



# Global Information Assurance Certification Paper

Copyright SANS Institute  
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permitted without express written permission.

## Interested in learning more?

Check out the list of upcoming events offering  
"Advanced Incident Response, Threat Hunting, and Digital Forensics (Forensics  
at <http://www.giac.org/registration/gcfa>

# Analysis of Diskette Image and Compromised Systems

GIAC Certified Forensic Analyst Practical Assignment  
Version 1.5

Merlin Namuth  
Submitted: December 22, 2004

© SANS Institute 2005, Author retains full rights.

## Abstract

This paper was written to meet the requirements for the SANS GIAC Certified Forensic Analyst practical assignment Version 1.5.

Part 1 is a scenario about a company, Ballard Industries, which manufactures fuel cells. The company has noticed their competitor, Rift, Inc., has been receiving orders for the same fuel cell, which was once unique to Ballard. The thought is that somehow Rift has received company proprietary information from Ballard. One day, an employee, Robert John Leszczynski, Jr., was “caught” leaving the company R&D lab with a diskette, which is against company policy. A full investigation ensued, which I cover in the remainder of my paper.

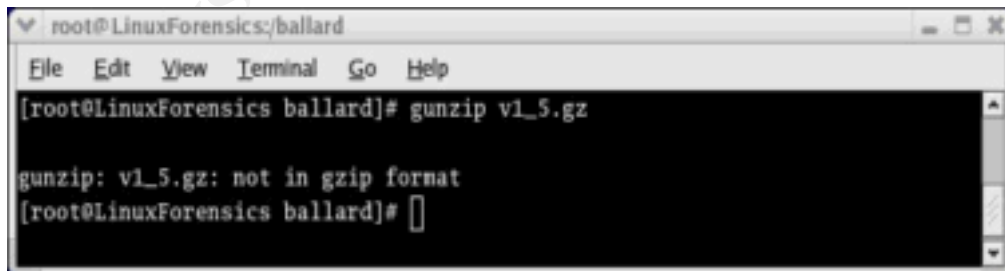
Part 2 Option 1, is about an actual investigation I conducted within my company. It was discovered that a commercially available key logger was installed on two manager’s systems. An investigation on these two systems took place as well as an investigation on someone suspected of installing the key logger. A lot of correlation took place between the analysis of the three systems.

## Part 1 – Analyze an Unknown Image

I downloaded the forensic image from ([http://www.giac.org/gcfa/v1\\_5.gz](http://www.giac.org/gcfa/v1_5.gz)) to my linux forensics workstation. I saved the image in the /ballard directory. The chain of custody provided in this assignment is:

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

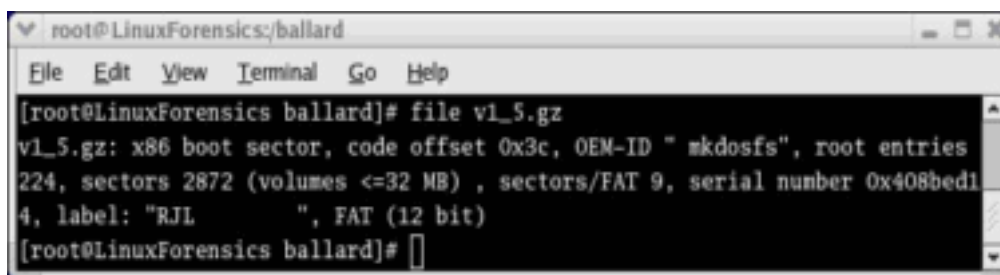
After downloading the forensic image, I attempted to uncompress it.



```
root@LinuxForensics:/ballard
File Edit View Terminal Go Help
[root@LinuxForensics ballard]# gunzip v1_5.gz

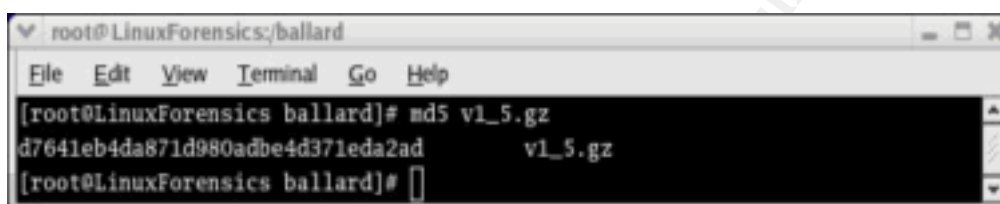
gunzip: v1_5.gz: not in gzip format
[root@LinuxForensics ballard]#
```

As shown above from the error message generated, this file is not in gzip format, even though it is named with a .gz extension. I needed to determine what this file type was. I ran the following to determine it is a FAT12 file.



```
root@LinuxForensics:/ballard
File Edit View Terminal Go Help
[root@LinuxForensics ballard]# file v1_5.gz
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 0x408bed1
4, label: "RJL      ", FAT (12 bit)
[root@LinuxForensics ballard]#
```

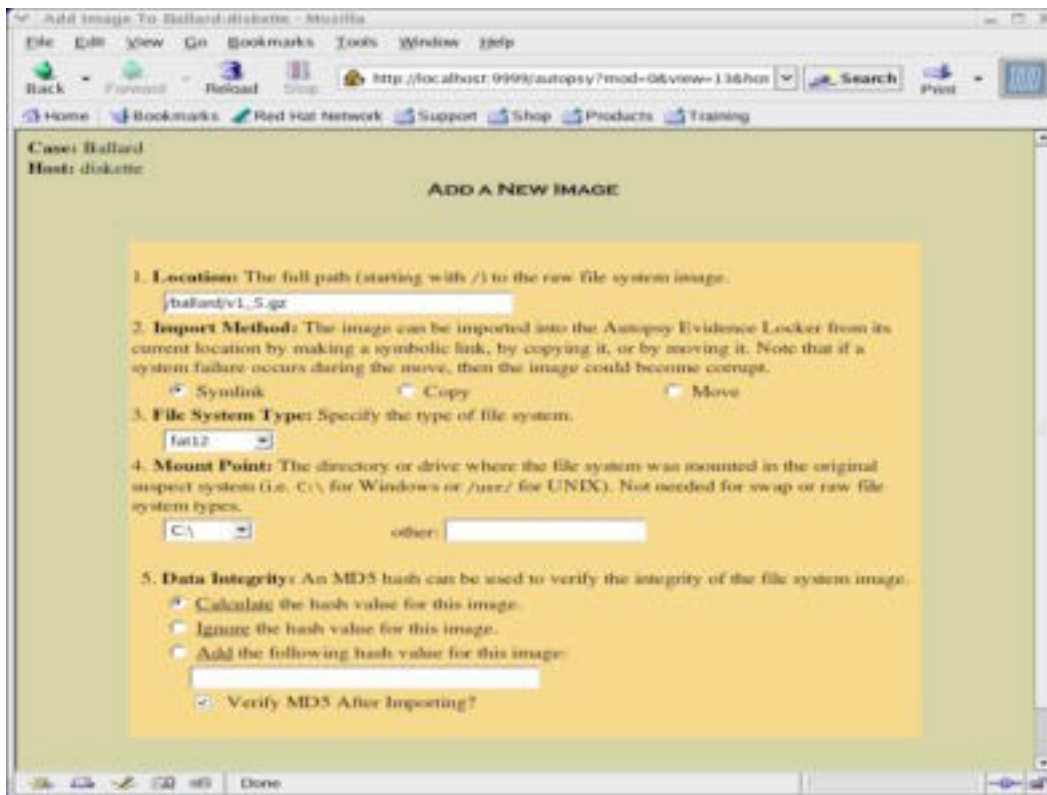
Now that I determine that the file really isn't compressed and is a FAT12 format, I had another problem. This file name is not the same as that specified in the chain of custody provided. I ran an md5 on the file. The md5 checksum is the same as that provided in the chain of custody.



```
root@LinuxForensics:/ballard
File Edit View Terminal Go Help
[root@LinuxForensics ballard]# md5 v1_5.gz
d7641eb4da871d980adbe4d371eda2ad      v1_5.gz
[root@LinuxForensics ballard]#
```

If I encountered this problem in a real-world investigation, I would report the problem to the person before myself who had the evidence. "If the information collected during an investigation should be used in legal proceedings, the prosecution is responsible for proving that what is presented in court is what was originally collected" (Mandia 92). It could be argued in this case that the evidence has been tampered with since the file names are not the same. Since the MD5 checksums are the same, it may be possible the evidence won't be thrown out. Regardless, in a real-world investigation, it is crucial that the chain of custody is maintained.

After downloading the image and verifying the MD5 checksums, I was ready to use Autopsy (<http://www.sleuthkit.org/autopsy/desc.php>). The first step was to create a new case. I called the case "Ballard." I then loaded the v1\_5.gz image into Autopsy. I selected the file type as FAT12 and the mount point as C:. Autopsy doesn't have a value for A:.



For item 5 on this screen, I selected the option for an MD5 value be taken after the image is imported. This verifies the integrity of the image after Autopsy has loaded it.

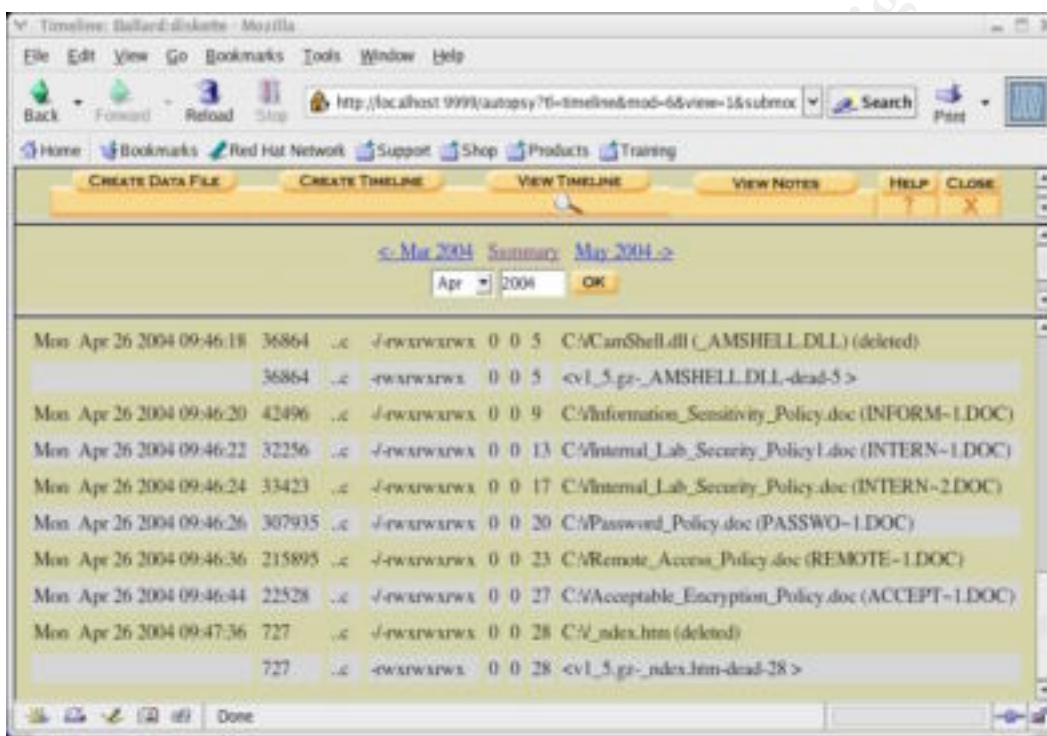


As seen above, the MD5 value calculated after Autopsy loaded the image is the same as the MD5 value from the chain of custody form.

The next step in my analysis of the image was to create a timeline within Autopsy with the following steps:

- Create Data File
  - Select: C:\ /ballard/v1\_5.gz

- Select all data types
- Select to create an MD5
- Create
  - Starting date: not selected
  - Ending date: not selected
- View Timeline
  - The earliest event found was February 3, 2001. The file “CamShell.dll” was deleted.
  - Selected Summary. I noticed most of the events occurred on April 2004.



In the above screen shot, on April 26, 2004 starting at 9:46:18, is the list of the files listed created, as indicated by the “c” before the permissions settings.

The files in the image were:

- Camshell.dll
  - 36864 bytes at create time on April 26, 2004, 09:46:18
- V1\_5.gz-AMSHHELL.DLL-dead-5
  - 36864 bytes at create time on April 26, 2004, 09:46:18
- Information\_Sensitivity\_Policy.doc
  - 42496 bytes at create time on April 26, 2004, 09:46:20
- Internal\_Lab\_Security\_Policy1.doc
  - 32256 bytes at create time on April 26, 2004, 09:46:22
- Internal\_Lab\_Security\_Policy.doc
  - 33423 bytes at create time on April 26, 2004, 09:46:24
- Password\_Policy.doc
  - 307935 bytes at create time on April 26, 2004, 09:46:26

- Remote\_Access\_Policy.doc
  - 215895 bytes at create time on April 26, 2004, 09:46:36
- Acceptable\_Encryption\_Policy.doc
  - 22528 bytes at create time on April 26, 2004, 09:46:44
- \_ndex.htm
  - 727 bytes at create time on April 26, 2004, 09:47:36
- v1\_5.gz-\_ndex.htm-dead-28
  - 727 bytes at create time on April 26, 2004, 09:47:36

I then selected “File Type” within Autopsy. This sorts the files based on what type of file they are. Plus, it gives me the inode number associated with each file.

