006, from <http://www.sysinternals.com/Utilities/PsPasswd.html>.
 - . (April 10, 2006). *PsService*. Retrieved October 28, 2006, from <http://www.sysinternals.com/Utilities/PsService.html>.
 - . (March 27, 2006). *PsShutdown*. Retrieved October 28, 2006, from <http://www.sysinternals.com/Utilities/PsShutdown.html>.
 - . (October 2, 2003). *PsSuspend*. Retrieved October 28, 2006, from <http://www.sysinternals.com/Utilities/PsSuspend.html>.
 - . (July 10, 2006). *PsTools*. Retrieved October 28, 2006, from <http://www.sysinternals.com/Utilities/PsTools.html>.
 - . (October 15, 2003). *SDelete*. Retrieved October 28, 2006, from <http://www.sysinternals.com/Utilities/SDelete.html>.
- Russinovich, M., & Cogswell, B. (July 10, 2006). *Autoruns*. Retrieved October 28, 2006, from <http://www.sysinternals.com/Utilities/Autoruns.html>.
- . (July 13, 2006). *Filemon for Windows*. Retrieved October 28, 2006, from <http://www.sysinternals.com/Utilities/Filemon.html>.