Quick and Effective Windows System Baselining and Comparative Analysis for Troubleshooting and Incident Response

Kevin Fuller

Quick and Effective Windows System Baselining and Comparative Analysis for Troubleshooting and Incident Response

GIAC (GCIH) Gold Certification

Author: Kevin Fuller, 2a373a@hughes.net
Advisor: Antonios Atlasis

Accepted: 22nd January 2012

Abstract

Baselining is an important tool for troubleshooting, audit, incident response and forensics. It involves documenting the features of a known good state of a system. This can be used to do a comparative analysis of the current system state to determine what has changed and how it was changed. In practice, doing a complete and documented system baseline can be time consuming and cumbersome and is infrequently done. The goal of this paper is to highlight a pair of tools and a methodology that can provide the ability to baseline a Windows system in a more complete manner across a number of areas. More importantly, the methodology will demonstrate the ability to do a quick, effective comparative analysis of a baseline and the current state to determine what has changed.

1. Introduction

What is a baseline? The primary definition of baseline is that it is a line that is a basis of measurement (Farlex Inc, 2011).

In general, a baseline is a well-defined well-documented version of the solution at some point in its lifecycle. It is used as a foundation to support other activities including follow on measurement. In actual use, a baseline may be defined as a more specific subtype (U.S. DOT, 2011).

- Functional Baseline: Initial specifications established; contract, etc.
- Allocated Baseline: State of work products once requirements are approved.
- Developmental Baseline: State of work products amid development.
- Product Baseline: Contains the releasable contents of the project.
- Others based upon proprietary business practices.

The concept of a baseline is standard across a number of business functions. In construction the baseline can be used to measure scheduling, management, construction methods and results. In a business that produces a product or service the can be a development baseline, a maintenance baseline, an operations baseline. If a business is purchasing or acquiring something there is an acquisition baseline and a performance baseline.

In information technology and security the recommended use of baselines has far reaching effects. From software development to networks and systems, baselines provide a starting point from which various measurements can be used for processes like troubleshooting, Software or Solution Development, Life Cycle management, auditing, etc.

A system baseline refers to a measurement of system or solution component. A good analogy that has been used in the past is that it's like leaving the teenage kids home alone for the weekend and taking a picture of every room in the house before leaving and

after returning (Adams, 2009). Comparative analysis is viewing the before and after pictures and determining what has changed and whether the changes were routine or the result of unknown or unauthorized activity (like a teenage party).

In Information Technology (IT), baselines are an important component of configuration management. In some circles, configuration management is viewed as another way to describe change management. Configuration management is also viewed as a subset of change management.

The Information Technology Infrastructure Library (ITIL) goes one step further and defines configuration management and change management as distinct processes under the Service Support volume of best practices (Klosterboer, 2008). As a separate process, change management is an enterprise function that manages changes at a high level. Configuration management addresses those changes at the system or solution level. A change to a configuration setting on a system that does not have an enterprise impact would be managed through configuration management. A configuration change that has an enterprise impact would be addressed and documented in configuration management and change management. This paper will view the configuration management process as it is defined above.

2. The Baseline Process and Common Windows Tools

Information Technology auditing in involves measuring IT processes, including security, against defined frameworks and processes that spell out the requirements through a series of internal controls. Most frameworks like COBIT spell out high level areas and break those areas down into objectives and activities that support those objectives (Davis, Schiller, & Wheeler, 2011).

The system baseline measurement in auditing may be against these objectives. The activities that support the objectives will be defined in a document of best practices and configuration settings for functionality and security. Such documents can be found on vender websites, third party and government websites such as the National Security Agency (NSA, 2009). Each guide will apply to a specific system. The comparative

analysis will consist of measuring the current system or solution state against those baseline documents, evaluating and reporting on the differences.

In system administration, the same best practices documents that are used in auditing would help in configuring a system baseline. The comparative analysis of a system baseline would be between the current state and the most recent baseline resulting from a system change like a software installation. The goal is to view what has changed as a result of the installation. If the goal is to troubleshoot the problem, the analysis becomes more focused on what has changed that could be the cause of the problem.

In information security the system baseline would be used with managing security configuration but has its real value in incident response and forensics. In these processes the value of a system baseline is when an incident occurs and the goal is to find out what has changed on the system as a consequence of incident.

The Incident Response methodology (Mandia., Prosise, & Pepe, 2003) consists of seven major components:

- Pre-incident preparation.
- Initial response.
- Formulate a response strategy.
- Investigate the Incident.
- Reporting.
- Resolution.

When an incident occurs and the incident response team is activated, one of their first tasks in investigating the incident is data collection. To do so there are three primary challenges that need to be considered (Mandia et al., 2003).

1. The data collection must be done in a forensically sound manner (i.e. without destroying any evidence).
2. The volume of data collected will easily exceed a single person's ability to read it in a reasonable timeframe.

Author Name, email@address

3. Data collected must be handled in a manner that preserves its integrity.

When dealing with data collection on a server and desktop computer systems, the goal is to collect the volatile data with a minimum impact to the system state.

For system baselining there is a plethora of tools in the Windows world that can gather this information. Microsoft Security Baseline Analyzer (MSBA, 2010) provides information on patch and security status of Windows, Microsoft SQL, and additional Microsoft components. The built-in System Information utility can view hardware configuration and provide a granular view of operating system components like drivers, services. The table below highlights a number of other available tools that can be used to collect system information (McDougal 2007).

arp.exe	hunt.exe	ntlast.exe	reg.exe
attrib.exe	ipconfig.exe	openports.exe	regdump.exe
auditpol.exe	iplist.exe	pclip.exe	RookitRevealer.exe
autorunsc.exe	ipxroute.exe	promisdetect.exe	route.exe
cmd.exe	listdlls.exe	ps.exe	sc.exe
cmdline.exe	mac.exe	psfile.exe	servicelist.exe
dd.exe	mdmchk.exe	psinfo.exe	sniffer.exe
drivers.exe	mem.exe	pslist.exe	streams.exe
dumpel.exe	nbtstat.exe	psloggedone.exe	strings.exe
efsinfo.exe	net.exe	psloglist.exe	tlist.exe
fport.exe	netstat.exe	psservice.exe	uname.exe
handle.exe	netusers.exe	pstat.exe	uptime.exe
hfind.exe	now.exe	psuptime.exe	whoami.exe
hostname.exe	ntfs.exe	pulist.exe	

Table 1. Common Windows Information Tools

Additionally, Windows Powershell and WMIC commands combined into scripts can extract a broad cross section of Windows system information.

Author Name, email@address

However, a single software tool that can easily gather most or all of the system information is not common. Usually, creating a good baseline or creating a comparative baseline entails gathering information using several different tools and then trying to align their output formats into a common format for viewing or manipulating the data. Another consideration is the information that is gathered by each tool and which information is important for a system baseline. A tool may, for example, gather information about the contents of memory and the amount of memory installed. The former would be important for incident handling but the later may be more important to a troubleshooting function.

The bigger challenge is doing the comparative analysis of all the data. Aligning the two or more different versions of tool output and finding the differences between the versions can be a mixed bag. For open ports and services it would be fairly easy to note the differences. When dealing with changes to specific files and processes, the comparative analysis becomes more time consuming and challenging.

The average system administrator is likely to look at the challenge and give up trying to do so in the first place. They may be more inclined to trust that, in the event something happens, they can figure out what has changed based on their innate knowledge of the system and its settings.

Alternatively, the system administrator may choose to trust in a proven tool or two to baseline a specific part of the system. One tool category that comes to mind is file integrity checking. It is an accepted process with a number of proven tools for baselining the file system and doing a comparative analysis looking for changes. This process can cover a large part of the system footprint.

Numerous security compromises and much of the day to day activity for a system involve changes to files on the system. So, file integrity checking can provide a fairly comprehensive view of the system. However, it is not complete and some areas, such as memory resident activity or ports and services, will not be included. There is also still the need to separate normal file change activity from activity that may be specific to aid

Author Name, email@address

troubleshooting or to find a compromise. Most file integrity software detects a change to the file size or its checksum and not what has specifically changed in the file.

What is needed is a tool or tools and a process that can quickly baseline the majority of a system settings and configuration without creating a lot of overhead for the system administrator. For forensics, the tools and processes used must also minimize the impact to the system. While it is ideal to image the system hard drive and do analysis offline, the more likely scenario is that forensic information will have to be extracted from the running system at the time an incident is initialized. In this case, it is generally accepted that minimal changes to the system state are acceptable as long as they are documented and the collection methods are forensically sound (McDougal, 2007).

The "forensically sound" data collection requirement is where the few commercially available tools do not have a documented or observed capability. Most of these tools were designed and focused on system data collection from a configuration management perspective. They also focus on the collection of the data types associated primarily with system administration and performance. Much of this same data is also useful for incident handling and forensics. However, it appears from basic research that most of the commercial tools do not utilize integrity validation controls such as hashing the output. Without these controls the output is useful in incident response but would have minimal value if the forensics investigation resulted in administrative or legal action. A more comprehensive analysis of the features of these tools is beyond the scope of this paper.