An “Extension Mismatch” was found with CamShell.dll. Autopsy indicated this file is really an HTML document, but has a .dll extension. The six documents found in the image were indicated by Autopsy as in fact being documents and listed all of them under the “documents Category.” The remaining file \_ndex.html was listed under the “text Category” as being an HTML file.

The next step was going to the “File Analysis” section of Autopsy. I individually selected each file and selected “Strings Display.” After viewing the output of running strings on each file, I extracted each file from the image. I did this by going to the “Meta Data” section of Autopsy. I typed in the inode number for each file and extracted it. The steps in this paragraph are straight forward once in Autopsy. To get to this section in Autopsy, you select your case, then the image. After selecting the image, the window that comes up shows the image you have selected and at the bottom you’ll see an option for timeline. On this window, just click ok to open the image. The next window has the options indicated by quotes in this paragraph across the top of the Autopsy window.

The results of running strings and extracting the files are combined below:

- Camshell.dll
  - First several lines of the file contained HTML code for Macromedia (<http://sdc.shockwave.com/shockwave/download/download.cgi?>) shockwave.
  - The rest of the file appeared to be an actual .dll file. A .dll stands for dynamic link library. This is used by and associated with programs that run under the Windows operating system. I gathered the following information to help in my search for the program this .dll file was associated with.
    - Twisted Pear Productions produced the .dll
    - <http://www.camouflage.freeseve.co.uk> I tried to visit this website, but the name won’t resolve.
    - Documents\VB\Programs\Camouflage\Shell\Camouflage.vbp The .vbp file extension suggests this is some sort of Microsoft Visual Basic program.
    - Copyright 2000-2001
    - Product Version 1.01.0001
    - OriginalFilename Camshell.dll

- Information\_Sensitivity\_Policy
  - Nothing suspicious in strings display; saw normal document headers and footers
  - Opened the file in OpenOffice.org Writer (<http://www.openoffice.org/>) without any problems
  - Read through the document thoroughly to look for any clues; didn't see any
- Internal\_Lab\_Security\_Policy1
  - Nothing suspicious in strings display; saw normal document headers and footers
  - Opened the file in OpenOffice.org Writer without any problems
  - Read through the document thoroughly to look for any clues; didn't see any
- Internal\_Lab\_Security\_Policy
  - Strings display showed data after the normal document footer; couldn't read the data
  - Opened the file in OpenOffice.org Writer without any problems
  - Read through the document thoroughly to look for any clues; didn't see any
- Password\_Policy
  - Strings display showed data after the normal document footer; couldn't read the data
  - Opened the file in OpenOffice.org Writer without any problems
  - Read through the document thoroughly to look for any clues; didn't see any
- Remote\_Access\_Policy
  - Strings display showed data after the normal document footer; couldn't read the data
  - Opened the file in OpenOffice.org Writer without any problems
  - Read through the document thoroughly to look for any clues; didn't see any
- Acceptable\_Encryption\_Policy
  - Nothing suspicious in strings display; saw normal document headers and footers
  - Opened the file in OpenOffice.org Writer without any problems
  - Read through the document thoroughly to look for any clues; didn't see any
- \_index.html
  - Strings display showed HTML code for Macromedia shockwave
  - No suspicious data after the HTML code

At this point I had enough information from the camshell.dll file to start my search on the Internet. Based on the information above, I was going with the assumption that whatever program uses the camshell.dll file was used to embed the data at the end of the Internal Lab Security Policy, Password Policy, and Remote Access Policy documents.

A Google search for camshell.dll produced 1 hit. The website found, (<http://www.tranceaddict.com/forums/archive/topic/79627-1.html>), discusses a program called “Camouflage” that hides MP3’s within other files. It didn’t mention hiding documents within documents, though. I decided to pursue this avenue. I found a site, (<http://scifi.pages.at/yoda9k/cf/info.htm>), to download Camouflage 1.0.

I downloaded Camouflage 1.0 and extracted it. After viewing the “camouflage.txt” file, I realized this isn’t the Camouflage associated with the camshell.dll file from the diskette image. There is no mention of “Twisted Pear Productions.”

I downloaded Camouflage 1.04 from (<http://www.zcu.cz/ftp/pub/win/simtelnet/win95/secfile/camou104.zip>). I read the Readme.txt file that was included in the zip file. It listed the website: <http://www.camouflage.freemove.co.uk>, the same one that was listed in the camshell.dll file from the diskette image.

Using a separate test PC that is isolated on the network, I installed Camouflage 1.04. Prior to this, I searched the Internet for the source code of Camouflage 1.04. I could not find the source. This would have been the ideal way of analyzing what Camouflage 1.04 does. Another method for analyzing what Camouflage 1.04 does is to reverse engineer the program.

Upon installing it, I discovered Camouflage 1.04 includes a file called “CamouflageShell.dll.” No camshell.dll. I viewed the contents of CamouflageShell.dll. Its file version is 1.00. I uninstalled Camouflage 1.04 from the test PC and verified it was completely removed.

I downloaded Camouflage 1.21 from (<http://camouflage.unfiction.com/Download.html>). I installed this on the test PC. As with Camouflage 1.04, I searched the Internet for the source code of Camouflage 1.21. I could not find the source. Again, this would have been the ideal method for analyzing what Camouflage 1.21 does.

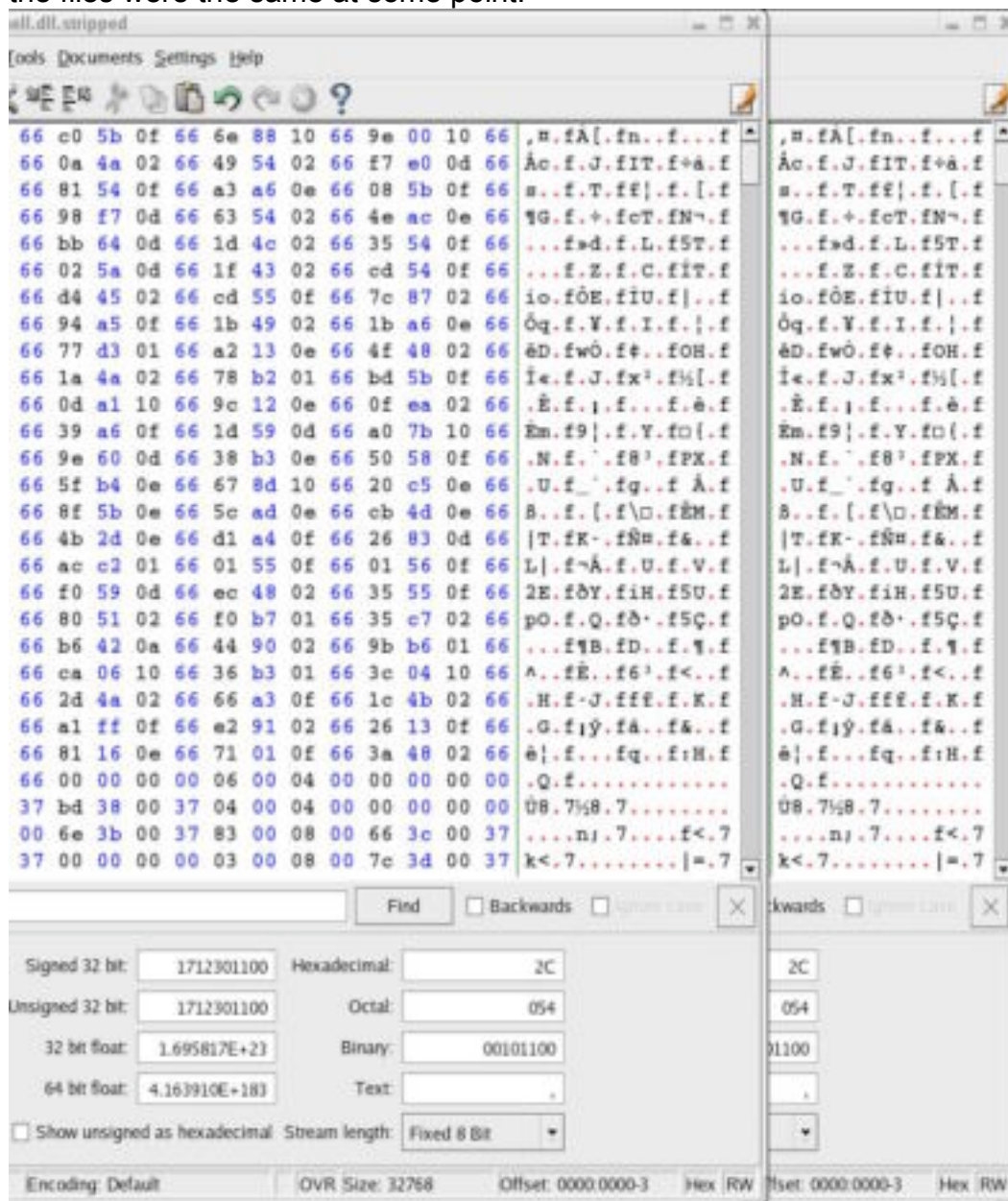
Upon installing it, I discovered Camouflage 1.21 includes a file called “CamShell.dll” located in the c:\Program Files\Camouflage directory. I viewed the contents of CamShell.dll. Its file version is 1.01.001, which is the same as the camshell.dll file found on the diskette image.

The next step in my analysis was to verify if the camshell.dll file from the diskette image came from Camouflage 1.21. From this point, I’ll refer to the camshell.dll from the diskette image as “camshell.dll.img.” I’ll refer to the camshell.dll from the installation of Camouflage 1.21 as “camshell.dll.orig.”

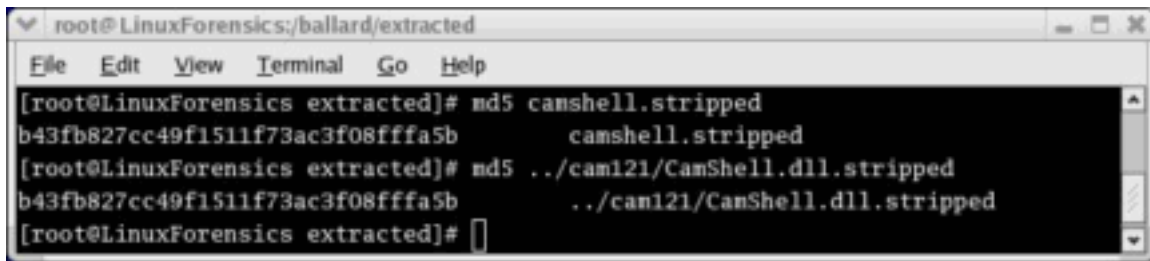
I opened each one in a hex editor on the forensics workstation. Right away I noticed differences. Camshell.dll.img has HTML code at the beginning of it. Camshell.dll.orig has a message stating, “This program cannot be run in DOS mode.” Camshell.dll.img does not include this message anywhere. This message is often found in Windows programs that cannot be run from within a DOS window.

Further comparison I found a lot of similarities. Both files included the reference to Twisted Pear Productions and the URL <http://www.camouflage.freemove.co.uk>. They also included the same path

Documents\VB\Programs\Camouflage\Shell. I didn't do an MD5 comparison, because I knew the MD5 values would be different as the Camshell.dll.orig included the reference to DOS mode that the Camshell.dll.img did not contain. However, because of the similarities I found, using a hex editor I discovered the starting point where both files have exact data. I wanted to test my theory that the files were the same at some point.



I cut out all data above this point and saved each camshell.dll. Then I ran an MD5 comparison of the two.

A screenshot of a terminal window titled 'root@LinuxForensics:/ballard/extracted'. The terminal shows the following commands and output:

```
[root@LinuxForensics extracted]# md5 camshell.stripped
b43fb827cc49f1511f73ac3f08fffa5b      camshell.stripped
[root@LinuxForensics extracted]# md5 ../cam121/CamShell.dll.stripped
b43fb827cc49f1511f73ac3f08fffa5b      ../cam121/CamShell.dll.stripped
[root@LinuxForensics extracted]#
```

As you can see, the MD5 checksums are exact, showing that the files from the point where I extracted the data within the hex editor to the end are identical matches. The likelihood these two files are different is nearly zero, since I personally verified portions of their contents match and the MD5 sums match, which is highly unlikely for any two different files.

