Global Information Assurance Certification Paper

GIAC Certified Forensic Analyst (GCFA)
Practical Assignment Version 1.5 (April 30th, 2004)

By Rob Ferrill

Table of Contents

SUMMARY

The purpose of this paper is to fulfill the requirement of the GIAC (Global Information Assurance Certification) program for the GCFA (Certified Forensic Analyst) certification. Throughout the paper I will demonstrate the knowledge and skills learned from the SANS classroom setting of Track 8: System Forensics, Investigation & Response.

The paper will be laid out into two parts. The first part is a scenario in which a company (Ballard Industries) suspects that their proprietary manufacturing designs have been stolen by a competitor (Rift, Inc.) who now is seemingly producing the exact same product and selling it to most of Ballard Industries former customers. After a full investigation ensues, the only possible evidence found in this case is a floppy disk that was removed from the briefcase of Robert Kyle Leszczynski while leaving the Research and Development labs of Ballard Industries. The security administrator asked me to review the floppy disk and provide a report of my findings. He also supplied chain of custody information for the disk.

In part two of this paper, an actual investigation from a real world case as analyzed by an information security analyst will be demonstrated. This case involved a disgruntled employee with a large amount of IT technical knowledge. This employee had developed a web-based application internal to one of the branch offices of ABC Corporation in order to automate some of the manual tasks at this office. When the employee feared that his termination was possible, it is suspected that he developed a logic bomb in his application to delete the entire application after he had not logged in for 14 days. This information was gleaned from a former coworker of his. All the steps taken to investigate this matter will be detailed in this section.

PART 1 – ANALYZE AN UNKNOWN IMAGE

Examination Details

The image obtained for this investigation was from a floppy disk seized by the on-staff security guard who is employed by Ballard Industries. The security administrator for Ballard Industries gave the following chain of custody details for the floppy disk.

- Tag# fl-260404-RJL1
- 3.5 inch TDK floppy disk
- MD5: d7641eb4da871d980adbe4d371eda2ad fl-260404-RJL1.img
- fl-260404-RJL1.img.gz

Upon obtaining the image from the SANS Institute for analysis, several steps were performed to verify the integrity of the image. I created a case file in

Autopsy, which is an open source forensics tool and part of the Sleuth Kit. It can be downloaded from <http://www.sleuthkit.org/autopsy/download.php>. Once you have created a case, you can add an image to be used for the case. When adding an image, you can supply the expected MD5 checksum and then have Autopsy verify the checksums match by choosing the checkbox that says "Verify MD5 After Importing?". The following MD5 sum was verified by Autopsy which validated that the image had not changed:

d7641eb4da871d980adbe4d371eda2ad *v1_5.gz

Even though the file name (v1_5.gz) is not the same as what was supplied in the chain of custody details (fl-260404-RJL1.img), you can see here that the MD5 sum derived by Autopsy is identical to the MD5 sum in the chain of custody details. This validates that the contents of the two files are identical and therefore the evidence has not changed in transit. Later in this paper, you will see that I used a windows program called md5deep to generate MD5 checksums that can be found at <http://md5deep.sourceforge.net/>. This program adheres to RFC 1321, which is the official MD5 specification. As we learned in Track 8, a program called MD5sum (which runs on Unix and Windows platforms) generates cryptographic hashes or strings of characters that are considered to be a signature of a file. This signature is calculated from the contents of the file plus some mathematic computations that create a 16-byte long string. We also learned that the MD5sum signature is repeatable, non-reversible and unique to each file, which makes it ideal to compare original files with copies and validate the integrity of the copy.

The following steps show my analysis of the floppy image in chronological order:

1. Initial steps were to download the image from the SANS website onto my RedHat forensics laptop. When conducting a forensic investigation of unknown files or binaries, the first thing I like to do in looking for clues is to run the strings utility which will show me any plain text that can be derived from the binary. So as I stated earlier about creating a case file in Autopsy, once you've imported the image you want to analyze, you can then click on the section titled "Keyword Search". Once in the keyword search screen, you can click the button to extract the strings of the allocated space on the image into a text file (which I did and the file was called v1_5.gz.str). By clicking another button on this screen, you can also create a file containing all the unallocated space (any clusters not currently in use by a file) on the image (which I did and the file was called v1_5.gz.dls), and then extract the strings from that into a text file (which I did and the file was called v1_5.gz.dls.str). You can then view both of these files from a command line by using your favorite text viewing command line tool in Unix. I like to use the more command so it doesn't scroll off the page (i.e. "more v1_5.gz.str" – this would scroll the output of this file to the bottom of page 1 and wait for you to move down a page or a line at a time).

When you get into the file analysis portion of the Autopsy program, you can display the strings of any file or see a report of the strings which will also give you other info about the file (i.e. the MD5 output, the directory entry number, the size, whether it's allocated or unallocated, the directory entry times (MAC times), and the sectors it takes up). I like to see the strings from a file because you can get a basic understanding of their functionality.

2. After viewing the strings output on the command line by typing "more v1_5.gz.str" and "more v1_5.gz.dls.str", I could tell that the image of the floppy disk contained what appeared to be several Microsoft Word documents. These documents appeared to be IT policy files relating to various topics like password usage, remote access, acceptable encryption, etc. The strings output also showed references to a program called Camouflage that were found at "C:\My Documents\VB Programs\Camouflage\Shell" so I immediately went to Google and did a search for "camouflage". One of the first search hits was a paper on the SANS website about a free steganography program called Camouflage. This led to a plethora of reading on many websites about steganography because I had not fully researched this area of information security and information hiding techniques.

3. I also performed a search on Google for one of the files seen in the strings output (lctxMenu.tlb), which turned out to be a Type Library. This did not lead me to any more relevant clues concerning steganography but I wanted to show that I researched this and other filenames that were indicated in the strings output to try and determine exactly what other clues I could find on this disk.

4. After reading about the Camouflage program from the SANS document, I suspected that the apparent policy documents on the floppy image actually contained hidden files with the use of steganography. In the strings output, it appeared the Word documents are similar in length (as far as number of pages) and all appear to be plaintext. Consideration will be given as to the possibility of installing Camouflage in order to extract the hidden files. Autopsy will be used to cut the files out individually. The image file appeared to be a gzip file but when trying to gunzip it I received an error that this file was not in gzip format. When the Unix 'file' command is run on the gzip file (i.e. "file v1_5.gz") to determine if it's truly a compressed file with the GNU utility gunzip, it states the following:

v1_5.gz: x86 boot sector, code offset 0x3c, OEM-ID " mkdosfs", root entries 224, sectors 2872 (volumes <=32 MB) , sectors/FAT 9, serial number 0x408bed14, label: "RJL ", FAT (12 bit)

When you run the file command on a normal gzip file, your output will say "gzip compressed data".

5. At this point the decision was made to have a closer look at the gzip image file with Autopsy. This revealed the 6 policy documents and two deleted files. The

first deleted file appears to be an HTTP document called index.htm and the second deleted file appears to be identical to the first but it has the name of CamShell.dll. These deleted files are proven to be identical later in my analysis when I show the MD5sum's of each file.

6. From another Google search, I found and downloaded the Camouflage software from <http://camouflage.unfiction.com/Camou121.exe>. After running it through my virus scanner, I installed this program on a Windows 2000 workstation for analysis since Camouflage is a Windows program. The Camouflage program is very easy to use. You simply right click on a file in Windows Explorer and you can choose from that menu to either Camouflage or Uncamouflage a file.

7. All 6 policy documents and 2 deleted files were extracted to the hard drive of my forensics laptop. ASCII output of some of the policy documents reveal lots of binary or encrypted data appended to the end of the policy (as it said it would do in the Camouflage documentation).

8. Running the un-camouflage functionality of the Camouflage program against all these files revealed one file without a Camouflage password (intern~2.doc). This revealed a file called "opportunity.txt" and within this file was the evidence of Mr. Leszczynski's intent to sell proprietary information belonging to Ballard Industries to a competitor for 5 million dollars. The Camouflage application also indicated that these files were created with Camouflage v1.2.1.

9. At this point, I ran into a quagmire because I realized there were at least two of the other documents camouflaged but I could not figure out the riddle in the file opportunity.txt as to the Camouflage password for these files. This conclusion about the other documents being camouflaged came from the fact that these Microsoft Word policy documents were all fairly similar in size (about 3 pages) but two of these documents were considerably larger in the number of bytes. For example, the Internal Lab Security Policy document was 3 pages in length and the file size was 33kb. However, the Password Policy was also 3 pages in length but was 301kb in size and all text (no graphics). The Remote Access Policy document was also 3 pages long but was 211kb in size. The following shows the contents of this file (Opportunity.txt):

I am willing to provide you with more information for a price. I have included a sample of our Client Authorized Table database. I have also provided you with our latest schematics not yet available. They are available as we discussed - "First Name".
My price is 5 million.
Robert J. Leszczynski

My gut reaction to this riddle was to try every possible combination of the word Robert as a password for the other camouflaged files as this was his "first name". This turned out to be futile as none my attempts worked.

10. This resulted in going back to the Internet once more to run a Google search looking for more information on how the Camouflage program really worked. There had to be a way to find out the password for these Camouflaged files without actually going to the painstaking effort of brute forcing my way into it. I needed to find how and where the passwords were actually stored in the Camouflaged file and/or what type of encryption might be used. I found a presentation done at BlackHat '04 (<http://www.blackhat.com/presentations/bh-usa-04/bh-us-04-raggo/bh-us-04-raggo-up.pdf>) on steganalysis by Michael T. Raggo - Principal Security Consultant for Verisign. In the presentation, he shows a case study on Camouflage and indicates how to find and decrypt the password for Camouflaged files. He told about a tool called BDHTool (Binary Decimal, Hexidecimal Tool) that can be used for XOR'ing the password.

This tool has quite a few other functions as seen at the website where I found it (<http://www.stormpages.com/bdhtool/>). This site states “LOGIC OPERATORS V1.2 is an 8-bit binary, decimal, hexadecimal conversion program. It can also perform the following logical operations on two 8-bit (bin, dec, hex) values: AND, NAND, OR, NOR, XOR, XNOR, NOT. Also, a truth table for each of the logical operators is displayed when the AND, NAND, OR, NOR, XOR, XNOR or NOT buttons are pressed. A basic 4-function (+, -, /, x) RPN calculator is also included for simple mathematical computations.”

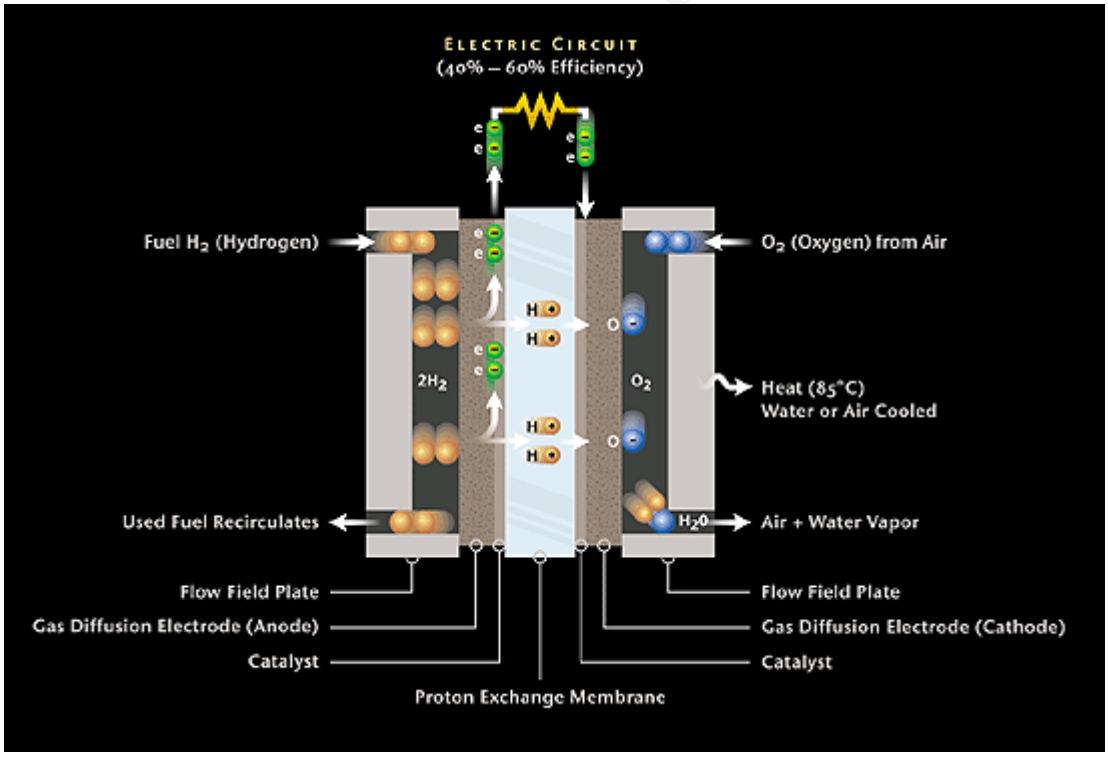
Credit was given to Guillermito at <http://www.guillermito2.net> for reverse engineering the Camouflage program to discover its password storing mechanisms.

11. From Guillermito’s website, he explained the full details on how the passwords are stored in Camouflaged files and how to retrieve that password. He also wrote his own program in assembly language to retrieve the password from Camouflaged files. Once I downloaded this program and ran it through my virus scanner for safety measures, I also looked at the actual assembly file that you can compile yourself. This *.asm file has notes at the beginning stating how easily the encryption is broken on a Camouflaged file so he states his programming is very simple in this assembly program he created.

12. I then used Guillermito’s program to crack the password to the two files I suspected were Camouflaged. The hint in the file opportunity.txt as to the password was "First Name". Initially I was trying every combination of “Robert” that I could think of to use as the password for these Camouflaged files. It turns out that after Guillermito’s program revealed the passwords to me, they turned out to be the first word in the name of the document. So for example, in the file called Remote Access Policy.doc, the password to uncamouflage this file was Remote (and it was case sensitive). The other Camouflaged file was called Password Policy.doc and hence the password to uncamouflage it was Password.

It appears that Mr. Leszczynski is the primary suspect in leaking proprietary company information as seen by the loss of business by Ballard Industries and the replication of their fuel cell batteries by Rift, Inc. Without a much more involved investigation of Mr. Leszczynski's personal computer and work computer, it would be impossible to prove beyond a shadow of a doubt that he is the guilty party. He could have been framed but initially all the evidence points to him.

