



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>



Abstract: Analyze a floppy image using strict forensic methodology and industry standard tools. This process reveals what is on the floppy and how it was used. I determine the following as detailed in my findings.

James Shewmaker

GCFA Practical v1.5 Part 1

Introduction

Mr. David Keen presents me with a floppy and a chain of custody form. I will analyze the floppy to determine the extent of its use and contents. The floppy is immediately imaged to prepare for analysis. The examination begins by from an external analysis of the floppy filesystem.

Examination and Analysis

My analysis machine is a portable Shuttle XPC that has Windows XP Professional on a 160 GB IDE hard drive. For testing I use a virtual machine running inside of VMware Workstation. This software includes everything needed to host a virtual operating system except for the operating system itself.¹ With this software I can keep a fresh image and revert to a clean environment when needed. For the analysis itself, I use a Helix 1.4 bootable CD². This Helix CD is a powerful and trusted Linux distribution that has Autopsy, the Sleuth Kit³, and other tools for use in forensic analysis. I will only be using the Sleuth Kit during this analysis, for finer control. I then use the 160 GB Western Digital drive for data storage during analysis.

I logged on to my Helix system and downloaded the image with wget (a http and ftp download command), saving the image as v1_5.img.

I ran gunzip on the image to uncompress it, then ran md5sum to compare the uncompressed image with the checksum in the chain of custody form.

```
MD5: d7641eb4da871d980adbe4d371eda2ad fl-260404-RJL1.img
```

```
MD5: d7641eb4da871d980adbe4d371eda2ad v1_5.img
```

Since these checksums match, I know that the contents are the same, and the file was not corrupt from the original received from Mr. Keen.

Media Analysis

I run the fsstat command from the Sleuth Kit to see the type of filesystem the floppy was formatted with the mkdosfs command. Most PC floppies are formatted FAT12, but I will verify to get a baseline time of when the floppy was formatted and used.

The results look typical of an ordinary floppy filesystem, but I will confirm the results with the file command. The file command determines what type of file by assessing patterns of the file's contents.

¹ More information is available at http://www.vmware.com/products/desktop/ws_features.html

² Helix is available for download at <http://www.e-fense.com/helix/>

³ The Sleuth Kit (TSK) is available at <http://sleuthkit.sourceforge.net/>

```
Terminal
[Helix (y)]# /KNOPPIX/usr/local/sleuthkit-1.70/bin/fsstat -f fat v1_5
FILE SYSTEM INFORMATION
-----
File System Type: FAT

OEM Name: mkdosfs
Volume ID: 0x408bed14
Volume Label (Boot Sector): RJI
Volume Label (Root Directory): RJI
File System Type Label: FAT12

Sectors before file system: 0

File System Layout (in sectors)
Total Range: 0 - 2871
* Reserved: 0 - 0
** Boot Sector: 0
* FAT 0: 1 - 9
* FAT 1: 10 - 18
* Data Area: 19 - 2871
** Root Directory: 19 - 32
** Cluster Area: 33 - 2871

META-DATA INFORMATION
-----
Range: 2 - 45426
Root Directory: 2

CONTENT-DATA INFORMATION
-----
Sector Size: 512
Cluster Size: 512
Total Cluster Range: 2 - 2840

FAT CONTENTS (in sectors)
-----
105-187 (83) -> EOF
188-250 (63) -> EOF
251-316 (66) -> EOF
317-918 (602) -> EOF
919-1340 (422) -> EOF
1341-1384 (44) -> EOF
[Helix (y)]#
```

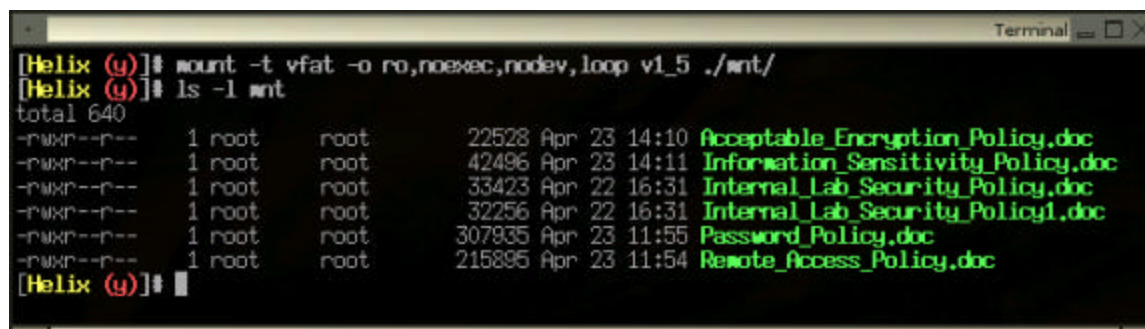
Figure 1

```
Terminal
[Helix (y)]# /KNOPPIX/usr/local/sleuthkit-1.70/bin/file v1_5
v1_5: x86 boot sector, code offset 0x3c, OEM-ID "mkdosfs", root entries 224, sectors 2872 (v
olumes <=32 MB), sectors/FAT 9, serial number 0x408bed14, label: "RJI", FAT (12 bit)
[Helix (y)]#
```

Figure 2

The results above (Figure 1, Figure 2) confirm what appears to be a normal image of a FAT12 formatted floppy filesystem. It happens to have a label of “RJL” and was created with a mkdosfs command, most likely from a Linux operating system.

Now I know how the floppy was formatted, and everything about the image looks normal, and I decide to mount the floppy in read only mode. I then take a look around and see if there is anything peculiar or interesting.



```
[Helix (u)]# mount -t vfat -o ro,noexec,nodev,loop v1_5 ./mnt/
[Helix (u)]# ls -l /mnt
total 640
-rwxr--r-- 1 root root 22528 Apr 23 14:10 Acceptable_Encryption_Policy.doc
-rwxr--r-- 1 root root 42496 Apr 23 14:11 Information_Sensitivity_Policy.doc
-rwxr--r-- 1 root root 33423 Apr 22 16:31 Internal_Lab_Security_Policy.doc
-rwxr--r-- 1 root root 32256 Apr 22 16:31 Internal_Lab_Security_Policy1.doc
-rwxr--r-- 1 root root 307935 Apr 23 11:55 Password_Policy.doc
-rwxr--r-- 1 root root 215895 Apr 23 11:54 Remote_Access_Policy.doc
[Helix (u)]#
```

Figure 3

I decide to look at these files (Figure 3) to see if there is anything obvious to help us in this investigation. Since the Password_Policy.doc file is one of the larger files, I open it with a khxedit to see if there are any obvious inconsistencies. Khxedit is a hex editor, which is a type of editor that displays the hexadecimal representation of each byte in a file. The only thing that seems odd--there is a reference at x6e17 to Cisco. It may be a bit odd that this word occurs in a password policy document when it isn't part of a technical discussion. Cisco is a common vendor though, so this may be nothing unusual.

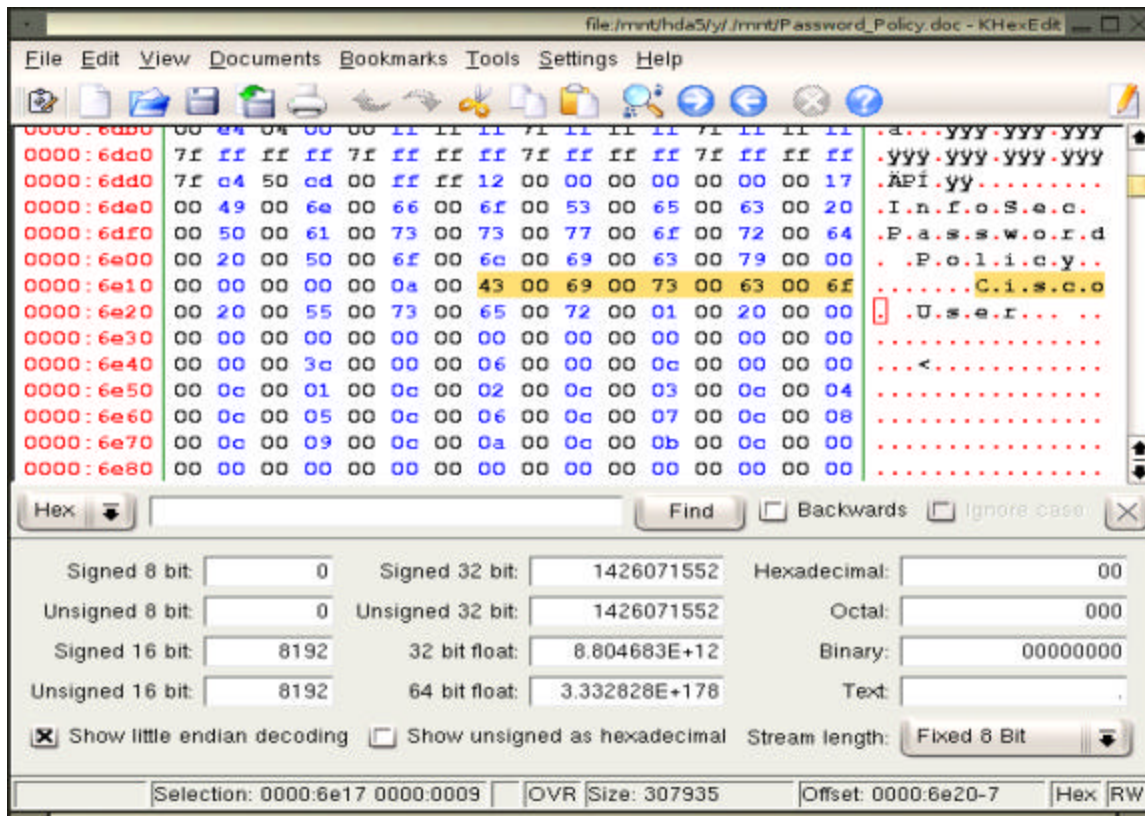


Figure 4

I play around with a copy of the Password_Policy.doc file (Figure 4), but I want to do it in a writable way, so I make a directory outside of the mounted image, name it “content” and copy the files there. Since I am finished mounting any images and will be working with copies of the files in the content directory, I type “exit” to return to my unprivileged shell account.

I try to open the same Password_Policy.doc file with abiword to look at the contents in a normal text view (Figure 5). Abiword is an open source editor that is capable of reading Microsoft Word formatted documents, and it comes installed on the Helix CD-ROM. Abiword prompts me for a password:

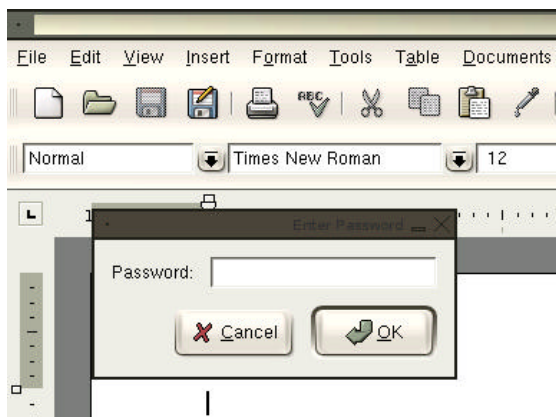


Figure 5

Clicking on “OK” displays an error message (Figure 6).

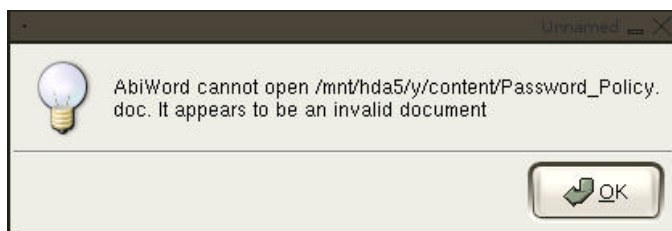


Figure 6

A blank password does not produce results, so I will put this idea on hold and examine another file. There is another file that behaves the same way, Remote_Access_Policy.doc. Now I have two interesting candidates, but not much in the way of obvious clues, so I decide to look around for deleted information

Since I finished mounting the image I use the dls command from the Sleuth Kit to dump the unallocated space from the FAT floppy image to a new v1_5.dls file (Figure 7).

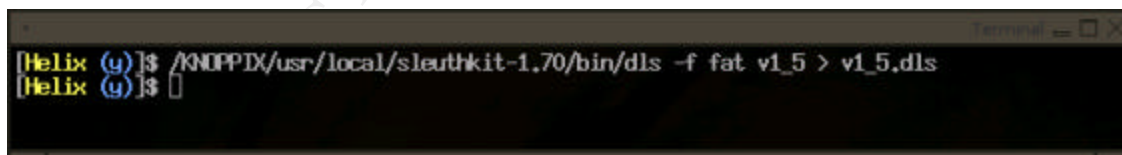


Figure 7

Then I use the command strings to pull out any human readable string of more than four characters on this deleted/unallocated space from the floppy (Figure 8).


```
Terminal
[Helix (y)]$ strings v1_5.dls | more
<HTML>
<HEAD>
<meta http-equiv=Content-Type content="text/html; charset=ISO-8859-1">
<TITLE>Ballard</TITLE>
</HEAD>
<BODY bgcolor="#EDED" >
<center>
<OBJECT classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=6,0,0,0"
WIDTH="800" HEIGHT="600" id="ballard" ALIGN=""
<PARAM NAME=movie VALUE="ballard.swf"> <PARAM NAME=quality VALUE=high> <PARAM NAME=bgcolor VALUE=#CCCCC> <EM
ED src="ballard.swf" quality=high bgcolor=#CCCCC WIDTH="800" HEIGHT="600" NAME="ballard" ALIGN=""
TYPE="application/x-shockwave-flash" PLUGINS PAGE="http://www.macromedia.com/go/getflashplayer"></EMBED>
</OBJECT>
</center>
</BODY>
</HTML>
11\SheCamouflageShell
ShellExt
VB5!
CamShell
BitmapShellMenu
CamouflageShell
CamouflageShell
Shell_Declares
Shell_Functions
ShellExt
modShellRegistry
kernel32
lstrcpyA
strlenA
ole32.dll
CLSIDFromProgID
StringFromGUID2
None
```

Figure 8

After the HTML code I see some references to “Camouflage” which sparks my interest. I also see the “VB5!” looks like a reference to Microsoft’s Visual Basic programming language. The “lstrcpyA” and “strlenA” look like function or sub routine names. The “ole32.dll” reference is not a surprise in this context, since it is a Microsoft library for interfacing files with applications. The word “Shell” that occurs often. This is interesting since shells are generally a user interface, either a graphical one like explorer in Windows or a command shell like bash on Linux.

Now I appear to have a footprint of a Visual Basic program for Windows in the unallocated space of the floppy. I will attempt to recover the deleted information. I have several key search words and pull up the v1_5.dls file in my hex editor, and can now move around and see the raw data easily.

I see typical company documentation that I saw with three of the files using abiword. I use my keyword list that contains some interesting strings I saw in the strings command output. Here are the hexadecimal locations and the keywords:

```
x1490  \.SheCamouflageShell
x1610  CamShell.BitmapShell Menu .. CamouflageShell
```



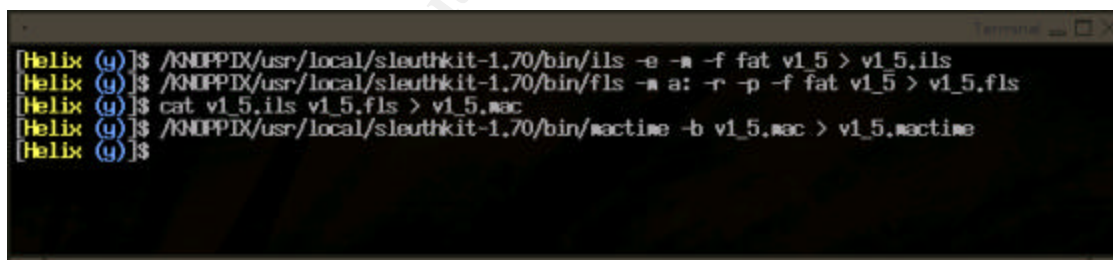
```
x1a26    CamouflageShell.vdp
x5050    CamShell.dll
```

At this point I have a few clues on the regular files. I proceed to recover the deleted files. Again, I make use of the excellent tools in The Sleuth Kit.

I need to first grab the structure of the image. I will use the Sleuth Kit's `ils` command to pull all of the metadata for each file (file name and other properties). To put things in perspective, I will grab all structures, not just the deleted ones by using the `-e` flag. I want to see when a file was last modified, accessed, or changed, using the `-m` flag. This is format is known as the mactime format. I save the output of the file in case I need it later, so I redirect the output to a new `v1_5.ils` file.