I did have to strip out data to the point where both of the camshell.dll files matched. This is because camshell.dll.img was partially overwritten with HTML code that appears to have come from \_ndex.html. I used the same process as above to compare the contents of the \_ndex.html and the beginning contents of the camshell.dll.img. I found where the files matched, cut out the remaining data in the camshell.dll.img. An MD5 comparison showed the checksums as being exact.

A theory is the suspect attempted to overwrite the camshell.dll.img file with the \_ndex.html file. The entire camshell.dll.img file was not overwritten as the \_ndex.html file is significantly smaller. The timeline doesn't clearly show this is what occurred. However, it is important to note that an MD5 comparison showed an exact match.

During my search for Camouflage, I came across the website (<http://guillermi2.net/stegano/camouflage/>) where the author talks about breaking into files created by Camouflage. It was from this page that I learned about tools to break into Camouflage files.

I downloaded the tool SetecAstronomy.pl (<http://packetstormsecurity.nl/crypt/stego/camouflage/>), which "...is a Perl script that can search files to identify where data has been hidden..." with Camouflage. If it detects such data, it displays how many hidden files are contained in the nonhidden file. "If a password was used to "protect" the hidden data, the password is printed out."

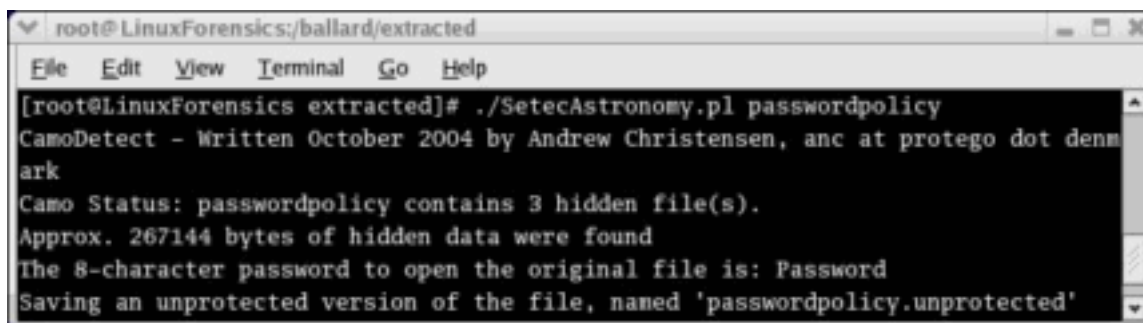
I ran the tool against the documents I found via the "strings display" in Autopsy that contained data after the document footer.

The document Internal\_Lab\_Security\_Policy.doc included one hidden file. There was no Camouflage password to extract the hidden file. If there was a password, SetecAstronomy.pl would have indicated this and displayed what the password was.

The document Remote\_Access\_Policy.doc included one hidden file. The Camouflage password was "Remote."

The document Password\_Policy.doc included three hidden files. The Camouflage password was "Password." The screen shot below shows the output of SetecAstronomy.pl ran against Password\_Policy.doc. When

Password\_Policy.doc was extracted from the image, it was saved as passwordpolicy.

A screenshot of a terminal window titled 'root@LinuxForensics:/ballard/extracted'. The window has a menu bar with 'File', 'Edit', 'View', 'Terminal', 'Go', and 'Help'. The terminal shows the command `./SetecAstronomy.pl passwordpolicy` and its output: `CamoDetect - Written October 2004 by Andrew Christensen, anc at protego dot denmark`, `Camo Status: passwordpolicy contains 3 hidden file(s).`, `Approx. 267144 bytes of hidden data were found`, `The 8-character password to open the original file is: Password`, and `Saving an unprotected version of the file, named 'passwordpolicy.unprotected'`.

Since I had great certainty that camshell.dll on the diskette image came from Camouflage 1.21, I used Camouflage 1.21 on the test PC to extract the hidden files.

After copying the 3 documents to the test PC and verifying their MD5 with the MD5 of each document on the forensics workstation, I proceeded to use Camouflage to extract the hidden files.

I right-clicked on Password\_Policy.doc. I chose "Uncamouflage." A box came up where I typed in the password, Password, which was discovered by the SetecAstronomy perl script. This extracted the following 2 .jpg's: PEM-fuel-cell-large.jpg, which was a picture of the design of a PEM Fuel Cell, and Hydrocarbon%20fuel%20cell%20page2.jpg, which was a picture of some chemical makeup and text describing it. A .gif was extracted: pem\_fuelcell.gif, which was a picture of the flow of chemical reactions to produce electricity.

Uncamouflage of Internal\_Lab\_Security\_Policy.doc yielded: Opportunity.txt The contents of this file were:

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

Uncamouflage of Remote\_Access\_Policy.doc yielded a Microsoft Access Database file: CAT.mdb. I opened the extracted file in Microsoft Access. According to the table name, this is a list of clients and their contact information. Furthermore, it included each of the client's account login name and account password.

Mr. Leszczynski attempted to steal company proprietary information from Ballard Industries. This information included design pictures of the PEM Fuel Cell, a client database, and a text file indicating his willingness to sell the information for 5 million. He was not successful in stealing the information. According to the timeline of the image, it shows Mr. Leszczynski creating the files

on April 26, 2004. The last file was created at 09:46:18. The diskette was seized on April 26, 2004 at approximately 4:45pm. This suggests that April 26, 2004, was the first day Mr. Leszczynski attempted to leave the R&D lab with the diskette containing the camouflaged files with company proprietary information.

I would suggest the Systems Administrators search every machine to determine if Mr. Leszczynski installed Camouflage on other machines. The default installation of Camouflage installs it to the Program Files\Camouflage directory. The Administrators could also search for the Camouflage executable if Mr. Leszczynski installed it in a non-default directory. Another recommendation is they search the systems for files that may include camouflaged files. One way to determine this is to look for unusually large Microsoft Word documents, as was the case with the documents on the diskette image.

The evidence does not support that Camouflage was run from the diskette. When I installed Camouflage onto a test PC, it installs the following files: Camouflage.exe, CamShell.dll, Readme.txt, and Uninst.isu. The timeline from the diskette does not show any other files from Camouflage as having ever been on the diskette other than CamShell.dll. I performed a test to camouflage a document with Camouflage, to see how the program works, as I couldn't locate the source code for Camouflage.

I right-clicked on the test file I wanted to camouflage. I chose: Camouflage. The next window showed the file I wanted to camouflage. The window after that prompted me to input the path and filename, document1, I wanted to camouflage my test file into. The next window prompted me for the path and filename I wanted to call the new file that included the test file and document1. The final window prompted me to input a password, which is an option. It is possible to camouflage without inputting a password.

I tested this with camouflaging to a diskette. It operates in the same manner as camouflaging files on a hard drive. It doesn't appear that Camouflage copies the CamShell.dll file to a diskette when camouflaging to a diskette.

A full forensic investigation on Mr. Leszczynski's workstation in the R&D lab is needed. I would look for Camouflage as being installed. I would also compare timelines on the workstation with those on the diskette image. It is necessary to gather all possible evidence. The forensic investigation I performed on the diskette image from the diskette found in Mr. Leszczynski's possession only proves that the diskette contained company proprietary information. Granted, there is a text document with Mr. Leszczynski's name on it. However, there is no proof Mr. Leszczynski knew about this camouflaged information on the diskette. It is possible that Mr. Leszczynski is being setup by another company employee. The scope of the investigation may increase if no malicious activity is detected on Mr. Leszczynski's workstation.

Once it is proved who intentionally was trying to steal company proprietary information, The Economic Espionage Act of 1996 (<http://www.usdoj.gov/criminal/cybercrime/ipmanual/08ipma.htm#VIII.B>.) is applicable with this case. 18 U.S.C. § 1832 specifically applies. This provision makes it illegal for someone to attempt to or actually sell trade secrets to another

entity where the money gained will not be benefiting the owner of the trade secrets.

The evidence gathered in this investigation clearly shows this was the intent. First, I gathered .jpg images of company proprietary information regarding the fuel cells. Second, I gathered a text document stating the intent to sell the trade secrets for 5 million.

Another thing to note is this provision states it is illegal to receive such trade secret information if you know it was stolen. Therefore, the scope of this investigation increases even further beyond everyone who works in and has access to the R&D lab. It is important for law enforcement to find whom this stolen information intended to be delivered to.

The penalty for violating this law is imprisonment “for up to 10 years and fined \$250,000 or both.” The penalty is even more severe if it is discovered that Ballard’s competitor, Rift, Inc., knew that the information they were receiving was stolen. In that case, Rift, Inc., if found guilty, could be fined “\$5 million for violating 18 U.S.C. § 1832.”

Other violations include violating Ballard’s internal security policies for taking information out of the R&D lab. The penalty for violating Ballard’s internal security policies is any disciplinary action up to termination of employment.

As I discussed earlier, there may be a problem with this investigation and upholding the evidence in court, since the chain of custody form does not match the diskette image name I received. In that case, it is important going forward as the investigation broadens that mistakes such as this are not repeated. Plus, the investigators need to keep in mind that the analysis of this diskette image may be thrown out of court and therefore need to find other evidence to support the notion that company proprietary information was stolen.

© SANS Institute 2005

## Part 2 – Option 1 – Perform Forensic Analysis on a system

My company was working on remediation efforts for a security incident that was declared two weeks earlier for virus activity. I declared a security incident for virus activity that was identified. This was detected in the IDS (Intrusion Detection System) and firewall logs. Declaring a security incident mobilized the necessary personnel, in this case the desktop support group, and communicated with IT management the steps undertaken for containment and clean up. The exposure was small with approximately 2% of the total workstations in my company's environment were infected. The environment is complex with hundreds of remote sites. All of the remote sites connect into the data center via frame relay for access to Company apps and Internet access.

The desktop technicians performing the virus clean up efforts ran the virus removal tools, which included McAfee VirusScan (<http://www.mcafeesecurity.com/us/products/mcafee/antivirus/category.htm>) and Trend Micro Housecall (<http://housecall.trendmicro.com/>). In addition, the technicians ran a spyware/adware removal tool, Ad-Aware (<http://www.lavasoft.com/software/adaware/>). These tools were installed, if not installed already, on the infected systems, which all run Microsoft Windows 2000.

On October 18, 2004, at 8:03am, one of the desktop technicians who was working on the virus clean up efforts contacted me. The technician discovered a non-standard directory in the \winnt\system32 directory, using the tools mentioned above.

I activated the Incident Response Plan. The IR Plan was developed and approved by IT senior management over a year ago. The first step of the IR Plan is the Determination Phase, in which the IR coordinator determines if malicious activity has occurred. In this case, I gathered as much detail as possible from the desktop technician as to what was discovered. The second step of the IR plan is the Initial Response Phase in which I notified management. Due to the size of the Company, the IR Plan calls for the 24x7 NOC to coordinate calling all needed individuals and schedule a conference call. The 24x7 NOC is utilized because they have off-hours contact information for everyone in the IT department. Instead of utilizing them for just off-hours security incidents, it is less confusing to use them for all security incidents.

A conference call with the VP of IT Operations, the Manager of Desktop Support, the team lead for the desktop support, and the desktop support technician was conducted. In this meeting, the first thing I stated is that all virus clean up efforts on the system that contained the non-standard directory needs to stop right away to preserve the system's state and prevent the further destruction of evidence.. Furthermore, as I didn't have enough detail yet as to what we were dealing with, I requested that all desktop support efforts for the remote location stop, including the virus clean up and any other desktop maintenance. I gathered further information from the desktop technician, including hostname, current IP address, system owner, and phone number. I also asked the desktop

technician everything he did to the system, so I could gather insight as to how the evidence has been touched on this system.

This directory was “PAL.” A Google search shows this directory and the contents belonging to a commercially available key logger, Key Log Pro (<http://www.keylogpro.com>). At this point, the technician contacted me, on October 18, 2004, at 8:00am.

On October 19, 2004 I contacted the end user affected to interview her about the incident. She said that she had strong reasons to believe someone in her office had installed the key logger. She said in the previous week, she and her manager had received an email from this particular individual (referred to as “suspect individual” for the rest of this paper), which contained a password she and her manager use to password protect Excel documents. She stated there is no reason the suspect individual should know this password.

I started out running forensics on the victim’s machine (victimpc1) and then decided to run forensics on her manager’s machine (victimpc2) as well. I expanded the scope of the investigation to include her manager’s machine, since he had received the email as well with the password used to protect Excel documents. This was a challenge as the office location for both individuals is over 1,000 miles away from the office where I work. Plus, the network connection between us is a 128k frame relay.

Due to the bandwidth constraints and the time constraints to determine who installed the key logger, imaging the victim’s machine remotely was not an option. Also, having the hands-on initial response on the console was not possible. An ideal environment is to have physical access to the compromised system where I can use the system console and the local cdrom drive to run various forensic tools. I describe throughout the remainder of this document steps I took to compensate for this challenge of gathering evidence on remote systems while maintaining forensic soundness.

