



Helix 1.7 for Beginners

by BJ Gleason and Drew Fahey

Manual Version 2006.03.07

HelixManual@gmail.com

Dear Helix User:

Thank you for taking the time to read this document. This is a work in progress, and hopefully, you will find it useful. The document has grown rapidly, and aside from being a guide for the beginner; it also contains reference materials for most of the commands included on the Helix CD.

As Helix is being updated, this manual will be updated, and I have just been informed by Drew Fahey, it will be included on the Helix CD. Interim updates, as needed, will be published on the <http://www.e-fense.com> website. There are several things that are not quite completed yet, but hopefully they will be in the next few revisions.

If you have something you would like to contribute to this documentation project, or if you have any suggestions, corrections, compliments or complaints, please send them to me at HelixManual@gmail.com. Please include the version number of the manual from the cover of the document, and any related page numbers.

While I could work on this forever, and never have it quite the way I want it, there comes a time in which you can not procrastinate any longer and have to get it out the door. So for the near future, I plan to release an update of the manual approximately every 2 months while all the kinks are being worked out.

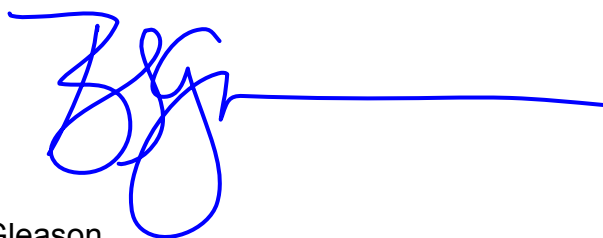
Some of the upcoming updates include:

- Documenting Windows command line tools
- More Examples
- More hands-on labs to allow users to practice and refine their skills
- Forensic Procedural Issues

I have learned so much from Helix, and from the many other forensic tools and websites that are out there. This is my opportunity to give back a little of what I have taken.

I'm looking forward to hearing what you think of this manual.










Respectfully,










BJ Gleason
07 March 2006, Seoul, Korea

A note on copyrights: The goal of this manual was to produce a single reference source for the tools included on the Helix CD. A large amount of the materials included in this manual have been copied from various websites, and attempts have been made to ensure that everything is properly documented and referenced. If you are the copyright owner of the material, and have issues with their inclusion in this document, please contact me at HelixManual@gmail.com, and we will resolve the issue. Thank you for your contributions, your patience, and your understanding.

Table of Content

Introduction	9
Thanks	9
Revision History	10
 Helix	12
What is Helix?	12
Why Helix Different	12
Operating Modes	12
The Windows Side	13
The Linux Side	14
 Getting Helix	16
Downloading	16
Checking the Download	16
Burning to a CD with Nero	17
Burning to a CD with Roxio EasyCD Creator	18
 Forensic Topics (Windows Side)	19
Write Protecting Media	19
Hardware Write Protection	20
Software Write Protection	21
Validating Write Protection Hardware / Software	21
 Helix Main Screen (Windows Side)	23
 Preview System Information	27
 Acquire a “live” image of a Windows System using dd	30
 Using dd	30
 FTK Imager	33
 Incident Response tools for Windows Systems	35

	Windows Forensic Toolchest (WFT) _____	37
	Incident Response Collection Report (IRCR2) _____	40
	First Responder's Evidence Disk (FRED) _____	42
	First Responder Utility (FRU) _____	45
	SecReport _____	47
	Md5 Generator _____	50
	Command Shell _____	52
	Rootkit Revealer _____	53
	File Recovery _____	57
	PuTTY SSH _____	63
	Screen Capture _____	64
	Messenger Password _____	65
	Mail Password Viewer _____	67
	Protect Storage Viewer _____	68
	Network Password Viewer _____	70
	Registry Viewer _____	71
	IE History Viewer _____	73
	Asterisk Logger _____	74