The other thing I need is the information on the actual file contents. The Sleuth Kit has an `fls` command that will list the files on the image, even the deleted ones. Although the deleted files may have some clues or even evidence, I need to examine them carefully, since the space on the filesystem was unused and possibly overwritten since the file's deletion. To accomplish this I use the `-r` flag to be sure I recover any subdirectories and their files, the `-p` flag to display their full path (in case of a subdirectory), and the `-a` flag to show me all the entries (even `"."` and `".."` for the current and parent directories). Since I want to save this output as well, I redirect its output to a new `v1_5.fl` file.

Now I would like to view this filesystem footprint, including both existing files and deleted file remnants. If the time of the source computer was off by a known value I could adjust it here. I do not know of any adjustments needed and I will generate the time line with the information on hand. The mactime tool from the Sleuth Kit was designed for this, so I combine the `v1_5.ils` and `v1_5.fl` files and process them into one timeline (Figure 9).



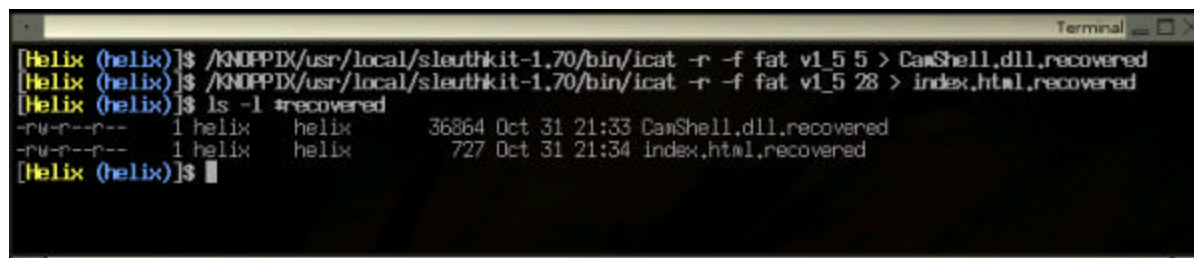
```
[Helix (u)]$ /KNOPPIX/usr/local/sleuthkit-1.70/bin/ils -e -m -f fat v1_5 > v1_5.ils
[Helix (u)]$ /KNOPPIX/usr/local/sleuthkit-1.70/bin/fls -m a: -r -p -f fat v1_5 > v1_5.fl
[Helix (u)]$ cat v1_5.ils v1_5.fl > v1_5.mactime
[Helix (u)]$ /KNOPPIX/usr/local/sleuthkit-1.70/bin/mactime -b v1_5.mactime > v1_5.mactime
[Helix (u)]$
```

Figure 9

The records that do not have a normal filename and use the `<>` symbols are the metadata structures. This helps us see the remnants of files that may have moved, changed, or deleted. I look at this timeline of deleted files and their contents in a nice list. See Appendix B. This list is formatted date, size, type, permission, owner, group, inode number, and name.

Recovering Deleted Files

From the timeline, I see two files deleted, named “CamShell.dll” and “index.htm”. I will use the `icat` command from the Sleuth Kit to recover any data that remains where the files used to be (Figure 10). I use the `-f` flag to specify a FAT system type as well as `-r` for recovering the data for the inodes that were in use by these files.



```
Terminal
[Helix (helix)]$ /KNOPPIX/usr/local/sleuthkit-1.70/bin/icat -r -f fat v1_5 5 > CamShell.dll.recovered
[Helix (helix)]$ /KNOPPIX/usr/local/sleuthkit-1.70/bin/icat -r -f fat v1_5 28 > index.html.recovered
[Helix (helix)]$ ls -l #recovered
-rw-r--r-- 1 helix helix 36864 Oct 31 21:33 CamShell.dll.recovered
-rw-r--r-- 1 helix helix 727 Oct 31 21:34 index.html.recovered
[Helix (helix)]$
```

Figure 10

When the CamShell.dll recovered file is examined with `khxedit`, I find that the first thousand bytes or so of the file is actually HTML. Using this `istat` utility from the Sleuth Kit shows why I see HTML in this file. At least one sector was used by both files and the HTML one was the most recent (Figure 11).

I try to recover the two deleted files from inodes 5, and 28 using the Sleuth Kit’s `icat` program. The `icat` program is designed to pull out data from an image by giving it an inode number. Unfortunately, the CamShell.dll inode 5 has a copy of the HTML file that is in 28 overwriting the first 1024 bytes (two clusters). I recovered these files with `icat`, but since the inode 5 file has been partially overwritten, I will revisit this only if I need more clues.

```
Terminal
[Helix (helix)]$ /KNOPPIX/usr/local/sleuthkit-1.70/bin/istat -f fat v1_5 5
Directory Entry: 5
Not Allocated
File Attributes: File, Archive
Size: 36864
Num of links: 0
Name: _AMSHLL.DLL

Directory Entry Times:
Written: Sat Feb 3 19:44:16 2001
Accessed: Mon Apr 26 00:00:00 2004
Created: Mon Apr 26 09:46:18 2004

Sectors:
33

Recovery:
33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48
49 50 51 52 53 54 55 56
57 58 59 60 61 62 63 64
65 66 67 68 69 70 71 72
73 74 75 76 77 78 79 80
81 82 83 84 85 86 87 88
89 90 91 92 93 94 95 96
97 98 99 100 101 102 103 104
[Helix (helix)]$ /KNOPPIX/usr/local/sleuthkit-1.70/bin/istat -f fat v1_5 28
Directory Entry: 28
Not Allocated
File Attributes: File, Archive
Size: 727
Num of links: 0
Name: _index.htm

Directory Entry Times:
Written: Fri Apr 23 10:53:56 2004
Accessed: Mon Apr 26 00:00:00 2004
Created: Mon Apr 26 09:47:36 2004

Sectors:
33

Recovery:
33 34
[Helix (helix)]$
```

Figure 11

Forensic Details

The timeline I generated from the metadata and filesystem show the index.html file copied to the floppy shortly after the CamShell.dll file. Normally, times of precisely midnight are typical symptoms of tampering. These things by themselves in a timeline are not indicative of any mischievousness, but there is a chance they direct us to more information. I will finish examining the timeline for more obvious clues before continuing in this direction.

By running the file command on the floppy image, it told us the filesystem was created with “mkdosfs”. This is a Gnu/Linux signature so I conclude the floppy was last formatted from a Linux distribution, and not a Microsoft Windows operating system.

The most interesting items from the timeline tell us the documents were last modified on April 23, and last opened (in any way) on April 26. The index.htm and CamShell.dll files were used in some way and then deleted on April 26. If assuming the date and time of any computer accessing this floppy were correct, then the floppy contained this CamShell.dll file since at least February 3, 2001. The disk label changed to “RJL” on April 25 and is significant--in that it identifies that Mr. Leszczynski, or at least somebody with the same initials, was in possession of the floppy at that time and actively using it.

Unfortunately, since this floppy was formatted with a FAT12 filesystem (standard for PC formatted floppies), no information that points to the user who owns the files. Any information I have with regards to ownership is the label of the volume and where I obtained the floppy from Mr. Leszczynski.

My keyword list has grown to include related terms and significant strings found in the analysis so far.

money	encrypt	secret	Leszczynski	cisco
pass	encode	camshell	Twisted Pear	Rift
fuel	stegano	camouflage	Robert	plan
cell	stegona	RJL	hide	schematic

Researching the Contents

Now that I have taken a glimpse of the contents, I will search a bit for additional information on-line. I use my favorite Internet search engine to search for “camshell.dll”. Google’s (<http://www.google.com/>) only match is a site that describes camouflaging .mp3 files: <http://www.tranceaddict.com/forums/archive/topic/79627-1.html>. Although there is not a link to where to find this Camouflage software, it does look like I am on the right track.

I return to Google and searched for the two words “camouflage” and “steganography” first (not realizing I misspelled steganography, which I will explain later). However, the search returned exactly two matches, which both were useful.⁴

I begin with link one, and follow it to www.camouflagesoftware.com, but this site at this current time is only advertisements. So I try looking in Google’s cache for any page on the site by using “site:www.camouflagesoftware.com” in the search box. This limits the search to pages found on that web site, and since I did not enter in any terms, it should return all pages that Google has indexed from the site. Google reports with a few links,

⁴ <http://www.sans.org/rr/papers/20/762.pdf> and <http://gaby.theridion.com/NotGaby.html>

but clicking on the cached copy does not provide any content. Time to try another cache.

I search for the site with the Wayback Machine at www.archive.org. This is a search engine with a partial archive of many sites on the Internet. I start from the current date moving backwards to find the last snapshot of the actual site. The one that starts to show us interesting information is November, 2002:

<http://web.archive.org/web/20021127203553/http://www.camouflagesoftware.com>

Now I have a page that claims Camouflage is a software package that can hide files in other files. On the download page there is a broken link to download the file from a different site at <http://www.camouflage.freemove.co.uk/Camou121.zip>. I go back to Google and search for the filename, and find a valid file at <ftp://ftp.sac.sk/pub/sac/comm/camou121.zip>. I continue to use Google to find more information, using “camouflage” and “download” as keywords, and I find what looks like an active mirror of the site⁵ which has a Camou121.exe file.

I have more software to try, and my search is confirmed, but not turning up any new information. I decide to try the software. Before leaving my Helix workstation, I use smbclient to copy the files from the image to my Windows XP test machine. This is the test machine running inside of VMware. As a final precaution, I change the VMware network settings to host-only mode, so that the system cannot act outside of its environment.

Once I shutdown Helix and switch to this Windows XP test environment created only for development and testing, I revert to the snapshot to ensure a clean installation. When my XP machine is finished booting, I adjust VMware so that the floppy drive points to the v1_5 image provided, so that Windows will treat it as a live floppy. I run McAfee Virus Scan on the floppy to ensure there is not any information that will confuse this analysis.

I decide to try opening up the five files with Microsoft Word XP. All files open normally, even the two that abiword prompted for a password. No abnormal characters or anything suspicious with this editor are apparent. The files all appear to have standard formatted text, not even a macro.

Software Analysis

I download the Camou121.exe file to the XP desktop and extract the file. See the Examination and Analysis section for a description of my test environment. A clean test environment is essential for a dynamic analysis.⁶ Since I am in a clean environment, I am not worried about potential damage to my installation, but I disable the network connectivity as a precaution.

⁵ <http://camouflage.unfiction.com/>

⁶ Skoudis, p. 576

I extract the Camou121.exe file to the C:\installTEMP\ folder and the self-extracting file automatically runs the setup.exe program. I accept all of the defaults in the installation dialog, and it installs Camouflage 1.22 successfully. A new program group was added to the start menu, and the only non-documentation shortcut in this new program folder is for Camouflage settings (Figure 12). Note the software contains a version discrepancy on some windows in the interface. I will determine later if this is the version that was used to hide any information on the floppy.

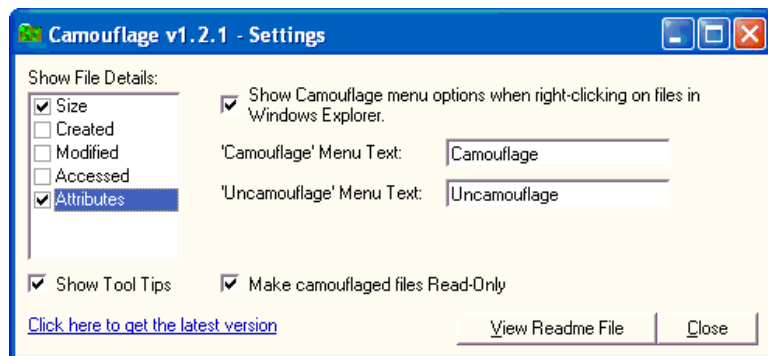


Figure 12

I notice the menu options by default come up when right clicking on a file in Windows Explorer, so I browse to the copy in the v1_5 folder on the desktop. I try a blank password on all the files, and Internal_Lab_Security.doc file was camouflaged with a blank password (Figure 13).

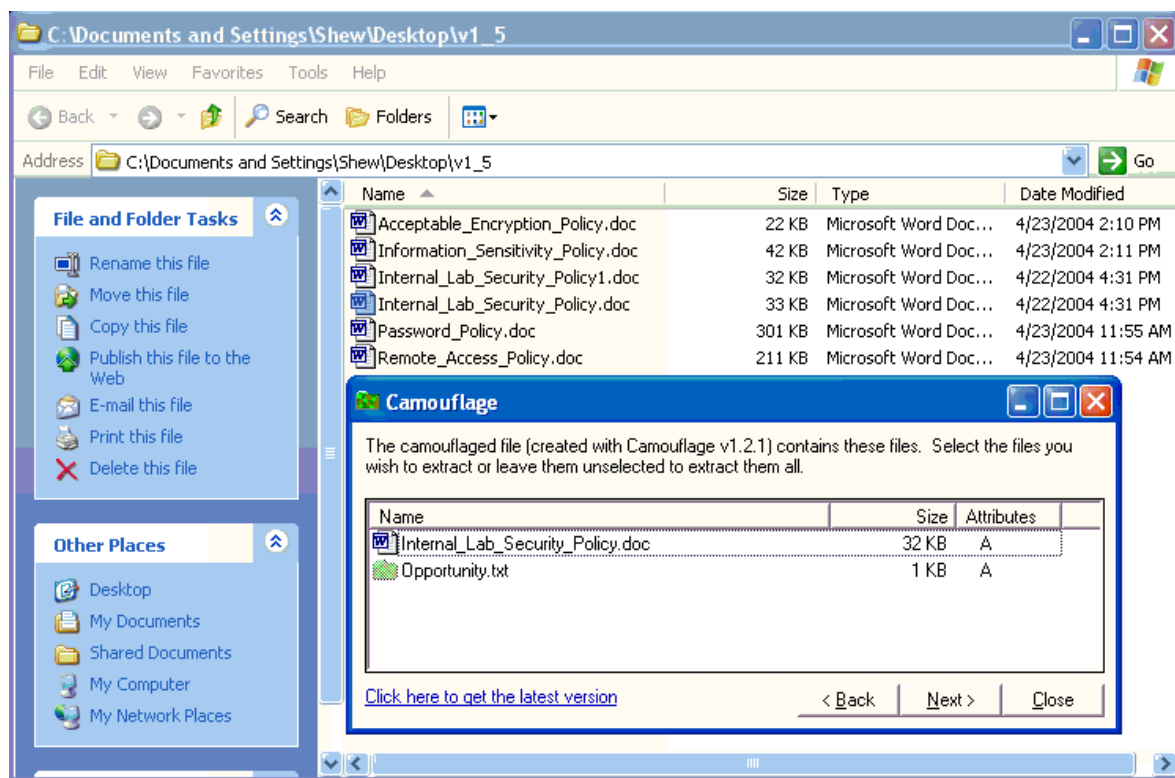


Figure 13

I extract both files to the desktop in a new “uncamouflaged” folder, and immediately run an md5sum on the file, and save the results. This text file extracts to a 1 KB file. I open notepad from my start menu and use File-Open to open this new file. Opening files in this method ensures that no file extension or MIME type trickery will interrupt our investigation. The contents of this text file:

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

I know a database and schematics exist, likely camouflaged among the other files. The reference to “First Name” is odd in this context. This concludes that the CamShell.dll clue I found on the floppy image was indeed used by Mr. Leszczynski to hide this message inside the host Word document.

The two larger Word documents reveal that nothing is camouflaged with a blank password. Acting on the clue from the recovered text file I start trying a few common

passwords in varying cases and literally Robert, since it the only first name in the text document. I also try both bad and good example passwords in the Password Policy document. So far, all of these passwords do not hide any additional files.

Taking a step back, I try similar passwords, starting with the keyword list, trying lower case, upper case, and first letter capitalized. By entering "Password" on the "Password_Policy.doc" file, I see new files (Figure 14). Apparently, the "First Name" meant the first name of the file, not the person's first name. I extract all files to the uncamouflaged folder and run md5sum on each file.