### Analysis of victimpc1

I gathered information from the victim as to what her machine name was and what the current IP address was. The company uses DHCP for addressing workstations. I wanted to verify I was connecting to victimpc1, the name I’m choosing to use for the purposes of this paper.

As local administrator on the victim’s machine, I ran the following trusted utilities from my Incident Response CD. The IR CD was in my laptop drive, shared to victimpc1. Once I was logged into victimpc1, I connected to the IR CD share, which then enabled me to run the trusted commands. If the system were located at the company location where I am, I would have imaged the machine to preserve system state. Given both victim machines were located in a remote location accessible only via a 64k frame-relay link, imaging the systems over the network was not an option. Plus, there aren’t any IT personnel located at this remote location who could have imaged the machine to another hard drive and sent the 2nd hard drive to me for analysis.

From victimpc1:

C:\ net use o: [\\laptop\IRCD](#)

Furthermore, I used netcat (<http://netcat.sourceforge.net/>) to tunnel the output from the trusted commands. A netcat listener was setup on the forensic workstation to gather the data.

```
# nc -l -p 31337 > /incident1/victimpc1/mtime (listener on the forensic workstation to receive the command for collecting Modification time)
```

```
dir /t:w /a /s /o:d | nc 10.10.10.10 31337 -w 3 (command for collecting Modification time on victimpc1 and piping the output to the forensic workstation)
```

For each command I ran on victimpc1, I stopped the previous listener, started a new one with the input redirected to a file name corresponding with the command to be run on victimpc1.

- `dir /t:w /a /s /o:d` collects file system timestamps for Modification Times
- `dir /t:a /a /s /o:d` collects file system timestamps for Access Times
- `dir /t:c /a /s /o:d` collects file system timestamps for Creation Times
  - MAC times = Modification, Access and Creation Times

It was important to collect MAC times before running any other commands to ensure that other investigative tools didn't change the MAC times before I had a chance to analyze the timeline. MAC times were gathered starting at the `\winnt\System32\PAL` and all sub directories and files.

- `fport` (<http://www.foundstone.com/resources/proddesc/fport.htm>). This utility shows the running process and the port it is using. Result: No suspicious processes were shown.

FPort v1.33 - TCP/IP Process to Port Mapper  
Copyright 2000 by Foundstone, Inc.  
<http://www.foundstone.com>

Pid	Process		Port	Proto	Path
416	svchost	->	135	TCP	C:\WINNT\system32\svchost.exe
8	System	->	139	TCP	
8	System	->	445	TCP	
736	residentagent	->	1029	TCP	C:\Program Files\LANDesk\Shared Files\residentagent.exe
880	MSTask	->	1030	TCP	C:\WINNT\system32\MSTask.exe
8	System	->	1033	TCP	
8	System	->	1303	TCP	
8	System	->	1350	TCP	
1136	wuser32	->	1761	TCP	C:\LDClient\wuser32.exe
1136	wuser32	->	1762	TCP	C:\LDClient\wuser32.exe
1760	aim	->	5180	TCP	C:\Program Files\AIM95\aim.exe
1892	DWRCS	->	6129	TCP	C:\WINNT\SYSTEM32\DWRCS.EXE
736	residentagent	->	9594	TCP	C:\Program Files\LANDesk\Shared Files\residentagent.exe
680	tmcsvc	->	33354	TCP	C:\LDClient\tmcsvc.exe

```

1448 MsgSys      -> 38292 TCP    C:\WINNT\system32\MsgSys.EXE

8      System    -> 137  UDP
8      System    -> 138  UDP
8      System    -> 445  UDP
232    lsass     -> 1044 UDP    C:\WINNT\system32\lsass.exe
1760   aim       -> 1083 UDP    C:\Program Files\AIM95\aim.exe
736    residentagent -> 9595 UDP    C:\Program Files\LANDesk\Shared
Files\residentagent.exe
680    tmcsvc    -> 33354 UDP    C:\LDCClient\tmcsvc.exe
680    tmcsvc    -> 33355 UDP    C:\LDCClient\tmcsvc.exe
1448   MsgSys    -> 38037 UDP    C:\WINNT\system32\MsgSys.EXE
632    pds       -> 38293 UDP    C:\WINNT\system32\cba\pds.exe
528    cvpnd     -> 62515 UDP    C:\Program Files\vpn\cvpnd.exe
528    cvpnd     -> 62517 UDP    C:\Program Files\vpn\cvpnd.exe
528    cvpnd     -> 62519 UDP    C:\Program Files\vpn\cvpnd.exe
528    cvpnd     -> 62521 UDP    C:\Program Files\vpn\cvpnd.exe
528    cvpnd     -> 62523 UDP    C:\Program Files\vpn\cvpnd.exe
528    cvpnd     -> 62524 UDP    C:\Program Files\vpn\cvpnd.exe

```

- netstat -- This command shows the open source port and destination port connections. The purpose of running this command is to determine if the key logger or another malicious process has an open connection to another system. For example, the key logger may not only be producing log files on the local hard drive, but may also be piping the output to another system. Result: The results of running this command did not support the above stated possibilities. The only ports I saw open were Windows NetBIOS port 139 and port 445.
- pslist (<http://www.sysinternals.com/ntw2k/freeware/pslist.shtml>). This utility shows the running processes on the system. Result: No suspicious processes were discovered.

PsList 1.26 - Process Information Lister  
 Copyright (C) 1999-2004 Mark Russinovich  
 Sysinternals - [www.sysinternals.com](http://www.sysinternals.com)

Process information for victimpcl:

Name	Pid	Pri	Thd	Hnd	VM	WS	Priv
Idle	0	0	1	0	0	16	0
System	8	8	36	206	1676	92	24
SMSS	148	11	6	33	5256	84	1076
WINLOGON	168	13	17	417	38112	2168	5992
services	220	9	37	749	49552	7612	8720
svchost	416	8	9	362	24596	2348	1640
SPOOLSV	452	8	12	181	35372	1652	3236
Ati2evxx	504	8	2	33	12892	360	268
Avsynmgr	516	8	4	100	27260	484	1264
VSStat	824	8	2	66	26044	1108	1300
vshwin32	896	8	7	172	60208	520	7240
Avconsol	988	8	2	63	29068	424	1536
cvpnd	528	8	4	143	31704	1336	2016
svchost	548	8	30	602	48792	3032	8632
LocalSch	596	8	6	137	26976	884	1136
PDS	632	8	5	138	32772	224	1616

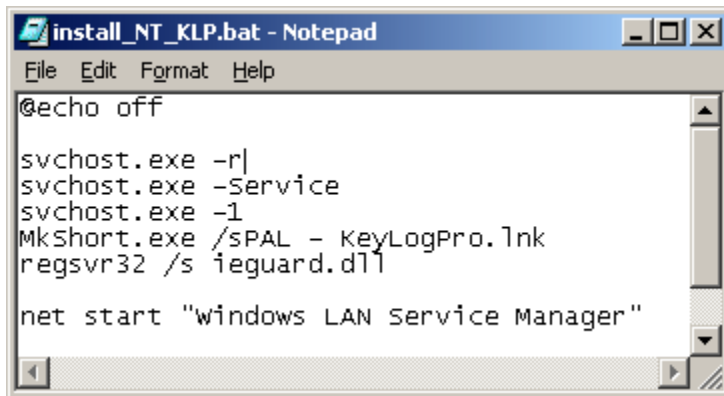
QIPCLNT	668	8	2	39	12364	260	328
tmcsvc	680	8	10	70	30004	364	3424
KodakCCS	696	8	3	62	23416	156	832
residentAgent.e	736	8	5	147	28676	164	1360
NetCfgSv	756	8	3	81	17328	492	600
ptssvc	836	8	3	99	17936	364	620
regsvc	872	8	2	29	9704	172	276
mstask	880	8	8	134	26832	3512	1436
ScsiAccess	912	8	2	27	11176	176	276
sp_SWIns	940	8	3	81	22776	508	720
stisvc	968	8	4	56	12684	180	484
svchost	1056	8	3	54	26204	420	800
winmgmt	1088	8	4	237	29496	1132	2212
mspmbspv	1100	8	2	52	12160	100	496
svchost	1116	8	6	176	28936	752	4040
wuser32	1136	8	16	216	49792	476	2408
MSGSYS	1448	8	13	158	37516	436	1676
svchost	1176	8	6	691	27908	1008	3700
MCSHIELD	1360	13	16	116	38280	4228	5880
DWRCS	1892	8	10	162	39084	5960	5368
LSASS	232	9	15	337	27332	1908	2392
SoftMon	1572	8	3	25	17376	680	508
csrss	172	13	10	531	20224	1620	1364
mcagent	436	8	3	109	33368	1172	1512
explorer	440	8	8	372	64636	9300	7964
atiptaxx	1372	8	2	83	25084	652	896
point32	1472	8	3	56	18284	448	688
hpcmpmgr	1480	8	3	173	36084	516	2008
hpwuSchd	1492	8	1	21	13984	252	244
cthi	1616	8	4	164	32576	1132	2128
HXIUL	1620	8	5	279	37188	792	4412
hpotdd01	1640	8	4	108	32392	1428	1580
yeqffet	1692	8	4	183	39020	732	1940
ViewMgr	1712	8	2	136	28088	2676	1312
TBPS	1716	8	7	214	45356	2228	2660
PIB	1784	24	2	72	29308	296	1308
rundll32	1724	8	3	103	32624	3532	5460
WToolsA	1748	8	7	167	50388	2068	2604
WSup	1700	8	3	110	42128	1328	2136
aim	1760	8	10	399	84732	6140	16100
AccessMgr	1852	8	3	151	81864	1600	7728
CMD	2032	8	1	23	11348	1044	332
pslist	1676	13	2	86	15592	1432	732

- **psloggedon** (<http://www.sysinternals.com/ntw2k/freeware/psloggedon.shtml>). This utility shows who is logged on locally and who is logged on via resource shares. Result: I saw myself logged on as local administrator.
- **psservice** (<http://www.sysinternals.com/ntw2k/freeware/psservice.shtml>). This utility is a “local and remote services viewer/controller”. Result: Looking through the output initially didn’t show anything out of the ordinary. However, a little further into the investigation shows the service “Windows LAN Service Manager” is the service the key logger creates and runs under. I discovered that when I analyzed the files in the PAL directory.

```
SERVICE_NAME: Windows LAN Service Manager
DISPLAY_NAME: Windows LAN Service Manager
```

```
(null)
      TYPE           : 110 WIN32_OWN_PROCESS INTERACTIVE_PROCESS
      STATE          : 4   RUNNING
                        (STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
      WIN32_EXIT_CODE : 0   (0x0)
      SERVICE_EXIT_CODE : 0   (0x0)
      CHECKPOINT      : 0x0
      WAIT_HINT       : 0x0
```

The installation file, install\_NT\_KLP.bat (below) shows the “Windows Lan Service Manager” service being linked to Key Log Pro.



```
install_NT_KLP.bat - Notepad
File Edit Format Help
@echo off
svchost.exe -r
svchost.exe -Service
svchost.exe -l
MkShort.exe /sPAL - KeyLogPro.lnk
regsvr32 /s ieguard.dll

net start "windows LAN Service Manager"
```

- dumpel  
(<http://www.microsoft.com/windows2000/techinfo/reskit/tools/existing/dumpel-o.asp>). This is a Microsoft tool for dumping Windows events into a file. I used it to dump the security, application and system logs. Result: I found instances of the suspect individual logged into the victim system. More investigation needed to determine if the access was legitimate or malicious, since the suspect individual performs IT duties.

An MD5 checksum was taken of all of the output and redirected into a file called: victimpc1md5. Then all of the output and victimpc1md5 files were tar'd up. Then an MD5 checksum was taken of the tar'd file and redirected into a file called: victimpc1tarmd5. The tar'd file and victimpc1tarmd5 file were saved to a CD-R. The CD-R was labeled, dated, and signed with my signature.

From the MAC times I was able to start putting together a time line of the PAL directory and all contents within the PAL directory. Under the PAL directory, the KLP directory was created on 07/07/2003 at 05:05pm. This gives us the starting time for when Key Log Pro was installed on victimpc1. The KLP directory contains the key logger installation and executable files. The output files from Key Log Pro are located in PAL\KLP\log\victimuser directory. According to MAC times, on 08/27/2003 at 01:06pm, I found 10 of the output text files from Key Log Pro accessed at the same time. This suggests some sort of bulk copy as so many files were accessed at the exact time.

After obtaining the MAC times for everything under the PAL directory, I zipped the PAL directory and copied it to my forensics workstation. I wanted to see if I could make a correlation between the output files of the key logger with

the password the victim and her manager use for password protecting Excel documents. I performed a grep <password name> on the output files from the key logger that capture the key strokes. These files were located in PAL\KLP\log\victimuser directory. I found the password in several Key Log Pro output files.