Furthermore, the commercial tools may not collect the right data because they have a focus on system administration and performance. One example is the Modified, Accessed, Changed (MAC) file information. This information can have value in incident handling forensics and even system troubleshooting but, is generally not necessary for systems administration and configuration management. One of the advantages with the tools and processes that will be discussed is that they will meet the requirements of forensically sound, comprehensive data collection and can provide a through view of the

Author Name, email@address

system state at the time the data is extracted. The tools presented here will be the Windows Forensics Toolkit (WFT) and KDiff3.

3. The Tools

3.1. Windows Forensics Toolkit

3.1.1. Background

WFT is a forensics analysis tool written by Monty McDougal when he was pursuing his GCFA certification. He wanted to create a tool that could automate the information gathering process in a forensically sound manner and that was self contained requiring little outside interaction by the user (McDougal, 2003).

At its core, WFT is a batch processing shell that calls a number of operating system specific and third party tools and executes them. It gathers the results into an HTML folder framework organized for easy viewing. It is now commercially available at www.foolman.net (McDougal, 2011).

3.1.2. Normal setup and use

To set up WFT to perform data collection it will need to do a onetime setup of the tools it will be using. This is done using the command **wft –fetchtools**. The program will read the wft.cfg file which has the location information for the tools it needs to install to do its work.

Once started, WFT will create a tools subfolder. Beneath the tools folder it will create the subfolder structure for organizing the tools and begin copying the tools. Appendix A contains a complete list of the tools and the information that each tool collects. WFT will first copy the OS specific tools for the Windows operating system it is running on. To complete the installation of the Windows tools for other versions of Windows, the setup will have to be re-run from each Windows operating system that is supported (Windows 2000 and later). The user will need to rerun WFT with the

"**fetchtools**" option while running on each applicable Windows version. Then, it will copy those tools into the corresponding subfolder.

For third party tools, the WFT readme file references using the Helix forensics CD as a source for copying some 3rd party tools. For other third party tools, WFT requires an Internet connection. It will connect to the Internet website for each tool it needs and download the tool. If an Internet connection is not available or WFT cannot locate and install a particular tool, then the user will have to manually download or copy the tool and place it in the correct location. Validating whether the setup was completely successful or not can be determined when performing the next step.

The next command to run is the **wft –fixcfg** command. This will inventory the tools folder, noting any missing tools, and perform a hash check on each tool discovered. The results are stored in a previously defined file that then must be copied to and overwrite the wft.cfg using the **move** command. WFT is now ready to do run against the system.

At this point, the entire WFT file structure must be copied to a USB drive so that it can be run independent of the target system. This is not absolutely necessary. WFT can run from the system drive. From an incident handling and forensics standpoint running from a USB drive mitigates the risk of unneeded changes to the target system which is important to incident handling. The USB drive also provides the flexibility to move WFT from machine to machine while insuring the tools that WFT accesses are system independent.

When executed without any switches, WFT starts in interactive mode. The tool has a number of parameter options available for customizing the way it runs (see Appendix B).

Author Name, email@address

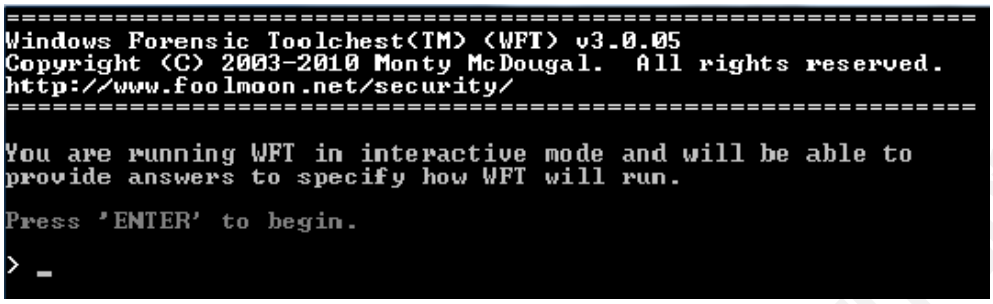

Figure 1: WFT interactive Mode

In this mode the user is prompted to answer questions that will configure some of the more common run parameters for the executable before it runs.

The questions WFT asks are located in Appendix C. The questions deal with the various parameters for running WFT and setting up the output for a forensics investigation. In most cases the defaults can be accepted with one exception. When working with a multi-drive or volume system it is a good idea to select the C: volume when asked unless circumstances dictate otherwise. Some tools like mac.exe will take hours to run on a large (100 GB+) drive. Restricting the WFT run to the C volume will reduce the overall time needed to complete the data collection.

Once the last question is answered, WFT begins to run the scripted routine to capture the system information.

WFT will create a specific output folder using the system name for the folder name and then a subfolder with the current date for the name and begin the collection process.

As WFT runs each tool, it will verify the tool's presence in the Tools repository and do a hash checksum check against the tool and then run it.

If WFT cannot find the tool in the tools repository, then it will return a "FILE_NOT_FOUND" error and skip that test (figure 2, 1). Once the tool has completed, WFT will hash the .html and .txt results file while writing them to their respective sub-folders in the system folder (figure 2, 2).

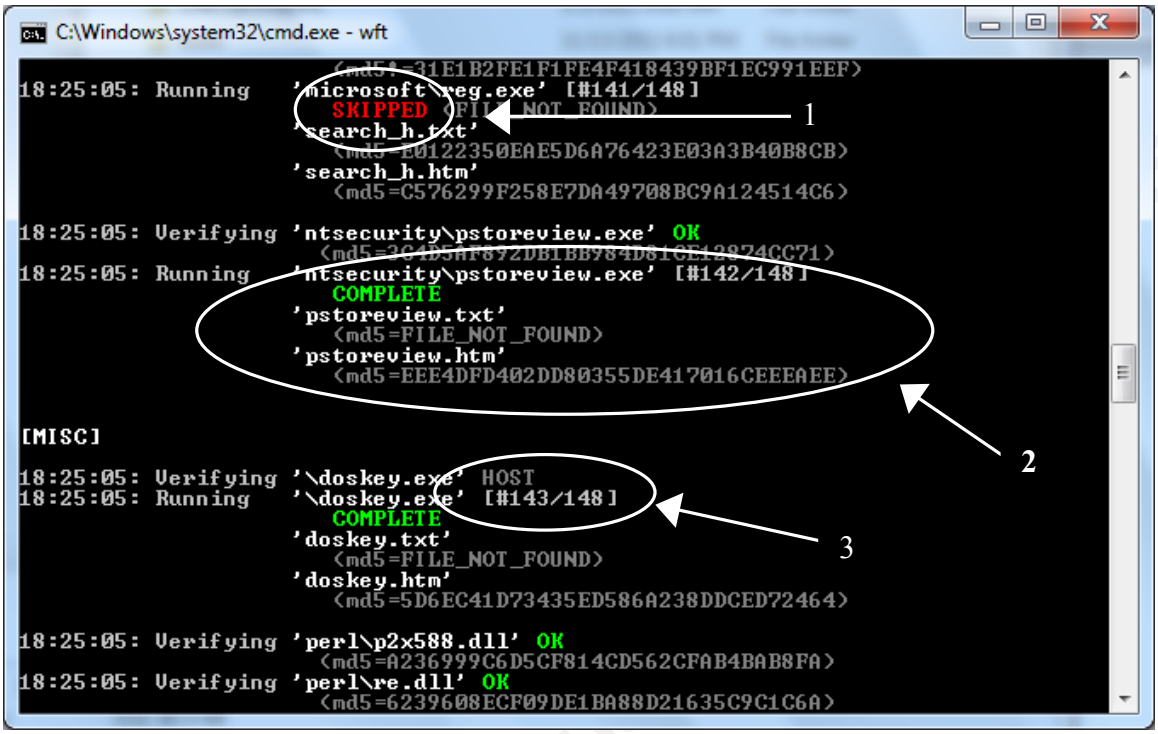

Figure 2: WFT Data Collection Run

A progress note to the right of the tool when running indicates which test WFT is currently on (figure 2, 3). Note that there are a couple of tools that will open their own window to run their check. One of these is the MAC time tool (\perl\Mac.exe) which reads the Modify/Access /Create times of each file on the system.

When complete, the output can be viewed in text or html format. Clicking on the link will bring up the html interface. Clicking on each tools link will open an html or text version of the output.

3.2 Kdiff3 V 09.9.96

3.2.1 Background

KDiff is a comparative analysis tool created by Joachim Eibl. He created KDiff after having difficulty doing 3-way merges using a commercial product. Like other similar tools, it can accept two versions of a file, a baseline file and updated file, and then, evaluate and highlight the differences between the two. It does a line by line, character by character analysis and displays the differences in an easy to read interface.

Author Name, email@address

It also has the ability to merge the differences into a single, updated file. One of the additional features of KDiff is the ability to do comparative analysis of directories as well as files. This feature is the reason why KDiff is uniquely useful when run against the output of WFT. KDiff3 is available at Sourceforge.net (Eibl, J 2011).