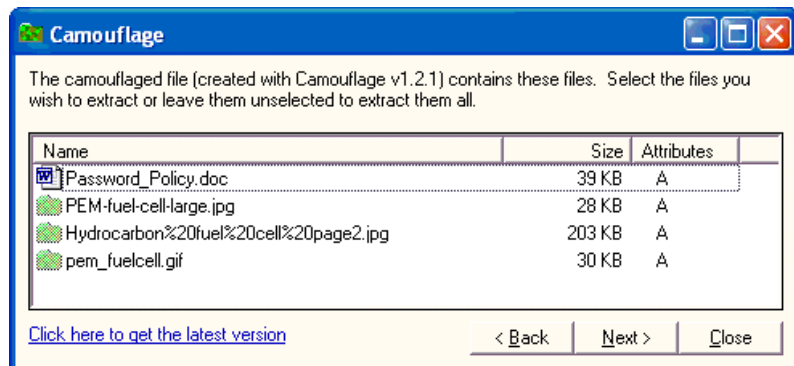


Figure 14

Once the "First Name" clue was discovered, I try all the other original files on the floppy image, but only the Remote_Access_Policy.doc file revealed any new camouflaged files (Figure 15). Again, I extract both files to the uncamouflaged folder and create md5sums.

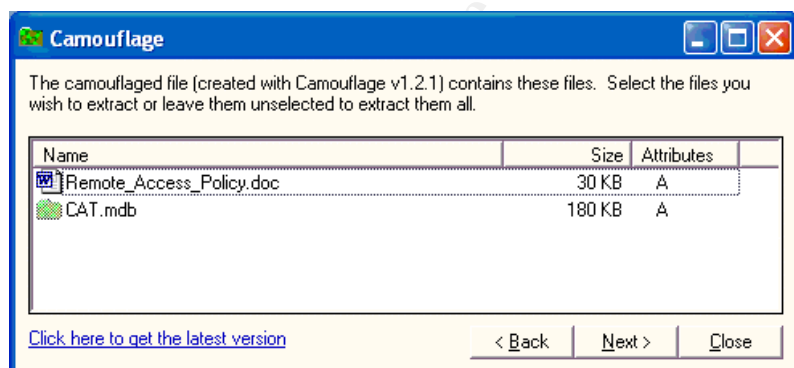


Figure 15

I carefully open these files to confirm their file types. The CAT.mdb file is a Microsoft Access Database that contains client passwords (Figure 16).

Microsoft Access - [Clients : Table]

Type a question for help

	Last	Phone	Company	Address	Address1	City	State	Zipcode	Account	Passwo
▶	Esposito	703-233-2048	Cook Labs	245 Main St		Alexandria	VA	20231	espomain	y4NSHM
	Jackson	410-677-7223	Double J's	11561 W. 27 St.		Baltimore	MD	20278	jack27st	JLbW3Pc
	Lee	866-554-0922	Tech Vision	300 Lone Grove Lane		Wichita	KS	30189	leetechv	O1A26a3
	Horton	800-234-king	King Labs, Inc.	700 King Labs Ave	Suite 900	Biloxi	MS	39533	hortking	Yk7Sr4p
	Jones	877-Get-done	Quick Printing	99 E. Grand View Dr		Omaha	NE	56098	joneeast	868y48RI
	Hayes	404-893-5521	Big Sky First	90 Old Saw Mill Rd		Billings	MT	59332	hayeolds	3R30bb7i
	Forrester	210-586-2312	TCFL	188 Greenville Rd		Austin	TX	77239	forrgree	si4OW8U
	Cash	212-562-0997	E & C Inc.	76 S. King St	Suite 300	Santa Barbara	CA	80124	cashking	Of8uQ1fC
	Bei	616-833-0129	Island Labs	65 Kiwi Way		Honolulu	HA	93991	beikiwiw	JDH20u2l
	Kelly		Data Movers	7256 Beerwah Ave.	Suite 110	Wetherby	U.K.	LS22 6RG	kellbeer	tmu0ENC
	Roy		The Magic Lam	4150 Regents Park	Row #170	Calgary	CAN	R4316DF	roythema	rJag6Q0C
*										

Record: 1 of 11

Datasheet View

NUM

start Camouflage H... 3 Windows E... Untitled - Not... 2 Microsoft ... 4:24 PM

Figure 16

I realized at this moment I misspelled steganography in my search earlier. For thoroughness, I Google for the keys words “camouflage” and “steganography”. The second link is an article about detecting camouflaged files and even has a link to a password tool. Unfortunately, I did not find the <http://www.guillermi2.net/stegano/camouflage/> link first; it would have saved a little time. This page describes important internals of the Camouflage software, and even a little utility to recover passwords: http://www.guillermi2.net/stegano/camouflage/Camouflage_Password_Finder_02.zip.

What new information does this give us? I learn two important things from this page: This page describes quite a bit of details on how Camouflage works, saving us the trouble of a binary analysis. The password utility confirms the three passwords I recovered (“”, “Password”, and “Remote”) but yields random characters for the rest of the files. This lets us know that no more information was available in those other files, the originals and the files that I pulled out. Trying the password finder on the uncamouflaged files, I am unable to recover any new files, therefore no files were double camouflaged.

Were the deleted files camouflaged? If they were, it would appear as extra encrypted data at the end of each file. I see the index.html.recovered file does not, and I have identified the differences in the CamShell.dll file to be an ASCII new line translation problem. If any extra information in the CamShell.dll file was recovered from the floppy, I would have seen it in the file difference. If a blank password exists, then nothing would show up in the recovery dialog.

The utility is simple, it has three possible outcomes: the password, a blank representing a blank password, and non-ASCII characters if it is not a camouflaged file. Examples of each to are shown in Figures 17, 18, and 19.



Figure 17

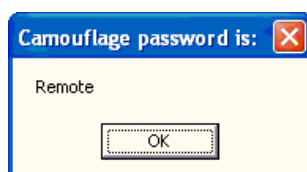


Figure 18

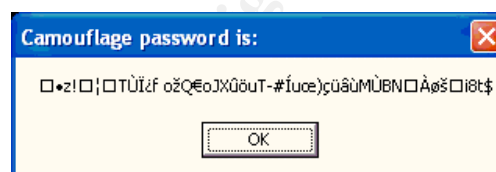


Figure 19

The only exception I found to this utility is that it cannot examine the CamShell.dll that the system uses. It will not provide any output in that instance. A copy of the file does show the non-ASCII characters. The utility monitored in the VMware test environment did not have any side effects of network, file, or registry activity.

Guillermi also has a link to <http://www.vikt0ry.com/> with another password utility. It is available under the Progs link and is designed to clear the password of a camouflaged file. This utility does not provide any new information to our investigation, but it can confirm the existence of steganography in a file.

Software Footprint

The Camouflage installer I downloaded made typical software installation changes to the registry. A log of these changes was created with the regshot tool.⁷ It extracts into a temporary directory before installing, then runs a typical installation dialog wizard. I monitor the installation with my Sysinternal tools, but see no abnormal traffic, registry, or file activity.

The temporary install directory contains 19 files. The default installation settings place four files in C:\Program Files\Camouflage\. Appendix A lists md5sums. One of the installed files is C:\Program Files\Camouflage\CamShell.dll, which appears identical to the deleted one from the floppy.

⁷ Regshot is a free tool to capture changes in the registry for Windows operating systems and is available at <http://regshot.yeah.net>

To confirm this file was precisely the one used to camouflage the content on the floppy, I use winhex, a hex editor to copy the bytes from 0x0000 to 0x0736 from the installed CamShell.dll and then paste them over the same amount of bytes of the CamShell.dll file recovered from the floppy. I repaired the obvious difference and can compare the remaining differences with md5sum.

If the files are identical to every byte, the md5sums should be the same. After running md5sum on the files, I find they do not match. At this point I open up ExamDiff Pro. This software specializes in showing both hexadecimal and ASCII representations of two files and compares any differences it finds.⁸

The obvious difference was the ASCII HTML code that replaced the beginning header information, and the second is what appear to be 0x0D and 0x0A changes. These represent carriage return and linefeed, respectively. This tells me that the file was copied between Windows and an operating system like UNIX that does not represent a new line with both characters. Other than these characters inserted, the files are now identical. This confirms the Camouflage software I installed and used to extract the hidden files, shows the CamShell.dll file once on the floppy.

Detection

If the Camouflage software was installed with default settings, it can be detected by searching for the CamShell.dll file. The utility allows the right-click menu text to be edited, so there may not be the words “Camouflage” and “Uncamouflage” when you right-click on a file in Explorer.

The easiest way to detect a Camouflaged file is to run the password recovery utility on the file. Likely candidates are files that appear larger than they should be, and non-text files. Plain text files are unlikely to be useful for camouflage since the end of the file will have encrypted data and is easy to notice.

Appendix C has a list of registry changes that can be examined to determine if the software was installed. The md5sums of the files created during installation are included in Appendix A.

Internal Policy Issues

According to Guillermito⁹, Camouflage uses a simple XOR method to hide the content of the files at the end of the host file. Since this is a trivial reversal to recover the original data, it is in violation of Ballard Industry’s Acceptable Encryption Policy, section 3.0.

⁸ ExamDiff Pro has a trial version available at http://www.prestosoft.com/ps.asp?page=edp_examdiffpro

⁹ Guillermito analyzed the encryption routine, and his results are available at <http://www.guillermito2.net/stegano/camouflage/>

The password choices of the three camouflaged documents are in violation of the Password Policy 4.2 guidelines. Also, in accordance with the Bright Industries Password Policy, the client accounts from the CAT.mdb database need to change passwords immediately.

The fact that the images and schematics have not been labeled in according to the Information Sensitivity Policy would also be a problem.

Legal Issues

Although the entire program was not on the floppy image as I downloaded it, I have confirmed it was used to hide proprietary data with the recovered CamShell.dll file.

Mr. Leszczynski has violated TITLE 18 , PART I , CHAPTER 90 , Sec. 1832 (a) (2) and (4) ¹⁰ and may be fined up to \$5,000,000 by attempting to communicate trade secrets.

Mr. Leszczynski has violated TITLE 18 , PART I , CHAPTER 47 , Sec. 1030 (a) (2) and (4) ¹¹ and may be fined and imprisoned up to 5 years by attempting to communicate trade secrets (economic espionage) for a purpose of financial gain.

Additionally, if Rift, Inc. or Mr. Leszczynski's contact qualifies as a foreign entity, then he has also violated TITLE 18 , PART I , CHAPTER 90 , Sec. 1831 (a) (2) and (4) ¹² and may be fined up to \$5,000,000 by attempting to communicate trade secrets with a foreign entity.

Mr. Leszczynski also released customer information in the CAT.mdb database. This username and password may provide access to more proprietary information. If there were Social Security or account numbers released with this database, then in the state of California SB 1386¹³ dictates that an announcement would have to be made about the breach in security.

Summary

This analysis shows Mr. Leszczynski attempted to release proprietary information from the company. Mr. Leszczynski also communicated his expectation of personal profit for this information. He purposely tried to hide this proprietary information to smuggle it to an outside source for financial gain.

¹⁰ <http://www4.law.cornell.edu/uscode/18/1832.html>

¹¹ <http://www4.law.cornell.edu/uscode/18/1030.html>

¹² <http://www4.law.cornell.edu/uscode/18/1831.html>

¹³ http://info.sen.ca.gov/pub/01-02/bill/sen/sb_1351-1400/sb_1386_bill_20020926_chaptered.html

REFERENCES

1. Bartlett, John. "The Ease of Steganography and Camouflage". March 12, 2002. URL: <http://www.sans.org/rr/papers/20/762.pdf>
2. Unknown. "Camouflage Home Page - Hide your files!". URL: <http://camouflage.unfiction.com/>
3. Skoudis, Ed. "Malware Fighting Malicious Code". 2004
4. Guillermito. "Breaking a Steganography Software". May 6, 2003. URL: <http://www.guillermito2.net/stegano/camouflage/>
5. "US CODE: Title 18,1832. Theft of trade secrets". URL: <http://www4.law.cornell.edu/uscode/18/1832.html>
6. "US CODE: Title 18,1030. Fraud and related activity in connection with computers". URL: <http://www4.law.cornell.edu/uscode/18/1030.html>
7. "US CODE: Title 18,1831. Economic espionage". URL: <http://www4.law.cornell.edu/uscode/18/1831.html>
8. "SB 1386 Senate Bill - CHAPTERED". URL: http://info.sen.ca.gov/pub/01-02/bill/sen/sb_1351-1400/sb_1386_bill_20020926_chaptered.html

© SANS Institute 2004, Author retains full rights.

Appendix A – MD5SUMS

Floppy Image:

d41d8cd98f00b204e9800998ecf8427e RJL (Volume Label Entry)
99c5dec518b142bd945e8d7d2fad2004 Information_Sensitivity_Policy.doc
(INFORM~1.DOC)
e0c43ef38884662f5f27d93098e1c607 Internal_Lab_Security_Policy1.doc
(INTERN~1.DOC)
b9387272b11aea86b60a487fbdclb336 Internal_Lab_Security_Policy.doc
(INTERN~2.DOC)
ac34c6177ebdcaf4adc41f0e181belbc Password_Policy.doc (PASSWO~1.DOC)
5b38d1ac1f94285db2d2246d28fd07e8 Remote_Access_Policy.doc (REMOTE~1.DOC)
f785ba1d99888e68f45dabeddb0b4541 Acceptable_Encryption_Policy.doc
(ACCEPT~1.DOC)
17282ea308940c530a86d07215473c79 _ndex.htm
6462fb3acca0301e52fc4ffa4ea5eff8 _amshell.dll

Uncamouflaged Files:

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

Temporary Install Direcotry:

6229a86a1d291c311da49a7d69a49a1f c:\\installTEMP_inst32i.ex_
51161bf79f25ff278912005078ad93d5 c:\\installTEMP_ISDel.exe
ecacc9ab09d7e8898799fe5c4ebbbdd2 c:\\installTEMP_Setup.dll
70422405572f674437236e6a055eff2d c:\\installTEMP_sys1.cab
917d579c98a04cd2d5e16b904aa0cbac c:\\installTEMP_sys1.hdr
b0be02f24f7509d04f039d54c4e28a18 c:\\installTEMP_user1.cab
0f190c4847d1d0980a4f8e8ca80deb7f c:\\installTEMP_user1.hdr
013f337c07031f189e270a336206b1d0 c:\\installTEMP\\DATA.TAG
97db73ed99a5892c490a89daaa588ec9 c:\\installTEMP\\data1.cab
615ac9433d57662c6fa2709e1e08c88c c:\\installTEMP\\data1.hdr
70627bd56fe92a5c97027cbbd88bacd0 c:\\installTEMP\\lang.dat
87032452941f0e75b4918a77a4ccdcef c:\\installTEMP\\layout.bin
478f65a0b922b6ba0a6ce99e1d15c336 c:\\installTEMP\\os.dat
b075733804ffedf19eb4b7180bbd2ed3 c:\\installTEMP\\Readme.txt
773efac7f1023b9a4879e36dbe72149a c:\\installTEMP\\Setup.bmp
71e6dd8a9de4a9baf89fca951768059a c:\\installTEMP\\Setup.exe
7ce7bc3c389619fe2ab23a0d11c73953 c:\\installTEMP\\SETUP.INI
eebb82eddf391b7f43b61cdf10da47be c:\\installTEMP\\setup.ins
1b79748e93a541cc1590505b6c72828a c:\\installTEMP\\setup.lid

Installed Software Files:

9f08258a80d578a0f1cc38fe4c2aebb5 c:\\Progra~1\\Camouflage\\Camouflage.exe
4e986ab0909d2946bed868b5f896906f c:\\Progra~1\\Camouflage\\CamShell.dll
0c25ad7792d555b6c8c37c77ceb9e224 c:\\Progra~1\\Camouflage\\Readme.txt
7ccb90076fde0d3d422f444b3b9fad12 c:\\Progra~1\\Camouflage\\Uninst.isu