I went back to look at the Windows Security Events log that I obtained through the “dumpel” command described above. I attempted to correlate any logins with the time the PAL directory was created. No such correlation was possible. The events in the Windows Security Events log did not go back that far. Therefore, no correlation existed between when the suspect was logged in on victimpc1 and when the PAL directory was created or when the apparent bulk copy occurred.

### Analysis of victimpc2

I performed the same forensics analysis on the manager’s system, victimpc2. I gathered information from the victim as to what his machine name was and what the current IP address was. The company uses DHCP for addressing workstations. I wanted to verify I was connecting to victimpc2. I found Key Log Pro installed on his machine as well.

As with the analysis of victimpc1, I used the same process for running trusted commands from the Incident Response CD and using netcat to pipe the output to the forensic workstation.

- dir /t:w /a /s /o:d collects file system timestamps for Modification Times
- dir /t:a /a /s /o:d collects file system timestamps for Access Times
- dir /t:c /a /s /o:d collects file system timestamps for Creation Times
- fport (<http://www.foundstone.com/resources/proddesc/fport.htm>). Result: No suspicious processes were shown.

FPort v1.33 - TCP/IP Process to Port Mapper  
Copyright 2000 by Foundstone, Inc.  
<http://www.foundstone.com>

Pid	Process	Port	Proto	Path
416	svchost	-> 135	TCP	C:\WINNT\system32\svchost.exe
8	System	-> 139	TCP	
8	System	-> 445	TCP	
692	residentagent	-> 1053	TCP	C:\Program Files\LANDesk\Shared Files\residentagent.exe
776	MSTask	-> 1057	TCP	C:\WINNT\system32\MSTask.exe
8	System	-> 1069	TCP	
1672	aim	-> 1188	TCP	C:\PROGRA~1\AIM95\aim.exe
1672	aim	-> 1200	TCP	C:\PROGRA~1\AIM95\aim.exe
8	System	-> 1440	TCP	
8	System	-> 1512	TCP	
1024	wuser32	-> 1761	TCP	C:\LDCClient\wuser32.exe
1024	wuser32	-> 1762	TCP	C:\LDCClient\wuser32.exe
1820	DWRCS	-> 6129	TCP	C:\WINNT\SYSTEM32\DWRCS.EXE
692	residentagent	-> 9594	TCP	C:\Program Files\LANDesk\Shared Files\residentagent.exe

```

676   tmcsvd      -> 33354 TCP   C:\LDCClient\tmcsvd.exe
1192  MsgSys      -> 38292 TCP   C:\WINNT\system32\MsgSys.EXE

8     System     -> 137   UDP
8     System     -> 138   UDP
8     System     -> 445   UDP
232   lsass      -> 500   UDP   C:\WINNT\system32\lsass.exe
232   lsass      -> 1029  UDP   C:\WINNT\system32\lsass.exe
184   winlogon   -> 1061  UDP   \??\C:\WINNT\system32\winlogon.exe
448   spoolsv    -> 1070  UDP   C:\WINNT\system32\spoolsv.exe
1384  Explorer    -> 1105  UDP   C:\WINNT\Explorer.EXE
1564  Weather    -> 1140  UDP
C:\PROGRA~1\AWS\WEATHE~1\Weather.exe
1672  aim        -> 1192  UDP   C:\PROGRA~1\AIM95\aim.exe
692   residentagent -> 9595  UDP   C:\Program Files\LANDesk\Shared
Files\residentagent.exe
676   tmcsvd      -> 33354 UDP   C:\LDCClient\tmcsvd.exe
676   tmcsvd      -> 33355 UDP   C:\LDCClient\tmcsvd.exe
1192  MsgSys      -> 38037 UDP   C:\WINNT\system32\MsgSys.EXE
628   pds        -> 38293 UDP   C:\WINNT\system32\cba\pds.exe
552   cvpnd      -> 62515 UDP   C:\Program Files\vpn\cvpnd.exe
552   cvpnd      -> 62517 UDP   C:\Program Files\vpn\cvpnd.exe
552   cvpnd      -> 62519 UDP   C:\Program Files\vpn\cvpnd.exe
552   cvpnd      -> 62521 UDP   C:\Program Files\vpn\cvpnd.exe
552   cvpnd      -> 62523 UDP   C:\Program Files\vpn\cvpnd.exe
552   cvpnd      -> 62524 UDP   C:\Program Files\vpn\cvpnd.exe

```

- **Netstat Result:** As with the first pc, victimpc1, I did not see any open ports to suggest the key logger is piping its output to another system. No other open ports appeared suspicious.
- **pslist** (<http://www.sysinternals.com/ntw2k/freeware/pslist.shtml>). Result: No suspicious processes were discovered.

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

Process information for victimpc2:

Name	Pid	Pri	Thd	Hnd	VM	WS	Priv
Idle	0	0	1	0	0	16	0
System	8	8	34	264	1676	212	24
SMSS	140	11	6	33	5256	376	1076
CSRSS	164	13	10	467	19744	1592	1364
WINLOGON	184	13	18	438	41488	1840	7192
SERVICES	212	9	36	605	34980	3124	2952
svchost	416	8	8	318	22172	1796	1428
SPOOLSV	448	8	13	160	27032	1208	2632
PackethSvc	476	8	4	113	22924	208	968
Ati2evxx	528	8	2	33	12892	372	264
Avsynmgr	540	8	4	99	27272	616	1312
VSStat	788	8	2	66	26056	940	1304
vshwin32	984	8	7	172	60288	1208	7316
Avconsol	1160	8	2	68	29108	1264	1548
cvpnd	552	8	4	143	31720	1080	1940
svchost	568	8	28	459	38396	3240	3212
LocalSch	588	8	5	56	15812	932	440
PDS	628	8	5	137	32776	224	1640

QIPCLNT	664	8	2	39	12364	600	332
tmcsvc	676	8	10	70	30008	416	3436
residentAgent.e	692	8	5	147	28680	392	1380
regsvc	756	8	2	29	9704	768	276
mstask	776	8	6	151	25116	1164	1132
sp_SWIns	848	8	3	81	22776	724	724
stisvc	940	8	4	56	12684	844	484
svchost	972	8	3	54	26216	492	880
WinMgmt	1008	8	5	252	41244	300	3868
wuser32	1024	8	16	217	49836	496	2440
MSGSYS	1192	8	13	154	37520	444	1688
MCSHIELD	1088	13	16	114	38344	5572	5780
DNTUS26	1788	8	3	46	16108	1280	336
DWRCS	1820	8	10	159	36940	5904	3048
LSASS	232	9	15	358	28520	2016	2540
SoftMon	1072	8	3	30	18316	1592	660
explorer	1384	8	14	388	42668	3772	5160
CMD	336	8	1	23	11348	1052	332
pslist	1704	13	2	76	15592	1408	696
AccessMgr	1276	8	3	143	78028	3940	6432
atiptaxx	1372	8	2	78	25016	1724	1004
dpps2	1532	8	1	30	20132	1468	2852
loadqm	1552	8	5	167	28588	2032	1680
msnmsgr	1560	8	4	178	41176	2612	5296
Weather	1564	8	9	326	92552	6736	5616
MWSOEMON	1640	8	1	26	14000	936	272
rundll32	1656	8	3	124	32008	4712	5408
aim	1672	8	9	348	73340	5748	14176
aoltray	1748	8	2	54	17152	1632	820
ssonsvr	1436	8	1	15	10264	500	192

- **psloggedon**  
(<http://www.sysinternals.com/ntw2k/freeware/psloggedon.shtml>). Result: In addition to seeing myself logged in as local administrator, I saw the suspect individual's account logged in via a resource share. This doesn't necessarily indicate suspicious activity as one of the tasks performed by the suspect individual was backups.
- **psservice** (<http://www.sysinternals.com/ntw2k/freeware/psservice.shtml>). Result: As with victimpc1, I also saw the key logger service.

```
SERVICE_NAME: Windows LAN Service Manager
DISPLAY_NAME: Windows LAN Service Manager
(null)
TYPE          : 110 WIN32_OWN_PROCESS INTERACTIVE_PROCESS
STATE         : 4   RUNNING
               (STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
WIN32_EXIT_CODE : 0   (0x0)
SERVICE_EXIT_CODE : 0   (0x0)
CHECKPOINT     : 0x0
WAIT_HINT     : 0x0
```

- **dumpel**  
(<http://www.microsoft.com/windows2000/techinfo/reskit/tools/existing/dumpel-o.asp>). Result: I found instances of the suspect individual logged into the victim system. More investigation was needed to determine if the

access was legitimate or malicious, since the suspect individual performs IT duties.

An MD5 checksum was taken of all of the output and redirected into a file called: victimpc2md5. Then all of the output and victimpc2md5 files were tar'd up. Then an MD5 checksum was taken of the tar'd file and redirected into a file called: victimpc2tar.md5. The tar'd file and victimpc2tar.md5 file were saved to a CD-R. The CD-R was labeled, dated, and signed with my signature.

As with victimpc1, from the MAC times I was able to start putting together a time line. Under the PAL directory, the KLP directory was created on 07/07/2003 at 04:43pm, about 22 minutes before it was created on victimpc1. This gives us the starting time for when Key Log Pro was installed on victimpc2. According to MAC times, on 06/14/2004 at 11:21am, I found 21 of the output text files from Key Log Pro accessed at the same time.

As with victimpc1, after obtaining the MAC times for everything under the PAL directory, I zipped the PAL directory and copied it down to my forensics workstation. I performed a grep <password name> on the output files from the key logger that capture the key strokes. These files were located in PAL\KLP\log\victimuser2 directory. I found the password used in password protecting Excel documents in several of the Key Log Pro output files.

I went back to look at the Windows Security Events log that I obtained through the "dumpel" command described above. I attempted to correlate any logins with the time the PAL directory was created. No such correlation was possible. The events in the Windows Security Events log did not go back that far, as with victimpc1.

## Correlated results of victimpc1 and victimpc2

Key Log Pro was installed on both systems located in the same physical remote office, on the same date, and within 22 minutes of each other. A follow-up interview with the lady and her manager indicated some possibilities why they were targeted. The lady is a manager and her boss is a director. They both have access to the Company financial system and other Company proprietary systems and applications.

There was no indication the output files from the key logger were being directed to another system as indicated in the output from the netstat, fport, and pslist commands. Plus, these commands would have shown any backdoors, if they existed. No correlation existed when numerous key logger output files were accessed on victimpc1 and when they were accessed on victimpc2, meaning the apparent bulk copy commands did not occur on the same date or time.

Was not able to determine who installed Key Log Pro on either system. The Windows Events logs did not go back that far.

## Investigation of suspectpc1

On October 22, 2004, a conference call was held between the victim, the manager, the IT VP, and myself. I reported my findings that I couldn't determine

who installed the key logger on victimpc1 and victimpc2. I stated that evidence might exist on the suspect's PC.

Management made the decision to place the suspect individual on administrative leave. The suspect individual's machine was shipped to me for forensics analysis. This machine was the suspect's workstation used for his job with our company. It was running Windows 2000 and connected to the company network via Ethernet. I instructed the manager to power off the suspect PC and on how to package the machine. He was to wrap it in bubble wrap and wrap tape around it such that the tape went around all 4 sides. "When the person closing the packaging seals the container, a signature across the seal will indicate that it has not been opened by anyone other than an authorized person" ( Kruse II 11). I instructed the manager to write his name all across the tape.

In addition, the suspect individual's domain account was disabled, which also disables VPN access. All users in this remote office were instructed to change their domain account passwords.

### Physical evidence collection

On October 25, 2004, at 11:32am I went to the shipping/receiving area to receive the package containing the suspect individual's PC (suspectpc1). I recorded the Airborne Express airbill number, which matched the number the manager gave me. Before moving the package, I took digital pictures of the box and of the surrounding area. Approximately 5 minutes later, I had the package in my office. I took more digital pictures of the box. At 11:40am I opened the box, witnessed by the VP involved with this incident and an IT manager. With the box open, I took digital pictures of the contents of the package. As I unpackaged the PC, I took more digital pictures. Once unpackaged, I contacted the manager and verified the system model and serial number.

- System: Dell OptiPlex GX1 with 3.5" floppy drive, PCI Ethernet card, PCI modem card
- Serial Number: abc123 (made-up for purposes of this paper)
- Evidence Tag Number: incident1-001

After recording the system and serial number, I opened the computer case. I took digital pictures of the case open and of the hard drive. I removed the hard drive.

- Hard drive: Maxtor model 8324OD3
- Serial Number: def456 (made-up for purposes of this paper)
- Evidence Tag Number: incident1-002

After recording the hard drive model and serial number, I burned the digital pictures to a CD-R. I labeled the CD-R, dated it, and signed it with my signature.