IE Cookie Viewer _____ 75



Mozilla Cookie Viewer _____ 76



Chain of Custody _____ 77



Preservation of Digital Evidence _____ 79



Linux Forensic Guide for Beginners _____ 79



Forensic Examination of Digital Evidence _____ 80



Browse contents of the CD-ROM and Host _____ 81



Scan for Pictures from a live system _____ 83



Exiting Helix _____ 85



Helix from the Command Line (Windows Side) _____ 86

Not Starting the GUI _____ 86

Starting a command shell _____ 86

Tools available from the Command Line _____ 87



Bootable Helix (Linux Side) _____ 90



Forensic Topics (Linux Side) _____ 91

Write Protecting Media _____ 91

Setting a USB device to Read/Write _____ 91

Using Helix in VMWare _____ 91

The Helix Filesystem _____ 92

Unionfs _____ 93

Copy-On-Write Unions _____ 93

Raid Essentials _____ 94

Understanding dd _____ 94

Traditional Acquisition (Dead Imaging) _____ 96

Imaging to a Netcat/Cryptcat Listener _____ 97

Imaging to a Samba Server _____ 97



Bootable Basics _____ **99**

F1 – Help and Cheat Codes _____ **100**

Default Options for the different boot modes _____ **107**

F2 – Language and Keyboard Layout Selection _____ **108**

F3 – Splash Mode Selection _____ **108**

F4 – Screen Resolution _____ **108**



Helix User Interface _____ **110**

The Desktop _____ **111**

Common Tasks _____ **112**

The File Manager _____ **113**



Helix Tools _____ **130**



Adepto _____ **133**



AIR: Automated Image and Restore _____ **137**



linen: EnCase Image Acquisition Tool. _____ **139**



Retriever _____ **141**



Autopsy _____ **144**



pyFlag _____ **147**



Regviewer _____ **149**



Hexeditor (GHex) _____ **151**



Xfce Diff _____ **152**



xhfs _____ **153**



ClamAV: ClamAV Anti Virus Scanner. _____ **184**



F-Prot _____ **186**



pyFlag _____ **188**



RAID-Reassembly _____ **188**



RAID-Reassembly _____ **189**



Partition-Info _____ **190**



Command Line Tools _____ **192**

2hash: MD5 & SHA1 parallel hashing. _____ 193

chkrootkit: Look for rootkits. _____ 199

chntpw: Change Windows passwords. _____ 202

dcfldd: dd replacement from the DCFL. _____ 204

e2recover: Recover deleted files in ext2 file systems. _____ 208

f-prot: F-Prot Anti Virus Scanner. _____ 209

fatback: Analyze and recover deleted FAT files. _____ 213

faust.pl: Analyze elf binaries and bash scripts. _____ 221

fenris: debugging, tracing, decompiling. _____ 222

foremost: Carve files based on header and footer. _____ 235

ftimes: A toolset for forensic data acquisition. _____ 238

galleta: Cookie analyzer for Internet Explorer. _____ 265

glimpse: Indexing and query system. _____ 266

grepmail: Grep through mailboxes. _____ 275

logfinder.py: EFF logfinder utility. _____ 277

logsh: Log your terminal session _____ 279

lshw: Hardware Lister. _____ 280

mac-robber: TCT's graverobber written in C. _____ 281

mac_grab.pl: e-fense MAC time utility. _____ 283

md5deep: Recursive md5sum with db lookups. _____ 284

outguess : Steganography detection suite. _____ 287

pasco: Forensic tool for Internet Explorer Analysis. _____ 291

rifiuti: "Recycle BIN" analyzer. _____ 292

rkhunter: Rootkit hunter. _____ 293

scalpel: Fast File Carver _____ 295

sdd: Specialized dd w/better performance. _____ 296

sha1deep: Recursive sha1sum with db lookups. _____ 300






sha256deep: Recursive sha1sum with db lookups. _____ 303

stegdetect: Steganography detection suite. _____ 306

wipe: Secure file deletion. _____ 309



Static Binaries _____ **314**

The Need for Static Binaries	314
 Windows	314
 Linux	315
 Solaris	316
 FAQ	317
Getting More Help	318
 Practice Labs	319
Lab 1a - Create an Image of a suspect Floppy Disk (Windows, Live Acquisition, dd)	320
Lab 1b - Create an Image of a suspect Floppy Disk (Windows, FTK Imager)	323
Lab 2 - Create a Floppy Disk from a suspect Image (Windows, FTK Imager)	328
Appendix 1 – Samba Forensic Config File	332
Appendix 2 – Linux Commands	333
References	339

Introduction

e-fense, Inc. developed Helix as an internal tool to provide the ability to acquire forensically sound images of many types of hard drives and partitions on systems running unique setups such as RAID arrays. It quickly grew to include many open source, and some closed source, tools for the forensic investigators at e-fense, and became the internal standard to image “live” systems as well as systems running RAID setups. This enabled us to easily deal with the issue in the corporate world that some systems could never be taken off-line to do a more traditional forensic acquisition. Since most corporate systems run Microsoft Windows, we developed a Windows functionality to facilitate the capture of live Windows systems’ volatile data, as well as to conduct a forensic acquisition while the system remained on-line.