Appendix B – MAC Timeline

```

Sat Feb 03 2001 19:44:16    36864 m.. -/-rwxrwxrwx 0      0      5
                          A:/CamShell.dll (_AMSHLL.DLL) (deleted)
                          36864 m.. -rwxrwxrwx 0      0      5
                          <v1_5-_AMSHLL.DLL-dead-5>
Thu Apr 22 2004 16:31:06    32256 m.. -/-rwxrwxrwx 0      0     13
                          A:/Internal_Lab_Security_Policy1.doc (INTERN~1.DOC)
                          33423 m.. -/-rwxrwxrwx 0      0     17
                          A:/Internal_Lab_Security_Policy.doc (INTERN~2.DOC)
Fri Apr 23 2004 10:53:56     727 m.. -/-rwxrwxrwx 0      0     28
                          A:/_ndex.htm (deleted)
                          727 m.. -rwxrwxrwx 0      0     28
                          <v1_5-_ndex.htm-dead-28>
Fri Apr 23 2004 11:54:32   215895 m.. -/-rwxrwxrwx 0      0     23
                          A:/Remote_Access_Policy.doc (REMOTE~1.DOC)
Fri Apr 23 2004 11:55:26   307935 m.. -/-rwxrwxrwx 0      0     20
                          A:/Password_Policy.doc (PASSWO~1.DOC)
Fri Apr 23 2004 14:10:50    22528 m.. -/-rwxrwxrwx 0      0     27
                          A:/Acceptable_Encryption_Policy.doc (ACCEPT~1.DOC)
Fri Apr 23 2004 14:11:10    42496 m.. -/-rwxrwxrwx 0      0      9
                          A:/Information_Sensitivity_Policy.doc (INFORM~1.DOC)
Sun Apr 25 2004 00:00:00      0 .a. -/-rwxrwxrwx 0      0      3
                          A:/RJL
                          (Volume Label Entry)
Sun Apr 25 2004 10:53:40      0 m.c -/-rwxrwxrwx 0      0      3
                          A:/RJL
                          (Volume Label Entry)
Mon Apr 26 2004 00:00:00    36864 .a. -rwxrwxrwx 0      0      5
                          <v1_5-_AMSHLL.DLL-dead-5>
                          22528 .a. -/-rwxrwxrwx 0      0     27
                          A:/Acceptable_Encryption_Policy.doc (ACCEPT~1.DOC)
                          307935 .a. -/-rwxrwxrwx 0      0     20
                          A:/Password_Policy.doc (PASSWO~1.DOC)
                          727 .a. -/-rwxrwxrwx 0      0     28
                          A:/_ndex.htm (deleted)
                          727 .a. -rwxrwxrwx 0      0     28
                          <v1_5-_ndex.htm-dead-28>
                          32256 .a. -/-rwxrwxrwx 0      0     13
                          A:/Internal_Lab_Security_Policy1.doc (INTERN~1.DOC)
                          36864 .a. -/-rwxrwxrwx 0      0      5
                          A:/CamShell.dll (_AMSHLL.DLL) (deleted)
                          42496 .a. -/-rwxrwxrwx 0      0      9
                          A:/Information_Sensitivity_Policy.doc (INFORM~1.DOC)
                          33423 .a. -/-rwxrwxrwx 0      0     17
                          A:/Internal_Lab_Security_Policy.doc (INTERN~2.DOC)
                          215895 .a. -/-rwxrwxrwx 0      0     23
                          A:/Remote_Access_Policy.doc (REMOTE~1.DOC)
Mon Apr 26 2004 09:46:18    36864 ..c -/-rwxrwxrwx 0      0      5
                          A:/CamShell.dll (_AMSHLL.DLL) (deleted)
                          36864 ..c -rwxrwxrwx 0      0      5
                          <v1_5-_AMSHLL.DLL-dead-5>
Mon Apr 26 2004 09:46:20    42496 ..c -/-rwxrwxrwx 0      0      9
                          A:/Information_Sensitivity_Policy.doc (INFORM~1.DOC)
Mon Apr 26 2004 09:46:22    32256 ..c -/-rwxrwxrwx 0      0     13
                          A:/Internal_Lab_Security_Policy1.doc (INTERN~1.DOC)
Mon Apr 26 2004 09:46:24    33423 ..c -/-rwxrwxrwx 0      0     17
                          A:/Internal_Lab_Security_Policy.doc (INTERN~2.DOC)

```

```

Mon Apr 26 2004 09:46:26    307935 ..c -/-rwxrwxrwx 0          0          20
                        A:/Password_Policy.doc (PASSWO~1.DOC)
Mon Apr 26 2004 09:46:36    215895 ..c -/-rwxrwxrwx 0          0          23
                        A:/Remote_Access_Policy.doc (REMOTE~1.DOC)
Mon Apr 26 2004 09:46:44    22528  ..c -/-rwxrwxrwx 0          0          27
                        A:/Acceptable_Encryption_Policy.doc (ACCEPT~1.DOC)
Mon Apr 26 2004 09:47:36      727  ..c -rwxrwxrwx 0          0          28
                        <v1_5-_ndex.htm-dead-28>
                        727  ..c -/-rwxrwxrwx 0          0          28
                        A:/_ndex.htm (deleted)

```

© SANS Institute 2004, Author retains full rights.

Appendix C – Camouflage Installation Regshot Output

REGSHOT LOG 1.61e5

Comments:

Datetime:2004/9/2 18:22:46 , 2004/9/2 18:23:56

Computer:TESTXP , TESTXP

Username:User , User

Keys deleted:1

HKEY_USERS\S-1-5-21-1482476501-413027322-839522115-
1003\Software\Microsoft\Windows\CurrentVersion\Explorer\Discardable\PostSetup
\ShellNew

Keys added:28

HKEY_LOCAL_MACHINE\SOFTWARE\Classes*\shellex\ContextMenuHandlers\Camouflage
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{29557489-990B-11D4-9413-
004095490AD4}
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{29557489-990B-11D4-9413-
004095490AD4}\Implemented Categories
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{29557489-990B-11D4-9413-
004095490AD4}\Implemented Categories\{40FC6ED5-2438-11CF-A3DB-080036F12502}
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{29557489-990B-11D4-9413-
004095490AD4}\InprocServer32
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{29557489-990B-11D4-9413-
004095490AD4}\ProgID
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{29557489-990B-11D4-9413-
004095490AD4}\Programmable
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{29557489-990B-11D4-9413-
004095490AD4}\TypeLib
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{29557489-990B-11D4-9413-
004095490AD4}\VERSION
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\Interface\{29557488-990B-11D4-9413-
004095490AD4}
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\Interface\{29557488-990B-11D4-9413-
004095490AD4}\ProxyStubClsid
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\Interface\{29557488-990B-11D4-9413-
004095490AD4}\ProxyStubClsid32
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\Interface\{29557488-990B-11D4-9413-
004095490AD4}\TypeLib
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\TypeLib\{35FE0039-0582-11D4-A337-
00805F49B06B}
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\TypeLib\{35FE0039-0582-11D4-A337-
00805F49B06B}\3.0
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\TypeLib\{35FE0039-0582-11D4-A337-
00805F49B06B}\3.0\0
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\TypeLib\{35FE0039-0582-11D4-A337-
00805F49B06B}\3.0\0\win32
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\TypeLib\{35FE0039-0582-11D4-A337-
00805F49B06B}\3.0\FLAGS
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\TypeLib\{35FE0039-0582-11D4-A337-
00805F49B06B}\3.0\HELPDIR
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CamouflageShell.ShellExt

```
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CamouflageShell.ShellExt\Clsid
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\App
Paths\Camouflage.exe
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\Camouf
lage
HKEY_LOCAL_MACHINE\SOFTWARE\Twisted Pear Productions
HKEY_LOCAL_MACHINE\SOFTWARE\Twisted Pear Productions\Camouflage
HKEY_LOCAL_MACHINE\SOFTWARE\Twisted Pear Productions\Camouflage\1.2.1
HKEY_USERS\S-1-5-21-1482476501-413027322-839522115-1003\Software\Camouflage
HKEY_USERS\S-1-5-21-1482476501-413027322-839522115-
1003\Software\Camouflage\Settings
```

Values deleted:28

```
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{C27CCE32-8596-11D1-B16A-
00C0F0283628}\InprocServer32\InprocServer32: '(f'^Vn-
}f(YR|eAR6.jiProductNonBootFiles>dbKx-lbmf(Gn,L[[Q~CN'
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{C27CCE33-8596-11D1-B16A-
00C0F0283628}\InprocServer32\InprocServer32: '(f'^Vn-
}f(YR|eAR6.jiProductNonBootFiles>dbKx-lbmf(Gn,L[[Q~CN'
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{C27CCE34-8596-11D1-B16A-
00C0F0283628}\InprocServer32\InprocServer32: '(f'^Vn-
}f(YR|eAR6.jiProductNonBootFiles>dbKx-lbmf(Gn,L[[Q~CN'
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{C27CCE35-8596-11D1-B16A-
00C0F0283628}\InprocServer32\InprocServer32: '(f'^Vn-
}f(YR|eAR6.jiProductNonBootFiles>dbKx-lbmf(Gn,L[[Q~CN'
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{C27CCE36-8596-11D1-B16A-
00C0F0283628}\InprocServer32\InprocServer32: '(f'^Vn-
}f(YR|eAR6.jiProductNonBootFiles>dbKx-lbmf(Gn,L[[Q~CN'
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{C27CCE37-8596-11D1-B16A-
00C0F0283628}\InprocServer32\InprocServer32: '(f'^Vn-
}f(YR|eAR6.jiProductNonBootFiles>dbKx-lbmf(Gn,L[[Q~CN'
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{C27CCE38-8596-11D1-B16A-
00C0F0283628}\InprocServer32\InprocServer32: '(f'^Vn-
}f(YR|eAR6.jiProductNonBootFiles>dbKx-lbmf(Gn,L[[Q~CN'
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{C27CCE39-8596-11D1-B16A-
00C0F0283628}\InprocServer32\InprocServer32: '(f'^Vn-
}f(YR|eAR6.jiProductNonBootFiles>dbKx-lbmf(Gn,L[[Q~CN'
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{C27CCE3A-8596-11D1-B16A-
00C0F0283628}\InprocServer32\InprocServer32: '(f'^Vn-
}f(YR|eAR6.jiProductNonBootFiles>dbKx-lbmf(Gn,L[[Q~CN'
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{C27CCE3B-8596-11D1-B16A-
00C0F0283628}\InprocServer32\InprocServer32: '(f'^Vn-
}f(YR|eAR6.jiProductNonBootFiles>dbKx-lbmf(Gn,L[[Q~CN'
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{C27CCE3C-8596-11D1-B16A-
00C0F0283628}\InprocServer32\InprocServer32: '(f'^Vn-
}f(YR|eAR6.jiProductNonBootFiles>dbKx-lbmf(Gn,L[[Q~CN'
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{C27CCE3D-8596-11D1-B16A-
00C0F0283628}\InprocServer32\InprocServer32: '(f'^Vn-
}f(YR|eAR6.jiProductNonBootFiles>dbKx-lbmf(Gn,L[[Q~CN'
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{C27CCE3E-8596-11D1-B16A-
00C0F0283628}\InprocServer32\InprocServer32: '(f'^Vn-
}f(YR|eAR6.jiProductNonBootFiles>dbKx-lbmf(Gn,L[[Q~CN'
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{C27CCE3F-8596-11D1-B16A-
00C0F0283628}\InprocServer32\InprocServer32: '(f'^Vn-
}f(YR|eAR6.jiProductNonBootFiles>dbKx-lbmf(Gn,L[[Q~CN'
```


[illegible]

[illegible]

HKEY_USERS\S-1-5-21-1482476501-413027322-839522115-

00 00 4D 00 69 00 63 00 72 00 6F 00 73 00 6F 00 66 00 74 00 20 00 41 00 63 00

HKEY_USERS\S-1-5-21-1482476501-413027322-839522115-

00 00 00 4D 00 69 00 63 00 72 00 6F 00 73 00 6F 00 66 00 74 00 20 00 50 00 6F

[illegible]

[illegible][illegible][illegible]

[illegible]


```

00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00
322-839522115-
ntVersion\Explorer\Discard
00 01 00 D4 07 09 00 04 00
322-839522115-
ntVersion\Explorer\Discard

```

```

00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00
322-839522115-
ntVersion\Explorer\Discard
00 01 00 D4 07 09 00 04 00
322-839522115-
ntVersion\Explorer\Discard

```

```

HKEY_LOCAL_MACHINE\SOFTWARE\Classes\TypeLib\{35FE0039-0582-11D4-A337-00805F49B06B}\3.0\:"CamouflageShell"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CamouflageShell.ShellExt\Clsid\:
"{29557489-990B-11D4-9413-004095490AD4}"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CamouflageShell.ShellExt\:
"CamouflageShell.ShellExt"
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\App
Paths\Camouflage.exe\Path: "C:\Program Files\Camouflage"
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\App
Paths\Camouflage.exe\: "C:\Program Files\Camouflage\Camouflage.exe"
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\Camouf
lage\UninstallString: "C:\WINDOWS\IsUninst.exe -f"C:\Program
Files\Camouflage\Uninst.isu""
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\Camouf
lage\DisplayName: "Camouflage"
HKEY_USERS\S-1-5-21-1482476501-413027322-839522115-
1003\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{75048700-
EF1F-11D0-9888-006097DEACF9}\Count\HRZR_EHACNGU:P:\Qbphzragf naq
Frggvatf\Hfre\Qrfgxgbc\Pnzbh121.rkr: 01 00 00 00 06 00 00 00 50 F0 C1 E1 19 91
C4 01
HKEY_USERS\S-1-5-21-1482476501-413027322-839522115-
1003\Software\Camouflage\Settings\Menu: 0xFFFFFFFF

```

Values modified:42

```

HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{1EFB6596-857C-11D1-B16A-00C0F0283628}\: "Microsoft TabStrip Control, version 6.0"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{1EFB6596-857C-11D1-B16A-00C0F0283628}\: "Microsoft TabStrip Control 6.0 (SP4)"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{2C247F23-8591-11D1-B16A-00C0F0283628}\: "Microsoft ImageList Control, version 6.0"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{2C247F23-8591-11D1-B16A-00C0F0283628}\: "Microsoft ImageList Control 6.0 (SP4)"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{35053A22-8589-11D1-B16A-00C0F0283628}\: "Microsoft ProgressBar Control, version 6.0"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{35053A22-8589-11D1-B16A-00C0F0283628}\: "Microsoft ProgressBar Control 6.0 (SP4)"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{66833FE6-8583-11D1-B16A-00C0F0283628}\: "Microsoft Toolbar Control, version 6.0"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{66833FE6-8583-11D1-B16A-00C0F0283628}\: "Microsoft Toolbar Control 6.0 (SP4)"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{8E3867A3-8586-11D1-B16A-00C0F0283628}\: "Microsoft StatusBar Control, version 6.0"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{8E3867A3-8586-11D1-B16A-00C0F0283628}\: "Microsoft StatusBar Control 6.0 (SP4)"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{BDD1F04B-858B-11D1-B16A-00C0F0283628}\: "Microsoft ListView Control, version 6.0"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{BDD1F04B-858B-11D1-B16A-00C0F0283628}\: "Microsoft ListView Control 6.0 (SP4)"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{C74190B6-8589-11D1-B16A-00C0F0283628}\: "Microsoft TreeView Control, version 6.0"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{C74190B6-8589-11D1-B16A-00C0F0283628}\: "Microsoft TreeView Control 6.0 (SP4)"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{D5DE8D20-5BB8-11D1-A1E3-00A0C90F2731}\InProcServer32\: "C:\WINDOWS\System32\msvbvm60.dll"

```

```

HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{D5DE8D20-5BB8-11D1-A1E3-00A0C90F2731}\InProcServer32\ : "C:\WINDOWS\System32\MSVBVM60.DLL"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{DD9DA666-8594-11D1-B16A-00C0F0283628}\ : "Microsoft ImageComboBox Control, version 6.0"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{DD9DA666-8594-11D1-B16A-00C0F0283628}\ : "Microsoft ImageComboBox Control 6.0 (SP4)"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{F08DF954-8592-11D1-B16A-00C0F0283628}\ : "Microsoft Slider Control, version 6.0"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\CLSID\{F08DF954-8592-11D1-B16A-00C0F0283628}\ : "Microsoft Slider Control 6.0 (SP4)"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\Interface\{A4C466B8-499F-101B-BB78-00AA00383CBB}\TypeLib\Version: "2.1"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\Interface\{A4C466B8-499F-101B-BB78-00AA00383CBB}\TypeLib\Version: "6.0"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\Interface\{A4C46780-499F-101B-BB78-00AA00383CBB}\TypeLib\Version: "2.1"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\Interface\{A4C46780-499F-101B-BB78-00AA00383CBB}\TypeLib\Version: "6.0"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\Interface\{BEF6E003-A874-101A-8BBA-00AA00300CAB}\TypeLib\ : "{AC2DE821-36A2-11CF-8053-00AA006009FA}"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\Interface\{BEF6E003-A874-101A-8BBA-00AA00300CAB}\TypeLib\ : "{00020430-0000-0000-C000-000000000046}"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\MSComctlLib.ImageComboCtl\ : "Microsoft ImageComboBox Control, version 6.0"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\MSComctlLib.ImageComboCtl\ : "Microsoft ImageComboBox Control 6.0 (SP4)"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\MSComctlLib.ImageComboCtl.2\ : "Microsoft ImageComboBox Control, version 6.0"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\MSComctlLib.ImageComboCtl.2\ : "Microsoft ImageComboBox Control 6.0 (SP4)"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\MSComctlLib.ImageListCtrl\ : "Microsoft ImageList Control, version 6.0"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\MSComctlLib.ImageListCtrl\ : "Microsoft ImageList Control 6.0 (SP4)"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\MSComctlLib.ImageListCtrl.2\ : "Microsoft ImageList Control, version 6.0"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\MSComctlLib.ImageListCtrl.2\ : "Microsoft ImageList Control 6.0 (SP4)"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\MSComctlLib.ListViewCtrl\ : "Microsoft ListView Control, version 6.0"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\MSComctlLib.ListViewCtrl\ : "Microsoft ListView Control 6.0 (SP4)"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\MSComctlLib.ListViewCtrl.2\ : "Microsoft ListView Control, version 6.0"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\MSComctlLib.ListViewCtrl.2\ : "Microsoft ListView Control 6.0 (SP4)"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\MSComctlLib.ProgCtrl\ : "Microsoft ProgressBar Control, version 6.0"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\MSComctlLib.ProgCtrl\ : "Microsoft ProgressBar Control 6.0 (SP4)"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\MSComctlLib.ProgCtrl.2\ : "Microsoft ProgressBar Control, version 6.0"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\MSComctlLib.ProgCtrl.2\ : "Microsoft ProgressBar Control 6.0 (SP4)"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\MSComctlLib.SBarCtrl\ : "Microsoft StatusBar Control, version 6.0"