### Imaging Suspect Hard Drive

I searched on the hard drive manufacturer's website for hard drive information.

([http://www.maxtor.com/portal/site/Maxtor/menuitem.5d2b41d3cef51dfe29dd10a191346068/?channelpath=/en\\_us/Support/Product%20Support/ATA%20Hard%20Drives/Desktop/DiamondMax%202160%20Ultra%20ATA](http://www.maxtor.com/portal/site/Maxtor/menuitem.5d2b41d3cef51dfe29dd10a191346068/?channelpath=/en_us/Support/Product%20Support/ATA%20Hard%20Drives/Desktop/DiamondMax%202160%20Ultra%20ATA)). I needed to find the correct jumper settings to make this drive ATA slave. Once I changed the jumpers, I placed the drive in the forensics workstation. Changing the jumpers enables the forensics workstation to boot from its own primary hard drive, but also enables it to access the hard drive from suspectpc1.

A good way to image a hard drive and prevent data change is to use a hardware write blocker. This prevents any changes to the data as it is being imaged from one hard drive to another. A hardware write blocker was not available to use in this case, but another method for preventing changes to the data was utilized. This procedure – mounting read only – is described a little later.

Before this incident, as I was putting together a forensics procedure document, I tested various ways to image a hard drive. The environment I was attempting to create was booting a suspect system off of a Knoppix-STD 0.1 CD (<http://www.knoppix-std.org/>) and imaging to an externally attached USB hard drive. I was using the “dcfldd” command to image the internal hard drive to the USB hard drive. It would get about 170MB completed before it completely locked up the system. I attempted various things, but was not successful. I sent an email to the contact on the website for Knoppix-STD outlining the problems I was experiencing.

The forensics workstation is a Dell OptiPlex. The operation system is RedHat 8.0. Several forensics tools are installed on the system, which will be described throughout the investigation.

After booting up the forensics workstation with the hard drive from suspectpc1 installed, the next step was getting the Linux operating system on the forensics workstation to recognize the hard drive. This is performed by using the “mount” command, but first I needed to see how the hard drive was laid out, specifically the partitions.

```
# fdisk -l /dev/hdb
```

This showed only 1 partition.

Next I ran the “mount” command.

```
# mount -ro,noexec /dev/hdb1 /mnt/hdb1
```

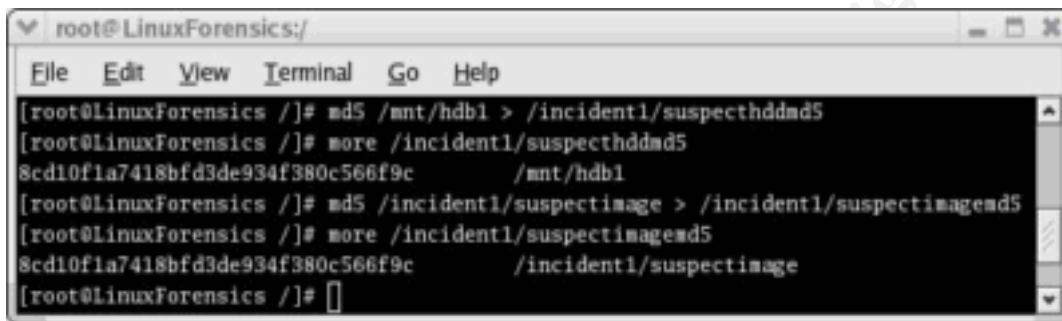
The -ro flags used with the “mount” command mounts the hard drive as read only. This was used to protect the data on the suspect hard drive. The “noexec” flag specifies that files won’t be executed on the mount hard drive.

I performed an MD5 checksum on the mounted suspect hard drive. This was saved in the following file: /incident1/suspecthddmd5.

The final step was to actually image the suspect hard drive. This was performed with the following command:

```
# dd if=/mnt/hdb1 of=/incident1/suspectimage
```

Once the imaging was complete, I performed an MD5 checksum of the suspectimage and saved it as: /incident1/suspectimagemd5. I tar'd the incident1 directory. Then an MD5 checksum was taken of the tar'd directory. The MD5 checksum of the suspectimagemd5 and of the tar'd directory were saved onto a floppy diskette, which remained in my possession at all times. Every day before I started working on suspectimagemd5, I took an MD5 checksum of the image and compared it with the MD5 checksum saved on the floppy. No discrepancies between the MD5 checksums ever came up during this investigation.



```
root@LinuxForensics:/  
File Edit View Terminal Go Help  
[root@LinuxForensics ~]# md5 /mnt/hdb1 > /incident1/suspecthddmd5  
[root@LinuxForensics ~]# more /incident1/suspecthddmd5  
8cd10f1a7418bfd3de934f380c566f9c /mnt/hdb1  
[root@LinuxForensics ~]# md5 /incident1/suspectimage > /incident1/suspectimagemd5  
[root@LinuxForensics ~]# more /incident1/suspectimagemd5  
8cd10f1a7418bfd3de934f380c566f9c /incident1/suspectimage  
[root@LinuxForensics ~]#
```

By this time, it was the end of the day. I placed the hard drive in an antistatic bag. I sealed the bag with packing tape and signed over the tape. The suspect hard drive was locked up for safe keeping until the end of the investigation.

## Image Analysis

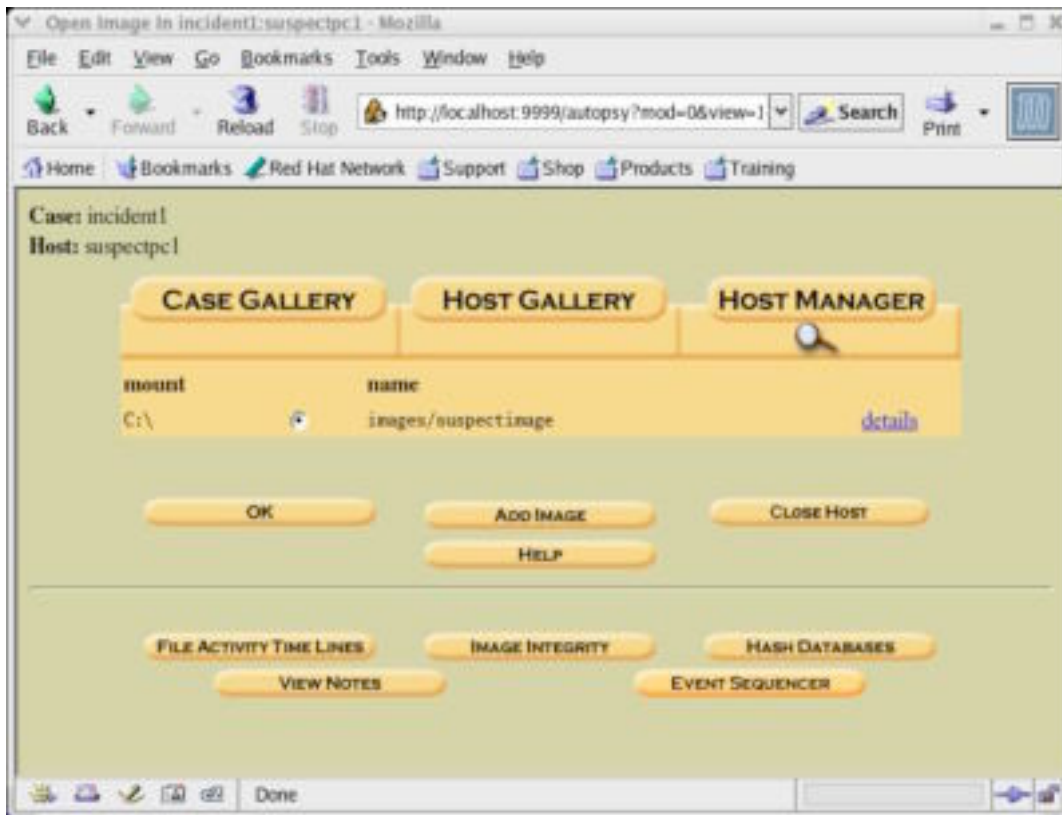
The overall goal of this analysis was to find any evidence linking the suspect to the key logger on the victimpc's. Since the suspect made it known that he knows the password the victim and her manager use for password protecting Excel documents, I was also searching for any evidence of how the suspect obtained that password.

On the forensics workstation, I used Autopsy Forensic Browser 2.01 (<http://www.sleuthkit.org/autopsy/desc.php>). To load the image from the suspect hard drive into Autopsy, I performed the following steps:

- Create New Case
  - Case Name: incident1
  - Investigator Names: Merlin Namuth
- Case Gallery
  - Select "incident1"
- Host Gallery
  - Add host
    - Host name: suspectpc1
- Host Manager
  - Add image
    - Location: /incident1/suspectimage
    - File System Type: ntfs
    - Mount Point: C:\

- Data Integrity: Calculate (This calculates the MD5 checksum, which then I compared with the MD5 checksum taken above. No discrepancies existed.)

The following screen shot shows the Case: incident1 Host: suspectpc1 and the suspectimage is loaded into this case.

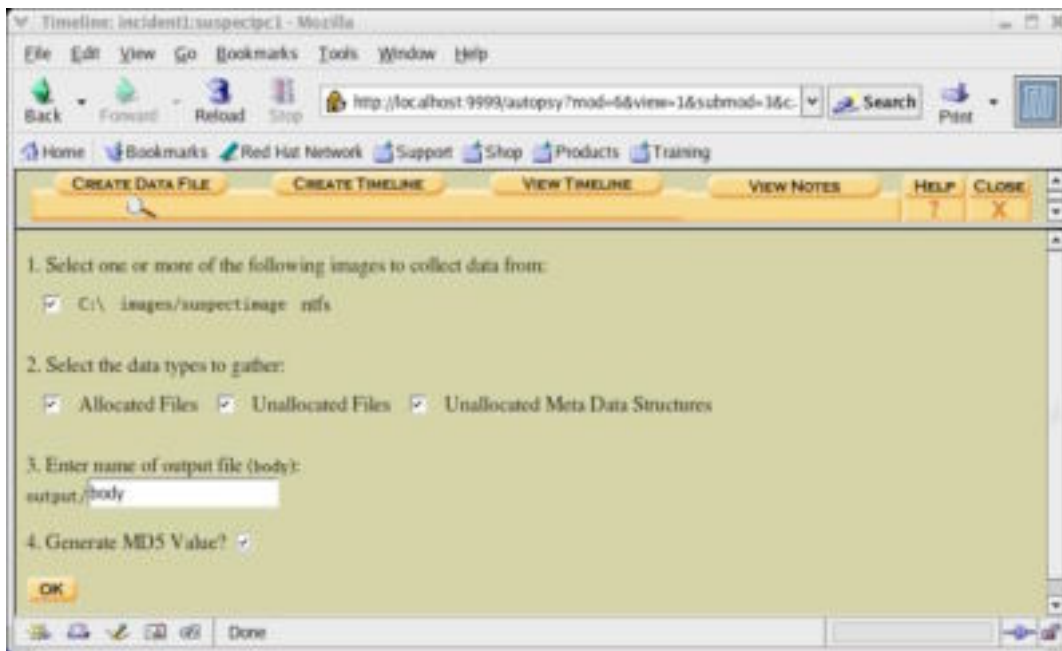


After the image was loaded, I stepped through Autopsy to a File Activity Time Line with the following steps:

- Create Data File
  - Select: C:\ images/suspectimage ntfs
  - Select all data types
  - Select to create an MD5

The following screen shot shows this step.