3.2.2 Normal setup and use

KDiff uses a standard Windows installer when being installed. The only additional options to consider are whether to include additional integration for the SVN Merge tool and Clearcase and whether to install KDiff for all users or not.

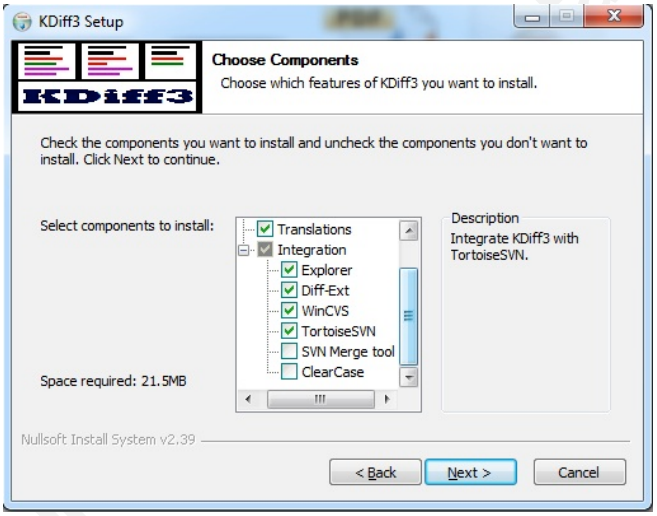

Figure 3: KDiff Install Window

Once complete the user can click on the program icon. They will first be presented with this screen where they can input the files or directories they want to work with and set some additional parameters. This screen also appears using the **File/Open** menu item. Once the files or folders are loaded the interface will appear as shown below.

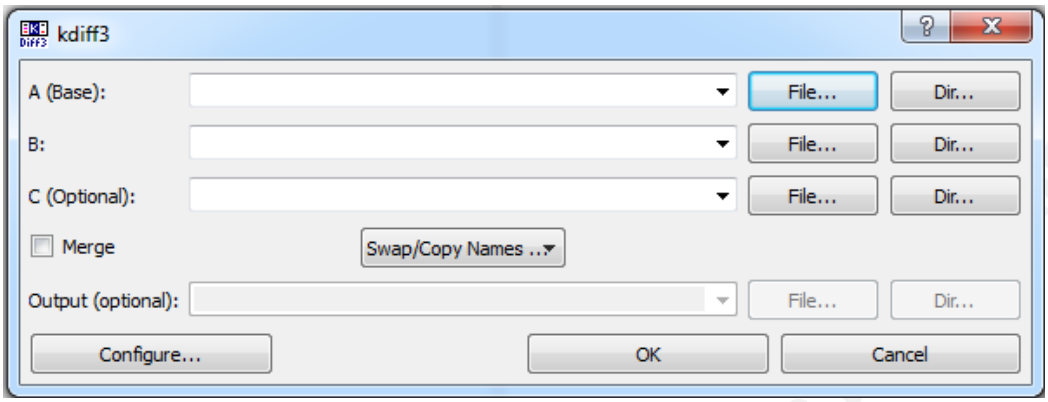

Figure 4: KDiff Load Window

This screen also appears using the **File/Open** menu item. Once the files or folders are loaded the interface will appear as shown below.

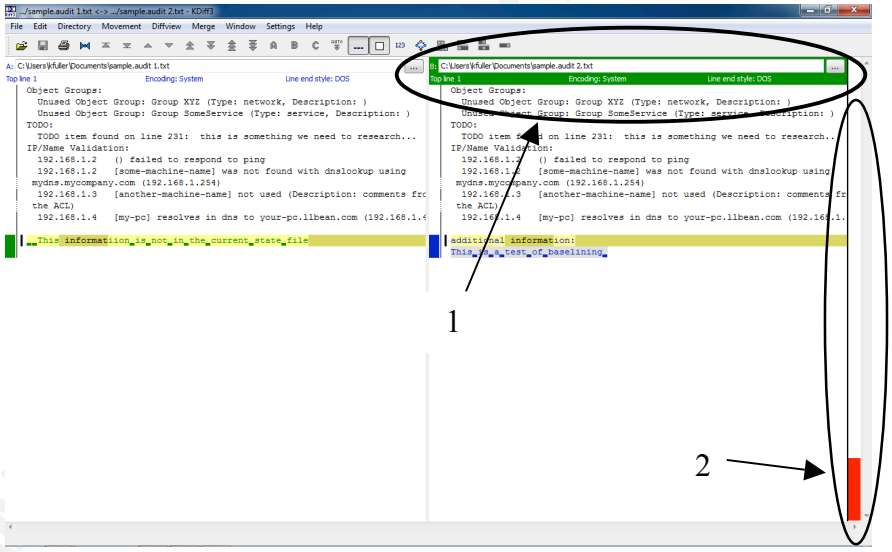

Figure 5: KDiff3 File Comparison Panel

The various colors used to highlight different parts of the output are detailed and can be re-configured from the **Settings/Config Kdiff3** menu item and the **Color** tab. Each file windows uses a different default color, blue for window A (the baseline window), green for Window B (The current state window) and magenta for Window C

Author Name, email@address

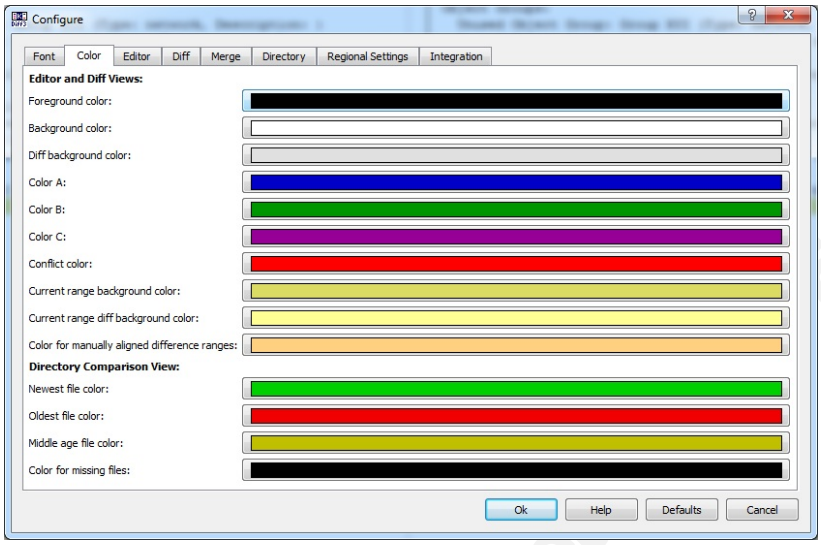

Figure 6: KDiff3 Color Tab

(When three way comparisons are being done). Clicking on the appropriate file window will result in the path window and headings line being highlighted in that windows applicable color (Figure 5, 1).

The change or difference information will be highlighted in the color of the file it is not located in and will be located in the file window of the file the changed information resides in. If the information is in the baseline file but not the current state file then the change will be highlighted in the baseline file window in the color of the current state file window. Inversely, if the information is not in the baseline file but is in the current state file, the information will reside in the current state window using the color of the baseline file window. KDiff3 expands upon the comparison feature by extending it to folders and the included files. Once the directories are identified and opened, Kdiff3 will review the directories and highlight the differences in the included files and their content. This feature is particularly useful for system baselining. System information tools, including those that WFT leverages, utilize multiple output files of information these can be placed in one directory for the baseline and another directory for the current state. One other important feature is the overview column one the right side of the comparison window. Here, red highlights where the conflicts between the two windows are located in the files (figure 5, 2).

Author Name, email@address

4. The Test Plan

The goal of the testing will be to run WFT and KDiff3 on each of three different Windows operating systems to determine if the tools operated the same on each system and to highlight differences and challenges. After doing so the focus will be to see, on one system, how effective the tools are in handling and identifying the changes made by an example backdoor Trojan.

4.1.1. System One

System one is Windows XP installed on an Acer Aspire netbook with an Atom Processor, 2 GB of Ram and a 160 GB drive. The system was modified to a C and D partition and the default Windows installation has been supplemented by several applications and tools installed on the D drive.

4.1.2. System Two

System two is a Windows Server 2003 R2 virtual machine created with VMWare workstation running on a an HP laptop with an AMD X2 processor, 4GB memory and an 120 GB drive. The image itself uses 512 MB of memory and a 20 GB hard drive.

4.1.3. System Three

System three is Windows 7 installed on an HP Laptop with an AMD A8 processor, 6 GB of memory and a 650GB drive. The system was fresh out of the box. The only changes from the default configuration was the removal of several HP "bloatware" utilities and games and the Norton Security Suite

4.2. The Plan

On each system the antivirus was uninstalled and then a WFT system baseline was run. The next step was to install AVG antivirus. The antivirus software was chosen since it will create and interact with processes as well as creates files. Additionally, Wireshark sniffer software was installed to create more differences between the default baseline and the current one to be run. Several files were modified to see if the changes

to their hashes would be detected. For the last test, a Trojan backdoor will be installed on the Windows 2003 server and then a WFT analysis will be run to compare against the baseline to see if the changes can be detected by KDiff3. For this exercise a back door is created using the Metasploit Framework (Metasploit, 2012). Invoking the command "**msfpayload windows/meterpreter/reverse_tcp LHOST=**(ip address) **LPORT=**(port number) **X > /root/DesktopBackdoor.exe"** creates an executable that will establish connection back to a listener on the address and port listed in the creation command. This will simulate a back door connection that a Trojan typically makes. After the Trojan is executed on the target system and the connection is made, the backdoor service will be migrated into an existing Windows process and a back door service is set up. The details on how to do this can be found on the Internet (Strand, 2009), (Metasploit Unleashed 2010).