Helix was first publicly released on 23 Nov 2003. Its popularity grew quickly, and Rob Lee started using it at SANS to teach the forensics track. Helix has been going strong ever since and has been downloaded countless times. Many Government agencies and Law Enforcement community across the globe have turned to Helix as their forensic acquisition standard due to its functionality and cost effectiveness (who can beat FREE)! The National White Collar Crime Center (NW3C) has chosen to use Helix to teach Law Enforcement Linux forensics on bootable CD’s.

The name Helix was chosen for no particular reason other than it fit with the sound of the name Linux. Also since forensics is a science the helix dna symbol seemed to apply. So Helix was born.

Helix is a work in progress and is not meant to be used by individuals without proper incident response and/or forensics training. While many complex commands are simplified with a GUI interface, it is the responsibility of the end user to know what these commands are doing so that you don’t inadvertently delete evidence, or so if called upon to testify, you don’t look like an idiot when you can’t explain your actions on the witness stand.

Helix is released under the terms of the GNU General Public License, version 2. Helix is distributed as is WITHOUT ANY WARRANTY; without the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

Thanks

This manual is a compilation of many sources. While most of this information can be found in other locations on the Internet, I wanted to pull them all together into a single reference manual for my students, and for anyone else who is interested in learning more about computer forensics.

The list of people who belong on this list is probably incomplete. If you think someone is missing, or if I have misidentified someone, please let me know...

Drew Fahey – Creator of Helix and the Helix Handbook

Klaus Knopper – Creator of Knoppix

Nirsoft – developers of many of the windows based tools

Jesse Kornblum, Harlan Carvey, Kevin Mandia, Chris Prosis, Matt Pepe, Brian Carrier

And to the countless others who have contributed their time and efforts developing these tools.

Revision History

Version	Details
2005.03.07	Release delayed to coincide with the new Helix 1.7 release Fixed FRU description to use the Forensic Server Project Updated some of the screen shots in Linux sections Added descriptions for the major tools on the Linux side. Added missing appendixes Added icons for each tool Various spelling corrections Added "Forensic Topics" for both the Windows and Linux Sides. Added Lab1b - Create an Image of a suspect Floppy Disk (Windows, FTK Imager) Added hardware and software media write protection features. Continuing to update references Added new Helix Logo Added "Getting Helix" Section Added "Using Helix in VMWare" Section Added "Static Binaries" Section Added FAQ Added Helix Boot options, cheat codes Added individual package descriptions for Linux side Added Lab 2 - Create a Floppy Disk from a suspect Image (Windows, FTK Imager)
2005.12.27	First Public Release
2005.10.05	Proof of concept to Drew Fahey

Thanks to all the Debuggers!

Harlan Carvey
Chris Cohen
Drew Fahey
Ian Marks

Advice to Beginners

Helix is a very powerful tool. But with great power comes great responsibility, and as a potential forensics investigator, it is your responsibility to learn how to use this tool properly. It is expected that if you have downloaded and created a bootable Helix disk, and that you have an interest in digital forensics.

But just as you can use a hammer to build a house, you can not build a house just using a hammer. To successfully build a house, you need architects, lawyers, construction workers, many tools, supplies, and inspectors. The same is true in the field of digital forensics. Before you examine any system, you need to make sure that you have permission to examine that system. You need to know the legal aspects of collection, documentation, and preservation of digital evidence. You need to know how to use the tools of the trade (such as those on the Helix CD).

Simple mistakes and good intentions can completely destroy digital evidence. It is strongly recommended that aspiring investigators learn about digital forensics, and practice on controlled systems before attempting to collect evidence from a real system.

