



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>

Forensic Analysis of a Misused System
David C. Shettler
GCFA Practical Assignment v2.0
4/5/2005

© SANS Institute 2000 - 2005, Author retains full rights.

Table of Contents

| | |
|--|----|
| <u>Executive Summary</u> | 3 |
| <u>Synopsis of Case Facts</u> | 3 |
| <u>The Alert</u> | 3 |
| <u>The Forensics Environment</u> | 4 |
| <u>The Suspect System</u> | 4 |
| <u>Imaging the System and Evidence Collection</u> | 6 |
| <u>Confirming the Suspect System</u> | 6 |
| <u>Imaging the Physical Memory</u> | 6 |
| <u>Imaging the Physical Drives</u> | 7 |
| <u>Running the Windows Forensic Toolchest</u> | 7 |
| <u>Seizing the Hardware</u> | 8 |
| <u>Backing up the Evidence</u> | 8 |
| <u>Analyzing the Evidence</u> | 9 |
| <u>Analyzing the Packet Capture</u> | 9 |
| <u>Image processing</u> | 11 |
| <u>Timeline Creation</u> | 12 |
| <u>Timeline Analysis</u> | 13 |
| <u>Mounting the Images</u> | 15 |
| <u>Antivirus Check</u> | 16 |
| <u>INTERNET EXPLORER™ History Analysis</u> | 16 |
| <u>Retrieving Deleted Files</u> | 18 |
| <u>TCT™ Lazarus – Memory Dump Analysis</u> | 20 |
| <u>NTUSER.DAT and Registry Analysis</u> | 20 |
| <u>Order of Events</u> | 21 |
| <u>String Searches</u> | 22 |
| <u>Loose Ends – Windows Forensic Toolchest Analysis & DEEP FREEZE™</u> | 24 |
| <u>Conclusion</u> | 24 |
| <u>Identifying the Culprit</u> | 24 |
| <u>Assessing Intent</u> | 25 |
| <u>Summary</u> | 26 |
| <u>Additional Information</u> | 26 |
| <u>Appendix A</u> | 28 |
| <u>Appendix B</u> | 29 |
| <u>Appendix C</u> | 32 |
| <u>Appendix D</u> | 33 |
| <u>References</u> | 35 |

Executive Summary

On January 10th, 2005, a system on the network at my organization was suspected of having downloaded child pornography; an activity in violation of the organization's policies, as well as the law. Data from the system was captured at the scene, and then the system was seized. Forensic analysis of the evidence captured at the scene, via analysis of the hard drive, memory, and network traffic showed that child pornography had indeed been viewed by an individual using a computer between approximately 9:30am EST and 10:00am EST.

The evidence indicated that the person viewing the child pornography had started viewing the material through email, then through browsing internet web pages viewing galleries of images. The email account where the activity originated from belonged to John Doe¹, the suspect.

Viewing or being in possession of child pornography is in violation of federal law, specifically 18 U.S.C. § 2252 (U.S.Code 2004) entitled "Certain activities relating to material involving the sexual exploitation of minors." Specifically, section (a)(4)(B) of the Code was violated. The act of possessing child pornography is also a violation of state law, specifically Chapter 181 of the Acts of 1997 in the state where the incident occurred.

Synopsis of Case Facts

The Alert

On January 10th, 2005, the administrator of our organization's intrusion detection system notified me of an incident involving a computer that had tripped policy rules geared towards catching child pornography on my organization systems. The administrator provided a compact disk containing a packet capture of the offending IP address using tcpdump, he initiated after noticing the alerts. The administrator had also notified onsite security personnel to find the machine and potential suspect, since the activity was ongoing.

The administrator stated that he noticed the alerts at approximately 09:40 EST, and began the capture sometime shortly thereafter. He stated that the copy he submitted to me and which had been burned on a CD was the packet capture file as of 10:50 EST. At that time the administrator had been notified by our onsite security personnel that the offending computer had been identified and quarantined from any user interaction. I arrived on the scene shortly after receiving the CD.

¹ "John Doe" is used herein to protect the identity of the individual observed during this investigation.

The Forensics Environment

Our forensics environment consists of a dedicated server running FEDORA™ Core 2. The server is not networked by default, but networking is enabled on the system when we need to transfer images to the system. Even when networking is enabled, the system is running an iptables firewall that restricts access to port 22 for setting up the remote connection, and port 9999 for netcat traffic, as will be discussed in more detail later. The forensics server runs VMWARE™ for windows forensic analysis, as well as testing unknown binaries.

The Suspect System

The system being analyzed was a publicly accessible computer, used by community members of the organization for various purposes, and located in a lab setting containing dozens of other computers. The suspect system was situated towards the rear of the room, with the monitor facing away from the door into the room. The chair in front of the suspect system had a coat on it, and a book bag was leaning against the table. The following pictures show the location in the room, as well as the system itself. The pictures are intentionally blurry.





The suspect system was powered on and connected to the organization's network. The system appeared to be running Microsoft's WINDOWS® XP operating system, and had two INTERNET EXPLORER™ browser windows open. The open window on the foreground was of our organization's email

system, logged in to the suspect's email account. The browser window in the background was accessing our organization's web site.

Imaging the System and Evidence Collection

System details had been identified through the tcpdump packet capture and intrusion detection alerts. The suspect system had a specific IP address, and it had been determined via switching and routing equipment that the system had the MAC address of 00-B0-D0-DF-DD-F2. Prior to gathering any evidence from the suspect system, I had to determine if this was in fact the offending system. Inserting HELIX™ Incident Response and Computer Forensics Live CD-ROM (e-fense 2004) in the suspect system's CD-ROM drive, the user interface launched automatically, and I selected the "Acquire a Live image of a Windows System using dd" option. The HELIX™ software then listed the available drives, which included two hard drives/hard drive partitions, C: and D:.

Confirming the Suspect System

Confirming the suspect system was necessary because this system was one out of approximately 30 systems in the room. From the user interface, I launched a shell. After running the *ipconfig /all* command, I found that the MAC address and IP address matched the MAC address and IP address provided to me, confirming this was indeed the suspect system.

Imaging the Physical Memory

Prior to seizing the suspect system, I decided I would image the Physical Memory of the system, as it could well contain important evidence. From my laptop, I connected to the remote forensics server to transfer the images. I launched the following command to receive the dd:

```
[root@server img]# nc -l -p 9999 | dcfldd of=2311-PhysicalMemory.dd hashwindow=0
hashlog=2311-PhysicalMemory.dd.md5
```

The nc command is netcat, a utility used to establish a network connection between two systems. In the above scenario, netcat is set to listen on port 9999, and when a connection is initiated, it is to hand off data to dcfldd. DCFL-dd (Biatchux/DoD 2002) is a clone of the popular dd imaging utility. DCFL-dd was written by the Department of Defense Computer Forensics Laboratory (DCFL). The arguments passed to the command above tell the program to output the image to the file named 2311-PhysicalMemory.dd. The next two arguments tell the program to calculate an MD5 hash of the image created at the end of the image, and to output that hash to the file named: 2311-PhysicalMemory.dd.md5.

After issuing that command, I issued the following command on the suspect computer:

```
11:33:02.24 E:\Shells> dd if=\\.\PhysicalMemory | nc server 9999
```

The above command initiates the dd program, which takes an image of the argument specified by if=, in this case, the system's Physical Memory. That data is then passed to netcat, which sends the data over to computer 'server' on the port 9999.

Imaging memory as a first step insures that memory will be as intact as possible, as further actions on the system can modify the contents of memory on the system, potentially erasing evidence. Memory is also volatile in that if the system loses power, the memory is cleared and the evidence is destroyed.

Imaging the Physical Drives

With the memory imaged, the next step was to request further information from the administrator of this system. The administrator confirmed the system serial number and internal asset tag, and also informed me of the hardware on the system, as well as the configuration of the system. The suspect system contained one internal drive with two partitions. One partition was approximately 16 gigabytes in size, while the other approximately 4 gigabytes in size. The C: partition housed the operating system and software, while the D: partition housed an image of the system to facilitate re-installation of the operating system and software.

Knowing this, I then decided to create a live image of the physical drive. I launched the following command on the networked server that I used previously to store the image of the system's physical memory:

```
[root@server img]# nc -l -p 9999 | dcfldd of=2311-PhysicalDrive0.dd hashwindow=0
hashlog=2311-PhysicalDrive0.dd.md5
```

I then launched the following command on the suspect computer to image the physical drive:

```
11:38:43.11 E:\Shells> dd if=\\.\PhysicalDrive0 | nc server 9999
```

Running the Windows Forensic Toolchest

After the imaging had completed, I ran Windows Forensics Tool Chest (WFT) (McDougal 2004) and sent the output to a windows share I had previously setup on a networked server. I knew there was a possibility that WFT could modify data on the suspect system, however, having already imaged both the memory and the system, I felt it was necessary to run. WFT gathers information about the system from various sources on the system, as well as by utilizing various tools to parse the data received. I chose not to run slow executables, and chose not to run executables that could write to the source machine when prompted to by the HELIX™ graphical user interface. I sent the output to a Netware file

share under my personal account directory where only I have permissions to view, modify, or otherwise manipulate data. Immediately after the WFT ended, I connected to the remote forensics server, and copied the files generated by WFT from the windows share to the forensics server. I then ran the following command on the remote forensics server:

```
ifconfig eth0 down
```

This command shuts down the networking interface on the server, removing it from the network and disconnecting all connections to the server.

Seizing the Hardware

After WFT had ended, I powered off the system by pulling the power plug from the power supply. Doing this insures no further information is written to the hard drive. I then opened the suspect system to confirm the existence of a single hard drive. Once that was confirmed, I placed evidence tags on the system and the system's hard drive, and turned the systems over to the security department for placement in their safe. See Appendix A for the list of evidence seized and tagged, and Appendix B for the chain of custody information regarding the evidence.

Backing up the Evidence

Prior to initiating my investigation on the acquired images, I executed the following command to break the large physical drive image up into pieces so they could be burned onto DVD media:

```
[root@server img]# dcfldd if=2311-PhysicalDrive0.dd hashwindow=0 hashlog=2311-PhysicalDrive0.dd.split.md5 | split -b 2048000000 - 2311-PhysicalDrive0.dd.split.
```

This generated the following files (output from is from a Linux directory listing):

```
-rw-r--r-- 1 root root 2048000000 Jan 10 13:13 2311-PhysicalDrive0.dd.split.aa
-rw-r--r-- 1 root root 2048000000 Jan 10 13:17 2311-PhysicalDrive0.dd.split.ab
-rw-r--r-- 1 root root 2048000000 Jan 10 13:21 2311-PhysicalDrive0.dd.split.ac
-rw-r--r-- 1 root root 2048000000 Jan 10 13:25 2311-PhysicalDrive0.dd.split.ad
-rw-r--r-- 1 root root 2048000000 Jan 10 13:29 2311-PhysicalDrive0.dd.split.ae
-rw-r--r-- 1 root root 2048000000 Jan 10 13:32 2311-PhysicalDrive0.dd.split.af
```

I then checked to determine that the MD5 hash files generated by the original

```
-rw-r--r-- 1 root root 2048000000 Jan 10 13:39 2311-PhysicalDrive0.dd.split.ah
-rw-r--r-- 1 root root 2048000000 Jan 10 13:43 2311-PhysicalDrive0.dd.split.ai
-rw-r--r-- 1 root root 1567997952 Jan 10 13:46 2311-PhysicalDrive0.dd.split.aj
-rw-r--r-- 1 root root 40 Jan 10 13:46 2311-PhysicalDrive0.dd.split.md5
```