After the changes were made a second WFT run was done to create the comparative analysis. The text directories from the output of each run were then loaded into KDiff3 to do the comparative analysis.

4.3 Execution

The tools were setup using a second Windows 7 laptop. Windows Forensics Toolchest was setup first. After the user license was configured, the **wft -fetchtools** command was invoked, WFT pulled the Windows 7 tools from the system and connected to the Internet and downloaded other tools that are needed. Windows 2000 Professional, Windows XP, Vista, and Server 2003 virtual machines were used to allow WFT to extract the tools for those operating systems version. Each virtual machine was started and the usb drive plugged in to the system so the virtual OS could recognize it and then the setup command was invoked. Once all the operating system tools were copied, there were a small number of tools that WFT could not find on the operating systems or the Internet. The script identified each tool during its run and referenced the Helix CD as the alternate location. The Helix version, Helix2008R1, was important. When a different version of Helix, version 3, 2009 was used, the applicable tools were not available on the

CD. Once the correct version was downloaded, burned to a CD and accessed by WFT the rest of the applicable tools were located and installed.

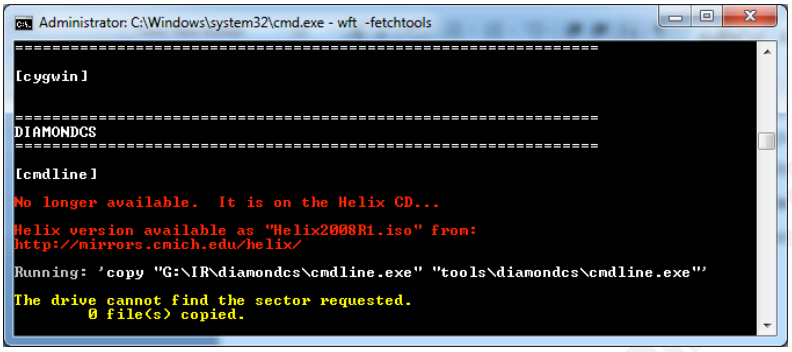

Figure 7: WFT, Helix Reference

After all tools were downloaded, the **wft –fixcfg** command was run and a few of the OS level tools were shown not installed. The tools folder was removed from the WFT folder and the **wft –fetchtools** command was rerun. Again, some OS level tools were not installed but this time it was different tools. So it appeared that the glitch was random. The missing tools were manually copied over.

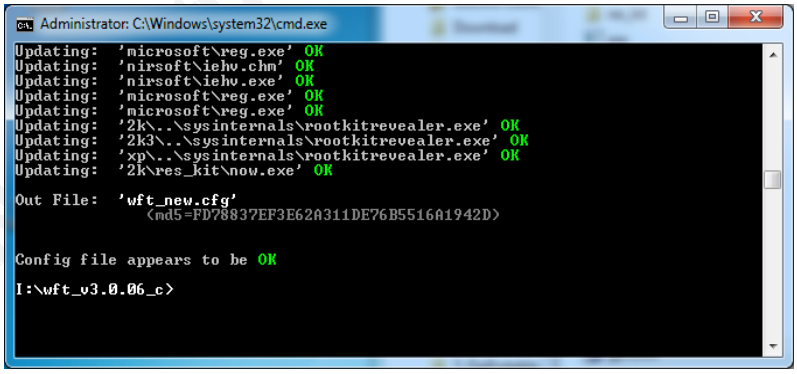

Figure 8: wft -fixcfg

One tool that the setup had a problem with was **gpresult.exe**. The tool was supposed to be extracted from the Windows 2000 instance which, it appeared, did not happen. Further research showed that it had been copied to the root of the tools folder instead of the win2k folder. Once it was moved to the correct location, WFT confirmed that it was installed. The WFT installation was copied to a USB drive. Next, KDiff3 was installed on the laptop. The install was routine with no surprises.

Author Name, email@address

4.3.1 System One Test

The first system to be baselined was the Windows XP system. While running the initial baseline, ipxproute.exe, failed to execute. It needed to access a library, winstream.dll. The dll was located in the System 32 folder and copied to the Tools\XP folder and the baseline was rerun. It then ran without errors. After installing the antivirus the current state baseline was run.

To load the baselines into KDiff3, the user must navigate to the system folder in the WFT folder and then, the date folder for the date the baseline was run. Each run for the date is located in a subfolder labeled by the time of the run. The txt folder is the folder that should be loaded and provides the cleanest view of the contents in the KDiff3 interface.

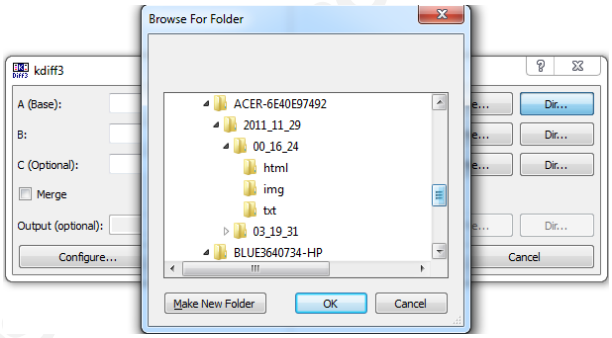

Figure 9: KDiff3, Browse to Directory

Looking at the interface there are several WFT files that have differences between the baseline and current state both routine and specific to the installations of Wireshark and AVG. Within the various output files KDiff3 highlighted numerous changes. Examples include the applog.txt output where there was evidence of the installation of AVG antivirus, attempts by Windows Update to download a root certificate update. The C_hidden file.txt output shows changes in the access times of many hidden files and directories. In the C_filestg.txt output there was a change to ntuser.dat from small letters to capital letters.

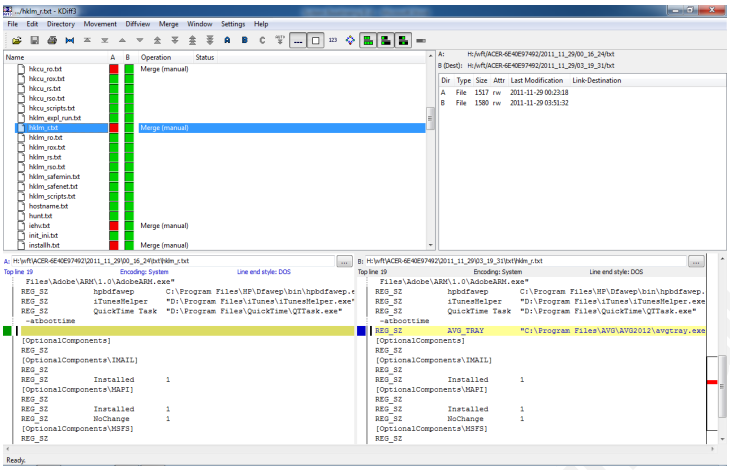

Figure 10: KDiff 3 File Comparison

Additionally, the AVG links and the Wireshark link to the Program Files folder as well as the files and folders created when AVG and Wireshark were installed. In the hklm_r.txt output, an entry was identified for the AVG Tray application that was added to the run.

4.3.2 System Two Test

The next system to be tested was the Server 2003 system. Like the XP system, the winstrm.dll had to be added to the wft\Tools\Win2k3 folder so ipxroute could run correctly. Once the baseline run was complete the antivirus and Wireshark were installed. For Server 2003 AVG antivirus would not work as it needed to connect to a management server during setup. Comodo's basic antivirus could install on Windows Server 2003 so it was installed instead.

The current state WFT run was completed and both runs were loaded into KDiff3. Again, KDiff3 highlighted the various changes between the two runs. Examples of the changes included the Comodo antivirus service that was highlighted in the autoruns.txt output, the removal of an environment variable in the Environm.txt output, and a change to the last time the Group Policy was applied in the gpusers.txt output.

It must be emphasized that the focus of these tools is data collection and initial comparative analysis. It still requires detailed analysis by the Information Technology or Security Specialist. This is illustrated in the C_ctimes.txt output. KDiff3 identified some

differences in files based on the order the information was collected by WFT. In this case, the same information was in both files but, in different locations based on the WFT collection. Hence, it was flagged by KDiff as a variation, creating a false positive.