Some recommended books on digital forensics include:

- Carrier, B. (2005). *File system forensic analysis*. Boston, Mass. ; London: Addison-Wesley.
- Carvey, H. A. (2005). *Windows forensics and incident recovery*. Boston: Addison-Wesley.
- Casey, E. (2004). *Digital evidence and computer crime : forensic science, computers, and the Internet* (2nd ed.). Amsterdam ; Boston: Academic Press.
- Farmer, D., & Venema, W. (2005). *Forensic discovery*. Upper Saddle River, NJ: Addison-Wesley.
- Jones, K. J. (2005). *Real digital forensics : computer security and incident response*. Indianapolis, IN: Addison Wesley Professional.
- Proise, C., & Mandia, K. (2003). *Incident response and computer forensics* (2nd ed.). New York, New York: McGraw-Hill/Osborne.
- Schweitzer, D. (2003). *Incident response : computer forensics toolkit*. Indianapolis, IN: Wiley.
- Solomon, M., Barrett, D., & Broom, N. (2005). *Computer forensics jumpstart*. San Francisco: Sybex.
- Vacca, J. R. (2005). *Computer forensics : computer crime scene investigation* (2nd ed.). Hingham, Mass.: Charles River Media.

I would also recommend that you create a home lab in which to practice with these tools. I recommend 2 systems, running Windows 2000 or XP, with a network connection between them, either via a switch or a crossover cable. Since some of these tools transfer data via the network, make sure you disable any firewalls, such as the one built-in to XP service pack 2, which can interfere with network connections. I would label one machine as "Suspect", and at the other as "Forensic". To experiment with disk imaging, I would recommend having machines with floppy disks, and the suspect system should have a small hard drive (4 gig) or less, since copying larger drives over a network can take a very long time, and require a lot of space on the forensic system. For the forensic system, I would recommend having two hard drives (or at least two partitions) – one for the operating system, and one for the collected evidence.



Helix

What is Helix?

Helix is a customization of the standard **Knoppix**¹ distribution. As such it owes everything to the work already done by Klaus Knopper and borrows heavily from work done by several individuals at www.knoppix.net. In fact, knoppix.net is the first place to go if you're looking for information on how to customize. There are many other LiveCD distributions available that have been around longer than Helix, but many of them are no longer maintained or are not updated on a frequent enough basis. Many of the original ideas for Helix originated from these distributions: Knoppix, Knoppix-STD², FIRE³, Morphix⁴, Insert⁵

Why Helix Different

Helix is a heavily modified version of Knoppix. However, while there are many variants on the original Knoppix, Helix is different in that much of the code has been tweaked for forensic purposes. Some of the code has been compiled from scratch and it is a distribution dedicated strictly for incident response and forensics.

Some of the major changes Helix incorporated into Knoppix are:

1. Helix will NEVER use swap space found on a system - even if forced.
2. The Helix automounter will set up drives it finds but will force a mount point to be ro, noatime, noexec, nodev, noauto, user
3. Helix sees all the filesystems identified by Knoppix (ext2, ext3, vfat, ntfs), but will also see xfs, resier, and jfs and more.
4. Helix incorporates as many open source forensics/incident response tools that could be found.
5. Added capability for "Knock and Talks." This feature allows a parole officer to preview a system for graphic images that may violate a parole.
6. Windows-side executable environment.
7. Added an overlay file system to allow writes to the CD.
8. Updated at least every 3 months to keep current.

Operating Modes

Helix operates in two different modes – Windows and Linux.

Helix is a forensically sound bootable Linux environment much like Knoppix, but a whole lot more. The "other side" of Helix, a Microsoft Windows executable feature, contains approximately 90 MB of incident response tools for Windows. The rationale behind this was that a majority of incidents require interaction with a live Windows system, the dominant operating system in the computer market.

¹ <http://www.knopper.net/knoppix/index-en.html>

² <http://www.knoppix-std.org/>

³ <http://fire.dmzs.com/>

⁴ <http://www.morphix.org/modules/news/>

⁵ http://www.inside-security.de/insert_en.html

As such Helix was broken down into the live response side and the bootable Linux OS side.

Windows: In the Windows Mode, it runs as a standard windows application used to collect information from “live” (still turned on and logged in) Windows system. It should be noted, that when a target system is live, its state is constantly changing. Not matter what tools you use on a live system, you will disturb the state of the live system – even doing “nothing” changes the state of a live system, since it is still running the operating system. However, since turn off the system can result in the lost of potentially important forensics information, the tools can be used to collect volatile information. It can also be used to collect information off of systems that can not be turned off, such as servers and other critical resources that can not be turned off. Finally, the Windows side of Helix can be used a portable forensic environment since it provides access to many windows based forensic utilities.