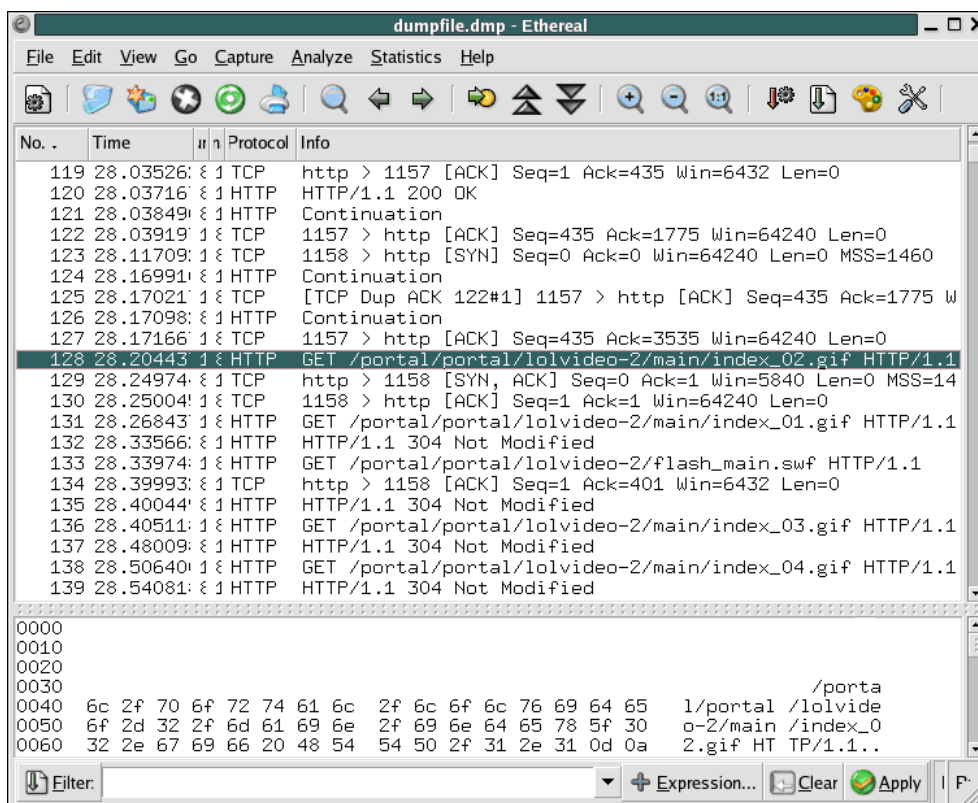
image and the new split image were identical, and confirmed that they were. I then burned the above files onto 5 DVD-R's. I tagged each of them sequentially 2311-2.1 through 2311-2.5. I then copied the tcpdump file off of the CDROM (given to me by the intrusion detection system administrator) onto my forensics workstation, and tagged the CDROM as tag #2311-3. Lastly, I burned the physical memory image file, as well as its MD5 hash file onto a CDROM and tagged it as 2311-4. I then made one set of copies of all 5 DVDs, and both CDROMs. I submitted the originals to the security department to store in their safe.

Analyzing the Evidence

Analyzing the Packet Capture

Having access to the packet capture file generated via tcpdump by the intrusion detection system administrator, I decided to analyze that first. Initially, I ran ETHEREAL™ network protocol analyzer (Combs 2004) against the packet capture. ETHEREAL™ is a traffic analyzer that has the ability to decode various protocols in a clear, readable way. The capture was extensive, containing several thousand packets. Scrolling through, it was apparent that adult websites were being visited and images were being downloaded due to the directory names in the captured http get requests as shown in the screenshot below.

© SANS Institute 2000 - 2005. All rights reserved. This document retains full rights.



In order to determine the nature of the images being downloaded, I needed to actually see the images. Extracting the images from the packet capture through ETHEREAL™ seemed a rather time consuming process, so instead, I ran a program called CHAOS READER™ software (Gregg 2004) against the packet capture. CHAOS READER™ extracts various file formats from a packet capture file, and also arranges them in a timeline as the packet capture occurred, and outputs everything in HTML format – rather convenient. The program is a completely automated way to extract data, particularly web traffic, from packet capture files, and as such, it appeared to be ideally suited for the task at hand.

CHAOS READER™ extracted 473 images from the packet capture, many of which were seemingly in violation of federal laws, specifically: 18 U.S.C. § 2252 (U.S. Code 2004) entitled “Certain activities relating to material involving the sexual exploitation of minors.” At that time, I created a compressed archive of the CHAOS READER™ output:

```

Server
File Edit View Terminal Tabs Help
-rw-r--r-- 1 dshettle dshettle 8230 Jan 30 06:58 session_0383.part_40.jpeg
-rw-r--r-- 1 dshettle dshettle 12178 Jan 30 06:58 session_0383.part_41.jpeg
-rw-r--r-- 1 dshettle dshettle 804 Jan 30 06:58 session_0383.part_42.gif
-rw-r--r-- 1 dshettle dshettle 1250 Jan 30 06:58 session_0383.part_43.gif
-rw-r--r-- 1 dshettle dshettle 224 Jan 30 06:59 session_0383.part_44.html
-rw-r--r-- 1 dshettle dshettle 1895 Jan 30 06:58 session_0383.part_45.gif
-rw-r--r-- 1 dshettle dshettle 1062 Jan 30 06:59 session_0383.part_46.gif
-rw-r--r-- 1 dshettle dshettle 1462 Jan 30 06:59 session_0384.http.html
-rw-r--r-- 1 dshettle dshettle 1004 Jan 30 06:58 stream_0259.domain.html
[LinuxForensics incident-1-10-2005]$ tar -cf chaosreader-output.tar chaosreader
[LinuxForensics incident-1-10-2005]$ gzip chaosreader-output.tar
[LinuxForensics incident-1-10-2005]$ md5sum chaosreader-output.tar.gz > chaosreader-output.tar.gz.md5
[LinuxForensics incident-1-10-2005]$

```

I then burned the file containing the compressed archive of the CHAOS READER™ output, and tagged the CDROM as 2311-5 and later handed over to the security department. The timeframe of the activity determined by CHAOS READER™ from the packet capture was from 09:44 until 10:00. This time frame was from the time the administrator started this packet capture, until the suspect using the IP address in question, stopped utilizing the network. Concerned over whether the timestamps from the packet capture were reliable or not, I contacted the person responsible for the server where the packet capture was initiated to determine whether the time on the server was synchronized with our organization's time server. The administrator confirmed that the time on the server was, and had been, synchronized with our time server at the time the capture took place via the network time protocol (NTP).

Having confirmed an incident did occur via the network packet capture, the next step was to begin analysis of the images captured.

Image processing

To establish the partitioning of the drive, the mmls command was issued. That command is a component of the SLEUTHKIT™ tool (Carrier 2005) collection of open source forensics tools. Several SLEUTHKIT™ components are used throughout the course of this investigation. SLEUTHKIT™ is used because it is a well documented, reliable, and affordable package for conducting computer forensics. It fully supports the file systems required for this analysis, and was readily available.

The mmls command essentially lists the partitions table of an image. Below is the output of the command.

```
[root@server img]# mmls -t dos 2311-PhysicalDrive0.dd
```

```
DOS Partition Table
```

```
Units are in 512-byte sectors
```

| Slot | Start | End | Length | Description |
|-----------|------------|------------|------------|-----------------------|
| 00: ---- | 0000000000 | 0000000000 | 0000000001 | Primary Table (#0) |
| 01: ---- | 0000000001 | 0000000062 | 0000000062 | Unallocated |
| 02: 00:00 | 0000000063 | 0030716279 | 0030716217 | NTFS (0x07) |
| 03: 00:01 | 0030716280 | 0039054014 | 0008337735 | Win95 Extended (0x0F) |
| 04: ---- | 0030716280 | 0030716280 | 0000000001 | Extended Table (#1) |
| 05: ---- | 0030716281 | 0030716342 | 0000000062 | Unallocated |
| 06: 01:00 | 0030716343 | 0039054014 | 0008337672 | Win95 FAT32 (0x0B) |

```
[root@server img]#
```

```
dcfldd if=2311-PhysicalDrive0.dd skip=63 count=30716217 of=2311-C-Drive.dd
```

```
hashwindow=0 hashlog=2311-C-Drive.dd.md5
```

```
dcfldd if=2311-PhysicalDrive0.dd skip=30716343 count=8337672 of=2311-D-Drive.dd
```

```
hashwindow=0 hashlog=2311-D-Drive.dd.md5
```

```
2311-C-Drive.dd.md5
```

```
2311-D-Drive.dd
```

```
2311-D-Drive.dd.md5
```

Timeline Creation

Now, with the C and D drive extracted, I began to analyze the images using more SLEUTHKIT™ tools. My first step in analyzing the images was to create a timeline. Timelines often highlight problems or areas of interest quicker than other methods of analysis. The chronological ordering of file accesses and modifications makes otherwise difficult to detect information stand out, particularly if the information doesn't belong in the chronological order it is found in under normal circumstances, and as such I've always found them to be a good way to begin an analysis of file systems. The first step is to use the SLEUTHKIT™ tool's ils and fls utilities.

The fls utility outputs filenames and directories from a given image for both

allocated and unallocated (deleted) files and directories, and is capable of outputting the information to a format readable by other utilities, namely the mactimes utility which will be discussed in more detail later. The following commands were used to extract the file and directory information from the images:

```
[root@server img]# fls -f ntfs -m "C:" -r 2311-C-Drive.dd > 2311-C-Drive.dd.flr  
[root@server img]# fls -f fat32 -m "D:" -r 2311-D-Drive.dd > 2311-D-Drive.dd.flr
```

The ils utility outputs information on deleted master file table (MFT) records. This information is useful as modification, access, and change times are still recoverable for the file which was identified by that MFT record. That data is stored in the MFT record itself. To gather that information, the following commands are issued:

```
[root@server img]# ils -f ntfs -m 2311-C-Drive.dd > 2311-C-Drive.dd.ils  
[root@server img]# ils -f fat32 -m 2311-D-Drive.dd > 2311-D-Drive.dd.ils
```

After the generation of these .ils and .flr files, concatenating them together is required for analysis by the mactimes utility. Mactimes, another component of SLEUTHKIT™, takes the modified, changed, and accessed information generated from both the ils and fls commands, and sorts them by chronological order, producing a timeline from the earliest time to the latest time. Given the nature of this case, I was able to jump to the date of the offense, and follow the activity that occurred on the file system as it happened. The following command is used to generate the timeline, using the mactimes program included in SLEUTHKIT™.

```
[root@server img]# mactime -b 2311-CD-Drive.mac > timeline.all
```

Timeline Analysis

With a complete timeline of the system's activity, I proceeded to look through the activity. The timeline was surprisingly terse. The suspect system appeared to have been installed on October 1st of 2002. A NOVELL NETWARE™ client was installed on that date, as was NOVELL GROUPWISE™, and other common applications used at our organization. WINDOWS™ XP Service Pack 1 was downloaded and installed on that date as well. The next activity occurred on October 10th, 2002, when QUICKTIME™ software was installed on the system. Various applications were installed throughout October, and then activity appeared to pretty much stop until September 3rd of 2003, when an application named DEEP FREEZE™ was installed.