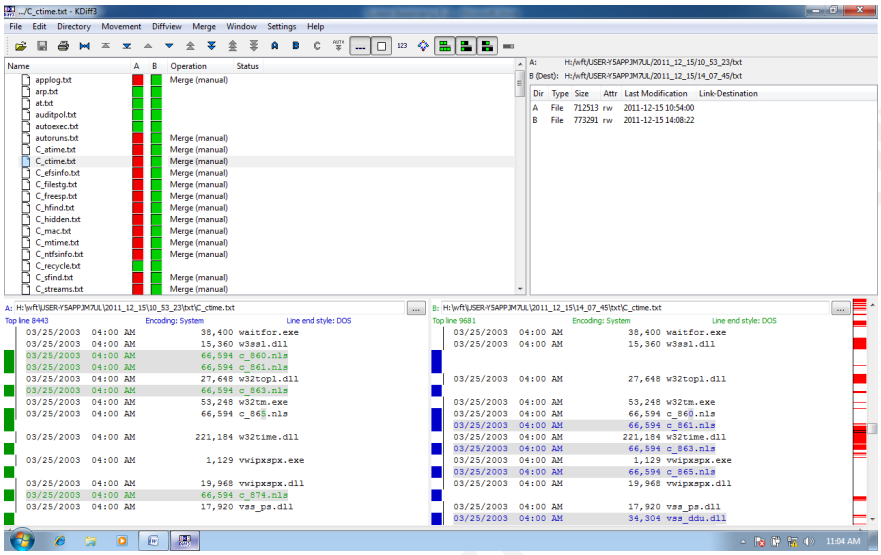

Figure 11: KDiff3 False Positives

Before starting the Trojan test a new Server 2003 virtual machine was started and a WFT baseline run was created. The Backdoor.exe was then copied to the desktop. After executing the application and verifying that the reverse connection was made, the **metsvc** command was invoked to create the backdoor service (Metasploit Unleashed, 2010). The second WFT run was created and both runs were loaded in KDiff3.

Analysis showed that activity for the both the Backdoor.exe and the metsvc were found in the output. In the netstat.txt the open ports and connection status for both backdoors were found. The dlls that metsvc process was utilizing were found in Listdlls.txt and procinterrogate.txt outputs. The disk location for Backdoor.exe was found in the C_atime and C_ctime.txt outputs. The path to the metsvc.exe, which

Author Name, email@address

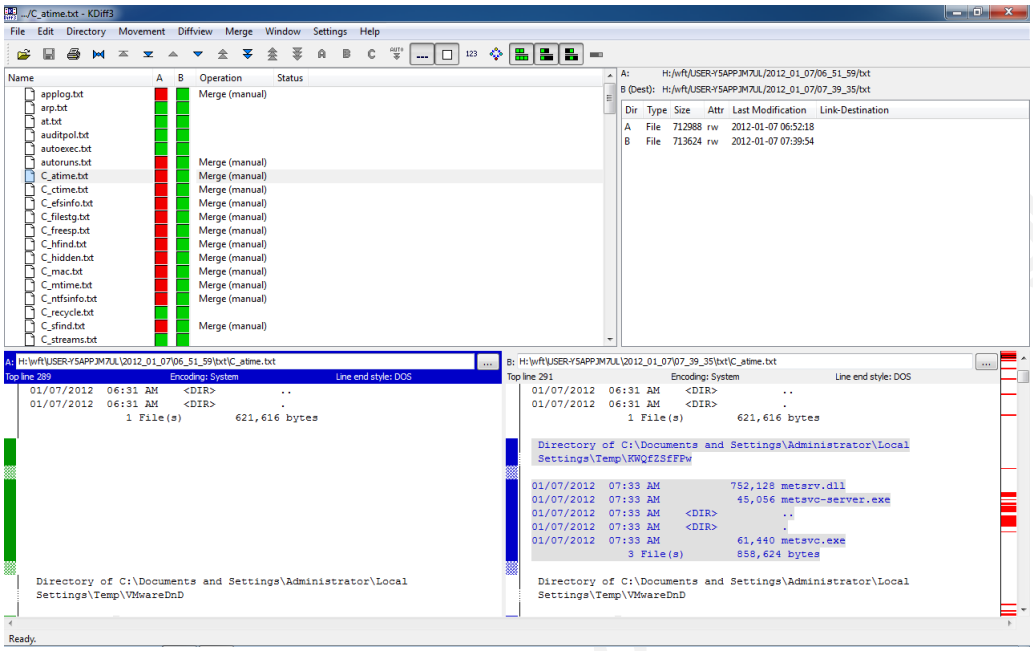

Figure 12: KDiff3 Metsvc Files

was saved to a folder in the user's \local settings\temp folder, was found in the output of several tool text files. Information for metsvc.exe and the service that it invoked was found in a couple of additional files. The results validate the ability of the two tools and the process to detect changes related backdoor Trojan like activity.

4.3.3 System Three Test

The third and last system to be tested was the Windows 7 system. As WFT began running the various tools, Windows 7 User Account Control (UAC) occasionally popped up requiring user interaction before the process would continue. Since one goal is for a minimum of user interaction during the collection process, UAC was turned off during

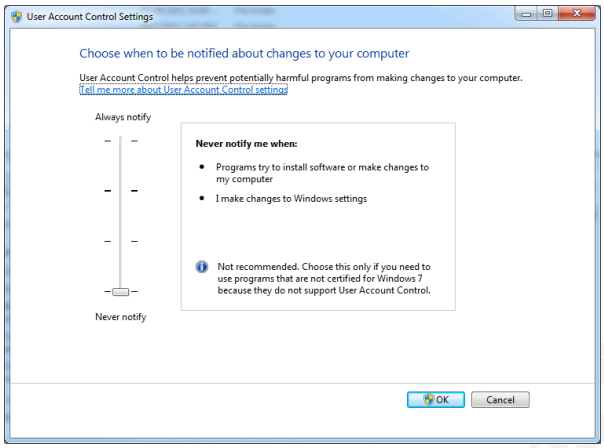

Figure 13: UAC Control Panel

The WFT data collection. This was done by going to **Control Panel/User Accounts/Change User Account Control Settings**. The popup window displayed contains a slider for controlling the UAC granularity. Sliding it to the bottom turns off the UAC control. Once UAC was turned off, WFT ran through and completed the baseline. After the baseline was completed UAC would normally be turned back on.

AVG Antivirus and Wireshark were installed and WFT was run again with UAC turned off to provide the current system state information.

During the data collection process several tools failed to operate correctly inside of WFT. Specifically, mem.exe could not run in Windows 7. Ipxroute, Pulist, streams and Rootkit Revealer stopped working right after starting.

In discussions with the author he had yet to optimize WFT to work with Windows 7. As a result, the workarounds used above were necessary and some outputs were not created or available. Since the same tools failed on the comparison WFT run there was not a potential problem with the accuracy of the results between baselines. The results from those tools were just not available for analysis.

For those tools that did work correctly the output was similar to the other two systems. Examples include the reboot after the AV installation, identified in the uptime.txt output, the changed text, dword, hex values and deleted and added information for registry keys identified in the regdump.txt output.

Author Name, email@address

For the tools that did not run or run correctly they still created output files. These files would need to be reviewed to verify their content.

4. Conclusion

As IT and security budgets become more constrained and the role of Information Technology and the associated threat continues to expand the need to streamline and automate process continues to grow. The concept of system baselining is one of those areas that is time consuming and can benefit of automation.

The use of Windows Forensics Toolchest and KDiff3 with the processes discussed can provide one way to automate the baselining and comparison process. In testing it was discovered that some tools did not run or run correctly. With some tools false positives were generated when the data collection sequence of the tool changed between the baseline and current state run. Overall, the two tools performed as expected and the issues discovered could be mitigated during the analysis. WFT and KDiff3 are not the end all in the baselining process. Detailed analysis still needs to be done on the collected data and comparison results. This must be done to address the issues mentioned and to quantify the differences discovered and determine why the differences exist. However, time can be saved in the collection of that data and initial analysis through the use of the processes presented.

Author Name, email@address

References

Adams, J. (2009, Aug 5). *What is a Baseline?* Retrieved on Aug 22, 2011 from Security Blanket Technical Blog: http://tcs-security-blanket.blogspot.com /2009/08/what-is-baseline.html

Davis, C., Schiller, M., & Wheeler, K. (2011). *IT Auditing: Using Controls to Protect Information Assets, 2nd Ed*. McGraw-Osborne Media

Eibl, J. (2011, Sept 3) KDiff3 (Version 09.9.96) Retrieved on 9 September 2011 from http://kdiff3.sourceforge.net

Fairlex.Inc, (2011). Baseline Definition. Retrieved on 12 Dec, 2011 from www.freedictionary.com/base+line

Klosterboer, L. (2008) *Implementing ITIL Configuration Management.* IBM Press

Mandia, K., Prosise, C., & Pepe, M (2003) *Incident Response & Computer Forensics 2nd Ed.* Emeryville: McGraw-Hill/Osborne Companies Inc.