```

```

HKEY_LOCAL_MACHINE\SOFTWARE\Classes\MSComctlLib.SBarCtrl\: "Microsoft
StatusBar Control 6.0 (SP4)"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\MSComctlLib.SBarCtrl.2\: "Microsoft
StatusBar Control, version 6.0"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\MSComctlLib.SBarCtrl.2\: "Microsoft
StatusBar Control 6.0 (SP4)"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\MSComctlLib.Slider\: "Microsoft Slider
Control, version 6.0"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\MSComctlLib.Slider\: "Microsoft Slider
Control 6.0 (SP4)"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\MSComctlLib.Slider.2\: "Microsoft Slider
Control, version 6.0"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\MSComctlLib.Slider.2\: "Microsoft Slider
Control 6.0 (SP4)"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\MSComctlLib.TabStrip\: "Microsoft
TabStrip Control, version 6.0"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\MSComctlLib.TabStrip\: "Microsoft
TabStrip Control 6.0 (SP4)"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\MSComctlLib.TabStrip.2\: "Microsoft
TabStrip Control, version 6.0"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\MSComctlLib.TabStrip.2\: "Microsoft
TabStrip Control 6.0 (SP4)"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\MSComctlLib.Toolbar\: "Microsoft Toolbar
Control, version 6.0"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\MSComctlLib.Toolbar\: "Microsoft Toolbar
Control 6.0 (SP4)"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\MSComctlLib.Toolbar.2\: "Microsoft
Toolbar Control, version 6.0"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\MSComctlLib.Toolbar.2\: "Microsoft
Toolbar Control 6.0 (SP4)"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\MSComctlLib.TreeCtrl\: "Microsoft
TreeView Control, version 6.0"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\MSComctlLib.TreeCtrl\: "Microsoft
TreeView Control 6.0 (SP4)"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\MSComctlLib.TreeCtrl.2\: "Microsoft
TreeView Control, version 6.0"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\MSComctlLib.TreeCtrl.2\: "Microsoft
TreeView Control 6.0 (SP4)"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\TypeLib\{000204EF-0000-0000-C000-
000000000046}\6.0\9\win32\: "C:\WINDOWS\System32\msvbvm60.dll"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\TypeLib\{000204EF-0000-0000-C000-
000000000046}\6.0\9\win32\: "C:\WINDOWS\System32\MSVBVM60.DLL"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\TypeLib\{831FDD16-0C5C-11D2-A9FC-
0000F8754DA1}\2.0\: "Microsoft Windows Common Controls 6.0"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\TypeLib\{831FDD16-0C5C-11D2-A9FC-
0000F8754DA1}\2.0\: "Microsoft Windows Common Controls 6.0 (SP4)"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\TypeLib\{831FDD16-0C5C-11D2-A9FC-
0000F8754DA1}\2.0\HELPDIR\: "C:\WINDOWS\System32\"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\TypeLib\{831FDD16-0C5C-11D2-A9FC-
0000F8754DA1}\2.0\HELPDIR\: ""
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\TypeLib\{EA544A21-C82D-11D1-A3E4-
00A0C90AEA82}\6.0\9\win32\: "C:\WINDOWS\System32\msvbvm60.dll\3"
HKEY_LOCAL_MACHINE\SOFTWARE\Classes\TypeLib\{EA544A21-C82D-11D1-A3E4-
00A0C90AEA82}\6.0\9\win32\: "C:\WINDOWS\System32\MSVBVM60.DLL\3"
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Cryptography\RNG\Seed: 51 0F 00 64 83
9D B3 B8 E4 D5 D6 40 EF A9 74 5F 14 19 B3 B0 F2 05 57 7B 62 EB 1C C4 11 F8 3C

```

```

E7 F9 EA AE ED F9 9B A2 67 C5 36 B7 1D 42 59 01 DB 3D A2 83 A0 D2 75 9E 9B E2
8A B9 B3 7C 89 6F 49 1B 67 94 EC 5F 00 9E 35 F0 37 D8 91 C2 C1 4B BA
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Cryptography\RNG\Seed: 70 0A E8 58 2F
42 20 02 68 B3 A5 60 2C C3 37 45 9F A3 64 70 FA A0 18 F7 F4 DD F2 41 94 16 BF
A3 C8 21 E1 47 E2 B3 18 B5 60 3C CA FB 92 BC F0 C6 71 E4 DD 32 6D 0B 2C C4 DE
0E BF 33 A8 64 97 A7 D9 AC 66 49 72 9B 11 64 2B 6F 8E D4 52 77 D1 E4
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\Eventlog\Application\VBRunti
me\EventMessageFile: "C:\WINDOWS\System32\msvbvm60.dll"
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\Eventlog\Application\VBRunti
me\EventMessageFile: "C:\WINDOWS\System32\MSVBVM60.DLL"
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\VBR
untime\EventMessageFile: "C:\WINDOWS\System32\msvbvm60.dll"
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\VBR
untime\EventMessageFile: "C:\WINDOWS\System32\MSVBVM60.DLL"
HKEY_USERS\S-1-5-21-1482476501-413027322-839522115-
1003\Software\Microsoft\Windows\CurrentVersion\Explorer\Discardable\PostSetup
\Component Categories\{00021493-0000-0000-C000-
000000000046}\Enum\Implementing: 1C 00 00 00 01 00 00 00 D4 07 09 00 04 00 02
00 12 00 09 00 28 00 F5 02 06 00 00 00 01 24 D0 30 81 6A D0 11 82 74 00 C0 4F
D5 AE 38 83 31 68 32 A0 48 1B 44 A3 42 7C 2A 44 0A 94 78 F3 31 EE C4 68 47 D2
11 BE 5C 00 A0 C9 A8 3D A1 61 4E A2 EF 78 B0 D0 11 89 E4 00 C0 4F C9 E2 6E 62
4E A2 EF 78 B0 D0 11 89 E4 00 C0 4F C9 E2 6E 64 4E A2 EF 78 B0 D0 11 89 E4 00
C0 4F C9 E2 6E
HKEY_USERS\S-1-5-21-1482476501-413027322-839522115-
1003\Software\Microsoft\Windows\CurrentVersion\Explorer\Discardable\PostSetup
\Component Categories\{00021493-0000-0000-C000-
000000000046}\Enum\Implementing: 1C 00 00 00 01 00 00 00 D4 07 09 00 04 00 02
00 12 00 17 00 1A 00 7E 01 06 00 00 00 01 24 D0 30 81 6A D0 11 82 74 00 C0 4F
D5 AE 38 83 31 68 32 A0 48 1B 44 A3 42 7C 2A 44 0A 94 78 F3 31 EE C4 68 47 D2
11 BE 5C 00 A0 C9 A8 3D A1 61 4E A2 EF 78 B0 D0 11 89 E4 00 C0 4F C9 E2 6E 62
4E A2 EF 78 B0 D0 11 89 E4 00 C0 4F C9 E2 6E 64 4E A2 EF 78 B0 D0 11 89 E4 00
C0 4F C9 E2 6E
HKEY_USERS\S-1-5-21-1482476501-413027322-839522115-
1003\Software\Microsoft\Windows\CurrentVersion\Explorer\Discardable\PostSetup
\Component Categories\{00021494-0000-0000-C000-
000000000046}\Enum\Implementing: 1C 00 00 00 01 00 00 00 D4 07 09 00 04 00 02
00 12 00 09 00 28 00 E0 03 02 00 00 00 25 8C 5C 4D 75 D0 D0 11 B4 16 00 C0 4F
B9 03 76 7F DE EA BD 65 C2 D0 11 BC ED 00 A0 C9 0A B5 0F
HKEY_USERS\S-1-5-21-1482476501-413027322-839522115-
1003\Software\Microsoft\Windows\CurrentVersion\Explorer\Discardable\PostSetup
\Component Categories\{00021494-0000-0000-C000-
000000000046}\Enum\Implementing: 1C 00 00 00 01 00 00 00 D4 07 09 00 04 00 02
00 12 00 17 00 25 00 65 00 02 00 00 00 25 8C 5C 4D 75 D0 D0 11 B4 16 00 C0 4F
B9 03 76 7F DE EA BD 65 C2 D0 11 BC ED 00 A0 C9 0A B5 0F
HKEY_USERS\S-1-5-21-1482476501-413027322-839522115-
1003\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{75048700-
EF1F-11D0-9888-006097DEACF9}\Count\HRZR_EHACNGU: 01 00 00 00 0B 00 00 00 D0
2F B4 CB 19 91 C4 01
HKEY_USERS\S-1-5-21-1482476501-413027322-839522115-
1003\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{75048700-
EF1F-11D0-9888-006097DEACF9}\Count\HRZR_EHACNGU: 01 00 00 00 0C 00 00 00 50
F0 C1 E1 19 91 C4 01
HKEY_USERS\S-1-5-21-1482476501-413027322-839522115-
1003\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{75048700-
EF1F-11D0-9888-006097DEACF9}\Count\HRZR_HVFPHG: 01 00 00 00 08 00 00 00 A0 4F
6A 30 19 91 C4 01

```

```
HKEY_USERS\S-1-5-21-1482476501-413027322-839522115-  
1003\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{75048700-  
EF1F-11D0-9888-006097DEACF9}\Count\HRZR_HVFPHG: 01 00 00 00 09 00 00 00 B0 CA  
9B E1 19 91 C4 01
```

```
-----  
Total changes:123  
-----
```

© SANS Institute 2004, Author retains full rights.

Analysis of a Red Hat Honeypot

Abstract: Analyze a compromised honeypot to determine the extent of damage and point of compromise. I use several forensic tools to discover the techniques and processes deployed against the system. This process explores and documents the intrusive activities.

James Shewmaker

GCFA Practical v1.5 Part 2 Option 1

Synopsis of Case Facts

A Red Hat 6.2 honeypot was placed into service on a residential DSL line on July 25, 2004. Typical services were enabled and content was copied to the machine. A second PC provided firewall network restrictions and bandwidth alarm running on FreeBSD.

Traffic monitoring alerts triggered on August 29, 2004 due to increased outgoing traffic. A copy of the contents of memory was acquired with the help of a CD-ROM response kit. The system's power was disabled. The drives were imaged after booting with Helix and transferred to an analysis machine to be examined.

System Description

The system's hostname is node12.bluenotch.net. It is a Linux Red Hat 6.2 honeypot. The partitioning configuration was taken from the system's /etc/fstab file before deployment.

/dev/hda5	/	ext2	defaults	1	1
/dev/hda1	/boot	ext2	defaults	1	2
/dev/hdb	/mnt/cdrom	iso9660	noauto,owner,ro	0	0
/dev/hda7	/tmp	ext2	defaults	1	2
/dev/hda8	/usr	ext2	defaults	1	2
/dev/hda6	/var	ext2	defaults	1	2
/dev/hdc1	/data	ext2	defaults	0	0
/dev/fd0	/mnt/floppy	auto	noauto,owner	0	0
none	/proc	proc	defaults	0	0
none	/dev/pts	devpts	gid=5,mode=620	0	0
/dev/hdc2	swap	swap	defaults	1	1

The honeypot was installed from a bootable CD-ROM. The default installation was selected as well as various additional packages, most importantly apache, mysqld, nfs, telnetd, wuftp, and XFree86. The following software was downloaded and compiled. Compiling some of the applications from source helped make the target more interested than a noticeably old operating system with a default configuration. The applications that were compiled were then current versions of the utilities ddclient, lame, lynx, screen, and wget. The current version of openssh was also downloaded, configured with incorrect settings, but not installed due to obvious compile errors. All third party source was left on the machine. The root password was set and one user account named "jim" was created.

The system was configured for DHCP and connected to a FreeBSD 4.9-STABLE bastion host. The FreeBSD machine was configured as a bridge and firewall. The firewall was configured to drop all packets from the honeypot or the outside directly to it, and used the "Dummynet" functionality to restrict outgoing traffic from the honeypot to 1 KB per second. This safety mechanism limits potential problems that may occur during

a slow incident response. A script on this firewall was configured to monitor the incoming traffic to the honeypot. This script's purpose is to send off an alarm once 6 MB of bandwidth is used. This firewall was connected to an eight port Asante 10 MB hub and then to a residential DSL line. The Internet provider's DHCP would provide the honeypot with its networking information.

The ddclient script on the honeypot was configured to update the IP address of the DNS name shew.gotdns.com. The hostname was set to node12.bluenotch.net and a CNAME record was created that pointed node12.bluenotch.net to shew.gotdns.com.

I took four personal websites and placed them on the data drive. I asked a musician friend of mine if I could place some of his music files on the system, and did so in both .wav and .mp3 format. Now this server looked like a small test or development server for websites, perhaps an older one that was never quite faded out.

Once the firewall and honeypot were ready, I imaged the honeypot with dd as a full backup. Then I connected to various IRC channels and asked for help getting .mp3 files on this node12.bluenotch.net server.

Hardware Description

The honeypot is on personal hardware and has no asset tags, therefore was not tagged for evidence.

Item	Item Type	Item Description
1	CPU	Pentium 233 MMX
2	RAM	2 x 8 MB SIMM and 2 x 32 MB SIMM = 80 MB
3	Removable Media	3.5" Mitsumi floppy (Master) and 8x Mitsumi CD-ROM (Primary IDE Slave)
4	System Drive	Quantum P/N CY43A011 S/N 164710921716 = 4 GB (Primary IDE Master)
5	Data Drive	Maxtor P/N 81750A4 S/N B40C0K1A = 4 GB (Secondary IDE Master)
6	PCI Video Adapter	
7	PCI Network Adapter	
8	PCI USB 1.1 Adapter	
9	Form Factor	ATX Mid-Tower

Image Media

Since the server was on the premises, it was easy to gather the volatile information. I logged on to my forensic analysis machine and started a netcat listener. I had anticipated a compromise of a machine on this LAN of honeypots and had the analysis machine (shew.bluenotch.net) configured and plugged into the same LAN. I turned on the machine, logged on, and started a netcat listener on port 4000. I switched the KVM interface to node12 and logged on as root so I could gather the memory contents as quickly as possible.

I immediately inserted my response kit CD-ROM into node12, and executed the following commands:

```
mount /dev/hdb /mnt/cdrom && \  
/mnt/cdrom/response_kit/linux_x86_static/dcfldd if=/dev/mem \  
hashwindow=0 | nc shew.bluenotch.net 4000
```

This command mounted the CD-ROM, ran dcfldd to collect every bit from the memory device, compute the md5sum of the contents, and sends it across the LAN to shew.bluenotch.net's port 4000 where the analysis machine saved every bit to a file. When it was completed I closed the netcat listener on shew.bluenotch.net and ran the following command to verify and save the md5sum of the acquired image.

```
md5sum mem.img > mem.img.md5
```

This stores an md5 fingerprint of the memory image. If I change any bit in the mem.img file and run md5sum again, it will change the fingerprint, and I will know the mem.img file is no longer the pristine copy I gather from the compromised box.

Now that I have a copy of the memory contents safe, I pull the power plug and remove the drives. This keeps the machine from changing any of the files, swapping memory out, or for the machine from being misused by the attacker during the analysis. As I gathered the state of the machine that I would lose during a power cycle, I can safely gather the rest of the evidence from our analysis machine.