It was determined at this time that an interview with those responsible for deploying and maintaining these systems would be helpful. The interview

provided some valuable information. The suspect system had been purchased in July of 2003, installed in August of 2003 with a system image of an operating system created in the fall of 2002. Then, the system was “frozen” with the DEEP FREEZE™ application in September. DEEP FREEZE™, according to the administrators of the system, reverts the system to the state that it was in prior to the installation of DEEP FREEZE™ and upon reboot, essentially nullifies any changes that had occurred on the system in the time span between reboots. This software is typically used to facilitate keeping the systems clean of spyware, adware, viruses, and other annoyances – as well as cleaning off anything downloaded and installed on the system without permission. According to the administrators, DEEP FREEZE™ was disabled on occasion when patches or service packs were applied to the system, or when new software was required. In addition, DEEP FREEZE™ was not enabled on the D drive, allowing some permanent storage.

Returning to the timeline, the activity on the suspect system, or lack thereof, now made some degree of sense. D drive activity was somewhat common, but activity on the C Drive was rare, and when noticed, very specific: software upgrades, installations, operating system patches. The machine is a publicly accessible machine, and is reportedly extensively used during the year, but the DEEP FREEZE™ software eliminates all traces of that activity – except on the D Drive.

The trend of limited timeline activity all changes on January 7th of 2005, when actual C Drive activity, not related to patching or upgrades, takes place. The user named ‘crossroads’, a default user account, logs in to the system. Internet browsing begins taking place on that day as evidenced by the creation of files in the temporary internet files directories on the suspect system. The anti-virus system automatically updated its signatures on that date. Several programs and files were accessed on the system that are consistent with the purpose of that system – the system is used primarily for foreign language instruction, and contains several multimedia files relating to the subject. These files were used throughout the day on January 7th. A virus scan appears to run as well on this date, changing the access times of nearly every file on the system.

Activity following the virus scan appears consistent with an idling machine, with small changes occurring over time, until on January 9th when the user logged out. The system then continued to idle until January 10th at 09:28, when the ‘crossroads’ default user logged into the system again. INTERNET EXPLORER™ activity, likewise, began again. The timeline below shows its output listing filenames of deleted INTERNET EXPLORER™ cache files, recognizable to me as being files from the organization’s web page:

```

File Edit View Terminal Tabs Help
Mon Jan 10 2005 09:30:00 168 n.c -rw-rw-rw- 0 0 7113 <2311-C-Drive.dd-academics_new[1].gif-dead-7113>
106 n.c -rw-rw-rw- 0 0 7115 <2311-C-Drive.dd-purple_spacer3[1].gif-dead-7115>
124 n.c -rw-rw-rw- 0 0 7029 <2311-C-Drive.dd-site_index_new[1].gif-dead-7029>
242 n.c -rw-rw-rw- 0 0 7098 <2311-C-Drive.dd-purple_spacer[1].gif-dead-7098>
31488 .a. -rw-rw-rw- 0 0 833-128-3 C:/WINDOWS/system32/drivers/cruse.sys
150 n.c -rw-rw-rw- 0 0 7137 <2311-C-Drive.dd-athletics_new[1].gif-dead-7137>
13184 .a. -rw-rw-rw- 0 0 507-128-3 C:/WINDOWS/system32/drivers/diskdump.sys
33792 .a. -rw-rw-rw- 0 0 146-128-3 C:/WINDOWS/system32/drivers/disk.sys
46336 .a. -rw-rw-rw- 0 0 147-128-3 C:/WINDOWS/system32/drivers/classpnp.sys
52709 .a. -rw-rw-rw- 0 0 22844-128-3 C:/WINDOWS/system32/drivers/DepFzLo.sys
16384 .a. -rw-rw-rw- 0 0 9616-128-3 C:/Documents and Settings/crossroads/Local Settings/Application Data/Microsoft/Internet Explorer/MSINGSI2.DAT
64 n.c -rw-rw-rw- 0 0 7026 <2311-C-Drive.dd-white_nav_filler_new[1].gif-dead-7026>
149 n.c -rw-rw-rw- 0 0 7048 <2311-C-Drive.dd-web_services_new[1].gif-dead-7048>
87 n.c -rw-rw-rw- 0 0 7053 <2311-C-Drive.dd-top_left_corner2[1].gif-dead-7053>
104 n.c -rw-rw-rw- 0 0 7138 <2311-C-Drive.dd-purple_spacer[1].gif-dead-7138>
349636 .a. -rw-rw-rw- 0 0 12381-128-3 C:/WINDOWS/Fonts/times.ttf
380969 .a. -rw-rw-rw- 0 0 15695-128-4 C:/Program Files/Real/RealOne Player/rpplugins/cdp13210.dll
196 n.c -rw-rw-rw- 0 0 7120 <2311-C-Drive.dd-administration_new[1].gif-dead-7120>
80 n.c -rw-rw-rw- 0 0 7076 <2311-C-Drive.dd-top_right_corner[1].gif-dead-7076>
102 n.c -rw-rw-rw- 0 0 7120 <2311-C-Drive.dd-purple_spacer4[1].gif-dead-7120>
Pattern not found (press RETURN)

```

At 09:30:19, the timeline shows the organization's GROUPWISE WEBACCESS™ email access site was browsed, based on the names of the cache files created. For the next several seconds, cache files were generated, consistent with navigating the email access site of the organization.

At 09:31:24, the timeline shows five cache files created, highlighted in the below screenshot, that are consistent with an email being opened in GroupWise Web Access for viewing.

```

File Edit View Terminal Tabs Help
150 mac -rw-rw-rw- 0 0 7889-128-1 C:/Documents and Settings/crossroads/Local Settings/
Temporary Internet Files/Content.IE5/OY4FG9EF/fl_cal[1].gif (deleted)
466 mac -rw-rw-rw- 0 0 8185-128-1 C:/WINDOWS/system32/dllcache/mmcndmgr.dll (deleted)
799 mac -rw-rw-rw- 0 0 8116 <2311-C-Drive.dd-black1[1].gif-dead-8116>
119 mac -rw-rw-rw- 0 0 7892-128-1 C:/WINDOWS/system32/dllcache/inetcp1.cpl (deleted)
150 mac -rw-rw-rw- 0 0 7889 <2311-C-Drive.dd-fl_cal[1].gif-dead-7889>
175 mac -rw-rw-rw- 0 0 7900 <2311-C-Drive.dd-trash[1].gif-dead-7900>
Mon Jan 10 2005 09:30:43 1350656 .a. -rw-rw-rw- 0 0 1741-128-3 C:/WINDOWS/system32/mshtml.tlb
Mon Jan 10 2005 09:30:44 18108 m.c -rw-rw-rw- 0 0 8193-128-4 C:/WINDOWS/system32/dllcache/mmmdd.dll (deleted)
18108 m.c -rw-rw-rw- 0 0 8193 <2311-C-Drive.dd-webacc[2]-dead-8193>
Mon Jan 10 2005 09:31:22 18108 .a. -rw-rw-rw- 0 0 8193-128-4 C:/WINDOWS/system32/dllcache/mmmdd.dll (deleted)
18108 .a. -rw-rw-rw- 0 0 8193 <2311-C-Drive.dd-webacc[2]-dead-8193>
Mon Jan 10 2005 09:31:24 21021 m.c -rw-rw-rw- 0 0 8209-128-4 C:/WINDOWS/system32/dllcache/mplay32.exe (deleted)
111 mac -rw-rw-rw- 0 0 8218 <2311-C-Drive.dd-imgattach[1].gif-dead-8218>
21021 m.c -rw-rw-rw- 0 0 8209 <2311-C-Drive.dd-webacc[2]-dead-8209>
495 .a. -rw-rw-rw- 0 0 7796 <2311-C-Drive.dd-logo[1].gif-dead-7796>
1963 m.c -rw-rw-rw- 0 0 8217 <2311-C-Drive.dd-HREF[1]-dead-8217>
111 mac -rw-rw-rw- 0 0 8218-128-1 C:/WINDOWS/system32/dllcache/mqad.dll (deleted)
111 mac -rw-rw-rw- 0 0 8218-128-1 C:/Documents and Settings/crossroads/Local Settings/
Temporary Internet Files/Content.IE5/BNHJKVVY/imgattach[1].gif (deleted)

```

At 09:31:32, cache files were created that appear to be from an adult web site. Some of the filenames contain the term “Lolita”, a commonly used term in pornography for young models (incidentally, a term later used as a keyword in string searches). The cache file indicating the first web page outside of the internal network visited is named “adult-portal.hk[1].htm”, possibly indicating the name of the domain for the website.

| Terminal | | | | | | | | | |
|--------------------------|-------|------|--------------|-----|------|------------|---|--|--|
| File | Edit | View | Terminal | Tab | Help | | | | |
| Mon Jan 10 2005 09:31:32 | 440 | mac | -/-rwXrwXrwX | 0 | 0 | 8223-128-1 | C:/WINDOWS/system32/dllcache/mqise.dll (deleted) | | |
| | 1269 | mac | -/-rwXrwXrwX | 0 | 0 | 8304-128-4 | C:/Documents and Settings/crossroads/Local Settings/Temporary Internet Files/Content.IE5/CP27CTYJ/lolita[1].css (deleted) | | |
| | 1269 | mac | -/-rwXrwXrwX | 0 | 0 | 8304 | <2311-C-Drive.dd-lolita[1].css-dead-8304> | | |
| | 440 | mac | -/-rwXrwXrwX | 0 | 0 | 8223 | <2311-C-Drive.dd-adult-portal.hk[1].htm-dead-8223> | | |
| Mon Jan 10 2005 09:31:33 | 13737 | mac | -/-rwXrwXrwX | 0 | 0 | 8308 | <2311-C-Drive.dd-0_1[1].jpg-dead-8308> | | |
| Mon Jan 10 2005 09:31:34 | 14833 | mac | -/-rwXrwXrwX | 0 | 0 | 8310 | <2311-C-Drive.dd-0_2[1].jpg-dead-8310> | | |
| Mon Jan 10 2005 09:31:35 | 3736 | mac | -/-rwXrwXrwX | 0 | 0 | 8314 | <2311-C-Drive.dd-1[1].jpg-dead-8314> | | |
| | 3736 | mac | -/-rwXrwXrwX | 0 | 0 | 8314-128-4 | C:/Documents and Settings/crossroads/Local Settings/Temporary Internet Files/Content.IE5/CP27CTYJ/1[1].jpg (deleted) | | |
| | 990 | mac | -/-rwXrwXrwX | 0 | 0 | 8320-128-4 | C:/WINDOWS/system32/dllcache/msoert2.dll (deleted) | | |
| | 3491 | mac | -/-rwXrwXrwX | 0 | 0 | 8325 | <2311-C-Drive.dd-b[1].jpg-dead-8325> | | |
| | 990 | mac | -/-rwXrwXrwX | 0 | 0 | 8320 | <2311-C-Drive.dd-0_3[1].jpg-dead-8320> | | |
| Mon Jan 10 2005 09:31:36 | 25495 | mac | -/-rwXrwXrwX | 0 | 0 | 8335 | <2311-C-Drive.dd-gentleangelsan[1].jpg-dead-8335> | | |
| | 25495 | mac | -/-rwXrwXrwX | 0 | 0 | 8335-128-4 | C:/WINDOWS/system32/dllcache/msscp.dll (deleted) | | |
| | 13396 | mac | -/-rwXrwXrwX | 0 | 0 | 8334-128-4 | C:/Documents and Settings/crossroads/Local Settings/Temporary Internet Files/Content.IE5/OY4FG9EF/flylolitas[1].jpg (deleted) | | |
| | 13396 | mac | -/-rwXrwXrwX | 0 | 0 | 8334-128-4 | C:/WINDOWS/system32/dllcache/msscds32.ax (deleted) | | |
| | 13396 | mac | -/-rwXrwXrwX | 0 | 0 | 8334 | <2311-C-Drive.dd-flylolitas[1].jpg-dead-8334> | | |
| Mon Jan 10 2005 09:31:37 | 14563 | mac | -/-rwXrwXrwX | 0 | 0 | 8345-128-4 | C:/Documents and Settings/crossroads/Local Settings/ | | |

The timeline continues on with new cache files being created, mostly suggesting pornographic web browsing, until 10:01:44, when the internet browsing activity stopped. Little activity occurred on the timeline until 11:14, the time when I began working on the suspect system myself.