Microsoft Security Baseline Analyzer (MSBA v2.2). (2010, Sept 14) Retrieved on 08 December, 2011 from http://technet.microsoft.com/en-us/security/cc184923

Metasploit Framework, (2012, January 4). Retrieved on 15, September, 2011 from http://metasploit.com/download/

McDougal, Monty (2003, Oct 6) *Forensic Analysis: Windows Forensic Toolchest (WFT)* Gold Research Paper, Sans Institute

McDougal, Monty (2011, July 23). Windows Forensics Toolchest (WFT). Retrieved on September 9, 2011, from Fool Moon Software & Security website: http://www.foolmoon.net/security/wft/

McDougal, Monty (2007, July 30).What's *New with Windows Forensics Toolchest* (WFT) v3.0. BOF presentation, Sansfire 2007. Retrieved on Aug 8, 2011 from Fool Moon Software & Security website: http://www.foolmoon.net/security/presentations

Metasploit Unleashed (2010, October 19) *Meterpreter Backdoor Service*. Retrieved on 23 December. 2011 from http://www.offensive- security.com/metasploit-unleashed/Meterpreter_Backdoor_Service

Author Name, email@address

National Security Agency (NSA) (2009). *Security Configuration Guides* retrieved on December 12, 2011 from http://www.nsa.gov/ia/mitigation_guidance/security_configuration_guides/index.shtml

Strand, John (2009). *Metasploit Meterpreter Reverse exe.* Video retrieved on 23 December, 2011 from http://vimeo.com/1975301

U.S DOT (2011). Configuration Management for Transportation Management Systems: a Primer. Retrieved on Sept 26, 2011 from www.ops.fhwa.dot/freewaymgmt/publications/cm-baseline.htm

Author Name, email@address

5 APPENDIX A

The Windows Forensics Toolkit (wft) utilizes a number of operating system and third party tools to extract system information. What follows is extracted from the wft.cfg file.

The tools are listed in their order of execution as well as the information that they are trying to extract. The extra syntax, hyperlinks and comments were removed to make the output readable. Where possible, the original syntax was retained. In some cases the format of the switches on some tools was changed from the format used in the file to a command line format to make it easier to interpret the switch being used. It is recommended that the wft.cfg file be reviewed prior to use to verify the full commands and to review the informational comments that aid in ensuring correct execution.

##########################

OS SPECIFIC FILES: *These Windows operating system files are contained in each source OS folder; 2k\XP\Vista\w2k3*

##

(folder)\cmd.exe: *Shell from a trusted system*

(folder)\mem.exe: *Displays status of programs currently loaded in memory*

(folder)\hostname.exe: *Print name of current host system*

(folder)\net.exe: *Lists domain info for the computer*

(folder)\ipconfig.exe*: show network interface configuration information*

(folder\arp.exe: *Displays entries in the Address Resolution Protocol (ARP) cache*

(folder)\route.exe: *Displays routing table information*

(folder)\netstat.exe: *Displays all connections and listening ports*

(folder)\ipxroute.exe: *Show the IPX routing tables.*

(folder)\nbtstat.exe: *Displays the NetBIOS name table of the local computer*

Author Name, email@address

(folder)\at.exe: *Shows user scheduled tasks to be performed at a later date and time*

(folder)\doskey.exe*: Displays MS-DOS command history for a system*

(folder)\tasklist.exe: *Displays a list of application(s) and associated task(s)/process*

(folder)\schtasks.exe: *Displays all scheduled tasks*

(folder)\gpresult.exe: *Displays information about how Group Policy has affected the current computer and any users who are logged on to the current computer*

#########

 # START #

##

2k\res_kit\now.exe: *Displays the current date and time to stdout with optional message*

foolmoon\fmnow.exe: *Displays the current date and time to stdout with optional message*

##########

 # MEMORY #

##

2k\..\fau\dd.exe: *Copies physical memory (or partitions) to a file*

(folder)\..\sysinternals\strings.exe: *Finds UNICODE (or ASCII) strings of a default length of 3 or more UNICODE (or ASCII) characters*

unxutils\pclip.exe*: Displays the content of the Windows clipboard*

foolmoon\fmclip.exe*: Displays the content of the Windows clipboard*

<%os%>\mem.exe mem_p Displays status of programs currently loaded in memory

<%os%>\mem.exe Displays status of programs, internal drivers, and other information

Author Name, email@address

#############

 # MAC TIME #

###

<%shell%> /C dir <%drive%>:\ /S /OD /TA: *Show last access time based file listing*

<%shell%> /C dir <%drive%>:\ /S /OD /TC: *Show last created time based file listing*

<%shell%> /C dir <%drive%>:\ /S /OD /TW > *Show last modified (written) time based file listing*

perl\mac.exe: *Retrieves file MAC times from Windows systems*

###############

 # SYSTEM INFO #

###

sysinternals\psinfo.exe: *List information about a system including disks, hotfixes, and installed software*

unxutils\uname.exe: *Identify the current system*

<%shell%> /C ver: *Show the operating system version number<*

<%shell%> /C set environm: *Displays environment variables*

foolmoon\fmuptime.exe: *Show how long system has been up*

(folder)\..\microsoft\uptime.exe: *Show how long system has been up*

sysinternals\psuptime.exe: *Shows you how long a system has been running since its last reboot*

unxutils\whoami.exe: *Display the effective current username*

(folder)\net.exe NET DOMAIN: *Lists domain info for the computer*

(folder)\net.exe NET USER: *Lists the user accounts for the computer*

Author Name, email@address

(folder)\net.exe NET GROUP: *Displays the groups for the domain*

(folder)\net.exe NET LOCALGROUP: *Displays the groups for the local computer*

(folder)\net.exe NET ACCOUNTS: *Displays the current settings for password, logon limitations, and domain information*

(folder)\net.exe NET DOMAIN ACCOUNTS: *Displays the current domain settings for password, logon limitations, and domain information*

2k\res_kit\auditpol.exe: *AUDIT POLICY: Enables the user to modify the audit policy of the local computer or of any remote computer*

##############

 # PROCESSES #

###

sysinternals\pslist.exe: *List detailed information about processes*

2k\res_kit\pulist.exe: *Displays processes running on local or remote computers*

sysinternals\listdlls.exe*: List all the DLLs that are currently loaded, their location, and version numbers*

2k\res_kit\pstat.exe: *Lists all running threads and displays their status*

diamondcs\cmdline.exe: *List running processes, the full path to the executable, and any command line parameters*

sysinternals\handle.exe: *List all processes and open handles*

winfingerprint\procinterrogate.exe: *Lists all processes, process ids, and their associated dlls*

#############

 # SERVICES #

###

sysinternals\psservice.exe: *View and control services*

Author Name, email@address

2k\res_kit\sc.exe: *List extended information for installed services*

<%os%>\net.exe NET START: *Lists running services*

netlatency\servicelist.exe: *List running services on a system*

(folder)\tasklist.exe tasklist v: *Displays a list of application(s) and associated task(s)/process(es) verbosely*

(folder)\tasklist.exe tasklist svc: *Displays a list of services(s) and associated task(s)/process(es)*

############

 # DRIVERS #

###

2k\res_kit\drivers.exe: *List information for installed drivers*

#################

 # NETWORK INFO #

###

<%os%>\ipconfig.exe: *Show network interface configuration information*

diamondcs\iplist.exe: *List all IP interfaces*

<%os%>\arp.exe: *Displays entries in the Address Resolution Protocol (ARP) cache*

<%os%>\route.exe: route print: *Displays routing table information*

<%os%>\netstat.exe netstat -a: *Displays all connections and listening ports*

<%os%>\netstat.exe netstat -an*: Displays all connections addresses and listening ports in numeric form*

foundstone\fport.exe: *Displays open ports and maps them to the associated application (sorted by port)*

(folder)\..\diamondcs\openports.exe: *Identify unknown open ports and their associated applications*

Author Name, email@address

(folder)\ipxroute.exe: *Show the IPX routing tables*

<%os%>\nbtstat.exe nbtstat -n: *Displays the NetBIOS name table of the local computer*

<%os%>\nbtstat.exe nbtstat -c: *Displays the contents of the NetBIOS name cache for remote machines*

<%os%>\nbtstat.exe nbtstat -s: *Displays NetBIOS client and server sessions*

foundstone\hunt.exe: *SMB share enumerator and admin finder*

<%os%>\net.exe NET SHARE: *Lists information about all resources being shared on the computer*

<%os%>\net.exe NET USE: *Lists the computer's connections*

<%os%>\net.exe: NET VIEW: *Lists the computers in the current domain*

<%os%>\net.exe NET SESSION*: Displays information about all client and server sessions for the local machine*

ntsecurity\promiscdetect.exe: *Checks if a local network adapter(s) is running in promiscuous mode (may indicate a sniffer)*

##########

 # LOGINS #

##

sysinternals\psloggedon.exe: *See who's logged on locally and via resource sharing*

systemtools\netusers.exe: *See who's logged on locally and via resource sharing*

systemtools\netusers.exe netusers /history*: See all users who have logged on locally and via resource sharing*

foundstone\ntlast.exe: *Show last successful logons*

foundstone\ntlast.exe ntlast -f: *Show last failed logons*

foundstone\ntlast.exe ntlast -i: *Show last interactive logons*

foundstone\ntlast.exe ntlast -r*: Show last remote logons*

Author Name, email@address

###############

 # EVENT LOGS #

###
#####

2k\res_kit\dumpel.exe dumpel -l system: *Dumps the System event log to a tab-separated text file*