The files that were uncamouflaged from the floppy disk in Mr. Leszczynski's briefcase also show evidence of the proprietary company info being leaked to competetors. The original file called Password Policy.doc actually contained 3 other hidden files. One of them was a file called Hydrocarbon%20fuel%20cell%20page2.jpg that contained schematics and formulas on the design of the fuel cell. Another of the three hidden files was called pem_fuelcell.gif, which is another drawing of how the fuel cell works. The last of the three hidden files was called PEM-fuel-cell-large.jpg, which is another drawing of the design of PEM fuel cells. Here's the picture last of these three files for posterity.



The original file called Remote_Access_Policy.doc contained one extra file when uncamouflaged. Its name was CAT.mdb and this was a Microsoft Access database that contained a table with the authorized clients of Ballard Industries. Here's the table converted to MS Excel for posterity.

First	Last	Phone	Company	Address	Address1	City	State	Zipcode	Account	Password
Bob	Esposito	703-233-2048	Cook Labs	245 Main St		Alexandria	VA	20231	Espomain	y4NSHMNf
Jerry	Jackson	410-677-7223	Double J's	11561 W. 27 St.		Baltimore	MD	20278	Jack27st	JLbW3Pq5
David	Lee	866-554-0922	Tech Vision	300 Lone Grove Lane		Wichita	KS	30189	Leetechv	O1A26a3k
Marie	Horton	800-234-king	King Labs, Inc.	700 King Labs Ave	Suite 900	Biloxi	MS	39533	Hortking	Yk7Sr4pA
Lenny	Jones	877-Get-done	Quick Printing	99 E. Grand View Dr		Omaha	NE	56098	Joneeast	868y48RH
Jeff	Hayes	404-893-5521	Big Sky First	90 Old Saw Mill Rd		Billings	MT	59332	Hayeolds	3R30bb7i
Roger	Forrester	210-586-2312	TCFL	188 Greenville Rd		Austin	TX	77239	Forrgree	si4OW8UV
Edward	Cash	212-562-0997	E & C Inc.	76 S. King St	Suite 300	Santa Barbara	CA	80124	Cashking	Of8uQ1fC
Steve	Bei	616-833-0129	Island Labs	65 Kiwi Way		Honolulu	HA	93991	Beikiwiw	JDH20u26
Jodie	Kelly		Data Movers	7256 Beerwah Ave.	Suite 110	Wetherby	U.K.	LS22 6RG	Kellbeer	tmu0ENOk
Patrick	Roy		The Magic Lamp	4150 Regents Park	Row #170	Calgary	CAN	R4316DF	Roythema	rJag6Q0O

It should also be noted here that the file creation times of the Camouflaged files, as shown by the timeline analysis later in this document, were around 9:46 am on April 26th 2004 which is the morning of the same day that the security guard seized the floppy disk from Mr. Leszczynski. This would provide further investigative specifics to enable forensic specialists to narrow down the timeframe of searches through other computers within Ballard Industries when looking for this Camouflage program.

As for advice to the system administrators at Ballard Industries, I would first state that my professional opinion is that without a doubt they have had a major criminal incident occur and that they need to follow their incident response procedures to the letter so that any evidence gathered would be submissible in court (especially chain of custody documentation). If they are not up to speed on how to maintain a chain of custody I would educate them but apparently they are from the information received from David Keen (Security Administrator). I would also advise them to get law enforcement involved immediately now that sufficient evidence has been gathered regarding illegal activity.

As for advice to the system administrators to detect whether or not Mr. Leszczynski tampered with their systems, I would recommend several steps:

- Gather all available logs (event logs from workstations and servers, firewall logs, IDS logs, etc.) and look for anything surrounding his userid and/or the IP address of his workstation or computers he was known to use. The IDS logs might have caught any malicious types of activity he might have been engaged in. The firewall logs would show what systems he has accessed outside of Ballard Industries and would also show what websites he might have visited for researching his scheme. Event logs from workstations and servers might show when he accessed these systems either remotely or at the console. They might also prove which files he accessed and show whether or not he had access rights to these. It's possible that he didn't have access rights to these files and would therefore have to use someone else's credentials to get the access or use some type of system vulnerability to escalate his privileges on the system.

This could also potentially leave traces behind for investigators to use in prosecuting him.

- Do a thorough forensic investigation of his workstation to look for evidence in this case. This would require obtaining a forensic image (a bit for bit copy) of his original hard drive and poring over it with a variety of forensic tools. The original drive would need to be locked up in a safe with limited access and include documented chain of custody information so that this could be used in court in his trial. Obvious things to look for in this case are keyword searches for the filenames on the floppy disk (the policy documents and especially the proprietary files). Using data recovery techniques (tools such as Encase, DLS from the Sleuthkit, Lazarus from The Coroner's Toolkit, Foremost from <http://www.sourceforge.net/projects/foremost>) you would want to try and recover any deleted files from his machine that might be used against him. If his machine is a Windows workstation, you will want to look through the unallocated space on his drive for INFO2 files. These files are records that are kept by the "Recycle Bin" that show file names that are placed there. Even if an investigator cannot recover a certain deleted file, the INFO2 file may show that it was in fact in the "Recycle Bin" at one point proving it existed on his computer. There is also the possibility of recovering file fragments with the various recovery tools previously mentioned. This would again provide investigators with evidence that this was on the system at one point and could provide further direction or implication of the suspect.
- Recover any old emails that he might have sent or received while on the job at Ballard. In this case you would want to look for the existence of several file types if this is a Windows computer. If he used Outlook Express for his email correspondence you will want to look for *.DBX files as these are used by this program. If he used Microsoft Outlook, you would want to look for *.PST or *.OST files which are Personal Folders and Offline Stores of Outlook email. If by chance he was using a web mail program to send and receive email you would want to search through his temporary Internet files for evidence of this. It would also be good to go through his Internet history to find all web sites he had visited.
- Gather some snapshot details of all running processes on critical servers as well as any open ports that are listening for a connection. You can use tools like pslist (from the PSTOOLS programs) to list running processes and fport (from <www.foundstone.com/resources/proddesc/fport.htm>) to show open ports and which programs have these ports open. You could also do a netstat –a to show all ports that are listening on a system. Depending on his technical savvy, he could have gone to a variety of lengths to hide his actions and / or maintain future access to systems. For example, he could have installed backdoors on systems, planted trojan horse applications for further access to systems or even installed root kits to hide certain processes and system information from the sysadmins

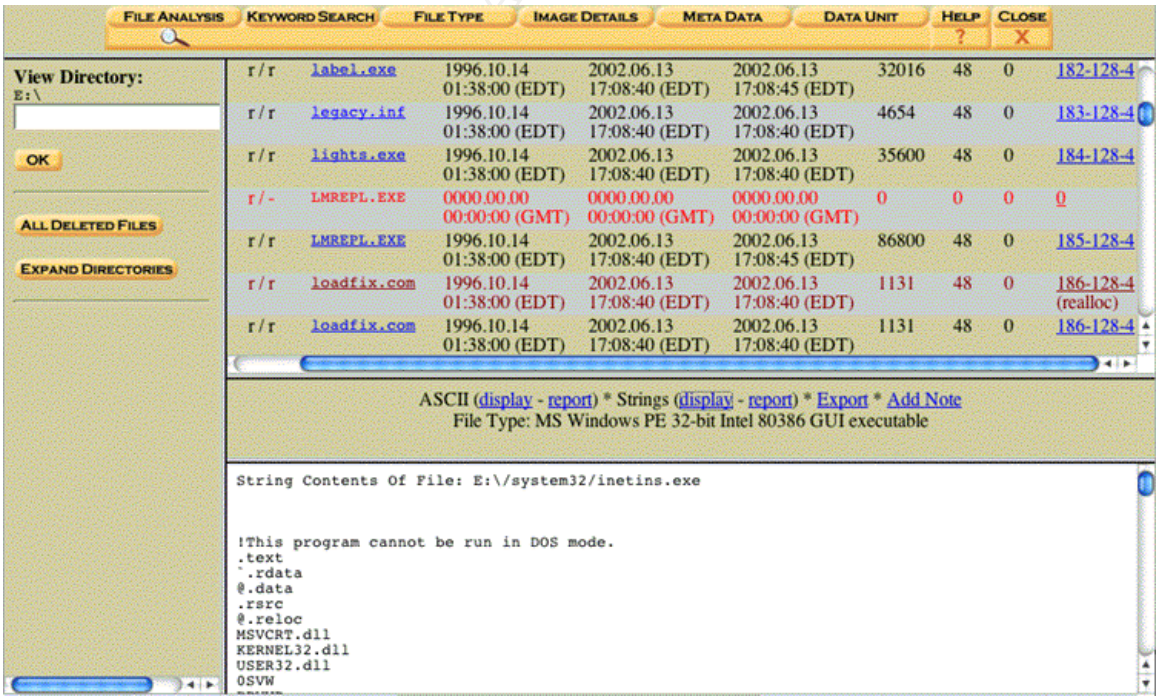
running those servers. I would use tools to check for root kits on all Ballard Industries computers. There is a tool that can be downloaded from <http://www.chkrootkit.com> that will check for known root kits. If he didn't want to go to the trouble of hiding things with a root kit, he may have installed a Trojan horse or backdoor on one or more systems that would show up in the list of processes. I would run an up-to-date virus scanner on all systems to look for known malware. The snapshot of running processes and open ports would enable system administrators to try and determine if abnormalities exist on these machines that should not be there.

- Sweep all systems within Ballard Industries looking for the images and documents that were stolen to get a feel for where he might have gotten them from and to see if he left any evidence behind. There would be multiple ways to accomplish this but one way would be to map network drives to all the machines and conduct searches for the file names. As you will see in the timeline analysis later, there is no evidence of the uncamouflaged files on the floppy disk. This indicates that the actual Camouflage process took place on another system and then the files were copied to the floppy disk. Another potential method to sweep computer systems looking for certain files would entail creating md5 hashes of all files on all computers at Ballard Industries in order to search the resulting hashes for known files. One open source tool that automates this process is a program called ftimes and it can be found at <http://ftimes.sourceforge.net/>. From the info on this website, we see that ftimes is really considered a system baselining and evidence collection tool designed to support content integrity monitoring, incident response, intrusion analysis, and computer forensics. This tool would effectively allow the system administrators at Ballard Industries to determine which systems (Enterprise-wide) would have potential evidence for this case by using one of it's two general capabilities (file topography and string search). With file topography, you are basically mapping out the file system on the machine and with string search, you are looking for a string of text or bytes in sequence.
- Sweep all Windows computers on the Ballard network to look in the registry for Camouflage keys. Even after being uninstalled, the registry is still littered with entries from the installation of Camouflage. For example, one of the many keys that still existed after the program was uninstalled was \HKEY_CURRENT_USER\Software\Camouflage. This again could be accomplished with the ftimes forensic utility just described. The existence of Camouflage being installed on any system or the existence of any left over registry keys after Camouflage was uninstalled would indicate a potential system that the suspect was using for his dirty work. This could potentially indicate that someone else may have been helping him as an accomplice and the normal every day users of these systems would need to be questioned thoroughly.

- Search all computers within Ballard Industries for network or packet sniffers. This would be something Mr. Leszczynski may have employed to capture network passwords for elevating his privileges on the network. There are many tools on the Internet available to do this and can be found by simply searching the web for "discover sniffers". One such tool discovered was called Anti-Sniff that can be found at <http://packetstormsecurity.nl/sniffers/antisniff/>. You can also manually look for any Ethernet network interface cards on Unix systems that are in promiscuous mode. This would show the PROMISC variable when running a simple "ifconfig" command on a Unix system.

Image Details

- Below is a listing of all the files that were retrieved from the floppy disk image (using Autopsy) including the uncamouflaged files (which are in subdirectories). In the file analysis section of Autopsy, you can click on the hyperlink of any file name and it will display the ASCII contents of that file in the bottom window. Also in that bottom window you can click on a link to display the strings or to export the file. This is how these files were exported and saved to a floppy disk for analysis on my Windows 2000 workstation. Below is a screen shot of the file analysis screen in Autopsy that I got from <http://www.sleuthkit.org/autopsy/images/file1.gif>.



- Since the Camouflage program was installed on my Windows 2000 workstation, the directory listings below are from this computer after I copied the files over from my Linux forensics workstation. The two files

that begin with an underscore were the deleted files from the disk and a review of their contents shows that they are very similar at the beginning of the file but the camshell.dll file appears to have a lot of binary and encrypted data at the end of this file. After extracting it with Autopsy and trying to run the Camouflage password crack program, it does not appear that this is a Camouflaged file. Below are the files:

Win2k Forensic Workstation - Directory of C:\My Documents\SANS\GCFW\Evidence Files

7/28/2004	02:52p	22,528	accept~1.doc
7/28/2004	02:52p	42,496	inform~1.doc
7/28/2004	02:52p	32,256	intern~1.doc
7/28/2004	02:52p	33,423	intern~2.doc
7/28/2004	02:52p	307,935	passwo~1.doc
7/28/2004	02:49p	215,895	remote~1.doc
9/17/2004	02:43p	<DIR>	Uncamouflaged
7/28/2004	02:49p	36,864	_amshell.dll
7/28/2004	02:49p	727	_ndex.htm

Win2k Forensic Workstation - Directory of C:\My Documents\SANS\GCFW\Evidence Files\Uncamouflaged

9/17/2004	02:42p	<DIR>	Internal_Lab_Security_Policy
9/17/2004	02:42p	<DIR>	Password_Policy
9/17/2004	02:43p	<DIR>	Remote_Access_Policy

Win2k Forensic Workstation - Directory of C:\My Documents\SANS\GCFW\Evidence Files\Uncamouflaged\Internal_Lab_Security_Policy

4/22/2004	05:31p	32,256	Internal_Lab_Security_Policy.doc
4/23/2004	03:03p	312	Opportunity.txt

Win2k Forensic Workstation - Directory of C:\My Documents\SANS\GCFW\Evidence Files\Uncamouflaged\Password_Policy

4/23/2004	11:21a	208,127	Hydrocarbon%20fuel%20cell%20page2.jpg
4/23/2004	12:55p	39,936	Password_Policy.doc
4/23/2004	11:23a	28,167	PEM-fuel-cell-large.jpg
4/23/2004	11:15a	30,264	pem_fuelcell.gif

Win2k Forensic Workstation - Directory of C:\My Documents\SANS\GCFW\Evidence Files\Uncamouflaged\Remote_Access_Policy

4/23/2004	12:21p	184,320	CAT.mdb
4/23/2004	12:54p	30,720	Remote_Access_Policy.doc

- The true name of the program used by Mr. Leszczynski to hide his criminal activities was called Camouflage (a steganography program used for hiding files inside of other files).
- Below is the File/MACTime information from image which was obtained through the Autopsy program by clicking on the “File Activity Time Lines” button after you import the image into the case. Once you have done this, the next screen requires you to create a data file for your timeline analysis

and you have to select the image you want to collect data from. In this case, there is only one image so you select it. You also have the option to choose the data types you want to gather (Allocated Files, Unallocated Files, and Unallocated Meta Data Structures. You must also choose the name of the output file and whether or not you want to generate an MD5 value for this file. Once you've created a data file, you use this for the next screen where you create a timeline. On this screen, you have to select the data file to use, the starting and ending dates for your timeline, the file name for the output, and whether or not to generate an MD5 value. You may then view your timeline either through Autopsy or your favorite text editor. The Unix editor "vi" is recommended.