The timeline proved beneficial in several ways. It showed that the system had very little activity on it, and helped us learn more about the system by forcing us to interview the administrator of the system. It also gave us an indication of the time with which the potentially illegal activity in question began, and when the activity ended.

The timeline was not only helpful, but necessary as well. Reviewing the timeline assisted us in identifying if perhaps a virus had infected the system, or perhaps spyware had gotten a foothold on the system. From the timeline, there did not appear to be any such activity – the machine idled normally and no suspicious files were created, modified, or accessed throughout the timeline, though further testing would be required to confirm or debunk that theory – specifically, a virus scan would be conducted.

Mounting the Images

The following commands mounted the C and D Drive image onto the mount points specified for further analysis:

```
mount -t ntfs -o loop,ro,umask=0222,noatime,noexec /home/dshettle/incidents/incident-1-10-2005/2311-C-Drive.dd /mnt/hack/windows_mount/CDrive
mount -t ntfs -o loop,ro,umask=0222,noatime,noexec /home/dshettle/incidents/incident-1-10-2005/2311-D-Drive.dd /mnt/hack/windows_mount/DDrive
```

Mounting the images allows for traversal of the drive images as if they were an operating file system. This in turn allows for more traditional actions against the images, such as running a virus scan, or copying files from the images. The images are mounted in such a way that neither the images themselves nor the

contents of the images are modified by any operation in the mounted drives – they are read-only. Access times to the files on the images do not change either. The images remain intact regardless of actions taken while the images are mounted.

Antivirus Check

After having mounted the drives, I decided to run an antivirus check to confirm that the activity noticed was not the result of a virus or malware, as is often the defense in such cases. Using CLAMAV™ software (ClamAV 2005), I first updated the AV signatures using *freshclam*, CLAMAV™ software's update engine, and then issued the following commands to initiate the scans:

```
clamscan -i -r --log=/tmp/2311-clamscan.log --database=/var/lib/clamav CDrive &  
clamscan -i -r --log=/tmp/2311-clamscan.log --database=/var/lib/clamav DDrive &
```

The scans ran for a significant amount of time, and in the end found no viruses.

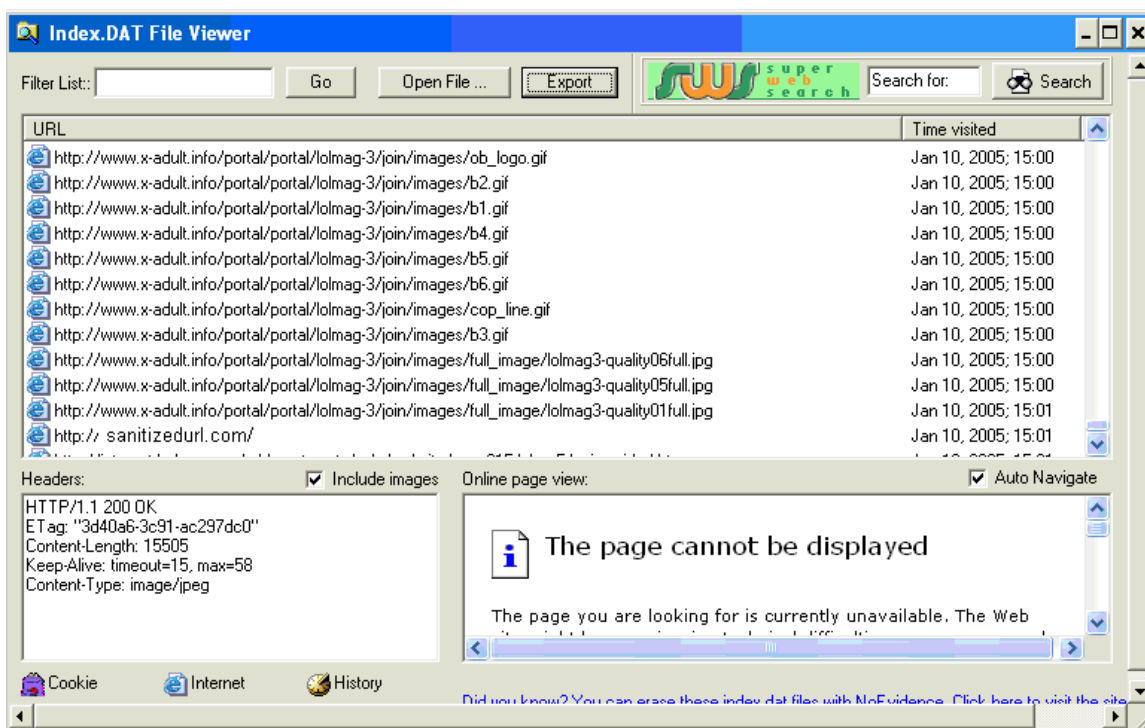
INTERNET EXPLORER™ History Analysis

The next step in my investigation was to make copies of some files, in particular files relating to INTERNET EXPLORER™ history. I copied the following files to shared network drive on a WINDOWS™ forensics analysis machine:

```
/mnt/hack/windows_mount/CDrive/Documents and Settings/crossroads/Local  
Settings/Temporary Internet Files/Content.IE5/index.dat  
/mnt/hack/windows_mount/CDrive/Documents and Settings/crossroads/Local  
Settings/History/History.IE5/index.dat  
/mnt/hack/windows_mount/CDrive/Documents and Settings/crossroads/Cookies/index.dat
```

where cached images and other web content is located in the temporary internet directory structure. Analysis of index.dat could, theoretically, create a full and accurate timeline of INTERNET EXPLORER™ utilization history, even more so than MACtimes analysis of the hard drive can create in that the index.dat file will list actual URL's accessed, not simply cache file names downloaded.

On the WINDOWS™ forensics analysis machine, I installed the INDEX.DAT VIEWER™ software (Marcovich 2003). The software parses index.dat files, and allows their contents to be exported into a comma separated file, which can then be manipulated by a spreadsheet application. Below is a screen-capture of the results of the parse prior to exporting the data (times are in GMT).



The index.dat file contained information from three dates.

| | |
|------------|--|
| 11/03/2003 | <ol style="list-style-type: none"> 1. windowsupdate.microsoft.com accessed 2. updates seemingly downloaded. |
| 11/23/2003 | <ol style="list-style-type: none"> 1. The organizations website accessed, specifically going to one department's website, then the activity stops. |
| 01/10/2005 | <ol style="list-style-type: none"> 1. The organizations website accessed. 2. The organization's email system was accessed. 3. A series of several apparently adult-themed websites were accessed over a period of time. |

Exploring the index.dat from the History.IE5 directory yielded similar results, though the file only contained activity from January 10th, 2005, and rather than listing information for every file downloaded, it listed only information based on clicks and actual URL's visited, as opposed to URL's visited and every file contained within those URL's.

The index.dat file from the cookies directory revealed two cookies set, both from a different time period than the one under investigation, and both apparently legitimate and non-malicious cookies.

Going back to the mounted file systems, I explored the temporary internet directories in more detail. I went into the directory and listed the files recursively inside the temporary internet files directories.

```

Terminal
File Edit View Terminal Tabs Help
[root@LinuxForensics Local Settings]# cd Temporary\ Internet\ Files\ Content.IE5\
[root@LinuxForensics Content.IE5]# find . -ls
 9433    0 dr-xr-xr-x  1 root   root         0 Jan 10 11:12 .
6688 148 dr-xr-xr-x  1 root   root    151552 Jan 10 11:12 ./0Y4FG9EF
5396   4 -r-xr-xr-x  2 root   root     1146 Jan 10 09:30 ./0Y4FG9EF/webaccedeb61b8[1]
7440   0 -r-xr-xr-x  2 root   root      259 Jan 10 09:30 ./0Y4FG9EF/webacc[1]
7094 156 dr-xr-xr-x  1 root   root    159744 Jan 10 11:12 ./89ABCDEF
7387   0 -r-xr-xr-x  2 root   root      580 Jan 10 09:30 ./89ABCDEF/webacc[1]
7794   4 -r-xr-xr-x  2 root   root      857 Jan 10 09:30 ./89ABCDEF/webacc[2]
9060   0 -r-xr-xr-x  2 root   root      479 Jan 10 09:30 ./89ABCDEF/index[1].htm
7030 148 dr-xr-xr-x  1 root   root    151552 Jan 10 11:12 ./BNHJKVVY
9062   4 -r-xr-xr-x  2 root   root     1202 Jan 10 09:30 ./BNHJKVVY/globalmenu[1].htm
7835   8 -r-xr-xr-x  2 root   root     7257 Jan 10 09:30 ./BNHJKVVY/webacc[1]
9065   4 -r-xr-xr-x  2 root   root     2295 Jan 10 09:30 ./BNHJKVVY/coursemenu[1].htm
5732 160 dr-xr-xr-x  1 root   root    163840 Jan 10 11:12 ./CP27CTYJ
7909  48 -r-xr-xr-x  2 root   root     29568 Jan 10 09:30 ./CP27CTYJ/webacc[4]
7784  16 -r-xr-xr-x  2 root   root      9800 Jan 10 09:30 ./CP27CTYJ/webacc[1]
22484   8 -r-xr-xr-x  2 root   root      4230 Jan 10 10:01 ./CP27CTYJ/universidad[1].htm
5429  16 -r-xr-xr-x  2 root   root     12076 Jan 10 09:30 ./CP27CTYJ/webacc[3]
 207 864 -r-xr-xr-x  1 root   root    884736 Jan 10 09:29 ./index.dat
[root@LinuxForensics Content.IE5]#

```

Apparently the temporary internet files had been deleted; this was also evident from the timeline previously captured. The next step was to retrieve the deleted cache files.