2k\res_kit\dumpel.exe dumpel -l application: *Dumps the Application event log to a tab-separated text file*

2k\res_kit\dumpel.exe dumpel -l security: *Dumps the Security event log to a tab-separated text file*

sysinternals\psloglist.exe psloglist -s system*: Dump System event log records*

sysinternals\psloglist.exe psloglist -s application: *Dump application event log records*

sysinternals\psloglist.exe psloglist -s security: *Dump security event log records*

################

 # FILE SYSTEM #

###

sysinternals\ntfsinfo.exe: *Shows you information about NTFS volumes*

sysinternals\psfile.exe: *Shows files opened remotely*

<%os%>\net.exe NET FILE: *Lists the open files on a server*

<%shell%>/C tree <%drive%>:\ /F /A*: The location of every file on the system*

foundstone\hfind.exe: *Hidden file finder with last access times*

<%shell%>/C dir <%drive%>:\ /S /A:H /T:A: *Show the hidden files on a system*

sysinternals\streams.exe: *View NTFS file stream information*

foundstone\sfind.exe: *View NTFS file stream information*

Author Name, email@address

2k\res_kit\efsinfo.exe: *Shows information about EFS-encrypted files*

<%shell%> /C dir "%UserProfile%\Recent": *Lists recently opened files*

<%shell%> /C dir <%drive%>:\recycler /S /OD /TA: *Lists files in the recycle bin*

<%shell%> /C dir "%SystemRoot%\prefetch": *Used for boot and application launch prefetching on XP / 2003*

netlatency\freespace.exe: *List how much free space exists on a drive or network share*

###############

 # AUTO START #

###

sysinternals\autorunsc.exe: *Shows you what programs are configured to run during system boot up or login, and shows you the entries in the order Windows processes them*

<%shell%> /C type "%SystemDrive%\autoexec.bat": *Starts every time system boots at DOS level*

<%shell%> /C type "%SystemRoot%\win.ini": *Starts every time Windows starts (look for... Shell=)*

<%shell%> /C type "%SystemRoot%\winstart.bat": *Starts every time Windows starts (operates as normal .bat file)*

<%shell%> /C type "%SystemRoot%\wininit.ini": *Used by setup programs; if file exists, it is run once and deleted by Windows*

<%shell%> /C dir "%SystemDrive%\Documents and Settings\All Users\Start Menu\Programs\Startup": *Show applications called from "All Users" Startup folder*

<%shell%> /C dir "%UserProfile%\Start Menu\Programs\Startup": *Show applications called from current user's Startup folder*

<%shell%> /C dir "%SystemRoot%\Tasks": *Show scheduled tasks submitted via at.exe or the Scheduled Tasks Wizard*

Author Name, email@address

<%os%>\at.exe: *Shows user scheduled tasks to be performed at a later date and time<P>*

(folder)\schtasks.exe: *Displays all scheduled tasks*

microsoft\reg.exe reg query HKLM\Software\Microsoft\Windows \CurrentVersion \Run /S: *Programs to be run when system starts*

microsoft\reg.exe reg query HKLM\Software\Microsoft\Windows\CurrentVersion \RunOnce /S: *Programs to be run once when the system starts and then the entry is removed*

microsoft\reg.exe reg query HKLM\Software\Microsoft\Windows\CurrentVersion \RunOnceEx /S: *Programs to be run once when the system starts and then the entry is removed*

microsoft\reg.exe reg query HKLM\Software\Microsoft\Windows\CurrentVersion \RunServices /S: *Services to be run once when the system starts and then the entry is removed*

microsoft\reg.exe reg query HKLM\Software\Policies\Microsoft\Windows \System \Scripts /S: *scripts to be run for various events (i.e., logon, logoff, shutdown, etc.)*

microsoft\reg.exe reg query ***CHKLM\SOFTWARE\Microsoft\Windows \CurrentVersion\ Policies\Explorer \Run\ /S:*** *Programs to be run when system starts*

microsoft\reg.exe reg query HKCU\Software\Microsoft\Windows\CurrentVersion \Run /S: *Programs to be run when system starts*

microsoft\reg.exe reg query HKCU\Software\Microsoft\Windows\CurrentVersion \RunOnce /S: *Programs to be run once when the system starts and then the entry is removed*

microsoft\reg.exe reg query HKCU\Software\Microsoft\Windows\CurrentVersion \RunOnceEx /S: *Programs to be run once when the system starts and then the entry is removed*

Author Name, email@address

microsoft\reg.exe reg query HKCU\Software\Microsoft\Windows\CurrentVersion \RunServices /S: *services to be run when system starts*

microsoft\reg.exe reg query HKCU\Software\Microsoft\Windows\CurrentVersion \RunServicesOnce /S: *services to be run once when the system starts and then the entry is removed*

microsoft\reg.exe reg query HKCU\Software\Policies\Microsoft\Windows\System \Scripts /S: *scripts to be run for various events (i.e., logon, logoff, shutdown, etc.)*

microsoft\reg.exe reg query HKCU\Software\Microsoft\Windows\CurrentVersion \Policies\Explorer\Run\ /S: *Programs to be run when system starts*

#############

 # REGISTRY #

###

ntsecurity\gplist.exe: *Lists information about the applied Group Policies*

<%os%>\gpresult.exe gpresult /v /scope user: *Displays information about how Group Policy has affected the current computer and any users who are logged on to the current computer*

<%os%>\gpresult.exe gpresult /v /scope system*: Displays information about how Group Policy has affected the current computer and any users who are logged on to the current computer*

foolmoon\fmrunlist.exe: *Lists commands typed into the Start|Run dialog*

microsoft\reg.exe reg query HKCU\Software\Microsoft\Windows\CurrentVersion \Explorer\RunMRU /S: *Lists commands typed into the Start|Run dialog*

foolmoon\fmrecentdocs.exe: *Lists Recent Docs*

microsoft\reg.exe reg query HKCU\Software\Microsoft\Windows\CurrentVersion\ Explorer\RecentDocs /S: *Lists Recent Docs*

foolmoon\fmopensave.exe: *Lists files opened/saved via explorer-style dialog boxes*

Author Name, email@address

microsoft\reg.exe reg query HKCU\Software\Microsoft\Windows\CurrentVersion \Explorer\ComDlg32\OpenSaveMRU /S: *Lists files opened/saved via explorer-style dialog boxes*

foolmoon\fminstallhistory.exe: *Lists applications that have been installed (if they have a registered unistaller)*

lists applications that have been installed (if they have a registered unistaller)

microsoft\regdmp.exe: *Dumps of all or part of the registry to stdout*

microsoft\reg.exe reg query HKLM\SYSTEM\CurrentControlSet\Control\SafeBoot \Minimal\ /S: *Performs add, change, import, export and other operations on registry subkeys*

microsoft\reg.exe reg query HKLM\SYSTEM\CurrentControlSet\Control\SafeBoot \Network\ /S: *Performs add, change, import, export and other operations on registry subkeys*

################

 # IE ACTIVITY #

################

nirsoft\iehv.chm: *Displays the list of all URLs that you have visited in IE over the last few days*

foolmoon\fmtypedurls.exe: *Lists URLs typed at the Internet Explorer address bar*

microsoft\reg.exe reg query "HKCU\Software\Microsoft\Internet Explorer\TypedURLs" /S: *Lists URLs typed at the Internet Explorer address bar*

microsoft\reg.exe reg query "HKCU\Software\Microsoft\Internet Explorer\Explorer Bars\{C4EE31F3-4768-11D2-BE5C-00A0C9A83DA1}" /S*: Lists explorer search history*

Author Name, email@address

ntsecurity\pstoreview.exe: *Lists the contents of the Protected Storage. It usually contains things like Internet Explorer username and password autocomplete, and Outlook account names and passwords.*

########

 # MISC #

##

<%os%>\doskey.exe doskey /history: *Displays MS-DOS command history for a system*

perl\mdm.exe: *Checks for the existence of a modem driver*

(folder) \..\sysinternals\rootkitrevealer.exe: *Root kit detection utility*

########

 # DONE #

##

2k\res_kit\now.exe: *Displays the current date and time to stdout with optional message (end)*

foolmoon\fmnow.exe: *Displays the current date and time to stdout with optional message (end)*

Author Name, email@address

6 Appendix B

These are the various command switches available in wft to customize and control how wft does its data collection and outputs the results and they can be viewed. This information is located in the usage.txt file in the wft folder.