Since we know that Autopsy is just a graphical front end to The Sleuth Kit (TSK) and other common tools, I wanted to make sure you understood this if you wanted to do it from the command line. The first step above to create the data file is done with the "fls" and "ils" commands from TSK and these are concatenated together. Then, this file is used as input when running the mactime tool to actually create the timeline. Notice in the results below, any deleted files have (deleted) at the end.

```
Sat Feb 03 2001 19:44:16      36864 m..  -rwxrwxrwx     0  0   5  <v1_5.gz-_AMSHELL.DLL-dead-5>
                              36864 m..  -/-rwxrwxrwx   0  0   5  a:\/CamShell.dll (_AMSHELL.DLL) (deleted)
Thu Apr 22 2004 16:31:06      33423 m..  -/-rwxrwxrwx   0  0  17  a:\/Internal_Lab_Security_Policy.doc (INTERN~2.DOC)
                              32256 m..  -/-rwxrwxrwx   0  0  13  a:\/Internal_Lab_Security_Policy1.doc (INTERN~1.DOC)
Fri Apr 23 2004 10:53:56        727 m..  -/-rwxrwxrwx   0  0  28  a:\/_ndex.htm (deleted)
                                727 m..  -rwxrwxrwx    0  0  28  <v1_5.gz-_ndex.htm-dead-28>
Fri Apr 23 2004 11:54:32     215895 m..  -/-rwxrwxrwx   0  0  23  a:\/Remote_Access_Policy.doc (REMOTE~1.DOC)
Fri Apr 23 2004 11:55:26     307935 m..  -/-rwxrwxrwx   0  0  20  a:\/Password_Policy.doc (PASSWO~1.DOC)
Fri Apr 23 2004 14:10:50      22528 m..  -/-rwxrwxrwx   0  0  27  a:\/Acceptable_Encryption_Policy.doc (ACCEPT~1.DOC)
Fri Apr 23 2004 14:11:10      42496 m..  -/-rwxrwxrwx   0  0   9  a:\/Information_Sensitivity_Policy.doc (INFORM~1.DOC)
Mon Apr 26 2004 00:00:00     215895 .a.  -/-rwxrwxrwx   0  0  23  a:\/Remote_Access_Policy.doc (REMOTE~1.DOC)
                                727 .a.  -/-rwxrwxrwx   0  0  28  a:\/_ndex.htm (deleted)
                             307935 .a.  -/-rwxrwxrwx   0  0  20  a:\/Password_Policy.doc (PASSWO~1.DOC)
                              33423 .a.  -/-rwxrwxrwx   0  0  17  a:\/Internal_Lab_Security_Policy.doc (INTERN~2.DOC)
                              42496 .a.  -/-rwxrwxrwx   0  0   9  a:\/Information_Sensitivity_Policy.doc (INFORM~1.DOC)
                              36864 .a.  -rwxrwxrwx     0  0   5  <v1_5.gz-_AMSHELL.DLL-dead-5>
                              32256 .a.  -/-rwxrwxrwx   0  0  13  a:\/Internal_Lab_Security_Policy1.doc (INTERN~1.DOC)
                              22528 .a.  -/-rwxrwxrwx   0  0  27  a:\/Acceptable_Encryption_Policy.doc (ACCEPT~1.DOC)
                                727 .a.  -rwxrwxrwx     0  0  28  <v1_5.gz-_ndex.htm-dead-28>
                              36864 .a.  -/-rwxrwxrwx   0  0   5  a:\/CamShell.dll (_AMSHELL.DLL) (deleted)
Mon Apr 26 2004 09:46:18      36864 ..c  -rwxrwxrwx     0  0   5  <v1_5.gz-_AMSHELL.DLL-dead-5>
                              36864 ..c  -/-rwxrwxrwx   0  0   5  a:\/CamShell.dll (_AMSHELL.DLL) (deleted)
Mon Apr 26 2004 09:46:20      42496 ..c  -/-rwxrwxrwx   0  0   9  a:\/Information_Sensitivity_Policy.doc (INFORM~1.DOC)
Mon Apr 26 2004 09:46:22      32256 ..c  -/-rwxrwxrwx   0  0  13  a:\/Internal_Lab_Security_Policy1.doc (INTERN~1.DOC)
Mon Apr 26 2004 09:46:24      33423 ..c  -/-rwxrwxrwx   0  0  17  a:\/Internal_Lab_Security_Policy.doc (INTERN~2.DOC)
Mon Apr 26 2004 09:46:26     307935 ..c  -/-rwxrwxrwx   0  0  20  a:\/Password_Policy.doc (PASSWO~1.DOC)
Mon Apr 26 2004 09:46:36     215895 ..c  -/-rwxrwxrwx   0  0  23  a:\/Remote_Access_Policy.doc (REMOTE~1.DOC)
```

```
Mon Apr 26 2004 09:46:44     22528..c   -/-rwxrwxrwx     0  0  27  a:\/Acceptable_Encryption_Policy.doc (ACCEPT~1.DOC)
Mon Apr 26 2004 09:47:36       727..c   -/-rwxrwxrwx     0  0  28  a:\/_ndex.htm (deleted)
                               727..c   -rwxrwxrwx       0  0  28  <v1_5.gz-_ndex.htm-dead-28>
```

- The owner and group info for all these files are zero's (0) which means that root is the owner and root is the group for all these files. Since floppy disks can not actually contain a full operating system, the owner and group (UID and GID) information would come from the operating system where the files were created. UID and GID are valid for the Unix environment but do not apply to the Windows environment. The sizes of all the files were listed above in the file listing. Another interesting observation in regards to the timeline analysis is that the file permissions on all these files appears to by 777 which is typically non-standard. I could not find anywhere in the Camouflage documentation that this is something the program actually does or if these permissions were set this way intentionally by the suspect.
- Below are the MD5 checksums of the files recovered from the image. You can see from the checksums that the first two files (the deleted files) are actually identical.

```
219f86a8ac9a33990f50c281462d689a  _amshell.dll
219f86a8ac9a33990f50c281462d689a  _ndex.htm
f785ba1d99888e68f45dabeddb0b4541  accept~1.doc
99c5dec518b142bd945e8d7d2fad2004  inform~1.doc
e0c43ef38884662f5f27d93098e1c607  intern~1.doc
b9387272b11aea86b60a487fbdc1b336  intern~2.doc
ac34c6177ebdcaf4adc41f0e181be1bc  passwo~1.doc
5b38d1ac1f94285db2d2246d28fd07e8  remote~1.doc
```

As for the uncamouflaged files, their MD5 checksums are as follows:

```
e0c43ef38884662f5f27d93098e1c607  Internal_Lab_Security_Policy.doc
3ebd8382a19c88c1d276645035e97ce9  Opportunity.txt
9da5d4c42fdf7a979ef5f09d33c0a444  Hydrocarbon%20fuel%20cell%20page2.jpg
e5066b0fb7b91add563a400f042766e4  Password_Policy.doc
5e39dcc44acccdca7bba0c15c6901c43  PEM-fuel-cell-large.jpg
864e397c2f38ccfb778f348817f98b91  pem_fuelcell.gif
c3a869ff6b71c7be3eb06b6635c864b1  CAT.mdb
2afb005271a93d44b6a8489dc4635c1c  Remote_Access_Policy.doc
```

- One of the key words discovered from the strings output that led to the discovery of the application being used are as follows:
    - 'CamouflageShell' which led me directly to the following website with a Google search (<http://www.sans.org/rr/papers/20/762.pdf>).

This was a paper done by Kyle Bartlett for his SANS GSEC certification titled “The Ease of Steganography and Camouflage”. This paper actually says you can get the software from <www.camouflagesoftware.com> but this site no longer exists.

- Another interesting keyword was found in the deleted index.htm file which was “ballard.swf”. A google search for this file lead me to <www.ballard.com> which has this shock wave flash on it’s Corporate website. This led me to ponder the question of whether or not Mr. Leszczynski might be using Ballard’s website to pass this information over to the competitor.
- Another interesting keyword found in the strings output was “vba6.dll” and many other references to Visual Basic type files. My thoughts on this are that since Camouflage is a Windows based program that it was written in Visual Basic (at least the screens and windows in the program).

Forensic Details

The name of the program used in this case is called Camouflage. This is a steganography program that is used for hiding files inside of other files to avoid detection. Based on the timeline analysis done in the Autopsy forensic program, it appears that the last time this program was used was Monday Apr 26th 2004 at 09:46:44. This is when the file Acceptable_Encryption_Policy.doc (Accept~1.doc) was created and this is the last file created.

Traditional steganography programs were limited to specific types of files that could be used to hide other files. For example, image files (JPG, GIF, etc.) were used to hide hidden text messages within the image itself. Camouflage however, takes any file you want to hide, scrambles it, and appends it to the end of any other file. The file you appended it to looks and behaves just like a normal file. However, if someone appended a hidden file to a text file with Camouflage, they would not be very smart because the binary data at the end of the file would be seen in Notepad (or whatever the default text editor is).

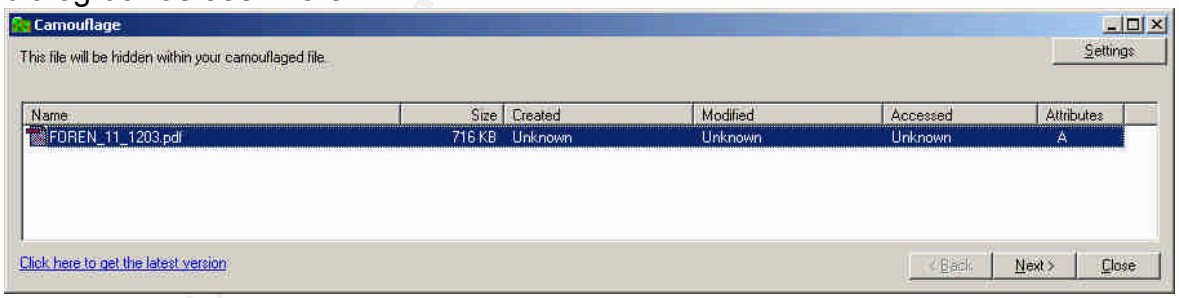
For example, let’s examine the recovered file Remote_Access_Policy.doc from the floppy disk. It looks and acts just like a normal Microsoft Word document. However, when you open this file in a text editor like Notepad or a Hex editor, you will see quite a bit of binary data at the end of this document. That is because the file CAT.mdb was hidden within this Word document using the Camouflage program. CAT.mdb is a Microsoft Access database containing the customer list for Ballard Industries. When uncamouflaged, the two files co-exist naturally and both work in their respective associated applications.

This case is slightly different from an investigation where you would be looking at an entire hard drive. There are typical procedures to follow for forensic methodologies but a lot of these do not apply to this case. For example, the

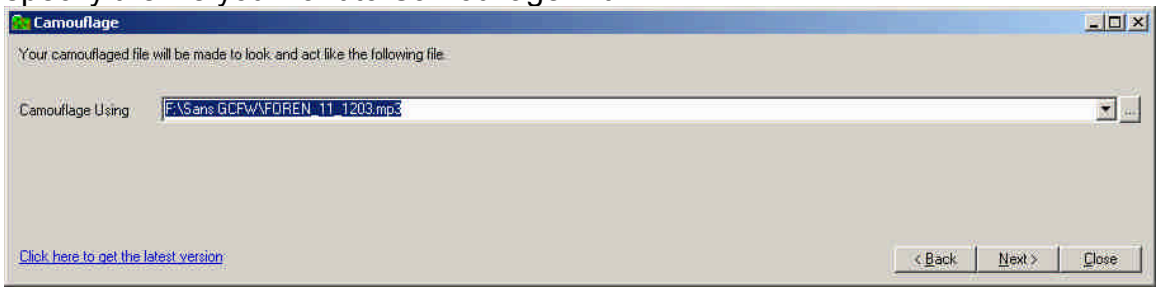
security guard at Ballard Industries that seized the floppy disk already handled the chain of custody information for us. However, we would still need to maintain documentation that will show what we did with the floppy while it was in our possession (preferably a locked cabinet or safe with limited access). Some of the traditional procedures would still apply and we did employ these in our investigation: string search, data recovery, and timeline creation… We could definitely form a dirty word list containing at least the file names involved in the information hiding process. Also, we made a forensic image (bit for bit copy) of this floppy disk for analysis because you never want to take the chance of altering the original data. Therefore, forensic analysis is typically done on forensic copies of the suspect media. We must always be ready to prove in a court of law that the original suspect drive was accessed in a "read-only" manner and therefore not written to and possibly tainted. File integrity is maintained within Autopsy under the "Image Integrity" section. When accessing this portion of Autopsy, you will see all the MD5 hashes you have generated within your case, and a button next to them that says validate. Clicking this button will compare the original MD5 sum with the current MD5 sum. This comparison proves the files have not changed during the forensic process.

The strings output led me directly to the Camouflage program and which was downloaded from <http://camouflage.unfiction.com/Camou121.exe>. Using MD5 checksums to compare the original image with the copy ensures the integrity of the original evidence.

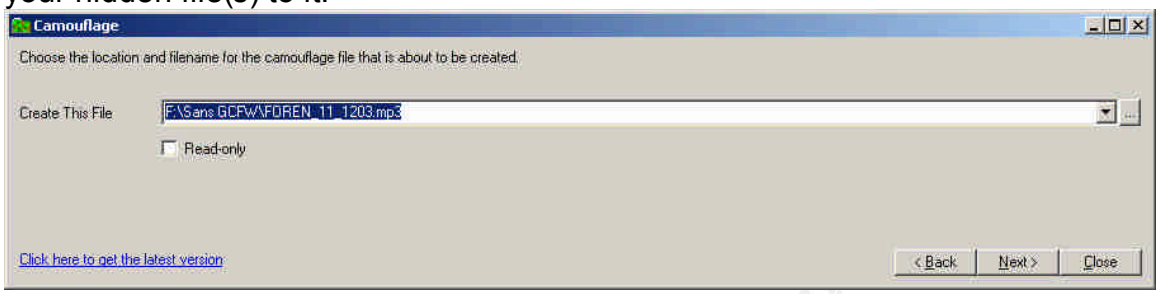
After Camouflage is installed, you will have two new options when you right click on any file in Windows Explorer (camouflage and uncamouflage). Once you right click on the file in Windows and choose camouflage, it brings up a Camouflage dialog box as seen here.