My analysis host machine is a Shuttle XPC with Athlon processor, 512 MB of RAM, floppy, DVD-ROM, and a Maxtor 160 GB hard drive. I use Windows XP Professional and a VMware Workstation¹⁴ virtual machine running Helix 1.4. Helix¹⁵ is based on Knoppix based on the Debian distribution of Linux. It conveniently places many reliable forensic tools on this disk and is designed to be a clean forensic environment.

I place the honeypot's primary master drive on the primary master of my analysis machine, so I can image to my 120 GB data drive. I boot into Helix and capture the physical image as well as a logical image of each partition. I prefer to acquire the drives this way when the system is already down. This way all of the images are all in one place and md5sums during the acquisition already computed.

I use dcfldd from the Helix CD-ROM to acquire these images as an exact copy of literally every bit of data, in addition to computing the md5sum of each file (Figure 1). This md5sum serves as an efficient fingerprint of the data. If any bit were to change in one of those images, the md5sum of that file would change.

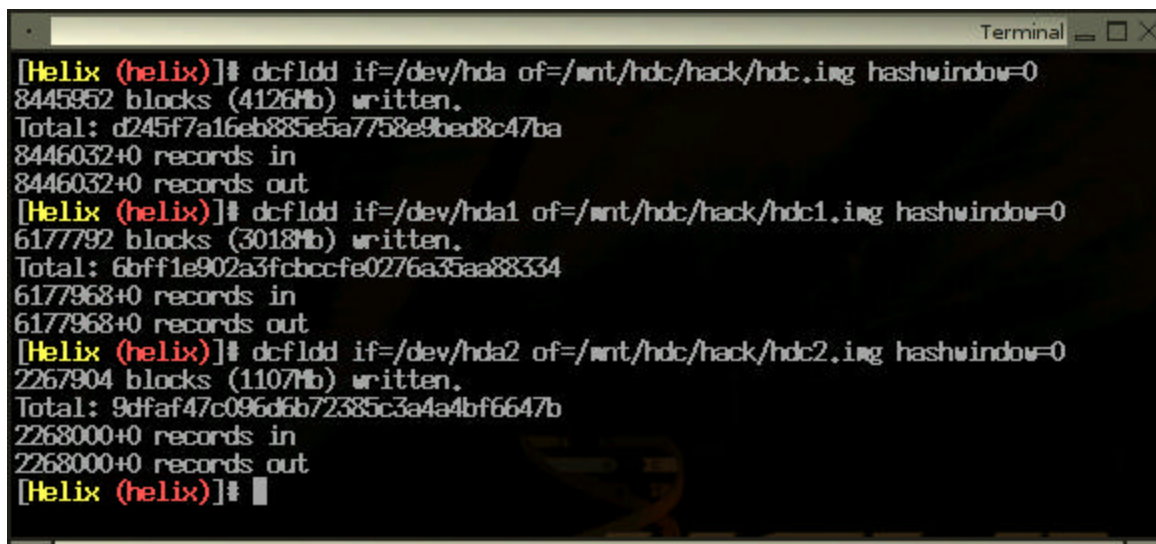
¹⁴ http://www.vmware.com/products/desktop/ws_features.html

¹⁵ <http://www.efense.com/helix/>

```
Terminal
[Helix (helix)]# dcfldd if=/dev/hda of=/mnt/hdc/hack/hda.img hashwindow=0
8467200 blocks (4136Mb) written.
Total: c9a34adac4ead609f8306add274d259e
8467200+0 records in
8467200+0 records out
[Helix (helix)]# dcfldd if=/dev/hda1 of=/mnt/hdc/hack/hda1.img hashwindow=0
513792 blocks (251Mb) written.
Total: 1a795060dfa384004aafda141aed51ba
514016+0 records in
514016+0 records out
[Helix (helix)]# dcfldd if=/dev/hda5 of=/mnt/hdc/hack/hda5.img hashwindow=0
513792 blocks (251Mb) written.
Total: 3a4b5671f5715afd564a6a5d41a8fc2b
514016+0 records in
514016+0 records out
[Helix (helix)]# dcfldd if=/dev/hda6 of=/mnt/hdc/hack/hda6.img hashwindow=0
513792 blocks (251Mb) written.
Total: e8b8962e719eb1d76c61e69e750b2125
514016+0 records in
514016+0 records out
[Helix (helix)]# dcfldd if=/dev/hda7 of=/mnt/hdc/hack/hda7.img hashwindow=0
513792 blocks (251Mb) written.
Total: ddff85c68064a85df87fd35ced9abdab
514016+0 records in
514016+0 records out
[Helix (helix)]# dcfldd if=/dev/hda8 of=/mnt/hdc/hack/hda8.img hashwindow=0
6409728 blocks (3131Mb) written.
Total: 109918be644e96d9ae567a38447b825b
6409872+0 records in
6409872+0 records out
[Helix (helix)]#
```

Figure 20

Once I have finished with the primary master of the honeypot. I shutdown and replace it with the secondary master from the honeypot. Booting into Helix once again, I acquire both the physical and logical images of every partition (Figure 2). Even though this drive was recognized as /dev/hdc by the honeypot, during this step it is /dev/hda because it is on the primary IDE channel.

A terminal window titled "Terminal" with a dark background and light-colored text. It shows three sequential commands and their outputs. The first command copies an image to hdc.img, the second to hdc1.img, and the third to hdc2.img. Each command uses the 'dcfldd' utility with 'if=/dev/hda' and 'of=' followed by the target file path. The output for each command shows the number of blocks written, the total data size in hexadecimal, and the number of records in and out.

```
[Helix (helix)]# dcfldd if=/dev/hda of=/mnt/hdc/hack/hdc.img hashwindow=0
8445952 blocks (4126Mb) written.
Total: d245f7a16eb885e5a7758e9bed8c47ba
8446032+0 records in
8446032+0 records out
[Helix (helix)]# dcfldd if=/dev/hda1 of=/mnt/hdc/hack/hdc1.img hashwindow=0
6177792 blocks (3018Mb) written.
Total: 6bfff1e902a3fcbccfe0276a35aa88334
6177968+0 records in
6177968+0 records out
[Helix (helix)]# dcfldd if=/dev/hda2 of=/mnt/hdc/hack/hdc2.img hashwindow=0
2267904 blocks (1107Mb) written.
Total: 9dfaf47c096d6b72385c3a4a4bf6647b
2268000+0 records in
2268000+0 records out
[Helix (helix)]#
```

Figure 21

To finish preparing the media for analysis, I return to Helix, but on VMware. I remove the honeypot drives and store them next to the honeypot machine. I boot my analysis machine into Windows XP to load Helix in a new virtual machine inside of VMware. I run the following commands to copy the two physical images over the virtual drives that VMware already created. Since VMware handles all drive activity in its own abstraction layer, the operating system does not need to behave any differently. I treat this virtual machine just as I would any PC, and use the `dcfldd` command again to copy from the images to the virtual drives.

I copied the entire environment of the honeypot, except for the contents of the memory. This enables us to examine the honeypot in greater precision. The revert feature allows us to return to the starting point after observing something on the system. This is highly beneficial, as examining a live system can change it. Another valuable side effect is that I am now running this honeypot on a much faster machine.

Media Analysis

Since I have duplicated the honeypot drives into VMware, I mount each partition to look for anything out of the ordinary (Figure 4). Note that I am using the files, not the partitions I imaged to this virtual machine.

```

[Helix (helix)]# mount -o ro,noexec,nodev,noatime,loop /mnt/hdb/hack/hda1.img /mnt/hda1
[Helix (helix)]# mount -o ro,noexec,nodev,noatime,loop /mnt/hdb/hack/hda5.img /mnt/hda5
[Helix (helix)]# mount -o ro,noexec,nodev,noatime,loop /mnt/hdb/hack/hda6.img /mnt/hda6
[Helix (helix)]# mount -o ro,noexec,nodev,noatime,loop /mnt/hdb/hack/hda7.img /mnt/hda7
[Helix (helix)]# mount -o ro,noexec,nodev,noatime,loop /mnt/hdb/hack/hda8.img /mnt/hda8
[Helix (helix)]# mount -o ro,noexec,nodev,noatime,loop /mnt/hdb/hack/hdc1.img /mnt/hdc1
[Helix (helix)]#

```

Figure 22

The /boot partition is stored in /mnt/hda1 and looks normal at first glance. The root partition is stored in /mnt/hda5. I immediately discover two suspicious files in the /root/test/ directory (Figure 5). I add “a.out” and “spoofer” to our keyword list and continue the media analysis.

```

[Helix (test)]# pwd
/mnt/hda5/root/test
[Helix (test)]# ls -al
total 21
drwxr-xr-x  2 root  root    1024 Aug 17 14:16 .
drwxr-xr-x  6 root  root    1024 Aug 11 19:34 ..
-rwxr-xr-x  1 root  root   13988 Aug 17 14:16 a.out
-rw-rw-r--  1 root  root    3547 Aug 17 14:16 udpspoof.c
[Helix (test)]#

```

Figure 23

It appears that the intruder did not attempt to hide these files, so I continue my cursory look at the filesystem. In the /home/jim/ directory I see more suspicious activity (Figure 6).

```

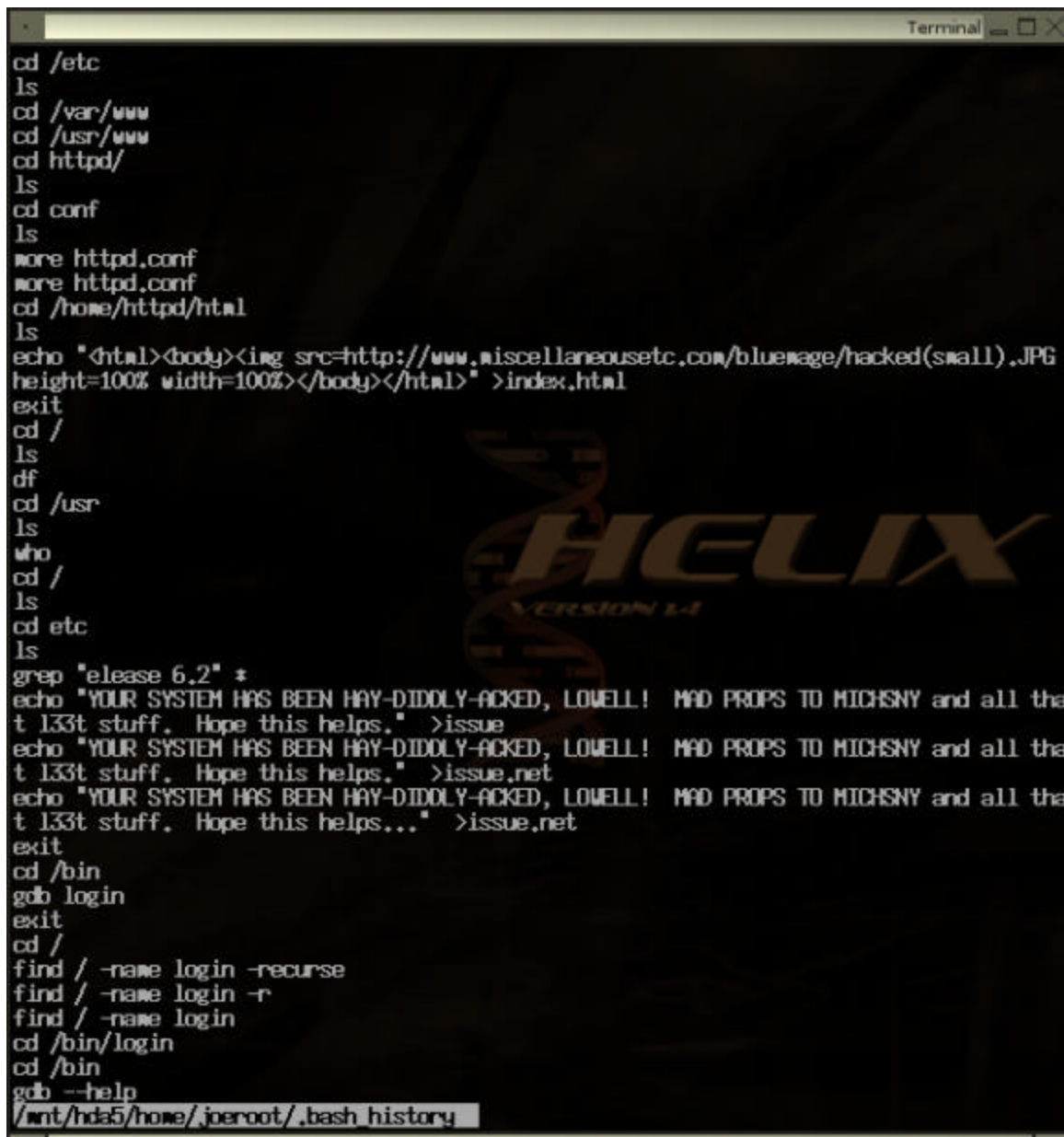
[Helix (hda5)]# ls -alt home
total 8
drwxr-xr-x  2 root  staff    1024 Aug 23 02:41 joeuser
drwx----- 4 500   500      1024 Aug 12 17:21 jim
drwx----- 2 501   501      1024 Aug 12 09:45 www
drwxr-xr-x  2 root  staff    1024 Aug 11 20:46 joeroot
drwxr-xr-x  8 root  root     1024 Aug 11 19:34 .
drwxr-xr-x  6 root  root     1024 Aug 11 19:24 ftp
drwxr-xr-x 18 root  root     1024 Aug  9 13:06 ..
drwxr-xr-x  6 root  root     1024 Jul 23 13:18 httpd
[Helix (hda5)]#

```

Figure 24

I know from the system’s timeline that the root and jim users were created well before August 11, and accounts for joeuser and joeroot have been added. Examining the

/home/joeuser/ directory reveals a bash history log. The .bash_history file contains up to 500 commands of the commands ran by that user¹⁶.



```
cd /etc
ls
cd /var/www
cd /usr/www
cd httpd/
ls
cd conf
ls
more httpd.conf
more httpd.conf
cd /home/httpd/html
ls
echo "<html><body><img src=http://www.misellaneousetc.com/bluewage/hacked(small).JPG
height=100% width=100%></body></html>" >index.html
exit
cd /
ls
df
cd /usr
ls
who
cd /
ls
cd etc
ls
grep "elease 6.2" *
echo "YOUR SYSTEM HAS BEEN HAY-DIDDLY-ACKED, LOWELL! MAD PROPS TO MICHSNY and all tha
t l33t stuff. Hope this helps." >issue
echo "YOUR SYSTEM HAS BEEN HAY-DIDDLY-ACKED, LOWELL! MAD PROPS TO MICHSNY and all tha
t l33t stuff. Hope this helps." >issue.net
echo "YOUR SYSTEM HAS BEEN HAY-DIDDLY-ACKED, LOWELL! MAD PROPS TO MICHSNY and all tha
t l33t stuff. Hope this helps..." >issue.net
exit
cd /bin
gdb login
exit
cd /
find / -name login -recurse
find / -name login -r
find / -name login
cd /bin/login
cd /bin
gdb --help
/rnt/hda5/home/joeuser/.bash_history
```

Figure 25

In Figure 8 I see a portion of the .bash_history file which logged vandalism of the telnet banner and a website defacement. I add "miscellaneousetc", "DIDDLY", "LOWELL", "MAD PROPS", and "MICHSNY" to the keyword list.