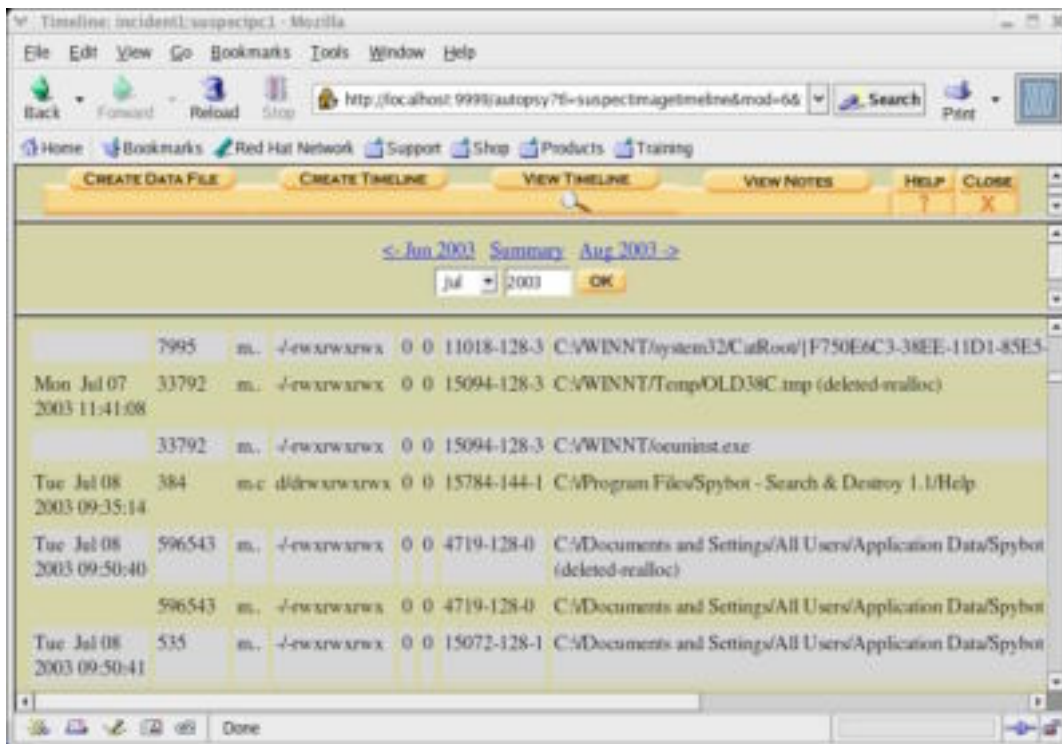




- Create Timeline (This is creating the MAC times. On both of the victim machines I calculated the MAC times using command line syntax. In these steps I'm configuring Autopsy to calculate the MAC times for all files on the suspectimage.) The timeline is included in a separate document, Merlin\_Namuth\_GCFA\_timeline.txt. The timeline is stripped down to meet the size requirements of this paper as the original size was 26MB
  - Starting date: July 1, 2003 (a few days prior to when the key logger was installed on victimpc1 and victimpc2)
  - Ending date: October 23, 2004 (the day the PC was seized and shipped to me for analysis)
- View Timeline
  - I searched on July 7, 2003 to see if I could find anything useful. July 7, 2003, is when the key logger was installed on victimpc1 and victimpc2

As shown in the following screen shot, no significant event occurred on July 7, 2003, that would correlate to when the key logger was installed on the victim machines. A significant event would include seeing the key logger installation file downloaded to suspectpc1. That would be indicated by a "c" for the create time. The screen shot below shows a Windows temp file that was deleted on July 7, 2003, indicating the file was deleted and the space the file utilized has been reallocated.

I viewed the time line going back 2 months to detect any activity associated to the key logger on victimpc1 and victimpc2. I didn't detect any such activity. This would be indicated by the key logger output files from victimpc1 and victimpc2 shown being created on suspectpc.

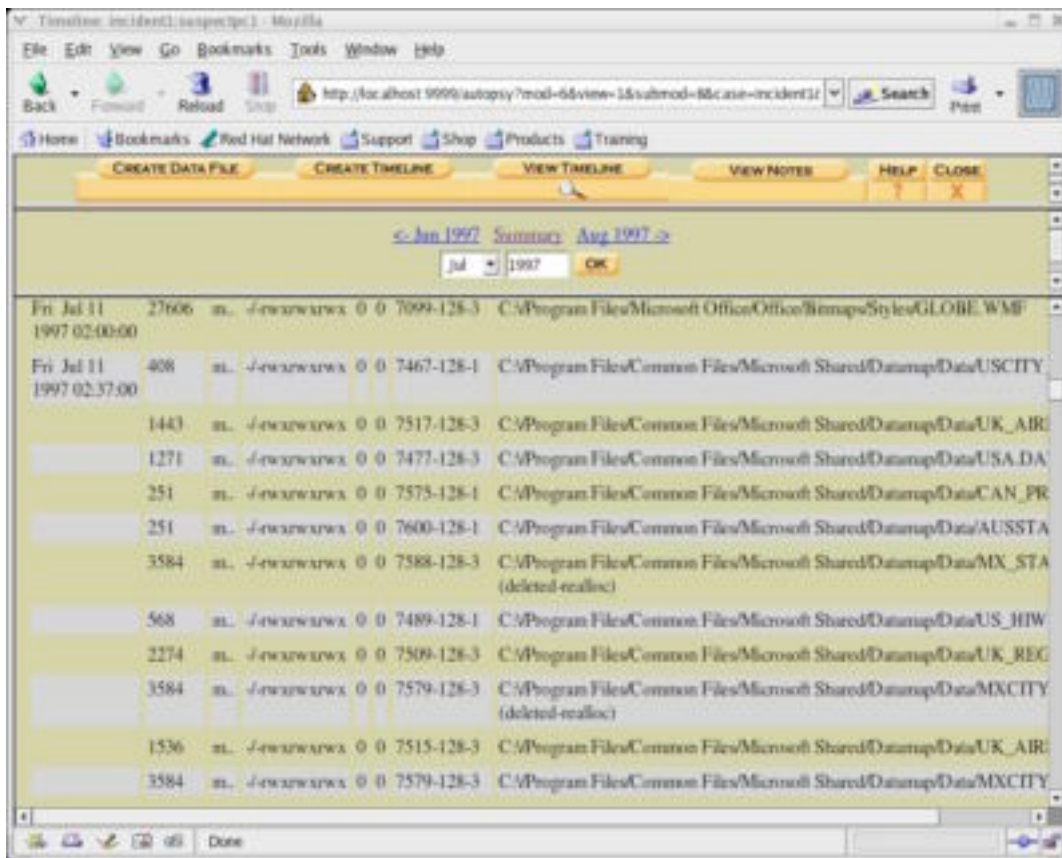


I analyzed the time line further to gain an understanding of major changes to the system, which might be helpful to the investigation.

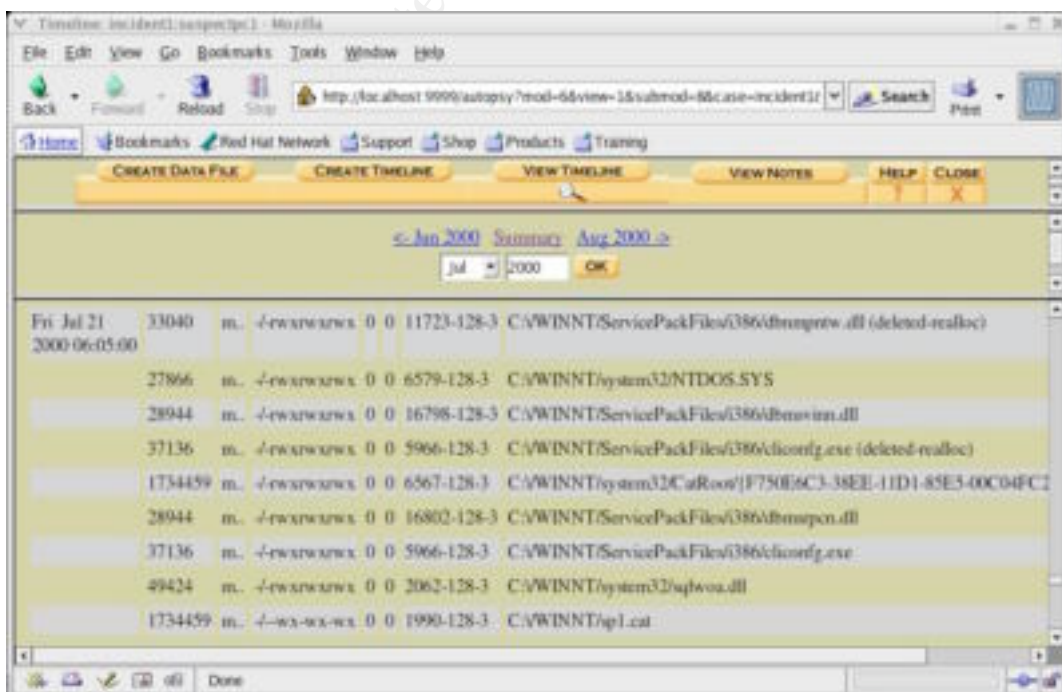
From the screen above, I clicked on "Summary". The resulting page shows the number of events broken down by month and day. I chose days that had a large number of events.

The earliest files were dated July 11, 1997.

© SANS Institute



On July 21, 2000, the timeline shows Service Pack 1 installing by modifying files. The last file in the following screen shot is “sp1.cat,” which is a file included in Service Pack 1.

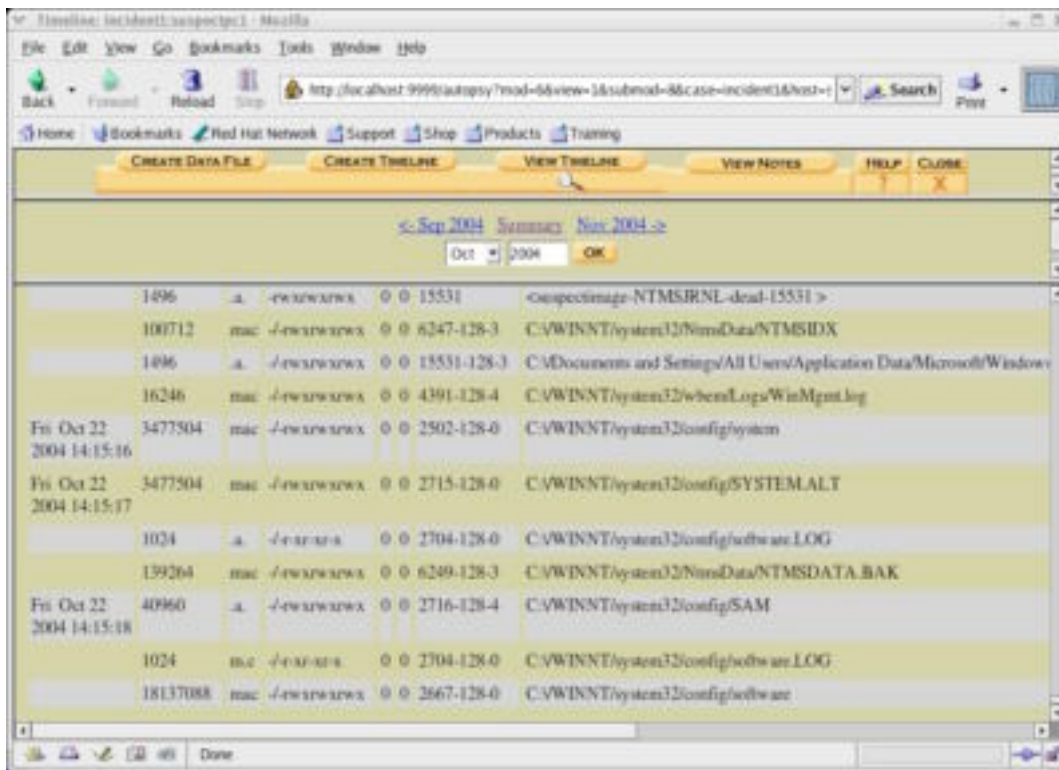


A large number significant number of file creates occurred on Jun 1, 2004. Several applications were removed and reinstalled. As shown below, the “c” indicates the creation of files and directories. The following screen shot shows the creation of files for Adobe Acrobat 5.0. Other applications included: Yahoo IM, AOL AIM, Spybot, and MSN messenger; not shown in this screen shot.

The screenshot shows a web-based timeline viewer interface. At the top, there's a menu bar with 'File', 'Edit', 'View', 'Go', 'Bookmarks', 'Tools', 'Window', and 'Help'. Below the menu is a navigation bar with buttons: 'CREATE DATA FILE', 'CREATE TIMELINE', 'VIEW TIMELINE', 'VIEW NOTES', 'HELP', and 'CLOSE'. A date selector shows '< May 2004 Summary Jul 2004 >' with a dropdown for 'Jun' and the year '2004'. The main area displays a list of file creation events. Each row includes a date, time, a small icon, a file path, and a description. The events are for Adobe Acrobat 5.0 files created on June 1, 2004, at various times between 12:04:18 and 10:33:13.

Date	Time	File Path	Description
Tue Jun 01 2004	12:04:18	C:\Program Files\Adobe\Acrobat 5.0\Help\ENU\MiniReader.pdf	
	28451	C:\Program Files\Adobe\Acrobat 5.0\Help\ENU\ACROBAT.PDF	
	1182048	<suspectimage-TrueImage.pdf-dead-21835>	
	397312	C:\Program Files\Adobe\Acrobat 5.0\Reader\AcroLinc.dll	
	53248	C:\Program Files\Adobe\Acrobat 5.0\Reader\ACROFX32.DLL	
	4082	C:\Program Files\Adobe\Acrobat 5.0\Help\ENU\DocBios.pdf	
Tue Jun 01 2004	16:79:36	C:\Program Files\Adobe\Acrobat 5.0\Reader\WHA Library.dll	
	4044	C:\Program Files\Adobe\Acrobat 5.0\Reader\ActiveX\pdf.tlb	
	384640	C:\Program Files\Adobe\Acrobat 5.0\Reader\ActiveX\pdf.ocx	
	147456	C:\Program Files\Adobe\Acrobat 5.0\Reader\Bin.dll	
	1138688	C:\Program Files\Adobe\Acrobat 5.0\Reader\Agm.dll	
	198	C:\Program Files\Adobe\Acrobat 5.0\Reader\RA\ENU.xml	
	1441792	C:\Program Files\Adobe\Acrobat 5.0\Reader\CoolType.dll	
	22270	C:\Program Files\Adobe\Acrobat 5.0\Reader\ReadMe.html	
Tue Jun 01	10:33:13	C:\Program Files\Adobe\Acrobat 5.0\Reader\Browsers\Browser\CS7.dll	

Last activity on the system was on October 22, 2004 at 14:15:18. October 22, 2004, is the day the suspect was put on administrative leave. This also indicates the last time any files on suspectpc1 were modified, accessed, and/or created, which is important to the integrity of the suspect hard drive and the image that was created from the suspect hard drive.