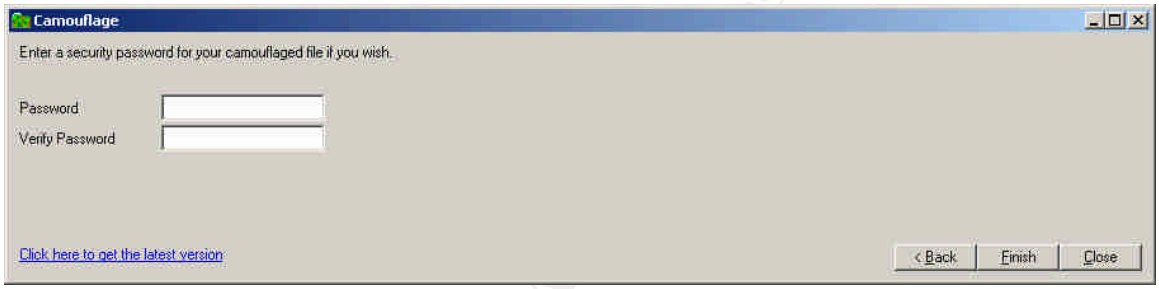
After hitting the next button you will get another window that asks for you to specify the file you want to Camouflage with.



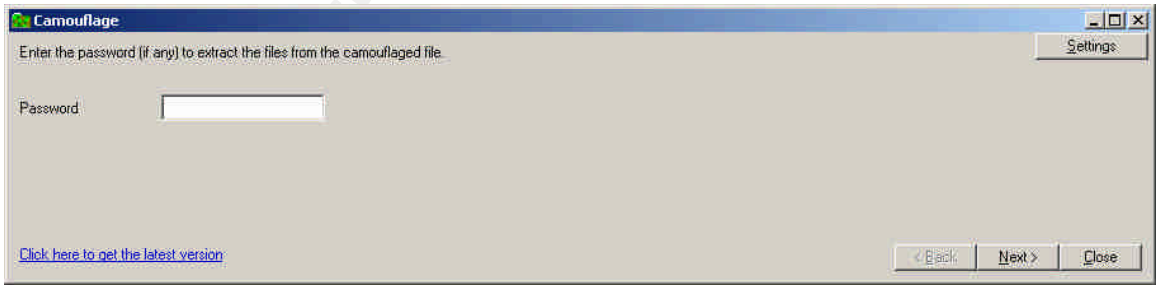
After that you hit next again and it asks you where to store the Camouflaged file. You must choose a different filename here than the original file. Camouflage does not modify the original file at all. It takes a copy of the original and appends your hidden file(s) to it.



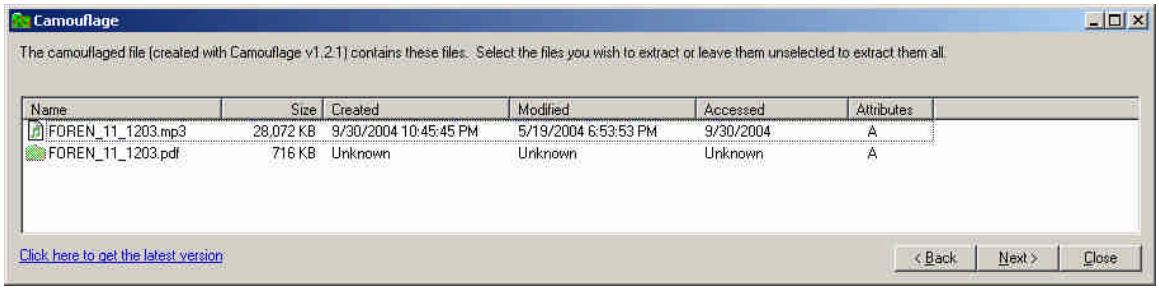
It then asks you if you want to specify a password for your newly Camouflaged file.



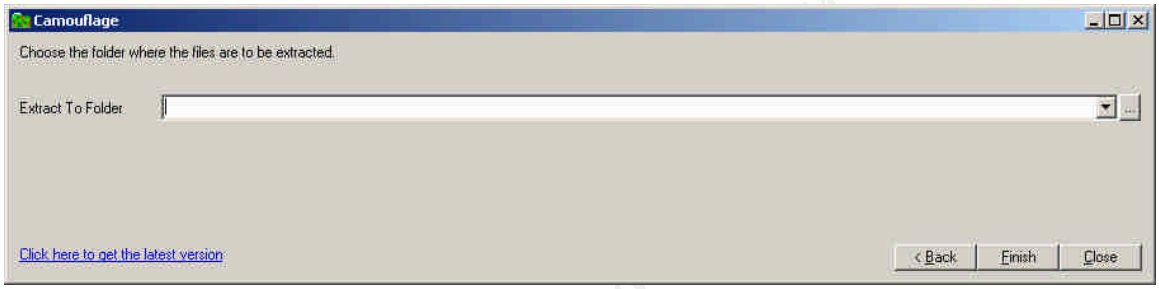
Once you click finish, it generates your new file for you with the hidden file inside. In this example, I chose an MP3 file to mask the PDF file. I executed the newly created MP3 file after Camouflage was finished doing its thing. The next thing I want to show is how to uncamouflage the file I just Camouflaged. If you right click on the file you Camouflaged, you can choose the option to uncamouflage and you will get the following screen.



Since there was no password on the file I created, I just hit next and got the following screen.



You can see above the two files that were used in the Camouflage process. Once I hit next here I get another window that asks where I want to extract this to as you see below.



Once I specify the path to extract these, I hit finish and the two files are extracted to the location I chose.

Program Identification

Since this program is no longer supported and the original website for Camouflage (<www.camouflagesoftware.com>) is not operational, the source code for this software does not appear to be accessible. Attempts were made to contact Steganography experts regarding this program and I was told that the source code for this program could not be found on any publicly accessible websites. From the other clues on the disk (camouflageshell keyword, etc.) that lead us to the Camouflage program to begin with, I was able to search the Internet and find this program to analyze it. This program was downloaded from the Internet at <http://camouflage.unfiction.com/Camou121.exe> and the MD5 sum generated for this executable on my Win2k forensics laptop is *9f08258a80d578a0f1cc38fe4c2aebb5*.

Legal Implications

There is little doubt after revealing the evidence found on the floppy disk in Mr. Leszczynski's briefcase that he appears to be the responsible party for the theft of Ballard Industries proprietary manufacturing designs. With the discovery of steganography usage, the intent to hide something was proven. With the un-hiding of the files that were hidden with Camouflage, we saw the file

Opportunity.txt that stated his apparent intent to share a client database and several proprietary schematics of Ballard's technology for producing fuel cells.

The investigation would require a much more intensive onsite inspection of company computers, physical access logs, video surveillance logs, etc. to prove that he is the responsible person. However, if convictions came solely from circumstantial evidence, he'd likely be found guilty today. It is impossible to prove which system Camouflage was actually executed on without investigating other computers at Ballard Industries. It's possible that Mr. Leszczynski could have used a personal laptop to Camouflage the files, which would mean there is no trace of this application on Ballard computers. I would suggest that if he has a personal laptop that it be seized as potential evidence and examined with the forensic steps I've mentioned earlier. I would request from law enforcement that his home be searched for evidence in this case and that Ballard Industries terminate his employment immediately.

By the fact that we were able to uncamouflage the hidden files, we know that this Steganography software was executed with intent to hide files. We cannot prove where the Camouflage program was installed at this point without further investigation. However, we do know from our research of Camouflage that the uninstall program is not very good at cleaning up after itself. As stated in the examination details previously, Camouflage leaves behind registry keys that would be easily identifiable unless Mr. Leszczynski knew to delete these.

As to the laws that apply to this case, there are several Federal laws that may apply. The case scenario does not state whether or not Ballard Industries had a patent on their fuel cell design but if they did, there would be obvious patent laws that apply. The law that becomes the front-runner for prosecuting this case would be the Economic Espionage Act of 1996 that criminalizes the theft of trade secrets. In this case, Federal prosecutors would want to pursue prosecution of Mr. Leszczynski under the US Code Title 18, Part 1, Chapter 90, section 1832 – Theft of Trade Secrets. To summarize the code that can be found at <http://www4.law.cornell.edu/uscode/18/1832.html>, anyone with intent to steal a trade secret for the benefit of anyone other than the owner will be eligible for fines and imprisonment up to 10 years. This also applies to organizations that commit this crime and the law states they will not be fined more than $5,000,000. In this case, Rift Inc. would be looked at here for prosecution under this law (Title 18, Part 1, Chapter 90, section 1832 (b)) for conspiring to steal the trade secrets of Ballard Industries. If Mr. Leszczynski was indeed proven the guilty party in this case for leaving Ballard Industries property with the proprietary fuel cell designs on a floppy disk in his briefcase, he would be eligible for this above stated imprisonment and fines for stealing the design of the fuel cells according to section 1832 (a)(1) of the above stated US Code.

If per se, Ballard Industries was headquartered in the state of Alabama, which is where I reside, there would potentially be state laws that the suspect would or

could be subjected to as well if found guilty of this crime. The state law of Alabama Code. §§ 8-27-1 et seq titled "The Alabama Trade Secrets Act" indicates a monetary restitution due to the theft or misappropriation of trade secrets but does not indicate jail time. This information can be found at <http://www.legislature.state.al.us/CodeofAlabama/1975/coatoc.htm>.

All organizations should have a section in their Acceptable Use Policy stipulating that the use of Steganography programs on any company computer is strictly prohibited.

Additional Information

The following links on the Internet were used as references to various articles on Steganography in general as well as the Camouflage program itself.

http://www.jjtc.com/Steganalysis/
http://www.securityfocus.com/infocus/1684
http://www.guillermito2.net/stegano/camouflage/index.html
http://www.blackhat.com/presentations/bh-usa-04/bh-us-04-raggo/bh-us-04-raggo-up.pdf

PART 2 – (OPTION 1) PERFORM FORENSIC ANALYSIS ON A SYSTEM

Synopsis of Case Facts

In this section of my paper I will provide details of an actual investigation that took place at an organization where I've worked.

Upon notification of potential employee sabotage of a home grown internal web application, myself and another IT Security staff member went to the branch office where the incident occurred. To preserve the confidentiality of the business, I will use fictitious names in the details of this investigation. The branch office notified us of allegations that former employee Kyle Williams had deleted files and or an entire application that he had designed. Kyle's employment had been recently terminated after a series of behavioral issues and policy violations. Kyle may have seen the writing on the wall as to his impending termination after several disciplinary actions and HR write-ups for his personnel file.

A friend and former co-worker of Kyle's informed local HR personnel at the branch office that Kyle had bragged that he had programmatically designed the custom web applications' files to be deleted after he had not been logged into the system for 14 days. Kyle had written several internal web applications to automate some of the day to day financial / operational processes for this office. The corporate IT department could in no way support home grown web

applications for each branch office because there are too many offices around the country and they are too limited on staff members as well. This is the reason that the corporate office mandates which software applications can be run at each office. Because there are multiple branch offices in all 50 states and no local IT staff, there was no way for the corporate office to know that Kyle Williams had written a rogue web application for this purpose. We learned after interviewing the local staff of this office that they had become quite dependent on this application for day-to-day operations and now had to conduct time sensitive tasks via paper again because the application Mr. Williams designed did not work anymore.

Upon arrival to the scene, we met with a gentleman by the name of Mark who was the local Office Manager for this facility. Mark directed us to three computers to focus our investigation on. The first was a Dell desktop running Windows 2000 Professional in the PBX/Phone room of the office. This system also served as the HVAC control system for the building. The second was a Compaq IPAQ running Windows 2000 Professional in the maintenance room that is actually where the web application was running before it quit functioning. The third is a Compaq desktop running Windows XP Professional that was the computer used daily by Kyle Williams and was located in the purchasing area of this office. This is the system that we will do our forensic analysis on. We were able to obtain evidence from the other two systems but do not have a forensic copy of the hard drives because they could not be taken offline long enough due to their criticality.

We immediately received local support from another local employee named Charlie who was very helpful in showing us the location of all the computers in the building. Using a forensic utility called Encase Forensic Edition from Guidance Software, we made a forensic (bit for bit) hard disk image of the and gathered evidence from the other two systems identified. It should be stated here that the use of Encase for submitting evidence in court cases has been challenged and validated numerous times and in numerous state courts. The following link will provide evidence to this <http://www.guidancesoftware.com/corporate/legal/index.shtm>.

We also did a complete inventory of the computers at this office and gathered some volatile evidence from the systems. We noted that Kyle Williams had created local administrator accounts on almost every computer in this building by the name of williamsj (which is against corporate policy). The data we gathered off of each computer included fport and netstat information on open ports, ipconfig information to list the network configuration, and pslist information to list all the processes running on each system. These tools were run from a floppy disk and their output was directed to the same floppy to prevent any data from being written to potentially suspect hard drives (although I do understand that there is still the potential here to modify certain access times of files accessed when running these commands from a floppy). We also noted each system that

had a modem installed and any hardware or software installed other than the standard PC configuration for the company. We requested telephone logs for the office to see if there was any indication of these systems being accessed through a modem by Mr. Williams after his termination. To date we have still not received the phone logs we requested.

It needs to be understood at this point in the paper that the most pertinent evidence to this case has been found on the Compaq IPAQ running the web server in the maintenance department and this is not the system we were able to obtain the forensic image from. The evidence gathered from the Encase analysis of this system while we were briefly able to access it proved to be the most pertinent to the case. It will be shown in the timeline analysis later that files were deleted on the system on 11/2/03 and the application was reported as inoperable on 11/3/03. It will be shown that the Windows Scripting Host called an Active-X control shortly before the web server files were deleted which could potentially be the method used for this deletion. It will be shown that the application and server logs from Windows 2000 appear to be inaccurate on the date / time stamps on and around 11/2/2003 indicating that the time may have been changed on the server or the event logs have been tampered with (modified).