Linux: In the Linux mode, it is a bootable, self-contained operating system that can be used for in-depth analysis of “dead” (powered-off) systems. When Helix boots, it runs entirely off the CD, and only mounts the hard drives in read-only mode, so they can not be modified. Aside from the standard Linux tools, this side includes numerous forensic analysis tools that can use to examine the target system.

The Windows Side

For a live response in any Windows environment, one can simply insert the Helix CD and “explore” the directories on the CD for the needed environment binaries. The binaries are static so they will run off the CD without any need for any additional libraries and or files. This makes a perfect trusted CD for an incident response where you cannot rely on the systems tools or programs.

The other option you have, at least in a Windows environment, is the Helix.exe. It will normally automatically load the menu if auto run is not disabled. Running Helix.exe or relying on auto run will bring up the Helix Windows environment in which several options become available. Of course these options are not new; a user could duplicate them manually. The Helix environment simply puts the options together in a forensically safe, easy to use, manner.

Let’s start by dissecting the live response side of Helix. The cornerstones of the live response side are the tools. Helix contains static binaries for Linux, Solaris, and Windows using GNU utilities and Cygwin tools. There are several other tools to include George Garners Forensic Acquisition Utilities suite, Sysinternal’s tools, Foundstone’s open source tools, the Windows Debugger, the Windows Forensic Toolchest, and many more. All of these tools have been tested and placed into Helix using a GUI.

The Helix GUI will only operate within a live Windows environment. It has been tested on Windows 98SE, Windows NT4, Windows 2000, and Windows XP. There are slight differences in running the Helix CD on each. The most important note to remember is that since Windows is required to run the above interface, many DLL files will be used by Helix from the operating system. This is not a problem however as long as you are aware of it. Some of the DLL files that will be used are shown in the table to the right.

These DLL files were not included on the CD because of the nature of the various versions of Windows. There is no way with the current build to have the Helix executable not access the built in Windows DLL files. The incident response / forensics tools included on the Helix CD are self sufficient - most of them will use their own libraries and not the libraries and/or file from the running system.

Helix can and will use other DLL files depending upon the system it is running. The other DLL files that are used are hooks into the specific hardware. So you must be aware of other files you may touch while using Helix in a live environment.

Some of the newer features in Helix on the Windows side include the ability for user input to some of the major tools like Windows Forensic Toolchest. Prior to Version 1.5 the input was static and could not be changed without remastering the Helix CD.

While many of the more common tools can be accessed via the Helix graphical user interface, many more tools can be accessed via a forensic command tool. Many of these tools are listed on the next page, however, since Helix is updated often, you can always check the /IR folder on the Helix CD to see what tools are included.

The Linux Side

One of the greatest benefits of Helix is the bootable environment. Helix will boot on all x86 architectures which make up a majority of the computers in the world. It is for this reason that Helix for the immediate future will remain on a CDROM. Almost every computer in the world has a CDROM, but most do not have DVD's, etc. While derivatives like Helix USB will be forthcoming, it is most stable on the CD platform. Helix, like Knoppix, will boot into a self contained Linux environment. This environment has been tweaked for forensic purpose. While there are many distributions of Knoppix such as Knoppix-STD, Helix only concentrates on Incident Response and Forensics.