Retrieving Deleted Files

Retrieving the deleted files could be accomplished in several ways. Knowing I was dealing with images and html files primarily, I chose to use the sorter to retrieve the deleted files, however I could have retrieved the files using icat, as the timeline contained all the MFT record numbers for the deleted files. I could have also retrieved the files from the autopsy forensics browser. Since the files I'm looking for are primarily images and html files, sorter seemed an easier, albeit perhaps more resource-intensive way to accomplish this. I created the following sorter configuration file from pieces found in the samples included with SLEUTHKIT™:

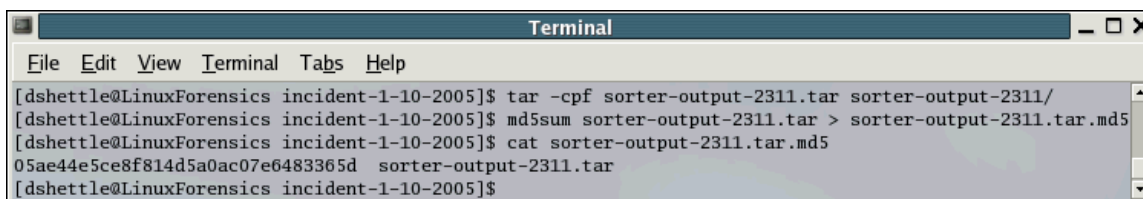
| | | |
|----------|----------------|--------------------|
| category | images | image data |
| ext | jpg, jpeg, jpe | JPEG image data |
| ext | gif | GIF image data |
| ext | tif | TIFF image data |
| ext | png | PNG image data |
| category | images | bitmap data |
| ext | bmp | PC bitmap data |
| category | text | HTML document text |
| ext | htm, html, hta | HTML document text |

```

sorter -h -m 'C:\' -d ./sorter-output-2311 -f ntfs -C ./sorter.conf -s -U ./2311-C-Drive.dd

```

Sorter recovered several thousand images and html pages. A quick analysis of the thumbnails confirmed the existence of contraband data in large quantities. I then created a tar file of the sorter output, and ran md5sum against the tar file, saving its output to a file and then burning the evidence to a CD, which I tagged as tag #2311-6, description: sorter output from C partition.



```

Terminal
File Edit View Terminal Tabs Help
[dshettle@LinuxForensics incident-1-10-2005]$ tar -cpf sorter-output-2311.tar sorter-output-2311/
[dshettle@LinuxForensics incident-1-10-2005]$ md5sum sorter-output-2311.tar > sorter-output-2311.tar.md5
[dshettle@LinuxForensics incident-1-10-2005]$ cat sorter-output-2311.tar.md5
05ae44e5ce8f814d5a0ac07e6483365d sorter-output-2311.tar
[dshettle@LinuxForensics incident-1-10-2005]$

```

Now, with the sorter output, the data from the timeline, as well as the CHAOS READER™ output and the data from the various index.dat files on the suspect system, establishing an order of events that occurred on the system was fairly straightforward.

I wanted to concretely link the order of events, and establish a purpose to the internet browsing activity. My goal was to recover at least the HTML files in the order they had been viewed. The timeline showed me the order of creation of the files, as did the index.dat analysis. I chose to tackle this the following way:

1. For each URL in index.dat, I would search the timeline for that file.
2. I would then get the MFT record number for that file from the timeline.
3. With that number, I would use the icat utility (another component of the SLEUTHKIT™ software) to extract that MFT record from the image file.
4. I would then view the extracted file, confirm that it is likely to be the cache file downloaded when accessing the given URL, then I would save that file in a directory with the original filename, preceded by a number representing the order in which that file was viewed.
5. In the cases where the html files retrieved were frames, I would keep the prominent frame where the following URL's link was, and discard the less-relevant data.

The purpose of this exercise was two fold -- when viewed in a web-browser, it creates a visual representation of the activity that occurred in the web browser. It also assists in establishing whether or not the page-views were solicited or not. If a link exists between the pages listed, then the websites were not likely accessed by automation (spyware, virus, etc). The process also facilitates determining what types of pages were visited (a portal site, a link page, a subscription page, etc.)

TCT™ Lazarus – Memory Dump Analysis

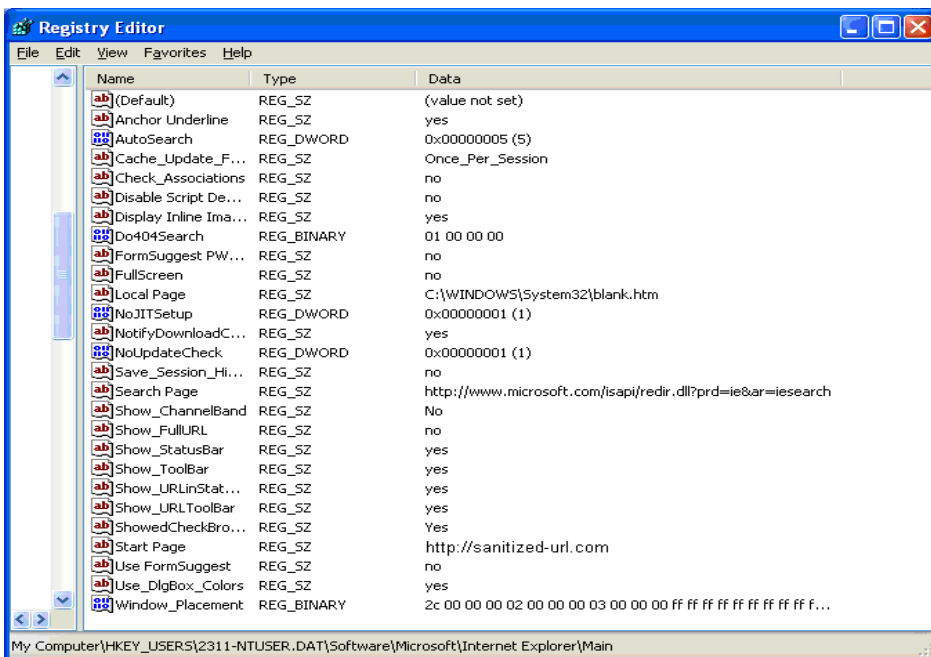
Having imaged the memory on the system, I was then able to analyze that data. At this time, I was confident that the activity that was suspected had indeed taken place, but memory analysis would add to that confidence and bolster the case. I would approach the memory analysis two ways: First, via Lazarus, a component of THE CORONER'S TOOLKIT™ (Farmer & Venema, 2005). Lazarus is a tool that analyzes raw data, like that contained in a memory dump, and attempts to categorize the data it finds into file types. It also conveniently

outputs the results into HTML, where you can browse through the findings. The results of Lazarus proved useful, as memory contained a great deal of the HTML pages visited by the suspect, as well as some images. Most of the information discovered had been already extracted from the file-system images via sorter, however this corroborated that data. See appendix D for screenshots of Lazarus and some of the data extracted.

NTUSER.DAT and Registry Analysis

Having seen the initial IE activity to be going to our organization's website, I wanted to determine whether the logged in user "crossroads" had its home page set in INTERNET EXPLORER™ to the organization's website. I suspected that to be the case, however I wanted proof. No longer having access to the running operating system on the suspect system, I looked to the NTUSER.DAT file for information. NTUSER.DAT contains all profile information for a given user. It should contain the home page setting for the "crossroads" user. An ASCII string search against the file yielded many results, but nothing that showed what would appear to be a home page. A Unicode string search did provide URL's, but not descriptions of what the URL's represented in the registry. Research was required to figure out how to dig deeper into the NTUSER.DAT file.

After some research (experts-exchange 2005), I copied the NTUSER.DAT file over to my WINDOWS™ forensics workstation to load the hive into regedit.exe for investigation. Regedit.exe is the windows registry edit, and it allows for importing offline registry files for evaluation. Launching regedit, I selected HKEY_USERS, then loaded the NTUSER.DAT copied over as a hive named 2311-NTUSER.DAT. I then searched through the imported hive for our organization's website. The value was found in Software\Microsoft\Internet Explorer\Main under the record Start Page: the registry entry for the home page setting in INTERNET EXPLORER™. This confirmed that INTERNET EXPLORER™, when launched by the suspect, would default to going to the organizations webpage. See the screenshot of regedit below:



Order of Events

At this point, I began creating the order of events timeline based on the information I had collected from all sources. I cross-referenced MFT record entries from the timeline with MFT record entries found in the sorter output, as well as the output gathered from the icat process extracting information based on index.dat and the timeline. The following is the order of events:

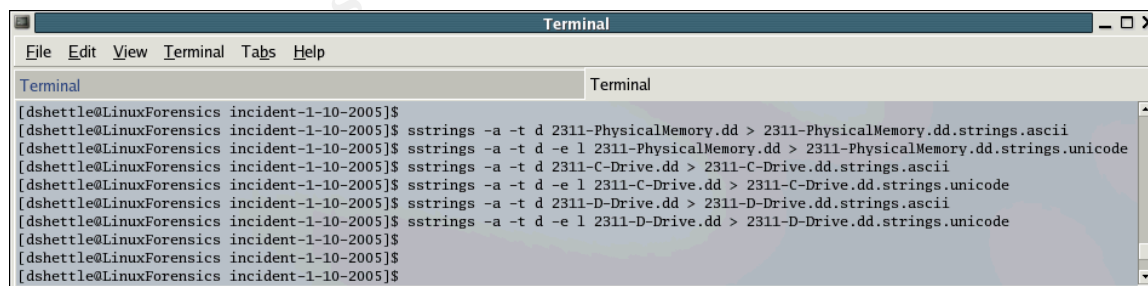
1. 9:28:45 - The suspect logged into the suspect system as the generic crossroads user.
2. 9:30:00 - The suspect opened an INTERNET EXPLORER™ browser.
3. 9:30:00 - The browser defaulted to going to the organization's website.
4. 9:30:15 - The organizations' email webaccess login page was accessed.
5. 9:30:21 - Academic section of website accessed.
6. 9:30:24 – Self Paced learning website accessed.
7. 9:30:25 – Courses webpage accessed.
8. 9:30:27 – Intermediate Spanish web page accessed.
9. 9:30:30 – Chapter menu webpage accessed
10. 9:30:40 – Organization email web access was logged into as John Doe.
11. 9:31:24 – An email is opened (see screenshot in appendix C).
12. 9:31:32 – First adult website containing contraband is accessed – a link from within the email.
13. 9:31:32 until 09:58 – several websites accessed. Several galleries of thumbnail images perused.
14. 9:58:33 – thumbnail expanded into full size image.
15. 9:58:49 – thumbnail expanded into full size image.
16. 9:59:12 – thumbnail expanded into full size image.

- 17.9:59:27 – thumbnail expanded into full size image.
- 18.9:59:44 – more contraband web pages loaded, galleries viewed.
- 19.10:00:45 – thumbnail expanded into full size image.
- 20.10:00:59 – thumbnail expanded into full size image.
- 21.10:01:16 – thumbnail expanded into full size image.
- 22.10:01:43 – Spanish chapter accessed (from step 9)
- 23.10:01:44 – Frame from chapter loaded.

At 10:01:44, the activity ended. From the above, as well as the evidence behind the above summary, it can be deduced that the suspect opened a browser, went to the organization's web email system, opened a new window, then navigated to a course webpage. The suspect then logged into the email system, and opened an email which launched a new window. The suspect then clicked on a link within the email, bringing them to a child pornography portal linking to several other sites. From the portal, the suspect jumped from site to site browsing galleries of child pornography thumbnails. Later on, the suspect began expanding the thumbnails and browsing full size high resolution images. The suspect then discontinued browsing and returned to the open window containing the foreign language instruction, clicked on a couple of links on the organizations website, and stopped browsing the internet, leaving both the email access window and the foreign language instruction window open.

String Searches

The session from login to idle lasted from 09:28:45 to 10:01:44, a total of 32 minutes and 59 seconds. At this time, confident that forensic analysis as it pertains to the case at hand was close to finished, I decided to run some string searches to tie up any loose ends. Using sstrings, the strings utility packaged with the SLEUTHKIT™, I extracted both ASCII and Unicode strings from both drives, as well as the physical memory images.



```

Terminal
File Edit View Terminal Tabs Help

[dshettle@LinuxForensics incident-1-10-2005]$ sstrings -a -t d 2311-PhysicalMemory.dd > 2311-PhysicalMemory.dd.strings.ascii
[dshettle@LinuxForensics incident-1-10-2005]$ sstrings -a -t d -e l 2311-PhysicalMemory.dd > 2311-PhysicalMemory.dd.strings.unicode
[dshettle@LinuxForensics incident-1-10-2005]$ sstrings -a -t d 2311-C-Drive.dd > 2311-C-Drive.dd.strings.ascii
[dshettle@LinuxForensics incident-1-10-2005]$ sstrings -a -t d -e l 2311-C-Drive.dd > 2311-C-Drive.dd.strings.unicode
[dshettle@LinuxForensics incident-1-10-2005]$ sstrings -a -t d 2311-D-Drive.dd > 2311-D-Drive.dd.strings.ascii
[dshettle@LinuxForensics incident-1-10-2005]$ sstrings -a -t d -e l 2311-D-Drive.dd > 2311-D-Drive.dd.strings.unicode
[dshettle@LinuxForensics incident-1-10-2005]$
[dshettle@LinuxForensics incident-1-10-2005]$
[dshettle@LinuxForensics incident-1-10-2005]$

```

At this time I created a keywords text file, containing keywords known to be associated with internet child pornography. Keywords were gleaned from meta-tags of recovered html files from the sorter results, as well as keywords established in the BLEEDINGSNORT™ software (BleedingSnort 2004) rule-set. The file contained keywords relative to the offense, and keywords relative to the suspect's name and username: lolita, r@ygold, zeps, preteen, pre-teen, pthc,

pedo, incest, underage, kiddy, "John Doe", jdoe.