Description of the Systems Being Analyzed

As previously stated, there were three systems investigated for this case. The system that we were able to seize and obtain the forensic image of was a Compaq Deskpro system used by Kyle Williams in the purchasing department of this office. This was the system that Kyle used on a daily basis for everyday work. Because of Kyle's self-taught IT skills, he was heavily relied upon by the local staff for IT support even though that was not his job. Therefore, he most likely worked on every computer in this building at one point or another. This that Kyle used every day was running Windows XP Professional.

Below are some of the system details obtained by Encase directly from the Windows registry and put into report format by a script within Encase. These scripts within Encase are called Enscripts and are developed by an open source community of Encase users. This particular Enscript is called "Initialize case" and will give the following information from the registry (Windows version and registration, Windows time zone settings, Windows networking information, Windows user information, Windows last shutdown time, Windows installed software, Windows hardware information, Windows Services, Permanently Mapped Drives, and Shared Folders). This script will also identify AOL information and AIM users. It can also identify all files with a given extension that you specify. To execute this script in Encase, you go to the tab labeled Scripts and double click on the Initialize Case script. Once you have double clicked it, you will then see the "Run" button light up so you can run the script. Below is the output from this "initialize case" EnScript being run against this drive that Mr. Williams used on a daily basis along with some descriptions of their meaning in parentheses. I will show N/A in the parentheses if these are not relevant.

Windows Version and Registration Information

- InstallDate: 09/11/03 03:55:09PM
- ProductName: Microsoft Windows XP
- RegisteredOrganization: ABC Corporation
- RegisteredOwner: ABC
- CurrentVersion: 5.1
- CurrentBuildNumber: 2600
- SystemRoot: C:\WINDOWS
- SourcePath: E:\I386
- PathName: C:\WINDOWS
- ProductId: 55274-xxx-xxxxxxx-23451
- CSDVersion: Service Pack 1

Windows Time Zone Settings and Active Time Bias

- Current control set is 001.
- Default control set is 001.
- Failed control set is 000.
- LastKnownGood control set is 002.
- Standard time bias is 5 hours offset from GMT.
- StandardName: Eastern Standard Time
- Standard time is set to change the Standard bias by 0 minutes.
- Standard time is set to change on Sunday of the 5th week of October, at 02:00 hours.
- DaylightName: Eastern Daylight Time
- Daylight savings is set to change the Standard bias by -60 minutes.
- Daylight savings time is set to change on Sunday of the 1st week of April, at 02:00 hours.
- Active time bias is 5 hours offset from GMT.
- The current time setting is 5 hours offset from GMT.
- The offset must be either added or subtracted from GMT depending on the time zone location.

Windows Network Settings

3Com 3C920B-EMB Integrated Fast Ethernet Controller

- IPAddress: 0.0.0.0 *(N/A)*
- SubnetMask: 0.0.0.0
- DefaultGateway:
- NameServer:
- DhcpIPAddress: 10.xxx.240.114 *(last IP assigned to this system)*

- DhcpSubnetMask: 255.255.255.0
- DhcpServer: 10.xxx.240.1
- Time lease was obtained: 11/07/03 09:18:11PM
- Time lease terminates: 11/10/03 09:18:11PM
- IPAutoconfigurationAddress: 0.0.0.0 *(N/A)*
- IPAutoconfigurationMask: 255.255.0.0 *(N/A)*
- The computer account name is "CPQUSXXXXXXXX"
- The primary domain name is "ABC"

Windows last shutdown time reported by the registry

- 11/07/03 09:37:06PM

Hardware

The one hard drive seized in this investigation came from a Compaq Deskpro system with an IBM Deskstar 40GB internal hard drive. This desktop computer also had the following components: internal 3.5" high density floppy drive, internal 100mb Iomega Zip drive, built-in Intel Pro/100 Ethernet adapter, integrated nVidia nForce AGP video adapter, and SoundMAX integrated digital audio. It was running Windows XP SP1 as seen in the previous section.

The hard drive was locked in a file cabinet in my office in which I am the only key holder. The hard drive for this case was assigned the following data for chain of custody tracking and a sticker was placed on the drive with this information as an evidence tag:

Case #	Tag #	Manufacturer	Model	Model #	Capacity	Serial #
0015	0015-01	IBM	Deskstar	IC35L040AVER07-0	41.0GB	TXN81411

Image Media

In this investigation I booted the suspect computer off of an Encase boot CD, which is a proprietary Dos-based image, provided by Guidance Software that will allow image acquisition in a read only mode. This prevents the original suspect drive from being written to and possibly damaging evidence. When you boot the suspect machine off of the Encase boot CD, you have four choices (Network Support, USB – Acquisition, USB – Destination, or Clean Boot). I chose Network Support since my only method to acquire the drive was with a crossover cable. At this point you have a menu to choose from to configure your network card. The option that I usually pick that is the easiest is "Auto". This tries to detect your Network Interface Card and set it up for you. The other menu choices are 1. Encase – to launch the program, 2. Manual – to configure the Network Interface Card yourself, 3. SCSI – to select from a SCSI card list, and 4. Scan – to scan the PCI bus for installed cards. Once I chose "Auto", it detects my 3Com Network Interface Card and says press any key to continue. It loads the Network

Interface Card driver and brings up the DOS version of Encase and waits for a connection.

At this point I would configure Encase on my forensic laptop to connect via the crossover cable by selecting add device and choose “Network Crossover”. Once the connection is established, you will see on the Encase suspect machine where the server piece is running that the “lock” feature is enabled. This means that the drive is mounted read only and it cannot be written to. This is a crucial piece of information from the legal perspective. You have to be able to prove to the courts that the original drive was not tampered with. After this I was able to acquire (bit for bit copy) this suspect drive over to my Encase forensics laptop via the crossover Ethernet cable.

During the acquisition of a drive through Encase, you have the option to do several necessary steps (search each file for keywords, verify file signatures, compute hash value, and re-compute hash value). When you compute the hash values during acquisition, this creates a hash for each file on the drive you are acquiring so that you can compare all these files against known hash databases. In Encase you can import known hash databases from various sources. One of these sources is the National Software Reference Library (NSRL) and they have a CD (called the Reference Data Set) containing hundreds of these hash sets that can be purchased. This can be found at <http://www.nsrl.nist.gov>. The hash sets can be used for purposes of eliminating known good files from searches thereby reducing the number of searchable files as well as the search time, or for finding known bad files such as hacker tools, viruses, malware, etc. It should also be noted that had I chosen Autopsy as the forensic tool of choice for this case that it also allows you to enter in hash databases for these same purposes. This comes in handy when using the file sorter in Autopsy because you can eliminate known good files and flag known bad files for further review.

Guidance Software also makes a read-only fast IDE device called a FastBloc device for faster drive acquisition but this will only install in a desktop computer system. For the traveling forensics investigator, Encase makes a portable FastBloc device to take on the road with you.

At this point, I was able to restore the forensic copy of this drive to another physical drive by right clicking on the evidence file icon under the case I created and choosing restore from the popup menu. The next popup menu will give you a list of available drives that you can restore the image to. Encase recommends that the target drive be slightly larger than the original drive. Encase will not allow you to restore an image to Drive0 since it thinks this is where the Operating System is likely installed. Since the new physical drive did not exactly match the drive geometry of the original drive, there were some write errors encountered as seen below. The reason for this is that the target drive was not the same CHS type (cylinders, sectors, and heads) as the source drive. Encase does have a mechanism to convert the drive geometry on the target drive to match that of the

source drive but I did not check that box, hence the write errors. However, before Encase starts the restore process, it computes the hash value of the existing image to use for comparison after the restore process is completed. As you can see below, the input hash value and the output hash value match exactly meaning that all files were restored to the target drive exactly as they were copied from the source drive.

Status: Completed
Start: 10/07/04 09:25:16PM
Stop: 10/07/04 11:26:25PM
Time: 2:01:00
Device: E
Total Sectors: 160,826,652
Read errors: 0
Write errors: 1,256,384
Input Hash: 906CF0A8BBF702D22D140E8FF58A0C86
Output Hash: 906CF0A8BBF702D22D140E8FF58A0C86
Verify errors: 1,256,384

When the restore had finished, I ran a separate hash command within Encase to recomputed the hash value of the restored drive. In the case view of Encase, the left hand side of the screen looks like Windows Explorer with a directory structure and plus signs beside each directory that has subdirectores. There is also a square box next to each directory and each file that can be blue checked. If you check the box at the top of the directory structure (which has a "C" for the C-drive), you can then click on the search button on the top toolbar to bring up the search window. On this window, you can uncheck the search for keywords and the verify file signatures and then check the compute has value and recompute hash values to generate a separate hash of this drive. As you can see below, the hash value matches exactly what the value was earlier.

Status: Completed
Start: 10/07/04 11:37:24PM
Stop: 10/07/04 11:59:55PM
Time: 0:22:31
Name: Kentucky_Purchasing
Start Sector: 63
Stop Sector: 78,155,279
Hash Value: 906CF0A8BBF702D22D140E8FF58A0C86

Media Analysis of System

For the purposes of analyzing the hard drive that we obtained a forensic image of, we will discuss the various areas of media analysis for only this system (the that he used daily). Since we don't have a forensic image of the other two systems we cannot fully explain the concept of media analysis.

The acquisition method of obtaining the forensic copy of the drive and the ability to do that in a read-only manner with Encase has already been described previously in this document. Some of the analysis concepts employed by forensic investigators are as follows:

- reviewing relevant files
  - email (PST files, Hotmail, Yahoo, etc.)
  - documents (MSOffice tracks locations of saved files)
  - Internet History
  - Registry (last commands executed, typed URL's, last files saved, last write time for registry keys)
  - Recycle Bin (INFO2 record analysis)
  - system and application log files (Windows event logs, etc.)
  - web server logs (IIS or Apache logs)a

The techniques actually used for this investigation are explained in the following bullets:

- An examination of the file system to look for modification to operating system software or configuration is necessary. The best way to do this is with the timeline analysis. As discussed previously, Encase comes with many built-in scripts for assisting the forensic examiner in media analysis. By viewing the scripts in Encase and double clicking on the one called timeline analysis, you can then click the run button to get a popup window for the timeline parameters. You can specify the start and stop dates you wish to use for your timeline. After you hit ok and the script runs, it creates a *.CSV file that you can open in Excel or any program that will parse a comma separated value file. Once you open this timeline, you should be able to look at key directories surrounding the operating system and deduce by their creation date when the operating system was installed. Once we have determined this, we can possibly conclude if specific system files have been tampered with. On a Windows platform, the create date of certain operating system files will change more frequently than say a Unix system due to the number of operating system patches that come out.
- With our suspect drive in this case containing a Windows file system (NTFS) we would not have to worry about looking for the existence of setuid or setgid files like we would on a Unix system. The existence of these files might indicate the method an attacker could use elevate his privileges in the operating system. We would however examine the file system for back doors by looking through a hash database in Encase of known backdoors to see if any of these files exist on the system. By using the hash analysis function in Encase, you can compare the hashes of all the files on the suspect drive against the hashes of files that are known backdoors. This is a much better method of detecting a malicious file because the attacker could rename a file to hide it but the hash signature would be the same because the contents of the file had not changed. Also in looking for backdoors, we would want to

review the registry keys that automatically start programs when the system boots. One such key would be \HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run and in this key you might find malicious backdoors that automatically start when Windows boots up.

- Looking for the existence of sniffer programs on the file system would be another important part of our analysis. On the Windows platform, which we have in this case, a network interface card requires additional software to be installed in order to function in promiscuous mode. Promiscuous mode is what allows the network interface card to sniff the traffic on the network. A program called WinPcap is typically used for this purpose on Windows systems. At this point a sniffer program such as Ethereal or Etherpeek would have to be installed. Therefore since we are doing analysis on a Windows system, we would be searching the file system, deleted files, and unallocated space for the existence of these programs. A keyword search was done in Encase for "sniffer", "ethereal", and "winpcap". A dirty word search in Autopsy would have accomplished the same purpose. Encase has a feature in the Enscript mentioned earlier called initialize case that will detect all installed programs. It does this by searching through several registry keys in the Software hive of the registry (the App paths key, the Uninstall key, and the Microsoft key). This script also looks at the NTUSER.DAT file that belongs to every user profile (meaning users that have logged onto this system) to determine which software is installed under each profile.
- Within the Enscripts available in Encase, you can run just a plain Internet history script or you can run a modified script that will actually look for keywords within the Internet history. This script searches through the drives in a case looking for index.dat files that contain URL's and timestamps of websites visited by the users of the system. Most end users are under the assumption that deleting the files in their temporary Internet files directory will remove the traces of sites that have been visited. However, this index.dat file stores this information also in a non-readable format. You have to have a forensics program or special file viewer to view this file. The Internet History Parser and column headings are explained next.

Time interpretations below:
Daily MSHist files (ie, MSHist011998100819981009index.dat -- one day between yyyymmddyyyymmdd dates):
Time 1: Last visited, (+/- GMT offset)
Time 2: Last visited, local time

Weekly MSHist files (ie, MSHist011998092819981005\index.dat -- one week between yyyymmddyyyymmdd dates):
Time 1: Last visit (+/- GMT offset)
Time 2: Time history file was created

Cookies:
Time 1: Cookie created
Time 2: Cookie last accessed

HistoryIndex.dat
Time 1: Last accessed
Time 2: Last accessed

Temporary Internet FilesContent.IE5index.dat:
Time 1: Server modified
Time 2: Last accessed

Times which are marked (+/- GMT offset): These times are actually offset by the opposite of the normal GMT offset. For example, PST is usually -8 hrs from GMT. So, if I see the date-time "10/31/02 05:50:53AM" in the first time column of a weekly history file, then the actual access time is +8 hours -- 1:50:53PM. As you can see, this will pull history information from index.dat files that live in multiple locations (one for the temporary internet files, one for the history, one for the cookies, as well as daily and weekly MSHist files – Microsoft application history files).