¹⁶ <http://www.die.net/doc/linux/man/man1/bash.1.html>

The other system directories show nothing out the ordinary; only the original services have start up scripts. There is no `.bash_history` file for the `joeuser` account. I proceed to the `/bin/` directory, and see that somebody has messed with the dates and times of the system binary files (Figure 9).

```

[Helix (bin)]# ls -alt /mnt/hda5/bin/ | more
total 5317
drwxr-xr-x 2 root root 1024 Aug 11 20:42 stuff
drwxr-xr-x 3 root root 2048 Aug 11 20:28 .
-rwxr-xr-x 1 root root 2612 Aug 11 19:34 arch
-rwxr-xr-x 1 root root 60592 Aug 11 19:34 ash
-rwxr-xr-x 1 root root 263064 Aug 11 19:34 ash.static
-rwxr-xr-x 1 root root 5756 Aug 11 19:34 basename
-rwxr-xr-x 1 root root 316848 Aug 11 19:34 bash
-rwxr-xr-x 1 root root 446800 Aug 11 19:34 bash2
-rwxr-xr-x 1 root root 9528 Aug 11 19:34 cat
-rwxr-xr-x 1 root root 12044 Aug 11 19:34 chgrp
-rwxr-xr-x 1 root root 13436 Aug 11 19:34 chmod
-rwxr-xr-x 1 root root 11952 Aug 11 19:34 chown
-rwxr-xr-x 1 root root 49680 Aug 11 19:34 consolechars
-rwxr-xr-x 1 root root 33392 Aug 11 19:34 cp
-rwxr-xr-x 1 root root 45712 Aug 11 19:34 cpio
-rwxr-xr-x 1 root root 25680 Aug 11 19:34 date
-rwxr-xr-x 1 root root 26576 Aug 11 19:34 dd
-rwxr-xr-x 1 root root 24816 Aug 11 19:34 df
-rwxr-xr-x 1 root root 4016 Aug 11 19:34 dmesg
-rwxr-xr-x 1 root root 2448 Aug 11 19:34 dovec
-rwxr-xr-x 1 root root 6792 Aug 11 19:34 echo
-rwxr-xr-x 1 root root 65520 Aug 11 19:34 ed
-rwxr-xr-x 1 root root 75600 Aug 11 19:34 egrep
-rwxr-xr-x 1 root root 4320 Aug 11 19:34 false
-rwxr-xr-x 1 root root 75600 Aug 11 19:34 fgrep
-rwxr-xr-x 2 root root 148848 Aug 11 19:34 gawk
-rwxr-xr-x 2 root root 148848 Aug 11 19:34 gawk-3.0.4
-rwxr-xr-x 1 root root 75600 Aug 11 19:34 grep
-rwxr-xr-x 3 root root 46384 Aug 11 19:34 gunzip
-rwxr-xr-x 3 root root 46384 Aug 11 19:34 gzip
-rwxr-xr-x 1 root root 8896 Aug 11 19:34 hostname
-rwxr-xr-x 1 root root 2891 Aug 11 19:34 igawk
-rwxr-xr-x 1 root root 19228 Aug 11 19:34 ipcalc
-rwxr-xr-x 1 root root 7952 Aug 11 19:34 kill
-rwxr-xr-x 1 root root 767016 Aug 11 19:34 linuxconf
-rwxr-xr-x 1 root root 20240 Aug 11 19:34 ln
-rwxr-xr-x 1 root root 76592 Aug 11 19:34 loadkeys
-rwxr-xr-x 1 root root 20452 Aug 11 19:34 login
-rwxr-xr-x 1 root root 20452 Aug 11 19:34 login.old
-rwxr-xr-x 1 root root 43024 Aug 11 19:34 ls
-rwxr-xr-x 1 root man 62384 Aug 11 19:34 mail
-rwxr-xr-x 1 root root 13696 Aug 11 19:34 mkdir
More--

```

Figure 26

Continuing looking, I examine the `/lib` directory, listing the most recently used files first, and see a suspicious time stamp on `libcrypt.so.1` (Figure 10).

```
Terminal
[Helix (hda5)]# ls -alt lib/ | more
total 10134
-rwxr-xr-x 1 root root 62153 Aug 12 00:13 libcrypt.so.1
drwxr-xr-x 4 root root 3072 Aug 11 23:45 .
drwxr-xr-x 18 root root 1024 Aug 9 13:06 ..
lrwxrwxrwx 1 root root 14 Jul 23 05:57 libdl.so.1 -> libdl.so.1.9.5
lrwxrwxrwx 1 root root 17 Jul 23 05:57 ld-linux.so.1 -> ld-linux.so.1
.9.5
drwxr-xr-x 3 root root 1024 Jul 23 05:56 modules
--More--
```

Figure 27

The /sbin contents appear to legitimate dates and times, but the directory itself shows August 11 at 19:34 for its modification time.

In analyzing the files on the /dev/hda6 device (mounted as /var on the honeypot), I see a significant amount of activity in the /var/log/ files. The summary of log contents:

- wuftp buffer overflow attempts
- rpc.statd attempts
- rlogin attempts
- telnet login attempts
- ftp login attempts
- pop3 login attempts
- many failed login attempts for users "guest", "someguy", "someguy2", "someuser2", "joeuser", "joeuser2", "joeroot2", "root", and "jim"
- passive portscans
- last entry in /var/log/messages is for August 12, 2004 at 09:46:07

The key points to note at this stage of the investigation are according to /var/log/messages are two exploits for wuftp and one for rpc.statd (Figure 11).



this stage the attacker is
does appear that the date
t of the /mnt/hda6 partition


```

[Helix (hda8)]# ls -alrt
total 117
438 drwxr-xr-x 14 root root 1024 Nov  3 13:19 ..
32909 drwxr-xr-x  3 root root 4096 Aug 11 20:52 stuff
  2 drwxr-xr-x 22 root root 4096 Aug 11 20:42 .
128258 drwxr-xr-x 16 root root 4096 Aug 10 16:49 local
128257 drwxr-xr-x 13 root root 4096 Aug  7 05:02 man
32065 drwxr-xr-x 145 root root 4096 Jul 23 06:04 doc
336673 drwxr-xr-x 35 root root 4096 Jul 23 06:04 include
96193 drwxr-xr-x 34 root root 8192 Jul 23 06:04 lib
304611 drwxr-xr-x  2 root root 4096 Jul 23 06:04 sbin
272545 drwxr-xr-x  6 root root 16384 Jul 23 06:04 bin
288577 drwxr-xr-x  2 root root 4096 Jul 23 06:04 dict
320643 drwxr-xr-x 23 root root 4096 Jul 23 06:04 share
352705 drwxr-xr-x  2 root root 8192 Jul 23 06:03 info
320641 drwxr-xr-x  2 root root 4096 Jul 23 06:03 games
336675 drwxr-xr-x  4 root root 4096 Jul 23 06:02 src
112561 drwxr-xr-x  3 root root 4096 Jul 23 05:57 i486-linux-libc5
112514 drwxr-xr-x  7 root root 4096 Jul 23 05:57 kerberos
48098 drwxr-xr-x  3 root root 4096 Jul 23 05:52 libexec
304750 drwxr-xr-x  4 root root 4096 Jul 23 05:51 i386-redhat-linux
176353 drwxr-xr-x  7 root root 4096 Jul 23 05:47 X11R6
  12 drwxr-xr-x  1 root root  10 Jul 23 05:47 tmp -> ../var/tmp
  11 drwxr-xr-x  2 root root 16384 Jul 23 05:46 lost+found
304609 drwxr-xr-x  2 root root 4096 Feb  6 1996 etc

[Helix (hda8)]# ls -alrt stuff/
total 12
288429 drwxr-xr-x  69 321 1002 4096 Aug 11 23:02 glibc-2.1.3
32909 drwxr-xr-x  3 root root 4096 Aug 11 20:52 .
  2 drwxr-xr-x 22 root root 4096 Aug 11 20:42 ..

[Helix (hda8)]#

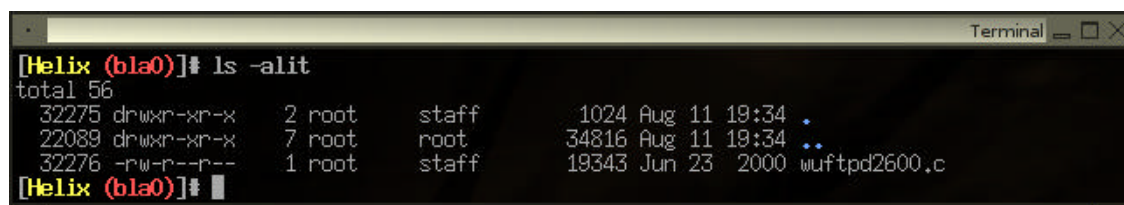
```

Figure 30

The /data partition was the first portion of the second IDE drive, and I found nothing interesting. The swap partition was /hdc2. This swap partition and memory snapshot I acquired before pulling the plug on the machine will be analyzed later. I finish walking through the system looking for obvious abnormalities and run a few commands to look for interesting files.

Command	Description	Results
find /mnt/ -perm 4000	Find any files that are setuid (when used they become the user that owns them, instead of the user running them)	None
find /mnt/ -perm 2000	Find any files that are setgid (when used they become the group that owns them, not the group running the file)	None
find /mnt/ -name "*core"	Find any coredump files produced when a process is killed dramatically	None
find /mnt/ -cnewer \hda5/etc/shadow-	Find any files created since the /etc/shadow- file (August 12, 09:45)	No new information
find /mnt/hda5/dev/ -type f	Find any userland files in the system devices	/mnt/hda5/dev/MAKDEV /mnt/hda5/dev/bla0/wuftpd2600.c
chkrootkit -r /mnt/	Find any known rootkits	None

The only new information I acquired was the /dev/bla0/wuftp2600.c file (Figure 14). Looking closer at that directory I see that this file claims to be created on August 11 at 19:45. This file is an exploit for the FTP server running on this honeypot. A detailed description of this vulnerability is in CVE-2000-0573¹⁷. This particular version of the exploit is available in a variety of places on the Internet.



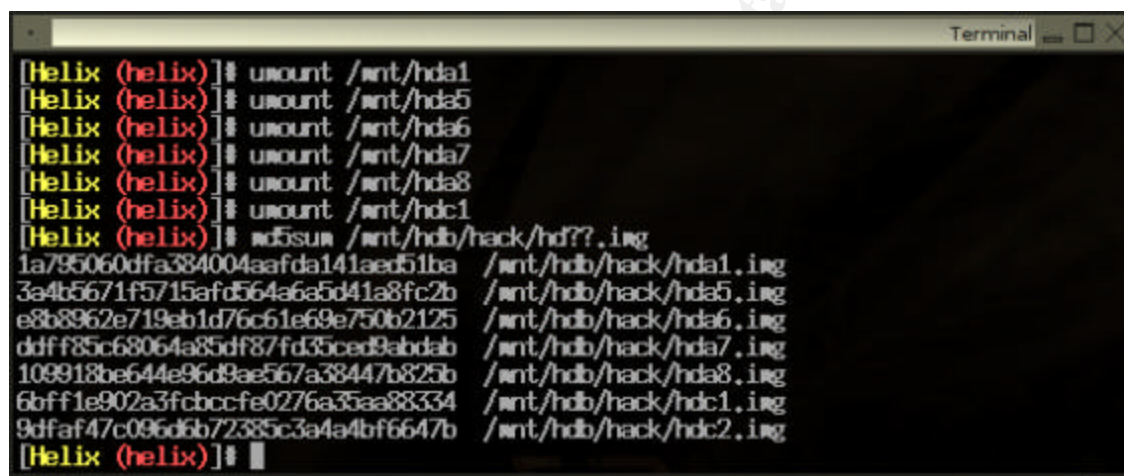
```

[Helix (bla0)]# ls -alit
total 56
 32275 drwxr-xr-x   2 root   staff   1024 Aug 11 19:34 .
 22089 drwxr-xr-x   7 root   root    34816 Aug 11 19:34 ..
 32276 -rw-r--r--   1 root   staff   19343 Jun 23  2000 wuftp2600.c
[Helix (bla0)]#

```

Figure 31

And to be sure I did not adversely affect our clean image, I unmount the drives and run md5sum on both drives to make sure I work on a clean copy (Figure 15).



```

[Helix (helix)]# umount /mnt/hda1
[Helix (helix)]# umount /mnt/hda5
[Helix (helix)]# umount /mnt/hda6
[Helix (helix)]# umount /mnt/hda7
[Helix (helix)]# umount /mnt/hda8
[Helix (helix)]# umount /mnt/hdc1
[Helix (helix)]# md5sum /mnt/hdb/hack/hd???.img
1a795060dfa384004aafda141aed51ba /mnt/hdb/hack/hda1.img
3a4b5671f5715afd564a6a5d41a8fc2b /mnt/hdb/hack/hda5.img
e8b8962e719eb1d76c61e69e750b2125 /mnt/hdb/hack/hda6.img
ddff85c68064a85df87fd35ced9abdb /mnt/hdb/hack/hda7.img
109918be644e96d9ae567a38447b825b /mnt/hdb/hack/hda8.img
6bfff1e902a3fcbccfe0276a35aa88334 /mnt/hdb/hack/hdc1.img
9dfaf47c096d6b72385c3a4a4bf6647b /mnt/hdb/hack/hdc2.img
[Helix (helix)]#

```

Figure 32

Timeline Analysis

Appendix A includes a complete timeline created from the files' dates and times on the system. Here is a summary of the key events. Since the time on the machine appears to have been altered, I have included both actual time and system time where available. This summary timeline was created from personal observation, various system logs, and file MAC times.

I was able to confirm the day of the time abnormality was August 23, 2004 because the weekly rotation of logs provided a second string of events for any given day that did not

¹⁷ <http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0573>

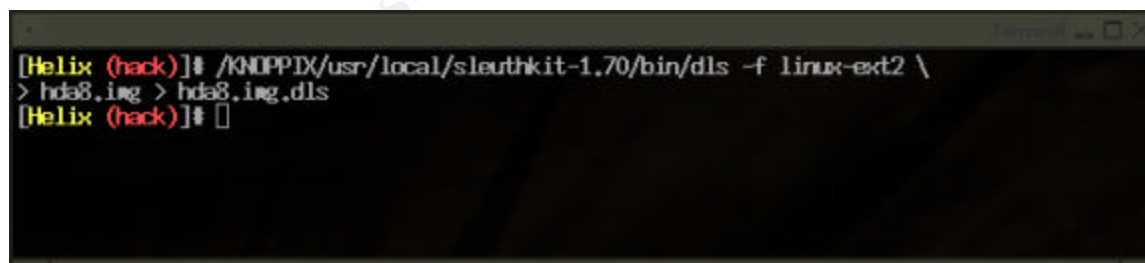
have any inconsistencies. I was also able to confirm the time the system was imaged on August 29, 2004 by booting a copy of the drive images in a VMware virtual machine.

Actual Time (Approximate)	System Time	Description
07/23/2004	07/23/2004	Red Hat 6.2 was installed on the main drive, configured, then imaged as a backup
07/25/2004 12:09:48	07/25/2004 12:09:48	Honeypot was exposed to the Internet, and loaded with copies of static websites – last changes made to the system
08/13/2004 12:14:40	08/13/2004 12:14:40	Attempt to login with rlogin from 58.106.74.235
08/19/2004 16:10:16	08/19/2004 16:10:16	Attempt to login with rlogin from 83.117.8.8
08/19/2004 23:32:11	08/19/2004 23:32:11	Multiple anonymous FTP logins from splat.nybbles.net
08/22/2004 21:39:07	08/22/2004 21:39:07	Exploit of rpc.statd
08/23/2004 15:20:33	08/23/2004 15:20:33	Last correct time entry from /var/log/messages (also appears that the honeypot rebooted)
08/23/2004 ~15:48:00	08/06/2004 12:48:10	First incorrect time entry from /var/log/messages
08/26/2004	08/09/2004 19:55:46	Anonymous FTP login from p508DF795.dip.t-dialin.net followed by multiple failed login attempts
08/26/2004	08/09/2004 19:56:07	Attempt to login as www user from same address
08/28/2004	08/11/2004 18:06:36	Attempt to login as someguy from 63.229.11.105
08/28/2004	08/11/2004 18:28:47	Attempt to login as joeroot2 from 63.229.11.105
08/28/2004	08/11/2004 18:28:59	Successful login as joeuser2 from 63.229.11.105 and successfully becomes joeroot2 (uid=0)
08/28/2004	08/11/2004 18:36:32	More multiple login failures from 70-56-68-132.tukw.qwest.net as someuser2
08/28/2004	08/11/2004 20:43:25	/usr/stuff/glibc-2.1.3/login/login.c changed

Actual Time (Approximate)	System Time	Description
08/28/2004	08/11/2004 22:30:56	joeuser2 tries to set password for jim account
08/28/2004	08/11/2004 23:10:03	Multiple attempts to login from 70-56-68-132.tukw.qwest.net as jim
08/29/2004	08/12/2004 00:13:13	/usr/stuff/glibc-1.3.2/md5-crypt/md5-crypt.c edited (from /home/joeroot/.bash_history and MAC timeline)
08/29/2004	08/12/2004 01:15:26	FTP exploit from 63.229.11.105
08/29/2004	08/12/2004 01:17:13	Second FTP exploit from 63.229.11.105
08/29/2004	08/12/2004 06:35:41	Root login on console to image memory – initial Incident Response
08/29/2004	08/12/2004 09:07	Pull the power plug from the honeypot