I conducted a string search on suspectimage. A string search is used to search for matches to what you are searching for. I performed the string search within Autopsy. It is called Keyword Search in autopsy. I searched on very specific words and phrases to determine if any correlation existed between suspectpc1 and the victim pc's.

The first word I searched for was "keylog." This result showed 20,718 occurrences of this word on the image. This search was too broad and needed to be more specific.

I searched for "klpf." This is the name of the executable file of the key logger installed on victimp1 and victimp2. The result showed 1 occurrence. This was found in Cluster 422742. I viewed the contents of that cluster and soon discovered other words associated with spyware and adware. This cluster contained a signature file for a spyware/adware removal tool. Therefore, the instance of finding klpf did not indicate that Key Log Pro was installed or had been installed on the suspect.

Victimp1 and victimp2 both had the same license key for Key Log Pro. I searched for the license string on the suspectimage. The result showed no occurrences.

The last string I searched for was the password the lady and her manager used for password protecting Excel documents. This password was supposed to be used between these 2 people and not shared with the suspect. I found 1 occurrence of this password on the suspectimage. This was found in Cluster 57412. This was contained in a configuration file for some sort of password cracking utility. The line in the configuration file this password was found read: "StartFrom = (password the lady and manager used)" Other key words in this

cluster included: BruteForce, MinPassLen\_Value, MaxPassLen\_Value. These are all words associated with a password cracking utility. I was unable to determine which password cracking utility was used based on just this configuration file.

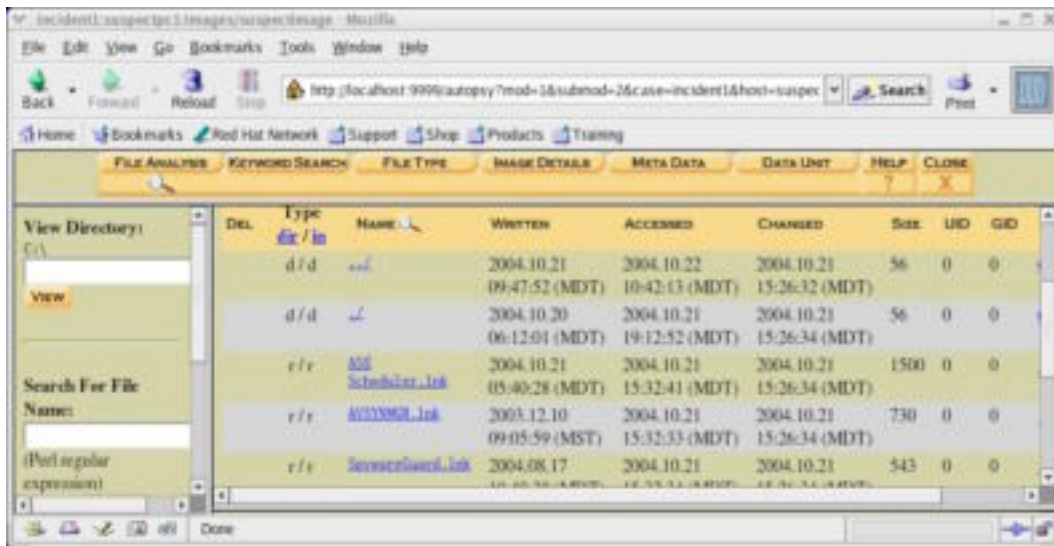
However, it was impossible to find and extract the program that uses this configuration file. This configuration file was located in a bad sector, which made it impossible to trace down the clusters containing the password cracking utility. The meta data for the cluster containing this configuration file pointed back to a special file in the ntfs file system called: \$BadClus:\$Bad. "This Metadata file contains a list of all the bad clusters on the volume." (<http://linux-ntfs.sourceforge.net/ntfs/files/badclus.html>).

Within Autopsy, I went to the File Analysis section. I proceeded down through the following files: \Documents and Settings\suspect\Local Settings\History\History.IE5 Within this directory I selected the file: index.dat. I then clicked on the link to export this file. I saved this under /incident1/suspectindex.dat I used pasco (<http://www.foundstone.com/resources/proddesc/pasco.htm>) to analyze this file. Paso is a forensics tool that enables you to view the information in the index.dat file. In this case, the index.dat file contains Internet Explorer history. I did not discover any suspicious URLs during my analysis. Suspicious URLs would include the URL for Key Log Pro and for password cracking utilities.

# pasco index.dat

I then proceeded down the following files: \Documents and Settings\Administrator\Local Settings\History\History.IE5. I exported the index.dat file from this directory. Again, I used pasco to view the contents of the index.dat file. I did not discover any suspicious URLs. I wanted to check the Internet history of the local administrator, just in case the suspect was browsing suspicious URLs while logged in at the local administrator. Suspicious URL's include the URL to download Key Log Pro and sites containing password cracking utilities.

I also viewed what programs startup when the pc starts. This is located in \Documents and Settings\suspect\Start Menu\ Programs\Startup. Nothing out of the ordinary was discovered. Suspicious programs starting up would include password cracking utilities.



I went to \RECYCLER. I clicked on each user SID and downloaded the INFO2 file for each user. The INFO2 file contains information about file(s) that was deleted.

I used a forensics tool, rifiuti (<http://www.foundstone.com/resources/proddesc/rifiuti.htm>), to view the contents of the INFO2 file for each user. I did not find any deleted files.

#### # rifiuti INFO2

In Autopsy, I went to the \Documents and Settings\suspect\My Documents folder. I found a deleted file named users. Autopsy shows files that were deleted in red. I clicked on the file users. Autopsy automatically switches to “Deleted File Recovery Mode” since the file had been deleted. It showed the file had been deleted on April 20, 2004.

I then clicked on Export to extract this file.

I then opened the extracted file in OpenOffice.org Writer. The document contained a list of users matched up with system names. All of the users were located in the same location as the suspect. Since the suspect performed IT tasks, having this information is appropriate.

## Conclusion

From the timeline I gathered some information as to the habits of the suspect. The suspect kept the system updated. From the timeline I found Windows service pack installs, antivirus updates, and Yahoo IM updates. In addition, the suspect used Spybot (<http://beam.to/spybotsd>), which is a spyware/adware removal tool. I saw frequent updates to Spybot, indicated by Spybot files being modified. I also saw a pop up blocker installed, Panicware (<http://www.panicware.com/popupstopper-popupkiller.html?hop=welcome1>). The suspect also frequented various technical websites.

All of the information gathered from the timeline mentioned above strongly indicates the suspect had a good understanding of how to maintain his Windows 2000 workstation.

There was no evidence of the key logger or the key logger output files from victimpc1 and victimpc2 currently on or having been on suspectpc1.

The only correlation from suspectpc1 and both victimpc1 and victimpc2 was the password cracking utility configuration file, on suspectpc1, containing a password known only to the owners of victimpc1 and victimpc2. It may be possible that the information and the password cracking utility may be recoverable using some 3<sup>rd</sup> party company who specializes in data recovery.

© SANS Institute 2005, Author retains full rights

## Works Cited

- “AntiVirus Software – McAfee Security System Protection.” December 2004.  
<<http://www.mcafeesecurity.com/us/products/mcafee/antivirus/category.htm>>.
- “Autopsy Forensic Browser: Description.” December 2004.  
<<http://www.sleuthkit.org/autopsy/desc.php>>.
- “\$BadClus (8) – File – NTFS Documentation.” December 2004. <<http://linux-ntfs.sourceforge.net/ntfs/files/badclus.html>>.
- “Breaking a steganography software: Camouflage.” December 2004.  
<<http://guillermi2.net/stegano/camouflage/>>.
- “[Camouflaged Mp3s Contain a Backdoor Beware – TranceAddict – A Site for Trance Music Fans!.” December 2004.  
<<http://www.tranceaddict.com/forums/archive/topic/79627-1.html>>.
- “Dumpel.exe: Dump Event Log.” December 2004.  
<<http://www.microsoft.com/windows2000/techinfo/reskit/tools/existing/dumpel-o.asp>>.
- “Foundstone, Inc.® Strategic Security.” December 2004.  
<<http://www.foundstone.com/resources/proddesc/fport.htm>>.
- “Foundstone, Inc.® Strategic Security.” December 2004.  
<<http://www.foundstone.com/resources/proddesc/pasco.htm>>.
- “Foundstone, Inc.® Strategic Security.” December 2004.  
<<http://www.foundstone.com/resources/proddesc/rifiuti.htm>>.
- “ftp.zcu.cz: /pub/win/simtelnet/win95/secfile/.” December 2004.  
<http://www.zcu.cz/ftp/pub/win/simtelnet/win95/secfile.camou104.zip>
- “GIAC: Global Information Assurance Certification – GIAC Certified Forensic Analyst (GCFA) Prac.” December 2004.  
<[http://www.giac.org/gcfa/v1\\_5.gz](http://www.giac.org/gcfa/v1_5.gz)>.
- “The GNU Netcat – Official homepage.” December 2004.  
<<http://netcat.sourceforge.net/>>.
- “Hoko info.” December 2004. <<http://scifi.pages.at/yoda9k/cf/info.htm>>.

“Intellectual Property Crimes: VIII. Theft of Commercial Trade Secrets.”  
December 2004.  
<<http://www.usdoj.gov/criminal/cybercrime/ipmanual/08ipma.htm#VIII.B>>.

“KeyLog Pro – Spy Software and Internet Monitoring Program.” December 2004.  
<<http://www.keylogpro.com/>>.

“Knoppix STD.” December 2004. <<http://www.knoppix-std.org/>>.

Kruse II, Warren G., Heiser, Jay G., Computer Forensics, Boston, Addison-Wesley, ©2002.

“Lavasoft.” December 2004. <<http://www.lavasoft.com/software/adaware/>>.

“Macromedia Shockwave Download Center.” December 2004.  
<<http://sdc.shockwave.com/shockwave/download/download.cgi?>>.

Mandia, Kevin, Prorise, Chris, Incident Response, New York, Osborne/McGraw-Hill, ©2001.

“Maxtor.com – DiamondMax 2160 Ultra ATA.” December 2004.  
<[http://www.maxtor.com/portal/site/Maxtor/menuitem.5d2b41d3cef51dfe29dd10a191346068/?channelpath=/en\\_us/Support/Product%20Support/ATA%20Hard%20Drives/Desktop/DiamondMax%202160%20Ultra%20ATA](http://www.maxtor.com/portal/site/Maxtor/menuitem.5d2b41d3cef51dfe29dd10a191346068/?channelpath=/en_us/Support/Product%20Support/ATA%20Hard%20Drives/Desktop/DiamondMax%202160%20Ultra%20ATA)>.

“OpenOffice.org: Home Page.” December 2004. <<http://www.openoffice.org/>>.

“.: [packet storm ]:. – <http://packetstormsecurity.org/>.” December 2004.  
<<http://packetstormsecurity.nl/crypt/stego/camouflage/>>.

“Panicware, Inc. – Stop pop up windows with Pop-Up Stopper Free Pop Up Killer.” December 2004. <<http://www.panicware.com/popupstopper-popupkiller.html?hop=welcome1>>.

“SpyBot – Search&Destroy.” December 2004. <<http://beam.to/spybotsd>>.

“Sysinternals Freeware - Information for Windows NT and Windows 2000 – PsList.” December 2004.  
<<http://www.sysinternals.com/ntw2k/freeware/pslist.shtml>>.

“Sysinternals Freeware – Information for Windows NT and Windows 2000 – PsLoggedOn.” December 2004.  
<<http://www.sysinternals.com/ntw2k/freeware/psloggedon.shtml>>.

“Sysinternals Freeware – Information for Windows NT and Windows 2000 – PsService.” December 2004.

<<http://www.sysinternals.com/ntw2k/freeware/pservice.shtml>>.

“Trend Micro – Free online virus Scan.” December 2004.

<<http://www.mcafeesecurity.com/us/products/mcafee/antivirus/category.htm>>.

© SANS Institute 2005, Author retains full rights.