I then ran grep against the files collected (partial output is shown in the following image). The string search produced too many results to be of any use in gathering anything of any importance that hadn't already been gathered, however, through the results of the search, it was easy to establish that someone had indeed used the suspect system to search for illicit materials.

```

Server
File Edit View Terminal Tabs Help

Server
[root@LinuxForensics img]$ grep -i -f keywords-childporn.txt 2311-PhysicalMemory.dd.strings.*
2311-PhysicalMemory.dd.strings.ascii: 10562616 NWDSitrTypeDown
2311-PhysicalMemory.dd.strings.ascii: 10823107 PEDO sit
2311-PhysicalMemory.dd.strings.ascii: 10952731 MyPreTeenURL
2311-PhysicalMemory.dd.strings.ascii: 47672605 dptHcpt
2311-PhysicalMemory.dd.strings.ascii: 47672637 dptHcpt
2311-PhysicalMemory.dd.strings.ascii: 72884290 <META name="description" content="! Adult SITES PORTAL ! Lolita
olita nymphets , lolita bbs many free preteen and lolita links. Zeps Guide lolita nymphets , lolita bbs , prete
ta , lolita sex lolita bbs russian lolita free lolita little lolita lolita bbs lolita lolita and bbs lolita l
bbs links lolit
2311-PhysicalMemory.dd.strings.ascii: 115681640 http://www.x-adult.info/portal/banners/preteensnature-02.jpg
2311-PhysicalMemory.dd.strings.ascii: 115681704 preteensnature-02[1].jpg
2311-PhysicalMemory.dd.strings.ascii: 115682024 http://www.x-adult.info/portal/banners/preteensnature-01.jpg
2311-PhysicalMemory.dd.strings.ascii: 115682088 preteensnature-01[1].jpg

```

Narrowing the keywords file down to just the suspect's username proved more interesting, however. A grep against the physical memory and C drive strings files revealed the suspect's mail password in a URL, apparently being posted to our organization's server for authentication.

```

Server
File Edit View Terminal Tabs Help

Server
2311-PhysicalMemory.dd.strings.ascii: 125541968 User.id=jdoe&User.password=nikkie&User.context=ip6gg6Pnbss2bj9P
mc&User.interface=frames&error=login&merge=webacc&action=User.Login&Url.hasJavaScript=1&Login.x=0&Login.y=0
2311-PhysicalMemory.dd.strings.ascii: 126110000 User.id=jdoe&User.password=nikkie&User.context=ip6gg6Pnbss2bj9P
mc&User.interface=frames&error=login&merge=webacc&action=User.Login&Url.hasJavaScript=1&Login.x=0&Login.y=05
2311-PhysicalMemory.dd.strings.unicode: 12473292 tp://mungedhost.net/servlet/webacc?User.context=ip6gg6Pnbss2b
j9Pmc&action=Item.Action&Item.drn=12091z1z0&Item.Read=&Url.Item.Reply=1&Url.Item.Reply.to=JCULL@mungedhost.net
,+jdoe.PO_HOLDEN.DOMAIN&Url.Item.Reply.cc=jbreilly.PO_ELMER.DOMAIN&Item.Reply=&merge=send
2311-PhysicalMemory.dd.strings.unicode: 50349348 http://mungedhost.net/servlet/webacc?User.context=ip6gg6Pnbss
2bj9Pmc&action=Item.Action&Item.drn=12091z1z0&Item.Read=&Url.Item.Reply=1&Url.Item.Reply.to=JCULL@mungedhost.n
et,+jdoe.PO_HOLDEN.DOMAIN&Url.Item.Reply.cc=jbreilly.PO_ELMER.DOMAIN&Item.Reply=&merge=send
2311-PhysicalMemory.dd.strings.unicode: 75117938 jdoe
[root@LinuxForensics img]$

```

Determining what file was holding that information required finding out the cluster size of the file system on the disk. That was achieved with SLEUTHKIT™'s fsstat utility. The utility tells you sector size, cluster size, as well as things like the location of the MFT. The fsstat utility revealed the cluster size to be 4096. On location on the drive where the password was found was at byte: 8572361008. The formula for determining the cluster of a byte location is : (location / clusterSize), or in this case, 8572361008 / 4096 = 2092861. To confirm that cluster is the location of the information found in the string search, the dcat utility is issued on that location and piped to strings. The string from the string search was indeed in that cluster. Then, to find out what file (or piece of

file) that cluster contained, the ifind utility was used. The ifind utility will locate the MFT record for the cluster specified (if one can be found), returning the MFT record number.

The ifind command returned 22996-128-3 as the MFT record number for the specified cluster. Next, the istat utility was used. The istat utility displayed information about an MFT record. It turns out that the file in particular was pagefile.sys, the WINDOWS™ paging file. WINDOWS™ utilizes this file to simulate having more memory for when the physical memory in a system is insufficient for the tasks being executed. Data found in pagefile.sys is data that had been, at one time, in physical memory, but had been swapped out to make room for other data requiring physical memory.

Repeating the above process on all the instances where the password was found in the strings file, revealed that the password was stored in RAM, but passed to the file system via the windows paging system. The password was not stored in a file or a cache file, rather it was likely a URL sent to the web server for authentication.

Loose Ends – Windows Forensic Toolchest Analysis & DEEP FREEZE™

Satisfied with the string search, there was one last problem that needed to be addressed with the suspect system. Given the nature of the DEEP FREEZE™ software installed on the system, it may be of significance to establish exactly when the system was last rebooted, as well as who had logged on to the system since that date. Having run the WINDOWS™ Forensic Toolchest at the scene, I decided to review its results to see if it had gathered the information needed, and perhaps other useful information.

The Toolchest had run the psinfo (Russinovich 2004) utility, which had captured the uptime of the suspect system. According to psinfo, the system had been up for 3 days 2 hours and 26 minutes, which corresponded with the timeline activity that had begun three days prior. Psinfo also reported the install date of the system as being 10/1/2002, which concurs with what the administrator had reported in the interview, as well as what had occurred in the timeline.

The Toolchest ran dozens of utilities gathering a significant amount of data. In reviewing the data, nothing stood out as being abnormal or unusual. I had hoped to find out a history of logins on the system, but that was not retrieved as auditing was not enabled on the system in question.

Conclusion

Identifying the Suspect

When the system was seized initially, the user John Doe was logged in to the

organization's email system. Later, data collected from the extracted image and html files confirmed that user John Doe had logged in to his email account, then initiated the actions that led to the investigation. Strings searches confirmed the presence of the user's full name and username in the physical memory of the system at the time of seizure. That is the extent of the evidence ascertained that tied this suspect to the events that took place. No direct link exists between the person and the actions that took place, from an electronic data perspective, only the email account and the actions that took place.

Assessing Intent

From experience, the typical defense in cases involving internet child pornography is often any one or more of the following: The suspect claims to have been a victim of malicious popup ads; the suspect claims to have received spam; and sometimes, the suspect received unsolicited emails and was attempting to unsubscribe. Malicious popup ads require the end-user to be browsing to websites that feed the malicious popup ads to the user, or require the users system to have malicious software installed on their system causing unsolicited popup ads.

In the present investigation, there was no evidence of any malicious software having been on the suspect system in question. Virus scans and the file activity timeline support the lack of evidence. Additionally, there was no evidence of web-browsing beyond the specific sites visited in succession and over time. The websites visited were adult websites exclusively geared towards child pornography, with the exception of the use of the organization's email system, and a few internal web pages.

The forensic data shows that an email was indeed received, but the email itself did not force any browsing activity. Recovery of the email from deleted cache files show that the email contained nothing that would trigger new web-pages to be loaded. The email was opened, a link clicked, and further links clicked for a period of over 20 minutes.

As for the unsubscribe defense, forensic evidence shows that the suspect using the system was browsing through galleries of thumbnails. The html files recovered and the MAC times associated with them highly suggest that the intent was browsing to view images. The order of events does not suggest that the suspect followed any logical path towards an unsubscribe option, in fact the suspect clicked on several thumbnails of images of child pornography to expand them into full size high resolution images (which I recovered from deleted cache files) – activity inconsistent with someone attempting to unsubscribe, and activity more consistent with someone with the intent to browse for the material under the scrutiny of this investigation. While the possibility for any of the defenses listed exists, the probability of them being true based on the forensic evidence gathered is low.

Summary

In conclusion, it is my belief based on the evidence discovered from forensic analysis of the system that the user of the system at the time of the packet capture, consistent with the times found in analysis of the system, viewed these web pages actively and with intent to view the material viewed. It is also my opinion that the materials viewed were illegal, and in violation of policies at my organization.

Additional Information

Below are links to resources I found useful throughout the course of this investigation. I've included of a brief description of the link.

Link: <http://securityfocus.org/infocus/1827>

Description:

This is a decent article by Keith J. Jones and Rohyt Belani on browser forensics. While I didn't use the methods detailed in their article, I could have and it would have facilitated things. WEB HISTORIAN™ is one tool in particular that I tested out after the investigation that looks very promising for IE history extraction.

Link: <http://public.findlaw.com/>

Description:

FINDLAW™ is an excellent research tool for finding information on laws at almost all levels of government, particularly state and federal. Searching is fast and efficient, and did throughout the course of this investigation, yield accurate results every time.

Link: <http://www.windowstlibrary.com/Content/121/07/1.html?Ad=1&>

Description:

Throughout the course of the investigation, we ran into several hurdles with our IT Acceptable Use Policy. We discovered that our policy had significant room for improvement. The article linked above, while dated, highlights a couple problems we encountered, and is a decent template from which to scribe an effective acceptable use policy, something we are currently re-assessing.

Link: <http://www.safehack.com/products/forensic/des.htm>

Description:

This article discusses the "Double-Edged Sword" nature of the DEEP FREEZE™ product, and its ability to wreck a forensics investigation. I ran into this article in the course of the investigation when researching the perils of the product. We were concerned over whether the legal end of the investigation could be hampered by the installation of the software. The article offers some insight into both the benefits and the dangers of

running DEEP FREEZE™.

Link: <http://www.law.duke.edu/journals/dltr/articles/2002dltr0019.html>

Description:

Questions arose during the course of the investigation as to whether or not the images discovered were real, or possibly virtual. The concept of virtual child pornography spurred some research into the subject, which is apparently a hot debate pitting the first amendment against the protection of children. This article, though somewhat opinionated, covers the basics of the debate.

© SANS Institute 2000 - 2005, Author retains full rights.