The below CSV file is the output from this Internet History Parser script analyzing the index.dat file from Kyle Williams user profile (most of these are evidence of the accessing the local web application he had written that ran on the computer called comp-9b4zdz:

EnCase Internet History Parser
Kentucky_PurchasingCompaq_Web_Server\Crossover·IPaq_C_Drive\C\Documents and Settings\williamsk\Local Settings\Temporary Internet Files\Content.IE5\index.dat

#	Type	Time 1	Time 2	Link
1	URL		08/04/03 03:16:10PM	http://localhost/uptime.asp
2	URL		08/04/03 03:16:43PM	http://localhost/central.supply/phone.book.asp
3	URL	08/12/03 10:25:24PM	09/25/03 03:08:40PM	http://comp-9b4zdz/extra/images/logo.jpg
4	URL	10/24/02 02:31:50PM	09/25/03 03:06:15PM	http://comp-9b4zdz/extra/images/index_02.gif
5	URL	10/24/02 02:31:50PM	09/25/03 03:06:15PM	http://comp-9b4zdz/extra/images/index_01.gif
6	URL	10/24/02 02:31:48PM	09/25/03 03:06:15PM	http://comp-9b4zdz/extra/images/index_03.gif
7	URL	10/24/02 02:31:50PM	09/25/03 03:06:15PM	http://comp-9b4zdz/extra/images/logo.gif
8	URL		08/04/03 03:17:06PM	http://localhost/central.supply/phone.book.asp307179a9
9	URL	10/24/02 02:31:50PM	09/25/03 03:06:15PM	http://comp-9b4zdz/extra/images/links.gif
10	URL	10/27/02 12:00:26PM	08/04/03 04:51:53PM	http://comp-9b4zdz/extra/images/index_04.gif
11	URL	10/24/02 02:31:50PM	09/25/03 03:06:15PM	http://comp-9b4zdz/extra/images/index_08.gif
12	URL		08/04/03 03:16:59PM	http://comp-9b4zdz/central.supply/vendor.details.asp?id=1749
13	URL		08/04/03 03:17:24PM	http://comp-9b4zdz/central.supply/phone.book.asp
14	URL		08/04/03 03:17:42PM	http://comp-9b4zdz/central.supply/phone.book.asp307179a9
15	URL		08/04/03 03:17:35PM	http://comp-9b4zdz/central.supply/vendor.details.asp?id=942
16	URL		08/04/03 03:20:59PM	http://comp-9b4zdz/pt/confirm.asp?id=84
17	URL		09/25/03 03:06:15PM	http://comp-9b4zdz/workorder/Workorderedit.asp?ID=-415463327

18	URL		08/04/03 04:50:47PM	http://comp-9b4zdz/Default.asp9941b92f
19	URL		09/25/03 03:04:04PM	http://comp-9b4zdz/Workorder/Workordersearch.asp
20	URL	03/29/03 10:45:19PM	08/04/03 03:33:52PM	http://localhost/extra/Common/default.css
21	URL	10/24/02 02:31:50PM	08/04/03 03:33:52PM	http://localhost/extra/images/logo.jpg
22	URL		08/04/03 03:19:07PM	http://comp-94bz92/Default.asp307179a9
23	URL	10/24/02 02:31:50PM	08/04/03 03:20:02PM	http://localhost/extra/images/index_01.gif
24	URL	10/24/02 02:31:48PM	08/04/03 03:20:02PM	http://localhost/extra/images/index_03.gif
25	URL	10/24/02 02:31:50PM	08/04/03 03:20:02PM	http://localhost/extra/images/index_02.gif
26	URL	10/27/02 12:00:26PM	08/04/03 03:20:02PM	http://localhost/extra/images/index_04.gif
27	URL	10/24/02 02:31:50PM	08/04/03 03:20:02PM	http://localhost/extra/images/logo.gif
28	URL	10/24/02 02:31:50PM	08/04/03 03:20:02PM	http://localhost/extra/images/links.gif
29	URL	10/24/02 02:31:50PM	08/04/03 03:20:02PM	http://localhost/extra/images/index_08.gif
30	URL		08/04/03 03:19:19PM	http://localhost/workorder/detaillist.asp
31	URL		08/20/03 11:38:21AM	http://comp-94bz92/
32	URL		08/04/03 03:20:02PM	http://comp-94bz92/Default.asp99cdb2d9
33	URL	10/24/02 02:31:50PM	08/04/03 03:20:02PM	http://comp-94bz92/extra/images/WorkOrder.gif
34	URL		08/04/03 03:20:05PM	http://localhost/WorkOrder/ReferenceSearch.asp
35	URL		08/04/03 03:20:19PM	http://localhost/WorkOrder/ReferenceSearch.asp99cdb2d9
36	URL		08/04/03 03:20:22PM	http://comp-9b4zdz/workorder/WorkOrderQuery.asp99e057d3
37	URL	10/24/02 02:31:48PM	08/04/03 03:20:29PM	http://comp-9b4zdz/extra/images/searchPIC.jpg
38	URL		08/04/03 03:20:29PM	http://comp-9b4zdz/workorder/WorkOrderQuery.asp99cdb2d9
39	URL	10/24/02 02:31:50PM	08/04/03 04:50:48PM	http://comp-9b4zdz/extra/images/WorkOrder.gif
40	URL		08/04/03 03:21:05PM	http://comp-9b4zdz/pt/confirm.asp?id=85
41	URL		08/04/03 03:21:10PM	http://comp-9b4zdz/pt/confirm.asp?id=52
42	URL		09/25/03 03:02:59PM	http://comp-9b4zdz/default.asp3dc6bce8
43	URL		08/04/03 04:50:19PM	http://comp-9b4zdz/Default.asp99cdb2d9
44	URL		08/04/03 04:47:41PM	http://comp-9b4zdz/pt/confirm.asp?id=55
45	URL		08/04/03 04:50:15PM	http://comp-9b4zdz/pt/confirm.asp?id=53
46	URL		09/25/03 03:03:30PM	http://comp-9b4zdz/msgs/displayMessage.asp?Body=4878&True
47	REDR		09/25/03 03:03:30PM	http://comp-9b4zdz/pt/census.asp
48	URL		08/04/03 04:51:52PM	http://comp-9b4zdz/extra/errors/access.denied.asp
49	URL		08/20/03 11:38:55AM	http://localhost/
50	URL	08/07/02 11:49:32AM	09/25/03 03:06:15PM	http://comp-9b4zdz/extra/common/popupmenu.js
51	URL	09/03/03 01:00:24PM	09/25/03 03:08:40PM	http://comp-9b4zdz/extra/Common/default.css
52	URL	09/09/03 11:22:16PM	09/25/03 03:06:15PM	http://comp-9b4zdz/extra/images/menu.gif
53	URL	08/11/03 10:43:14AM	09/25/03 03:05:45PM	http://comp-9b4zdz/extra/images/heading/main.gif
54	URL	08/17/03 07:25:02PM	09/25/03 03:05:45PM	http://comp-9b4zdz/extra/images/arrow.gif
55	URL	08/11/03 10:43:16AM	09/25/03 03:03:37PM	http://comp-9b4zdz/extra/images/heading/msg.gif
56	URL	09/02/03 03:00:46PM	09/25/03 03:03:35PM	http://comp-9b4zdz/extra/images/msg-read.gif
57	URL	09/08/03 02:15:43PM	09/25/03 03:03:35PM	http://comp-9b4zdz/extra/images/msg-system.gif
58	URL		09/25/03 03:03:37PM	http://comp-9b4zdz/msgs/displayMessage.asp?Body=4880&True
59	URL		09/25/03 03:05:51PM	http://comp-9b4zdz/workorder/detaillist.asp
60	URL	08/11/03 10:43:14AM	09/25/03 03:06:15PM	http://comp-9b4zdz/extra/images/heading/wo.gif
61	URL		09/25/03 03:05:45PM	http://comp-9b4zdz/default.asp
62	URL	10/24/02 02:31:50PM	09/25/03 03:06:15PM	http://comp-9b4zdz/extra/images/Printable_work_order.gif
63	URL		09/25/03 03:06:22PM	http://comp-9b4zdz/workorder/notes.asp?id=-415463327
64	URL		09/25/03 03:08:38PM	http://comp-9b4zdz/
65	URL	08/24/03 09:38:56PM	09/25/03 03:05:51PM	http://comp-9b4zdz/extra/images/wonote.gif

66	URL	05/03/03 06:35:41PM	http://localhost/Default.asp307179a9
67	REDR	05/03/03 06:35:41PM	http://comp-9b4zdz/extra/submit/addpoollog.aspdf335ec6
68	URL	05/03/03 06:37:08PM	http://comp-9b4zdz/maintenance/list.pool.logs.asp307179a9
69	URL	05/03/03 06:37:15PM	http://comp-9b4zdz/workorder/WorkOrderQuery.asp85329107
70	REDR	05/03/03 06:37:15PM	http://comp-9b4zdz/extra/submit/addboilerlog.aspdd9e2d4b
71	REDR	05/03/03 06:37:15PM	http://comp-9b4zdz/extra/submit/addsecuritylog.aspe60c2fa5

You can see from this output that Mr. Williams was developing this web application on his local computer, hence the <http://localhost> entries, and then copying the files to the remote web server and testing them there (hence the <http://comp-9b4zdz> entries).

- From the case view of the suspect file system in Encase, you can right click on one of the Windows Registry files in C:\$WINDIR\$SYSTEM\Config and choose the menu option that says view file structure. This will create the Registry keys in a subfolder type format just like you were in Regedit. If this were a Unix system, we would want to look at the /etc directory for configuration files that may have been edited or altered by the attacker.
- In this case, there does not appear to be any abnormal processes in the Registry key that starts processes every time the machine is booted. The key I am referencing is called… HKLM/Software/Microsoft/Windows/CurrentVersion/Run. On the suspect machine, the only entries here are benign and have to do with anti-virus, hot-key commands, etc.

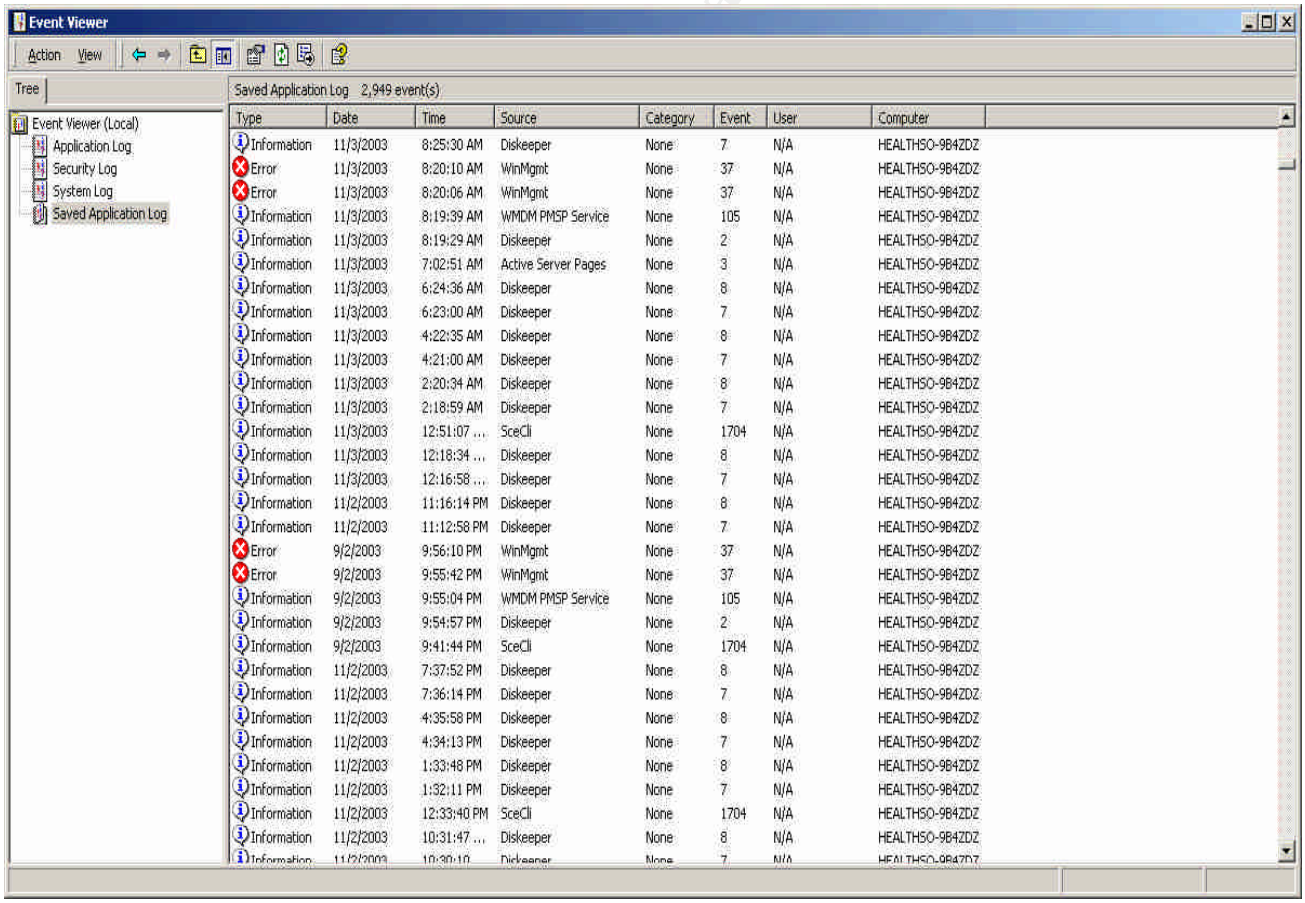
The below evidence was obtained from the Compaq IPAQ system that was running the web server that was allegedly tampered with. This IRC session is just substantiating evidence that Mr. Williams was indeed researching and learning about blackhat hacker skills.

- IRC session log from 01/22/2000 that is basic instruction on “How Servers are Cracked”.