Recover Deleted Files

To recover deleted files, I will use a combination of tools from the Sleuth Kit¹⁸. First I need to find out where on the hard drive the data from the /usr/stuff/glibc-1.3.2/ files are. The fls tool lists the filename and other key information about the file such as path, permissions, and inode.



```
[Helix (hack)]# /KNOPPIX/usr/local/sleuthkit-1.70/bin/dls -f linux-ext2 \
> hda8.img > hda8.img.dls
[Helix (hack)]#
```

Figure 33

Looking at the new hda8.img.dls file I made in Figure 16, I see more than just a few files that came from the /usr partition, so I decide to write a quick Perl script to help me recover all files from the fls hda8 results. This script opens the output of the previous

¹⁸ <http://www.sleuthkit.org/>

command, saves the filename and inode, and uses the Sleuth Kit's `icat` command to dump the contents of that inode a new file. I do not need a complete directory tree of every deleted file from that partition, I replace every slash in the path with an underscore. This keeps all the files in one place, and easier to process with `strings` or `grep`.

```
#!/usr/bin/perl -w

open (FLS,"<hda8.fls") || die;
while ($line=<FLS>) {
    @data = split(/\|/, $line);
    $file{$data[1]} = $data[3];
}
close (FLS);

foreach $filename (keys(%file)) {
    if ($filename =~ /\$/ ) {next;} # Skip directories
    $filename_new = $filename;
    $filename_new =~ s/\//_/g;
    print "Recovering $filename as $filename_new\n";
    system("/KNOPPIX/usr/local/sleuthkit-1.70/bin/icat " .
        "-r -f linux-ext2 /mnt/hdb/hack/hda8.img " .
        $file{$filename} . " > hda8.undeleted/$filename_new" );
}
```

This script recovers 40,035 files from the hda8 partition alone. Note this includes live files as well as dead files, since I used the default output of the `fls` command. More importantly, I recovered 15,443 files that were deleted out of the `/usr/stuff` directory. Examination of these files fails to come up with anything useful, even with a line-by-line examination with a hex editor for abnormalities or keywords (Figure 17).

```

[Helix (hda8,undeleted)]# ls *md5-crypt*
_usr_stuff_glibc-2.1.3_md5-crypt
_usr_stuff_glibc-2.1.3_md5-crypt_
_usr_stuff_glibc-2.1.3_md5-crypt_...
_usr_stuff_glibc-2.1.3_md5-crypt_Makefile
_usr_stuff_glibc-2.1.3_md5-crypt_Versions
_usr_stuff_glibc-2.1.3_md5-crypt_crypt-entry.d
_usr_stuff_glibc-2.1.3_md5-crypt_crypt-entry.o
_usr_stuff_glibc-2.1.3_md5-crypt_crypt-entry.op
_usr_stuff_glibc-2.1.3_md5-crypt_crypt-entry.os
_usr_stuff_glibc-2.1.3_md5-crypt_crypt.d
_usr_stuff_glibc-2.1.3_md5-crypt_crypt.o
_usr_stuff_glibc-2.1.3_md5-crypt_crypt.op
_usr_stuff_glibc-2.1.3_md5-crypt_crypt.os
_usr_stuff_glibc-2.1.3_md5-crypt_crypt_util.d
_usr_stuff_glibc-2.1.3_md5-crypt_crypt_util.o
_usr_stuff_glibc-2.1.3_md5-crypt_crypt_util.op
_usr_stuff_glibc-2.1.3_md5-crypt_crypt_util.os
_usr_stuff_glibc-2.1.3_md5-crypt_libcrypt.a
_usr_stuff_glibc-2.1.3_md5-crypt_libcrypt.so
_usr_stuff_glibc-2.1.3_md5-crypt_libcrypt.so.1
_usr_stuff_glibc-2.1.3_md5-crypt_libcrypt.p.a
_usr_stuff_glibc-2.1.3_md5-crypt_libcrypt.pic.a
_usr_stuff_glibc-2.1.3_md5-crypt_md5-crypt.c
_usr_stuff_glibc-2.1.3_md5-crypt_md5-crypt.d
_usr_stuff_glibc-2.1.3_md5-crypt_md5-crypt.o
_usr_stuff_glibc-2.1.3_md5-crypt_md5-crypt.op
_usr_stuff_glibc-2.1.3_md5-crypt_md5-crypt.os
_usr_stuff_glibc-2.1.3_md5-crypt_md5.c
_usr_stuff_glibc-2.1.3_md5-crypt_md5.d
_usr_stuff_glibc-2.1.3_md5-crypt_md5.h
_usr_stuff_glibc-2.1.3_md5-crypt_md5.o
_usr_stuff_glibc-2.1.3_md5-crypt_md5.op
_usr_stuff_glibc-2.1.3_md5-crypt_md5.os
_usr_stuff_glibc-2.1.3_md5-crypt_md5c-test.c
_usr_stuff_glibc-2.1.3_md5-crypt_md5c-test.d
_usr_stuff_glibc-2.1.3_md5-crypt_md5test.c
_usr_stuff_glibc-2.1.3_md5-crypt_md5test.d
_usr_stuff_glibc-2.1.3_md5-crypt_onlymd5-entry.c
_usr_stuff_glibc-2.1.3_md5-crypt_onlymd5-entry.d
_usr_stuff_glibc-2.1.3_md5-crypt_onlymd5-entry.o
_usr_stuff_glibc-2.1.3_md5-crypt_onlymd5-entry.op
_usr_stuff_glibc-2.1.3_md5-crypt_onlymd5-entry.os
_usr_stuff_glibc-2.1.3_md5-crypt_stamp.o
_usr_stuff_glibc-2.1.3_md5-crypt_stamp.os
_usr_stuff_glibc-2.1.3_md5-crypt_stamp.op
_usr_stuff_glibc-2.1.3_md5-crypt_stamp.os

[Helix (hda8,undeleted)]# grep joeuser *md5-crypt*
[Helix (hda8,undeleted)]# grep *login.c*
[Helix (hda8,undeleted)]# grep joe *login.c*
[Helix (hda8,undeleted)]# grep joe *md5-crypt*
[Helix (hda8,undeleted)]#

```

Figure 34

Recovering files from the other partitions in the same way, I was unable to recover any new information. The bulk of deleted files were created during installation or configuration of the honeypot.

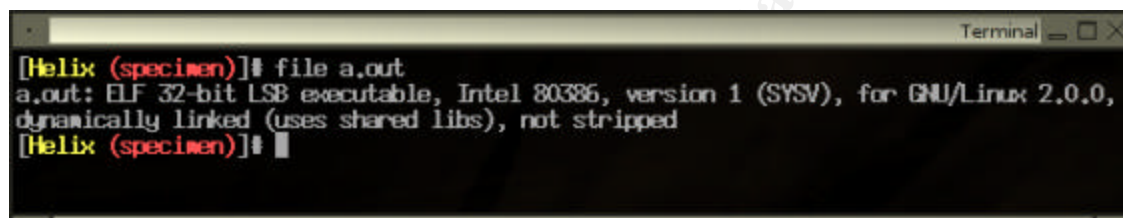
String Search

In the process of analyzing this honeypot, I built a list of keywords that may be used on the system by an intruder or be in a log file that provides additional information. I used this list to enter into Autopsy in a search for things that could be interesting to this investigation.

Keywords					
Rootkit	joe	wbar7.sea-4-12-024-250.dsl.verizon.net	lsanca2.dsl-verizon.net	68.235.193.29	63.227.131.230
Spoof	miscellaneousetc	p508DF795.dip.t-dialin.net	80.137.151.171	4.3.89.110	70.56.68.157
a.out	Diddly	80.141.247.149	4.14.144.53	162.105.204.75	70.56.68.132
Uftp	Lowell	63.229.11.105	162.105.204.75	generic-hostname.arrowneet.dk	4.26.183.157

God	Map props	tukw.qwest.net	4.63.234.158	jeannedarc-2-82-67-85-25.fbx.proxad.net	splat.nibbles.net
Sploit	Michsny	218.109.214.101	83.117.8.8	81.208.112.72	Bluemage
Exploit	Someguy	4.63.232.245	66.93.180.108	68.106.74.235	Someuser
Beaty and Beast	jim@bluenotch.net	stuff	trojan		

After performing an exhaustive search on the unallocated portions of each partition I was unable to find a single clue or file used by the attacker that did not still exist on the system in a live form. This is not too surprising as I have seen the attacker have several backup copies of the /etc/passwd and /etc/shadow files as well as the /usr/stuff directory.



```

[Helix (specimen)]# file a.out
a.out: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), for GNU/Linux 2.0.0,
dynamically linked (uses shared libs), not stripped
[Helix (specimen)]#

```

Figure 35

Using a string analysis on the /root/test/a.out file confirms that the source code in the same directory as is its true source. The file command gives various details about a file by examining some of bytes at the beginning. It shows me the symbols are intact and it appears to have been compiled from this source (Figure 18). There are two interesting items in the strings of this file (Figure 19), the phrase “Beauty and Beast”, and the email address jim@bluenotch.net.

Did the attacker assume because this box was node12.bluenotch.net that mail would get delivered even though there is no mail exchange record in DNS for this system? I add the new phrase, entire email address (since this address was never used) to my keyword list. As the strings matches the source file I have precisely, a complete debug of this binary file would be unnecessary.

```
Terminal
[Helix (specimen)]$ strings a.out
/lib/ld-linux.so.2
__gmon_start__
libc.so.6
putchar
printf
perror
socket
fprintf
__deregister_frame_info
setsockopt
sendto
bcopy
gethostbyname
stderr
htons
exit
atoi
_IO_stdin_used
__libc_start_main
__register_frame_info
GLIBC 2.0
PTRNL
us$<
Beauty and Beast/jim@bluenotch.net
usr: %s IP_origen puerto_origen IP_destino puerto_destino
imposible resolver ip origen
imposible resolver ip destino
fallo socket
Tienes IP_HDRINCL :-))
fallo setsockopt IP_HDRINCL
fallo sendto
datagrama mandado:
%02x
[Helix (specimen)]$
```

Figure 36

The last remaining point of interest I found was the `/lib/libcrypt.so.1` file. This file was also displayed earlier in the `/home/joeroot/.bash_history` file. Just looking at this file strings gives us nothing. To investigate further, I use `khxedit` to look at both the hexadecimal representation of the file and an ASCII representation. I do find an occurrence of the word “stuff” which reminds me of the path I found earlier in at `/usr/stuff/`. I will be careful to take a very close look at this item, it is often that attackers create a backdoor in such a place¹⁹ so that they can return later.

¹⁹ Skoudis, p. 190

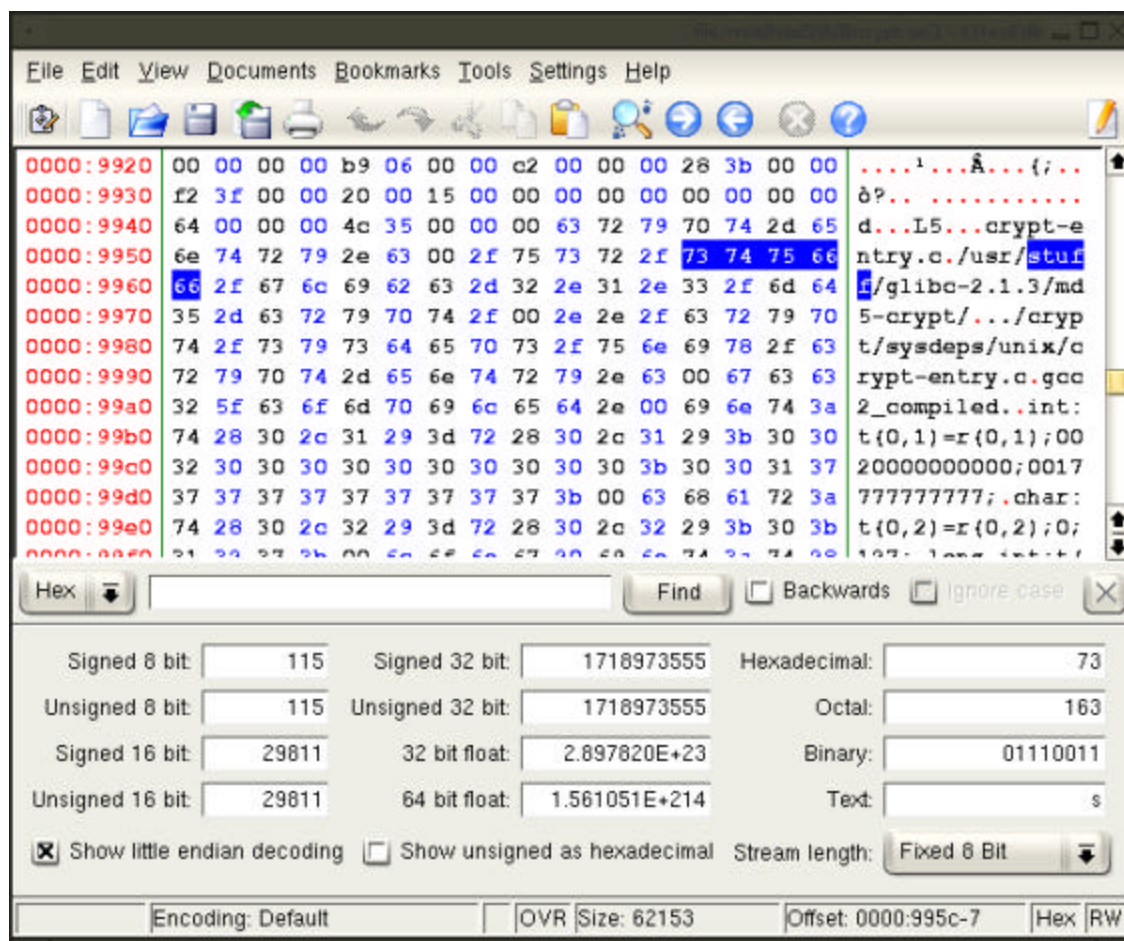


Figure 37

Since this file references the files I recovered from `/usr/stuff/` (Figure 20), I run a strings search on those recovered files using our keywords. This search provides no new information. Either the intruder hid his tracks very well or he was unable to finish establishing a back door on this host.

Conclusion

It appears the attacker exploited the CVE vulnerability in the NFS system's `rpc.statd` program²⁰. Access was also achieved by attacking the vulnerability in `wuftpd`. Accounts were created, the `/bin/login` was compiled in an attempt to backdoor the system. Attempts to break into the account named `jim` appear to be unsuccessful.

The website and telnet banner were defaced. The telnet banner was changed later, as I can see from the `joeroot` account's `.bash_history` file. The date and time of the honeypot was somehow changed to hide the tracks of the attacker. The `/bin` directory contents also had their time and date changed.

²⁰ <http://www.securityfocus.com/bid/1480/info/>

It appears that the attacker came in from several points, Verizon DSL in the Seattle area, another Verizon IP address (63.229.11.105), then from somewhere via Qwest. The other IP addresses I saw in our logs appear to be scans that were unrelated. The log files do not appear to have been sanitized, and the work area the attacker used was not hidden. This leads us to believe the attacker was not worried about getting caught. The attack appears to have a time frame of around three days.

The attacker attempted to backdoor the `/lib/libcrypt.so.1` library and the `/bin/login` executable. The attacker left most of the evidence in the logs after compromising the honeypot, as well as numerous backup copies of modified files.

Two new user accounts were made for the intruder's use. I find no trace of the intruder connecting to other machines except to download towards the goal of backdooring the system. There is no evidence of outgoing attacks.

The keyword list also turned up little specific information when used with online search engines. I did get a hit on "MICHSNY", but it appears to be unrelated, and does not contribute any new information to what I found on the system.

© SANS Institute 2004, Author retains full rights.

REFERENCES

9. "Products -- VMware Workstation 4.5 ". URL: http://www.vmware.com/products/desktop/ws_features.html
10. "Helix". URL: <http://www.efense.com/helix/>
11. "bash(1): GNU Bourne-Again SHell - Linux man page". URL: <http://www.die.net/doc/linux/man/man1/bash.1.html>
12. "CVE-2000-0573". URL: <http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0573>
13. "The Sleuth Kit & Autopsy: Forensics Tools for Linux and other Unixes". URL: <http://www.sleuthkit.org/>
14. Skoudis, Ed. "Malware Fighting Malicious Code". 2004
15. Guillermito. "Breaking a Stegonagraphy Software". May 6, 2003. URL: <http://www.guillermito2.net/stegano/camouflage/>
16. "SecurityFocus HOME Vulns Info: Multiple Linux Vendor rpc.statd Remote Format Stri". URL: <http://www.securityfocus.com/bid/1480/info/>

© SANS Institute 2004, Author retains full rights