Appendix A

Summary of Evidence Collected

Below is the list of evidence collected, seized, and secured as each item was tagged and documented in my notes:

| Tag Number | Description | Serial Number |
|------------|--|---------------|
| 2311-1 | Dell Optiplex GX-240: 2.2GHZ Processor, 256 megabytes of RAM, 1 internal hard drive, 1 internal CDROM drive, 1 internal floppy drive. | 49C8K01 |
| 2311-1.1 | 20 Gigabyte Maxtor Internal Hard Drive from 2311-1 | B8RJ5GSD |
| 2311-2.1 | DVD 1 of 5 containing files: 2311-PhysicalDrive0.dd.split.aa 2311-PhysicalDrive0.dd.split.ab | N/A |
| 2311-2.2 | DVD 2 of 5 containing files: 2311-PhysicalDrive0.dd.split.ac 2311-PhysicalDrive0.dd.split.ad | N/A |
| 2311-2.3 | DVD 3 of 5 containing files: 2311-PhysicalDrive0.dd.split.ae 2311-PhysicalDrive0.dd.split.af | N/A |
| 2311-2.4 | DVD 4 of 5 containing files: 2311-PhysicalDrive0.dd.split.ag 2311-PhysicalDrive0.dd.split.ah | N/A |
| 2311-2.5 | DVD 5 of 5 containing files: 2311-PhysicalDrive0.dd.split.ai 2311-PhysicalDrive0.dd.split.aj 2311-PhysicalDrive0.dd.split.md5 | N/A |
| 2311-3 | CDROM containing tcpdump file: incident-050110.dmp | N/A |
| 2311-4 | CDROM containing files: 2311-PhysicalMemory.dd 2311-Physicalmemory.dd.md5 | N/A |
| 2311-5 | CDROM containing files: chaosreader-output.tar.gz chaosreader-output.tar.gz.md5 | N/A |
| 2311-6 | CDROM containing the output of Sorter sorter-output-2311.tar sorter-output-2311.tar.md5 | N/A |

Appendix B

Evidence Chain of Custody

| Tag Number: 2311-1 | | | | |
|--------------------|------------------|--------------------------|--------------------------|---------------------------|
| Contact | Role | Date of custody transfer | Location Stored | Storage Security |
| David Shettler | Forensic Analyst | Initial | Scene of Incident | Security Officers in room |
| Lt. Jack Bower | Security Officer | 01/10/2005 – 13:10 EST | Security Department Safe | Locked by Key |

| Tag Number: 2311-1.1 | | | | |
|----------------------|------------------|--------------------------|--------------------------|---------------------------|
| Contact | Role | Date of custody transfer | Location Stored | Storage Security |
| David Shettler | Forensic Analyst | Initial | Scene of Incident | Security Officers in room |
| Lt. Jack Bower | Security Officer | 01/10/2005 – 13:10 EST | Security Department Safe | Locked by Key |

| Tag Number: 2311-2.1 | | | | |
|-----------------------|------------------|--------------------------|--------------------------|------------------|
| Contact | Role | Date of custody transfer | Location Stored | Storage Security |
| David Shettler | Forensic Analyst | 01/11/2005-12:00 EST | David's Office | |
| Captain Frank Furillo | Security Captain | 01/11/2005 – 15:10 EST | Security Department Safe | Locked by Key |

| Tag Number: 2311-2.2 | | | | |
|----------------------|------------------|--------------------------|-----------------|------------------|
| Contact | Role | Date of custody transfer | Location Stored | Storage Security |
| David Shettler | Forensic Analyst | 01/11/2005-12:00 EST | David's Office | |

| | | | | |
|--------------------------|---------------------|---------------------------|--------------------------------|------------------|
| Captain Frank Furillo | Security Captain | 01/11/2005 – 15:10 EST | Security Department Safe | Locked by Key |
|--------------------------|---------------------|---------------------------|--------------------------------|------------------|

© SANS Institute 2000 - 2005, Author retains full rights.

| Tag Number: 2311-2.3 | | | | |
|-----------------------|------------------|--------------------------|--------------------------|------------------|
| Contact | Role | Date of custody transfer | Location Stored | Storage Security |
| David Shettler | Forensic Analyst | 01/11/2005-12:00 EST | David's Office | |
| Captain Frank Furillo | Security Captain | 01/11/2005 – 15:10 EST | Security Department Safe | Locked by Key |

| Tag Number: 2311-2.4 | | | | |
|-----------------------|------------------|--------------------------|--------------------------|------------------|
| Contact | Role | Date of custody transfer | Location Stored | Storage Security |
| David Shettler | Forensic Analyst | 01/11/2005-12:00 EST | David's Office | |
| Captain Frank Furillo | Security Captain | 01/11/2005 – 15:10 EST | Security Department Safe | Locked by Key |

| Tag Number: 2311-2.5 | | | | |
|-----------------------|------------------|--------------------------|--------------------------|------------------|
| Contact | Role | Date of custody transfer | Location Stored | Storage Security |
| David Shettler | Forensic Analyst | 01/11/2005-12:00 EST | David's Office | |
| Captain Frank Furillo | Security Captain | 01/11/2005 – 15:10 EST | Security Department Safe | Locked by Key |

| Tag Number: 2311-3 | | | | |
|-----------------------|------------------|--------------------------|--------------------------|------------------|
| Contact | Role | Date of custody transfer | Location Stored | Storage Security |
| David Shettler | Forensic Analyst | 01/11/2005-12:00 EST | David's Office | |
| Captain Frank Furillo | Security Captain | 01/11/2005 – 15:10 EST | Security Department Safe | Locked by Key |

| Tag Number: 2311-4 | | | | |
|--------------------|--|--|--|--|
|--------------------|--|--|--|--|

| Contact | Role | Date of custody transfer | Location Stored | Storage Security |
|-----------------------|------------------|--------------------------|--------------------------|------------------|
| David Shettler | Forensic Analyst | 01/11/2005-12:00 EST | David's Office | |
| Captain Frank Furillo | Security Captain | 01/11/2005 – 15:10 EST | Security Department Safe | Locked by Key |

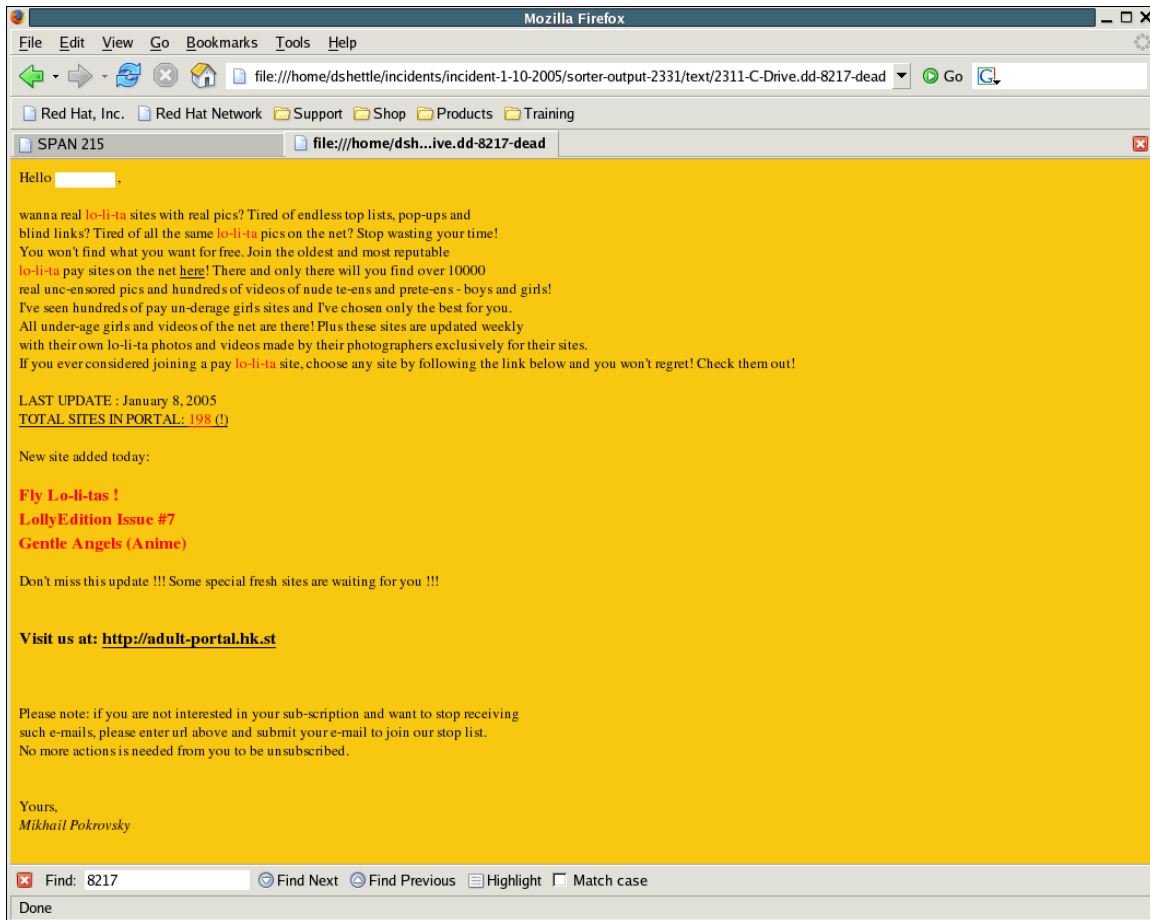
| Tag Number: 2311-5 | | | | |
|-----------------------|------------------|--------------------------|--------------------------|------------------|
| Contact | Role | Date of custody transfer | Location Stored | Storage Security |
| David Shettler | Forensic Analyst | 01/11/2005-12:00 EST | David's Office | |
| Captain Frank Furillo | Security Captain | 01/11/2005 – 15:10 EST | Security Department Safe | Locked by Key |

| Tag Number: 2311-6 | | | | |
|-----------------------|------------------|--------------------------|--------------------------|------------------|
| Contact | Role | Date of custody transfer | Location Stored | Storage Security |
| David Shettler | Forensic Analyst | 01/14/2005-08:00 EST | David's Office | |
| Captain Frank Furillo | Security Captain | 01/14/2005 – 09:30 EST | Security Department Safe | Locked by Key |