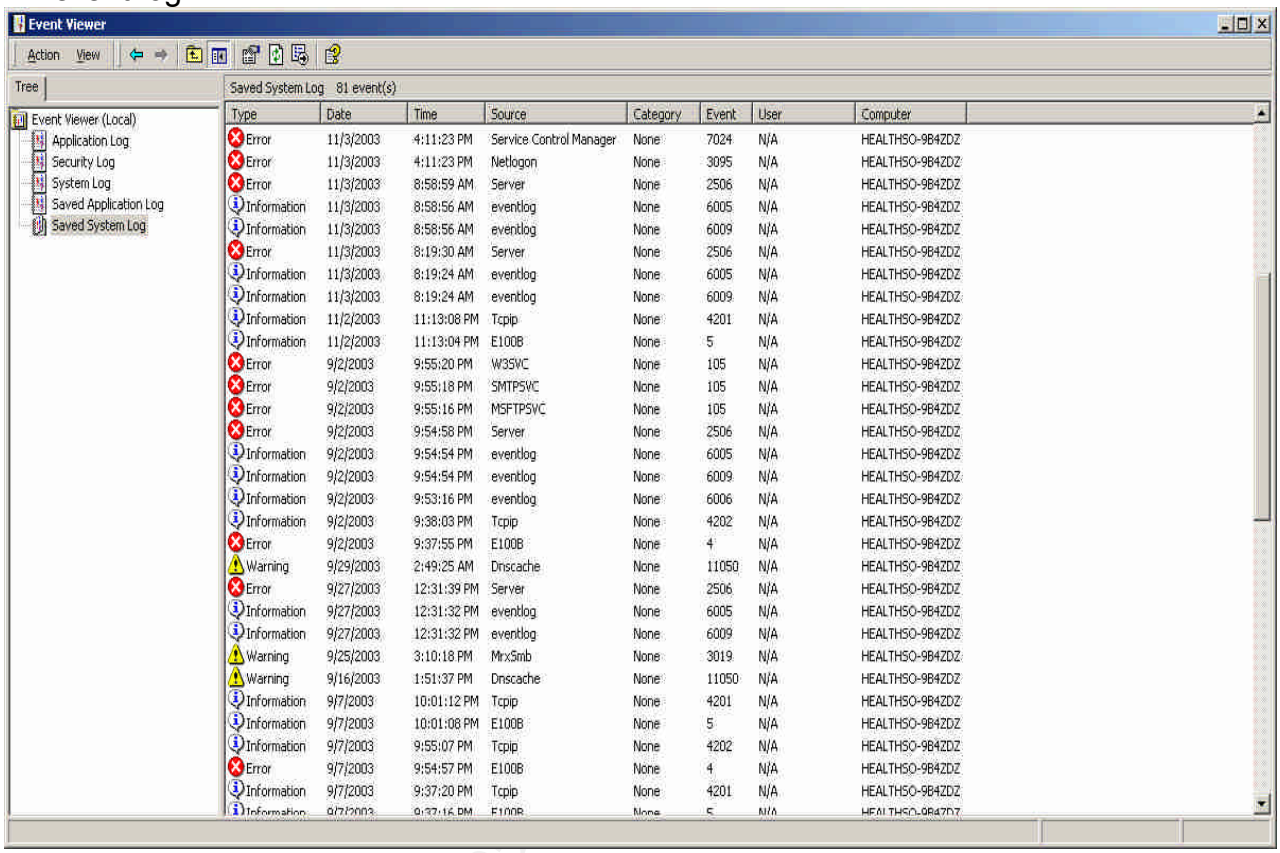
Session Start: Sat Jan 22 18:04:06 2000… excerpt below…
[18:04] *** Now talking in #bsrf
[18:04] *** Topic is 'Welcome to #bsrf | Our Website: http://blacksun.box.sk | Next IRC lecture: 'How Servers are Cracked' | See http://blacksun.box.sk/irc.html | Alright, I know the bot is down most of the time now, and that the channel is ultra insecure, so please don't abuse this... heh, yeah right·'
[18:04] *** Set by Raven on Wed Jan 19 07:22:58
[18:04] * #bsrf is being logged
[18:04] <INTJ> okie
[18:05] <INTJ> ready?
[18:05] <Raven> alright
[18:05] <Raven> is everyone ready?

- In reviewing the IIS web server logs, I noticed that the computer at 10.xxx.240.79 (NCRxxxxxx3287) created the following entry in the 11/03/2003 log file (since the logs are stored in GMT, this would be 12:07 am on 11/03/2003. This could potentially be effort of the suspect to determine if his plan succeeded to delete the web server files. A closer analysis of this 10.xxx.240.79 machine will need to be done to determine if it was accessed remotely or someone locally tried to open this website.
    - 06:07:49 10.xxx.240.79 GET / 403

The application event logs on the web server definitely seem to be somehow altered based on the missing entries on the date when the application stopped working. Notice the date discrepancies in the screen shot below. The web application no longer worked on the morning of 11/3/03.
From this it seems possible that the discrepancy between the date/time stamps in this screen shot could be due to the fact that someone changed the date/time on the server and then put it back to the correct one.



- You will also notice the discrepancy in the dates of the entries for the system event log.



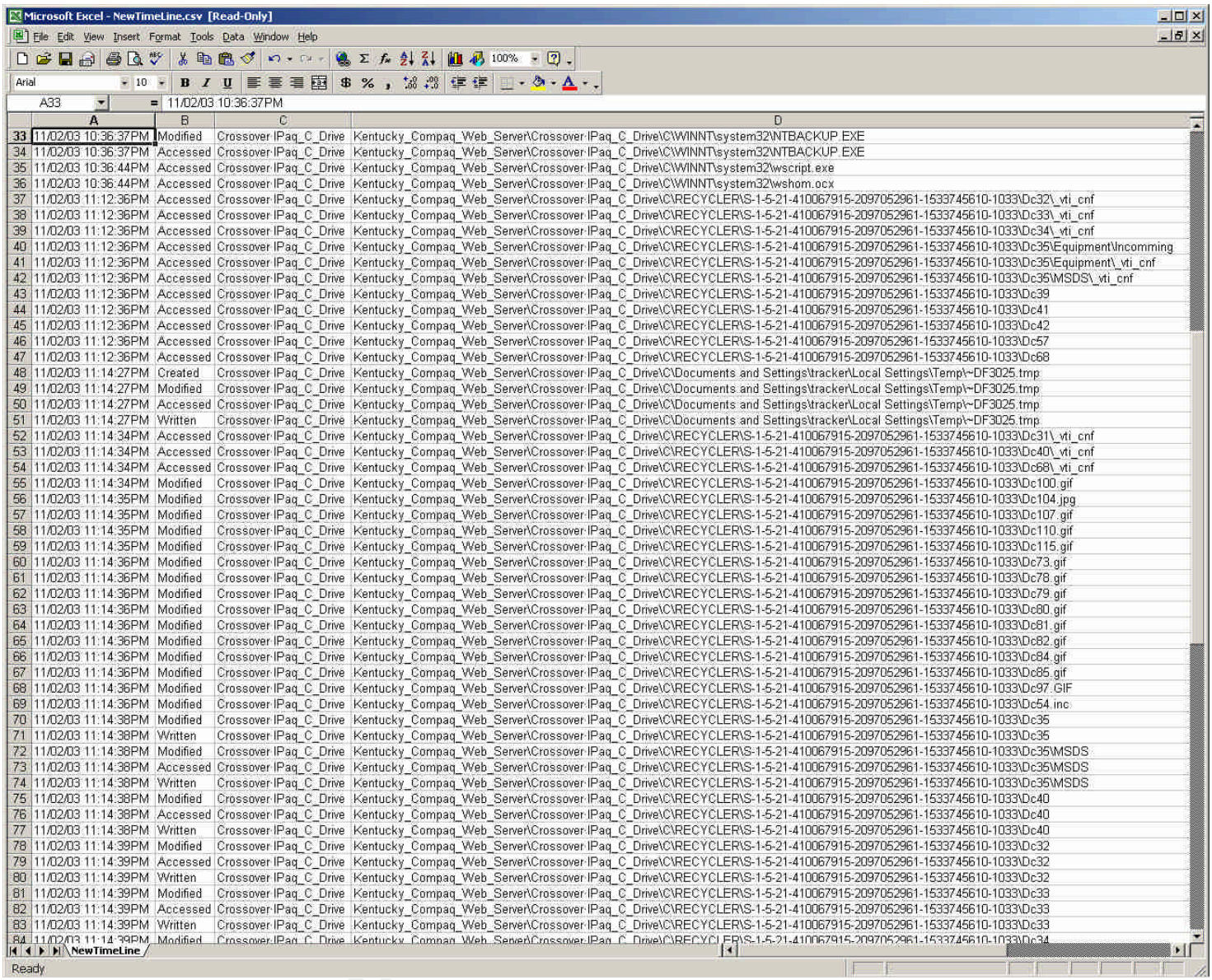
Type	Date	Time	Source	Category	Event	User	Computer
Error	11/3/2003	4:11:23 PM	Service Control Manager	None	7024	N/A	HEALTHSO-9B4ZDZ
Error	11/3/2003	4:11:23 PM	Netlogon	None	3095	N/A	HEALTHSO-9B4ZDZ
Error	11/3/2003	8:58:59 AM	Server	None	2506	N/A	HEALTHSO-9B4ZDZ
Information	11/3/2003	8:58:56 AM	eventlog	None	6005	N/A	HEALTHSO-9B4ZDZ
Information	11/3/2003	8:58:56 AM	eventlog	None	6009	N/A	HEALTHSO-9B4ZDZ
Error	11/3/2003	8:19:30 AM	Server	None	2506	N/A	HEALTHSO-9B4ZDZ
Information	11/3/2003	8:19:24 AM	eventlog	None	6005	N/A	HEALTHSO-9B4ZDZ
Information	11/3/2003	8:19:24 AM	eventlog	None	6009	N/A	HEALTHSO-9B4ZDZ
Information	11/2/2003	11:13:08 PM	Tcpip	None	4201	N/A	HEALTHSO-9B4ZDZ
Information	11/2/2003	11:13:04 PM	E100B	None	5	N/A	HEALTHSO-9B4ZDZ
Error	9/2/2003	9:55:20 PM	W3SVC	None	105	N/A	HEALTHSO-9B4ZDZ
Error	9/2/2003	9:55:18 PM	SMTPSVC	None	105	N/A	HEALTHSO-9B4ZDZ
Error	9/2/2003	9:55:16 PM	MSFTPSVC	None	105	N/A	HEALTHSO-9B4ZDZ
Error	9/2/2003	9:54:58 PM	Server	None	2506	N/A	HEALTHSO-9B4ZDZ
Information	9/2/2003	9:54:54 PM	eventlog	None	6005	N/A	HEALTHSO-9B4ZDZ
Information	9/2/2003	9:54:54 PM	eventlog	None	6009	N/A	HEALTHSO-9B4ZDZ
Information	9/2/2003	9:53:16 PM	eventlog	None	6006	N/A	HEALTHSO-9B4ZDZ
Information	9/2/2003	9:38:03 PM	Tcpip	None	4202	N/A	HEALTHSO-9B4ZDZ
Error	9/2/2003	9:37:55 PM	E100B	None	4	N/A	HEALTHSO-9B4ZDZ
Warning	9/29/2003	2:49:25 AM	Dnscache	None	11050	N/A	HEALTHSO-9B4ZDZ
Error	9/27/2003	12:31:39 PM	Server	None	2506	N/A	HEALTHSO-9B4ZDZ
Information	9/27/2003	12:31:32 PM	eventlog	None	6005	N/A	HEALTHSO-9B4ZDZ
Information	9/27/2003	12:31:32 PM	eventlog	None	6009	N/A	HEALTHSO-9B4ZDZ
Warning	9/25/2003	3:10:18 PM	MrxSmb	None	3019	N/A	HEALTHSO-9B4ZDZ
Warning	9/16/2003	1:51:37 PM	Dnscache	None	11050	N/A	HEALTHSO-9B4ZDZ
Information	9/7/2003	10:01:12 PM	Tcpip	None	4201	N/A	HEALTHSO-9B4ZDZ
Information	9/7/2003	10:01:08 PM	E100B	None	5	N/A	HEALTHSO-9B4ZDZ
Information	9/7/2003	9:55:07 PM	Tcpip	None	4202	N/A	HEALTHSO-9B4ZDZ
Error	9/7/2003	9:54:57 PM	E100B	None	4	N/A	HEALTHSO-9B4ZDZ
Information	9/7/2003	9:37:20 PM	Tcpip	None	4201	N/A	HEALTHSO-9B4ZDZ

- Upon reviewing the files that were accessed or modified on the evening of 11/2/03 which is when it appeared that the event logs were modified, we noticed a massive amount of files being sent to the recycle bin around 11:14 pm. One of the files at 10:36 pm was c:\winnt\system32\wscript.exe which is the Windows Scripting Host. This application could have executed a script file to perform this delete action. However, we have not found a script of this nature to this point.

Timeline Analysis

The timeline analysis that I will be showing here is not from the system that we were able to image. As stated previously, there were three machines involved in this investigation and two of them could not be taken offline long enough for us to image them. I was however able to analyze the affected web server long enough to run a timeline analysis and gather other clues. This system is not truly a server but really a desktop machine (a Compaq IPAQ) acting as a web server. Below is the screen shot of the timeline analysis on the web server showing all the files being sent to the recycle bin at the 11:14pm mark on 11/02/03. This would indicate the point in time when the files were actually deleted from the system and caused the web server to stop functioning as it did before.

You can also see from the timeline that files c:\winnt\system32\wscript.exe and c:\winnt\system32\wshom.ocx were accessed at 10:36 pm on 11/02/03 just before the files were sent to the recycle bin. As an investigator this would indicate to me that some process called the Windows Scripting Host and it accessed the file whsom.ocx, which is an Active-X control. Without sufficient training and experience with programming Active-X controls I am unable to determine at this point if this was definitely the cause of the files being deleted. I would need to engage a programmer/developer at this point to explain the details behind this control.