usage: **wft -usage**: Outputs these instructions to stdout
usage: **wft -about**: Outputs information about WFT to stdout
usage: **wft -license:** Outputs the WFT license to stdout
usage: **wft -md5 [**filename]: Outputs MD5 checksum for file filename to stdout
usage: **wft -sha1** [filename]: Outputs SHA1 checksum for file filename to stdout
usage: **wft -wfthash** [filename]: Outputs WFT checksum (MD5:SHA1) for file filename to stdout
usage: **wft -checkcfg incfgfile** [-toolpath path_to_tools]: Checks a config file for errors in format or checksum
usage: **wft -fixcfg incfgfile outcfgfile** [-toolpath path_to_tools]: Outputs a new config file with updated checksums
Note: Also updates v1.0 and v2.0 config files to the v3.0.06 format (except <%drive%> macros)
usage: **wft -genreport report_path** [-reg regfile]: Outputs WFT report for a previous WFT run
Note: The XML file wft_rpt.xml must exist in report_path
usage: **wft -update** Updates WFT and config file(s) via http
usage: **wft -fetchtools** [-toolpath path_to_tools] Downloads default WFT utilities via http\ftp
usage: **wft -browser** [browser]: Causes WFT to open the output 'index.htm' in user browser
Note: Browser defaults to system browser if not specified
Otherwise WFT executes [browser] with 'index.htm' argument
i.e. '[browser] dst\index.htm'

Author Name, email@address

No [browser] validation is performed, so a full path
is NOT required for execution
i.e. argument 'netscape' would use the default path

usage: **wft -case** [casename]: Specifies the casename of the case to be included in main page

Note: If casename has a space it will need to be in quotes for DOS
i.e. -case "Fluffy Bunny Attacks Again"

usage: **wft -color**: Causes WFT to use color in console output

usage: **wft -cfg** [cfgfile]: Uses cfgfile to determine which tools to run by WFT

Note: Cfgfile defaults to '.\wft.cfg' if not specified

usage: **wft -def** [defile]: Uses deffile to determine interactive defaults for WFT

Note: Deffile defaults to '.\wft.def' if not specified

usage: **wft -drive** [drive_letters]: Specifies the drives to be used by WFT

Note: Defaults to 'auto' which is all FIXED_DISKs

usage: **wft -dst** [destination]**:** Defines the path that WFT reports will be written to

Note: Destination defaults to '.\' directory if not specified
Destination directories will be created if they do not exist
Destination should be a remote file system or removable disk
i.e. '\\computer\share\directory\'
Destination can include command-line macros
i.e. $magic$ = expands to '$systemname$\$date$\$time$'
$systemname$ = SYSTEM NAME of the current computer
$date$ = current DATE in the format 'YYYY_MM_DD'
$time$ = current TIME in the format 'HH_MM_SS'

usage: **wft -hash** [hash]: Specifies the hash to be used by WFT

Note: Hash defaults to 'md5' if not specified
Supported hash values are 'md5', 'sha1', and 'none'

usage: **wft -interactive**: Causes WFT to run interactively

Note: Any additional command-line arguments become defaults

Author Name, email@address

usage: **wft -name** [investigator]: Specifies the name of the investigator to be included in reports

Note: If name has a space it will need to be in quotes for DOS

i.e. -name "Monty McDougal"

usage: **wft -nocolor**: Causes WFT not to use color in console output

usage: **wft -nodefault**: Causes WFT not to use default file

usage: **wft -nointeractive**: Causes WFT to not run interactively

Note: This overrides WFT defaults

usage: **wft -noprunetools**: Causes WFT not to prune tools list to remove tools skipped based on OS

Note: This overrides WFT defaults

usage: **wft -noreport**: Causes WFT not to create HTML (H) reports

Note: This overrides WFT defaults

It also overrides the -browser option if it is also specified

usage: **wft -noslow**: Causes WFT not to run slow (S) executables in cfgfile

Note: This overrides WFT defaults

usage: **wft -nowrite**: Causes WFT not to run executables that write to the source machine

Note: This overrides WFT defaults

usage: **wft -os** [host_os]: Specifies the OS string used by WFT for OS specific functions

Note: OS defaults to 'auto' for host_os auto detection

OS 'host' will use untrusted host paths / binaries

i.e. -toolpath for OS commands becomes 'C:\WINNT\system32\'

usage: **wft -prompt**: Causes WFT to prompt about running prompt (P) executables

Note: Argument -prompt does not override -noslow or -nowrite options

usage: **wft -prunetools**: Causes WFT to prune tools list to remove tools skipped based on OS

usage: **wft -reg** [regfile]: Uses regfile to determine which WFT functions are available

Note: Regfile defaults to '.\wft.reg' if not specified

usage: wft -report: Causes WFT to create HTML (H) reports

Author Name, email@address

usage: wft -shell cmdshell:: Redefines shell references from '<%os%>\cmd.exe' to cmdshell

Note: Cmdshell defaults to 'cmd.exe' for the specified '-os'

Cmdshell 'host' will use untrusted system shell

i.e. 'host' becomes 'C:\WINNT\system32\cmd.exe'

usage: **wft -slow**: Causes WFT to run slow (S) executables in cfgfile

usage: **wft -toolpath** [path_to_tools]: Defines the path where wft tools are stored

Note: Path_to_tools defaults to '.\' directory if not specified

Path_to_tools can be a remote file system or removable disk

i.e. '\\computer\share\directory\'

usage: **wft –write**: Causes WFT to run executables that write to the source machine

Author Name, email@address

7 Appendix C

The following question asked by wft when run in interactive mode. They cover a range of configuration settings that will commonly addressed when using wft for forensic analysis. Most of the questions can be useful in other applications of the software.

The first seven questions deal with setting up environment variables.

WFT uses a registration file to enable certain features during data collection. What is the path and filename for the registration file you would like to use?

Answer [<user specified file>] (Default='wft_reg.xml'):

_WFT supports the ability to specify an alternate toolpath which will be the base directory for all tools specified in the WFT config file.

What is the toolpath you would like to use?

Answer [<user specified path>] (Default='tools'):

WFT utilizes OS specific commands for tool execution. These should be known good binaries that match the OS WFT is collecting data from (this is important).

'host' will use the host's SYSTEM directory

'auto' will use the toolpath OS directory

What is the OS path for the commands you would like to use?

Answer [<user specified host_os>] (Default='auto'):

WFT utilizes the command shell (cmd.exe) for tool execution. This should be known good binary that matches the OS WFT is collecting data from (this is important).

'host' will use the host's cmd.exe

'auto' will use the toolpath OS directory's cmd.exe

What is the path and filename for the command shell you would like to use?

Author Name, email@address

Answer [<user specified shell>] (Default='auto'):

WFT supports the ability to specify the output destination path which can include dynamically generated macros at run time. Destination directories will be created if they do not exist. Destination should be a remote file system or removable disk.

i.e. '\\computer\share\directory\'

Destinations can also include command-line macros.

i.e. $magic$ = expands to '$systemname$\$date$\$time$'

$systemname$ = SYSTEM NAME of the current computer

$date$ = current DATE in the format 'YYYY_MM_DD'

$time$ = current TIME in the format 'HH_MM_SS'

What is the destination path you would like to use?

Answer [<user specified path>] (Default='$magic$'):

WFT supports the ability to run commands using a dynamically provided list of drives.

'auto' will include all FIXED_DISKs

Drives can also be a list of drive letters

i.e. 'CEF' would use drives C, E, and F

What drive(s) would you like to use?

Answer [<user specified drives>] (Default='auto'):

The next two questions involve configuring case information for forensics. In incident response, this may be important if the goal is to use the information for forensics evaluation relevant to a potential or actual criminal case.

WFT supports the ability to specify an investigator name which will be included as part of logs / reports.

What is the investigator name you would like to use?

Answer [<user specified investigator>] (Default='N/A'):

WFT supports the ability to specify a casename which will be included as part of logs / reports.

Author Name, email@address

What is the casename you would like to use?

Answer [<user specified casename>] (Default='N/A'):

WFT will run a checksum on every file it touches. The next question allows for choosing which hash format to use.

WFT can use either MD5 or SHA1 checksums for file integrity

Which hash format would you like to use?

Answer [MD5/SHA1/NONE] (Default='md5'):

Some tools can take a long time to run. Running 'slow' tools can take an hour or more to finish.

Do you want to run tools that are slow?

Answer [Y/N] (Default='Y'):

Some tools can alter the system they are being run on. This includes some tools in the default WFT config file. While these tools make only minimal impacts, you must understand they are modifying the state of the source system!

Do you want to run tools that can write to the source system?

Answer [Y/N] (Default='Y'):

WFT will produce HTML reports in addition to text reports.

Do you want to use HTML reporting?

Answer [Y/N] (Default='Y'):

WFT can prompt before running certain commands at run time.

Do you want to enable prompting?

Answer [Y/N] (Default='N'):

WFT supports the ability to open the output HTML report in an external browser at report completion.

Do you want to automatically open the completed report?

Answer [Y/N] (Default='N'):

Author Name, email@address

You are about to run the equivalent of:

wft.exe -case N/A -cfg wft.cfg -drive auto -dst $magic$ -hash md5 -name N/A –os auto -prunetools -reg wft_reg.xml -shell auto -toolpath tools

Is that OK?

Answer [Y/S/R/Q] (Default='Y', 'S' to Save, 'R' to Restart, 'Q' to Quit):

Author Name, email@address