© SANS Institute

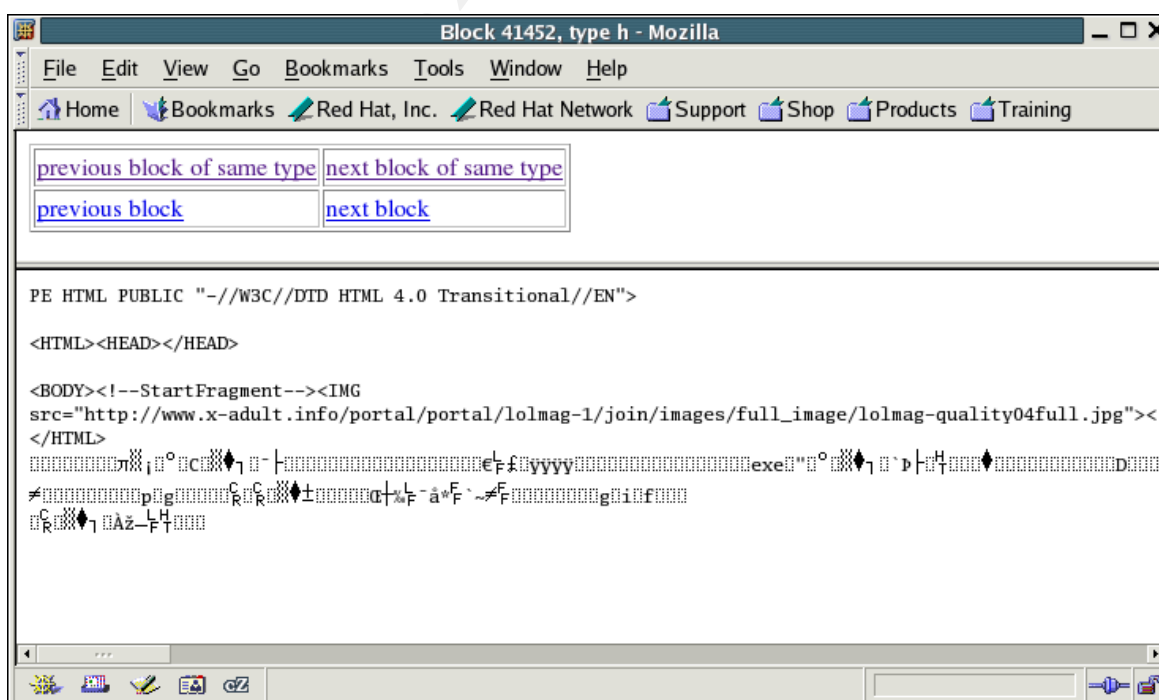
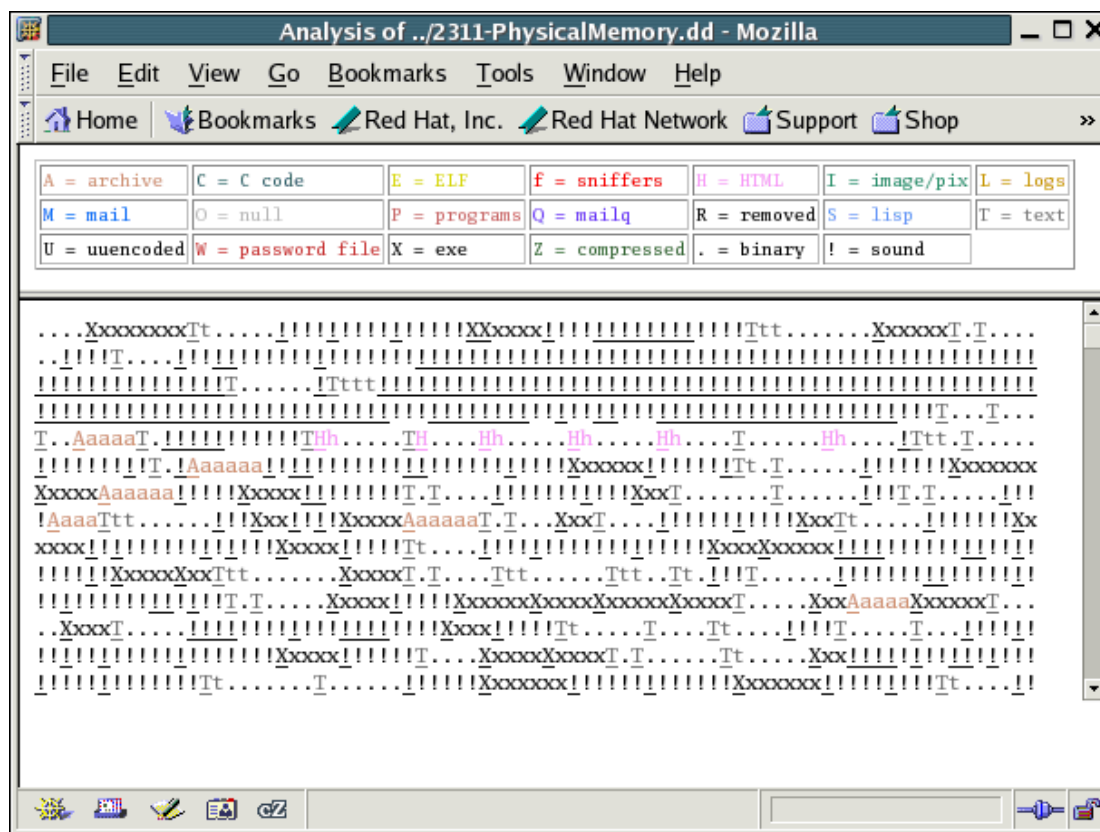
Appendix C

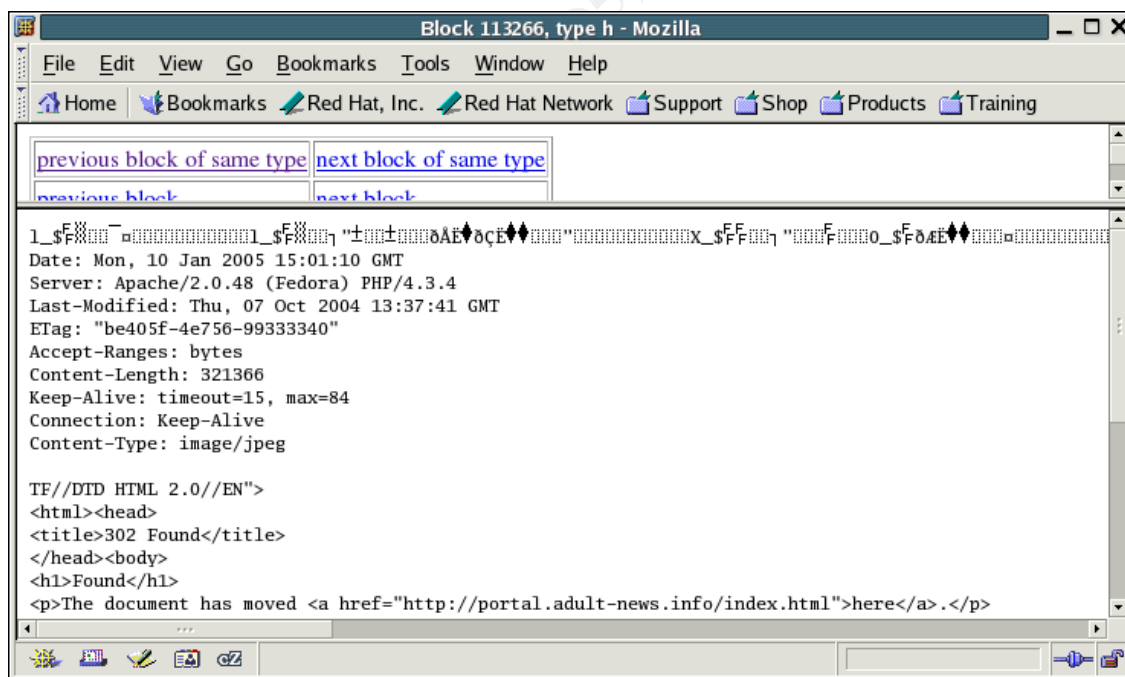
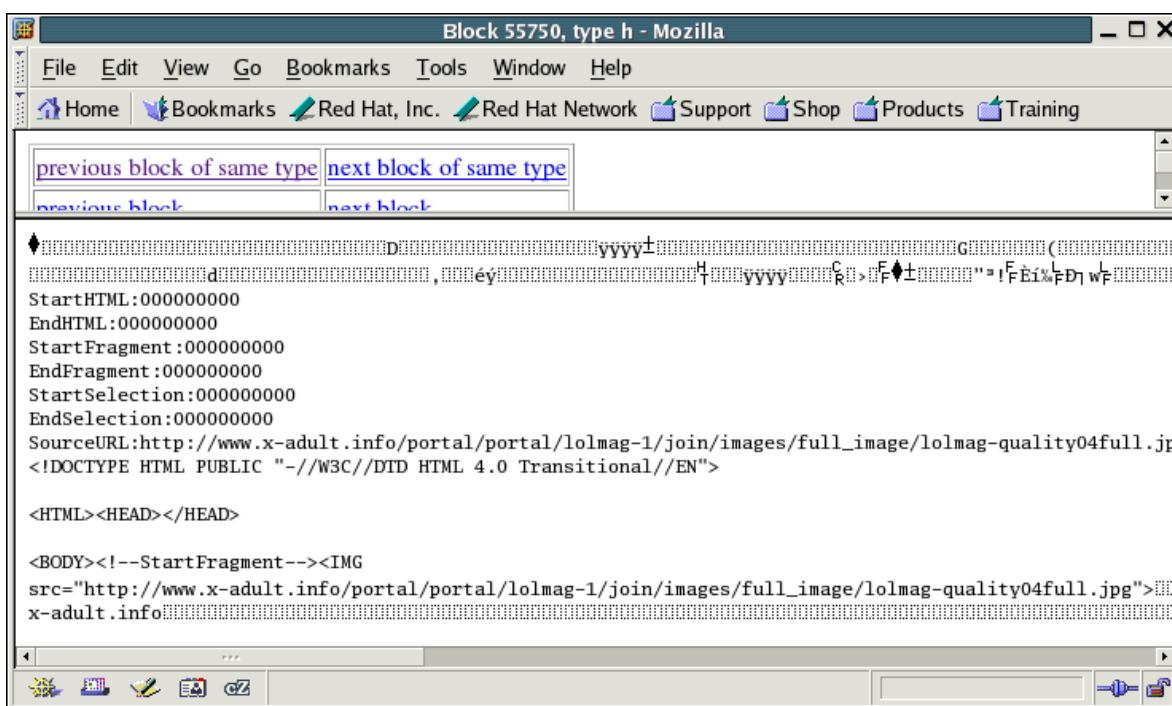
Screenshot of opened email (sanitized)



Appendix D

Lazarus Output





References

- Biatchux/DoD. (2002). "DCFL-DD." Retrieved 12/20, 2004, from http://sourceforge.net/project/showfiles.php?group_id=46038&package_id=44987.
- BleedingSnort. (2004). "Bleeding Snort Ruleset." Retrieved 12/17, 2004, from <http://www.bleedingsnort.com/>.
- Carrier, B. (2005). "The Sleuth Kit and The Autopsy Forensic Browser." Retrieved 12/20, 2004, from <http://www.sleuthkit.org/>.
- ClamAV. (2005). "Clam Antivirus Scanner." Retrieved 1/13, 2005, from <http://www.clamav.net>.
- Combs, G. et. al. (2004). "Ethereal: A Network Protocol Analyzer." 0.10.8. Retrieved 1/13, 2005, from <http://www.ethereal.com>.
- e-fense, I. (2004). "HELIX Incident Response and Computer Forensics Live CD." version 1.5. from <http://www.e-fense.com/helix/index2.html>.
- experts-exchange. (2005). "Finding Windows XP Serial off crashed computer." Retrieved 01/30, 2005, from http://www.experts-exchange.com/Operating_Systems/WinXP/Q_21276676.html.
- Farmer & Venema. (2005). "The Coroner's Toolkit" Retrieved 01/12, 2005, from <http://www.fish.com/tct/>.
- Gregg, B. (2004). "Chaosreader." version 0.93. from <http://users.tpg.com.au/bdgcvb/chaosreader.html>.
- Marcovich, J. (2003). "Index.DAT Viewer." version 2.1. from <http://www.exits.ro/index-dat-viewer.html>.
- McDougal, M. (2004). "Windows Forensic Toolchest." Retrieved 12/20, 2004, from <http://www.foolmoon.net/security/wft/>.
- Russinovich, M. (2004). "PsInfo." Version 1.62. Retrieved 12/20, 2004, from <http://www.sysinternals.com/ntw2k/freeware/psinfo.shtml>.
- U.S.Code. (2004). "U.S. Code Collection: Certain activities relating to material involving the sexual exploitation of minors." Retrieved 1/30, 2005, from http://www.law.cornell.edu/uscode/html/uscode18/usc_sec_18_00002252----000-.html.