	A	B	C	D
33	11/02/03 10:36:37PM	Modified	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\WINNT\system32\NTBACKUP.EXE
34	11/02/03 10:36:37PM	Accessed	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\WINNT\system32\NTBACKUP.EXE
35	11/02/03 10:36:44PM	Accessed	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\WINNT\system32\wscript.exe
36	11/02/03 10:36:44PM	Accessed	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\WINNT\system32\wshom.ocx
37	11/02/03 11:12:36PM	Accessed	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\RECYCLER\S-1-5-21-410067915-2097052961-1533745610-1033\Dc32_vti_cnf
38	11/02/03 11:12:36PM	Accessed	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\RECYCLER\S-1-5-21-410067915-2097052961-1533745610-1033\Dc33_vti_cnf
39	11/02/03 11:12:36PM	Accessed	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\RECYCLER\S-1-5-21-410067915-2097052961-1533745610-1033\Dc34_vti_cnf
40	11/02/03 11:12:36PM	Accessed	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\RECYCLER\S-1-5-21-410067915-2097052961-1533745610-1033\Dc35\Equipment\Incomming
41	11/02/03 11:12:36PM	Accessed	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\RECYCLER\S-1-5-21-410067915-2097052961-1533745610-1033\Dc35\Equipment_vti_cnf
42	11/02/03 11:12:36PM	Accessed	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\RECYCLER\S-1-5-21-410067915-2097052961-1533745610-1033\Dc35\MSDS_vti_cnf
43	11/02/03 11:12:36PM	Accessed	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\RECYCLER\S-1-5-21-410067915-2097052961-1533745610-1033\Dc39
44	11/02/03 11:12:36PM	Accessed	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\RECYCLER\S-1-5-21-410067915-2097052961-1533745610-1033\Dc41
45	11/02/03 11:12:36PM	Accessed	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\RECYCLER\S-1-5-21-410067915-2097052961-1533745610-1033\Dc42
46	11/02/03 11:12:36PM	Accessed	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\RECYCLER\S-1-5-21-410067915-2097052961-1533745610-1033\Dc57
47	11/02/03 11:12:36PM	Accessed	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\RECYCLER\S-1-5-21-410067915-2097052961-1533745610-1033\Dc68
48	11/02/03 11:14:27PM	Created	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\Documents and Settings\tracker\Local Settings\Temp\~DF3025.tmp
49	11/02/03 11:14:27PM	Modified	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\Documents and Settings\tracker\Local Settings\Temp\~DF3025.tmp
50	11/02/03 11:14:27PM	Accessed	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\Documents and Settings\tracker\Local Settings\Temp\~DF3025.tmp
51	11/02/03 11:14:27PM	Written	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\Documents and Settings\tracker\Local Settings\Temp\~DF3025.tmp
52	11/02/03 11:14:34PM	Accessed	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\RECYCLER\S-1-5-21-410067915-2097052961-1533745610-1033\Dc31_vti_cnf
53	11/02/03 11:14:34PM	Accessed	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\RECYCLER\S-1-5-21-410067915-2097052961-1533745610-1033\Dc40_vti_cnf
54	11/02/03 11:14:34PM	Accessed	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\RECYCLER\S-1-5-21-410067915-2097052961-1533745610-1033\Dc68_vti_cnf
55	11/02/03 11:14:34PM	Modified	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\RECYCLER\S-1-5-21-410067915-2097052961-1533745610-1033\Dc100.gif
56	11/02/03 11:14:35PM	Modified	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\RECYCLER\S-1-5-21-410067915-2097052961-1533745610-1033\Dc104.jpg
57	11/02/03 11:14:35PM	Modified	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\RECYCLER\S-1-5-21-410067915-2097052961-1533745610-1033\Dc107.gif
58	11/02/03 11:14:35PM	Modified	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\RECYCLER\S-1-5-21-410067915-2097052961-1533745610-1033\Dc110.gif
59	11/02/03 11:14:35PM	Modified	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\RECYCLER\S-1-5-21-410067915-2097052961-1533745610-1033\Dc115.gif
60	11/02/03 11:14:36PM	Modified	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\RECYCLER\S-1-5-21-410067915-2097052961-1533745610-1033\Dc73.gif
61	11/02/03 11:14:36PM	Modified	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\RECYCLER\S-1-5-21-410067915-2097052961-1533745610-1033\Dc78.gif
62	11/02/03 11:14:36PM	Modified	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\RECYCLER\S-1-5-21-410067915-2097052961-1533745610-1033\Dc79.gif
63	11/02/03 11:14:36PM	Modified	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\RECYCLER\S-1-5-21-410067915-2097052961-1533745610-1033\Dc80.gif
64	11/02/03 11:14:36PM	Modified	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\RECYCLER\S-1-5-21-410067915-2097052961-1533745610-1033\Dc81.gif
65	11/02/03 11:14:36PM	Modified	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\RECYCLER\S-1-5-21-410067915-2097052961-1533745610-1033\Dc82.gif
66	11/02/03 11:14:36PM	Modified	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\RECYCLER\S-1-5-21-410067915-2097052961-1533745610-1033\Dc84.gif
67	11/02/03 11:14:36PM	Modified	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\RECYCLER\S-1-5-21-410067915-2097052961-1533745610-1033\Dc85.gif
68	11/02/03 11:14:36PM	Modified	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\RECYCLER\S-1-5-21-410067915-2097052961-1533745610-1033\Dc97.GIF
69	11/02/03 11:14:36PM	Modified	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\RECYCLER\S-1-5-21-410067915-2097052961-1533745610-1033\Dc54.inc
70	11/02/03 11:14:38PM	Modified	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\RECYCLER\S-1-5-21-410067915-2097052961-1533745610-1033\Dc35
71	11/02/03 11:14:38PM	Written	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\RECYCLER\S-1-5-21-410067915-2097052961-1533745610-1033\Dc35
72	11/02/03 11:14:38PM	Modified	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\RECYCLER\S-1-5-21-410067915-2097052961-1533745610-1033\Dc35\MSDS
73	11/02/03 11:14:38PM	Accessed	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\RECYCLER\S-1-5-21-410067915-2097052961-1533745610-1033\Dc35\MSDS
74	11/02/03 11:14:38PM	Written	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\RECYCLER\S-1-5-21-410067915-2097052961-1533745610-1033\Dc35\MSDS
75	11/02/03 11:14:38PM	Modified	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\RECYCLER\S-1-5-21-410067915-2097052961-1533745610-1033\Dc40
76	11/02/03 11:14:38PM	Accessed	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\RECYCLER\S-1-5-21-410067915-2097052961-1533745610-1033\Dc40
77	11/02/03 11:14:38PM	Written	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\RECYCLER\S-1-5-21-410067915-2097052961-1533745610-1033\Dc40
78	11/02/03 11:14:39PM	Modified	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\RECYCLER\S-1-5-21-410067915-2097052961-1533745610-1033\Dc32
79	11/02/03 11:14:39PM	Accessed	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\RECYCLER\S-1-5-21-410067915-2097052961-1533745610-1033\Dc32
80	11/02/03 11:14:39PM	Written	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\RECYCLER\S-1-5-21-410067915-2097052961-1533745610-1033\Dc32
81	11/02/03 11:14:39PM	Modified	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\RECYCLER\S-1-5-21-410067915-2097052961-1533745610-1033\Dc33
82	11/02/03 11:14:39PM	Accessed	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\RECYCLER\S-1-5-21-410067915-2097052961-1533745610-1033\Dc33
83	11/02/03 11:14:39PM	Written	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\RECYCLER\S-1-5-21-410067915-2097052961-1533745610-1033\Dc33
84	11/02/03 11:14:39PM	Modified	Crossover IPaq_C_Drive	Kentucky_Compaq_Web_Server\Crossover IPaq_C_Drive\C\RECYCLER\S-1-5-21-410067915-2097052961-1533745610-1033\Dc34

In creating a timeline analysis for the drive that we actually imaged, Encase creates a file by default called timeline.csv when you run the Timeline Enscript. On Windows operating systems, *.CSV files automatically associate themselves with Microsoft Excel. You run into an issue when you want to see a timeline in Excel for an entire hard drive when the CSV file is 55mb and Microsoft Excel will only hold around 65,000 rows in one spreadsheet. You can open the document in Microsoft Word but there is no way to sort the data by date like you would be able to in Microsoft Excel. When I opened this CSV file in Microsoft Word, it was over 2000 pages long with ½ inch margins all around and Arial type 8 font. Needless to say, this is a lot of data. However, from searching this file we can see that the operating system appears to have been installed on the date below because this is when the Windows directory was created:

09/08/02
07:13:46AM;"Created";"Kentucky_Purchasing";"\Kentucky_Purchasing\C\WINDOWS";"WINDOWS"

You can also look for metadata like $MFT and $BOOT when trying to determine when an operating system was installed on a computer. This is accomplished easily through Autopsy but I have not seen this feature in Encase. There are many files that are created during a Windows installation where you could look for the creation date (i.e. registry files, system32 directory and many subdirectories).

You can see that a service pack was applied on 9/11/03 around 7:51 pm as the files in \Windows\ServicePackFiles\i386 were modified:

09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\msn238.mar
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\msjro.dll
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\msjet40.dll
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\mofd.dll
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\iphlpapi.dll
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\mstsc.exe
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\imagehlp.dll
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\itircl.dll
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\ixsso.dll
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\nmas.dll
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\msjetol1.dll
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\mslbui.dll
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\msdasc.dll
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\migwiz.inf
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\iuengine.dll
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\msdart.dll
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\msobshel.htm
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\msado25.tlb
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\irbus.sys

09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\msvfw32.dll
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\mnmdd.dll
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\netcfgx.dll
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\netip6.inf
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\netapi32.dll
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\licwmi.dll
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\netrtsnt.inf
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\msmsgs.exe
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\mpg4dmod.dll
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\msmsgsin.exe
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\mmcndmgr.dll
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\mqqm.dll
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\msadox.dll
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\msador15.dll
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\moricons.dll
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\migism.dll
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\msnmtllc.dll
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\migrate.obe
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\licmgr10.dll
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\msmsgs.inf
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\ncobjapi.dll
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\msutb.dll
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\mst120.dll
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\mstsweb.cat
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\ndisuio.sys
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\nic1394.sys
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\imgutil.dll
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\mdmlt3.inf
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\msscds32.ax
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\msadcf.dll
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\msihnd.dll
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\kd1394.dll
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\ieaksie.dll
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\netwlan2.inf
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\msn6.exe
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\lwadihid.sys
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\imm32.dll
09/11/03 07:51:17PM Modified Kentucky_Purchasing\Kentucky_Purchasing\C\WINDOWS\ServicePackFiles\i386\msdaosp.dll

Recover Deleted Files

Encase uses several different methods to recover deleted files and directories from hard drives. The easiest file system to recover files from is the FAT file system. Encase has a menu option when you right click on a “volume” that says recover folders. This command instructs Encase to search through the unallocated clusters on the FAT partition for the “dot, double dot” signature that is indicative of a deleted directory. This command usually has great success recovering files and folders on a drive that has been formatted.

With NTFS, UFS, EXT2/3 partitions, Encase handles the recovery of deleted files differently. When Encase detects a partition of this type, it automatically adds a gray folder called “Lost Files” to the case. In an NTFS file system, Encase parses the MFT (Master File Table) and looks for files that have no parent directory. It then places these files in the gray “Lost Files” folder for the examiners review. This is the same process for the other file system types listed above (UFS, EXT2/3).

When files have been placed in the Recycle Bin within Windows, Encase can indicate when the file was deleted. The Recycle Bin also creates a database of files that are deleted, which is stored in records called INFO2. Encase has a script included that can search the unallocated space and slack space for these INFO2 files. Even though this will not allow you to recover a deleted file itself, it will let you know of the existence of particular files at one time.

After reviewing the files recovered in the “Lost Files” folder in Encase, none of them contained any relevant information to this case and were therefore not recovered.

String Search

In searching this drive for evidence, I wanted to look for clues relating to the possible deletion of an application or particular files on a web server. The application involved here was a work order program so this gave some ideas for keyword searches. In addition, the drive we imaged was not the actual web server and therefore did not return desired results in searching for strings or keywords. The following list shows the types of searches done on the imaged drive.

- All graphics files
- Files modified or created since the termination of the suspect (timeline)
- Files modified or created around the time of the application failure
- Internet history
- Installed applications
- All email addresses identified on each system
- Keyword search for “Williamsj” on each system
- Keyword search for “jWilliams” on each system
- Keyword search for “work order” on each system
- Keyword search for “delete” on each system
- Keyword search for “del” on each system

The keyword searches were done by clicking view on the top menu bar in Encase and selecting the keyword menu item. You will then have a screen where you can add keywords that you want to search on. Once you’ve added your search words and put a check in the box next to them, you click on the search icon on the main toolbar. This brings up the search screen and you will need to select the checkbox that says to search only the selected keywords. The

web application that was deleted was a Work Order system developed by Mr. Williams hence the search for “delete”, “del”, “work order”, “jwilliams”, and “williamsj”. The Internet history might show time correlation for when he tried to access certain sections of the site. Installed applications would indicate to us what programs were used on the system for legitimate and potentially malicious work. The email search is to determine if he was communicating about his malicious intent to anyone outside the company. Graphics files would show what websites he had been to and could possibly show intent on his part. Out of all these searches, minimal clues to this case were found and those that were found have been previously noted.

Conclusions

After many man-hours on this project there has not been any conclusive evidence found that indicates Mr. Williams planted any logic bombs to destroy applications or any evidence that he accessed the network remotely after his termination. Since we never were able to obtain the phone records for the office, we do not know if he had accessed any machines remotely after his termination. We do know that his domain account had been disabled but do not know if he knew other employees credentials that he might have been using to access the network after he was terminated. There is a lot of circumstantial evidence as seen in the web server application files being sent to the Recycle Bin and the Application and Security Event logs being out of whack with their dates. However, none of this circumstantial evidence is proving that Mr. Williams is the guilty party. If more time had been allocated for this project and the availability of some web developers to pore over the code left behind by Mr. Williams, we could have potentially produced some evidence to aid in this logic bomb theory but the time, staff, and money needed to handle this type of request was just not there.

List of References

Bartlett, John. The Ease of Steganography and Camouflage. GSEC v1.3. 17 March, 2002. <http://www.sans.org/rr/papers/20/762.pdf>

SANS Institute. Track 8 – Forensic and Investigative Essentials. Volume 8.1. SANS Press, 2004

Raggo, Michael T. Steganography, Steganalysis, & Cryptanalysis. <http://www.blackhat.com/presentations/bh-usa-04-raggo/bh-us-04-raggo-up.pdf>

BDHTOOL home page. < http://www.stormpages.com/bdhtool/>

Guillermito ZONE. May 6th, 2003. < http://www.guillermito2.net/stegano/camouflage/index.html>

Camouflage Home Page – Hide your files! <http://camouflage.unfiction.com/Camou121.exe>

Foundstone, Inc. Strategic Security home page. 2004.
< http://www.foundstone.com/resources/proddesc/fport.htm>

chkrootkit – locally checks for signs of a rootkit. 2 Sept., 2004.
< http://www.chkrootkit.org>

Welcome to the Ftimes Project. 2004. <http://ftimes.sourceforge.net/Ftimes>

AntiSniff Version 1.0. Copyright 1999 L0pht Heavy Industries, Inc.
<http://packetstormsecurity.nl/sniffers/antisniff/>

Autopsy Forensic Browser. Copyright 2004.
<http://www.sleuthkit.org/autopsy/images/file1.gif>
<http://www.sleuthkit.org/autopsy/download.php>

Sourceforge.net: Project Info – Foremost. Copyright 2004.
<http://www.sourceforge.net/projects/foremost>

md5deep. 12 Oct. 2004. <http://md5deep.sourceforge.net>

Steganalysis – Attacks against Steganography and Watermarking – Countermeasures. Copyright 1995 – 2003.
<http://www.jjtc.com/Steganalysis/>

SecurityFocus HOME Infocus: Steganography Revealed. April 9th, 2003
<http://www.securityfocus.com/infocus/1684>

US CODE: Title 18,1832. Theft of trade secrets. August 6th, 2004.
<http://www4.law.cornell.edu/uscode/18/1832.html>

The Code of Alabama. 1975.
<http://www.legislature.state.al.us/CodeofAlabama/1975/coatoc.htm>

Encase Legal Resources. 2002 – 2004. Guidance Software, Inc.
<http://www.guidancesoftware.com/corporate/legal/index.shtm>