DLL Files used by Helix in Windows

```
C:\WINDOWS\system32\ADVAPI32.dll
C:\WINDOWS\system32\Apphelp.dll
C:\WINDOWS\system32\comctl32.dll
C:\WINDOWS\system32\comdlg32.dll
C:\WINDOWS\system32\CRYPT32.dll
C:\WINDOWS\system32\DNSAPI.dll
C:\WINDOWS\system32\dsound.dll
C:\WINDOWS\system32\GDI32.dll
C:\WINDOWS\system32\IMAGEHLP.dll
C:\WINDOWS\system32\kernel32.dll
C:\WINDOWS\system32\KsUser.dll
C:\WINDOWS\system32\midimap.dll
C:\WINDOWS\system32\MSACM32.dll
C:\WINDOWS\system32\msacm32.drv
C:\WINDOWS\system32\MSASN1.dll
C:\WINDOWS\system32\MSCTF.dll
C:\WINDOWS\system32\mslbui.dll
C:\WINDOWS\system32\msvcrt.dll
C:\WINDOWS\system32\NETAPI32.dll
C:\WINDOWS\system32\ntdll.dll
C:\WINDOWS\system32\ole32.dll
C:\WINDOWS\system32\OLEAUT32.dll
C:\WINDOWS\system32\oledlg.dll
C:\WINDOWS\system32\OLEPRO32.DLL
C:\WINDOWS\system32\rasadhlp.dll
C:\WINDOWS\system32\RPCRT4.dll
C:\WINDOWS\system32\Secur32.dll
C:\WINDOWS\system32\SETUPAPI.dll
C:\WINDOWS\system32\SHELL32.dll
C:\WINDOWS\system32\SHLWAPI.dll
C:\WINDOWS\system32\USER32.dll
C:\WINDOWS\system32\uxtheme.dll
C:\WINDOWS\system32\VERSION.dll
C:\WINDOWS\system32\wdmaud.drv
C:\WINDOWS\system32\WINMM.dll
C:\WINDOWS\system32\WINSPOOL.DRV
C:\WINDOWS\system32\WINTRUST.dll
C:\WINDOWS\system32\WS2_32.dll
C:\WINDOWS\system32\WS2HELP.dll
C:\WINDOWS\system32\WSOCK32.dll
```

Helix Tools

DRIVE ACQUISITION:

CYGWIN Tools: (dd, md5sum, shalsum, tee, split)
FAU by George Garner: (dd, nc, md5sum, volume-dump, wipe)
GNU Tools: (dd, md5sum, shalsum, tee, split)

INCIDENT RESPONSE TOOLS:

Windows NT: (cmdnt.exe, doskey.exe, ipconfig.exe, ...)
Windows 2000: (cmd2k.exe, doskey.exe, netstat.exe, net.exe, ...)
Windows XP: (cmdxp.exe, doskey.exe, ipconfig.exe, ...)

OTHER TOOLS:

Cygin: (version 2.427)
Somarsoft: (DUMPEVT, DUMPSEC, dumpreg)
wft: (Version: v1.0.03 (2003.09.20) by Monty McDougal)
getinfo: (Version: 3.02.10 by Alexander Kotkov)
Microsoft Debugging Tools: (Version: 6.2)
GNU-Win32 Static-Binaries
Linux Static-Binaries
Solaris Static-Binaries

FOUNDSTONE'S OPEN SOURCE TOOLS:

AFind	Audited	CIScan	DACLchk	DSScan	FPipe	FileStat
HFind	MessengerScan	MyDoomScanner	NetSchedScan	RPCScan2	SFind	SNScan
SQLScan	SuperScan4	Vision	attacker	bintext	bopping	ddosping
filewatch	galleta	pasco	rifiuti	showin	sl	trout

SYSINTERNALS OPEN SOURCE TOOLS:

AccessEnum	DiskView	EFSDUMP	Filemon	HOSTNAME	LogonSessions
NTFSINFO	Regmon	TDIMON	TOKENMON	Tcpview	autoruns
autorunsc	livekd	procexp	psexec	psgetsid	pskill
psloglist	pspasswd	pssshutdown	pssuspend	strings	tcpvcon

NTSECURITY OPEN SOURCE TOOLS:

browselist	dumpusers	efsview	etherchange	filehasher	gplist
gsd	listmodules	lns	macmatch	periscope	pmdump
promiscdetect	pstoreview	winfo	winrelay		

PERL TOOLS BY HARLEN CARVEY:

ads	bho	finfo	keytime	rights
service	share	sigs	ver	windata



Getting Helix

Helix is available as a free downloadable ISO image from <http://www.e-fense.com/helix/>. From the download page <http://www.e-fense.com/helix/downloads.php>, you will have a choice of mirrors sites from where you can download the file. For the best response, and fastest download, try to pick a mirror site close to you.

Downloading

While it is possible to download the image file with your browser, it is recommended that you use a download accelerate such as Download Express (<http://www.metaproducts.com/DE.html>), Download Accelerator Plus (<http://www.speedbit.com/>), or GetRight (<http://www.getright.com/>) to ensure that the large file, which is about 700MBs, downloads properly. These utilities can resume downloads that are interrupted, and can segment large files and simultaneously download the different segments for faster transfers.

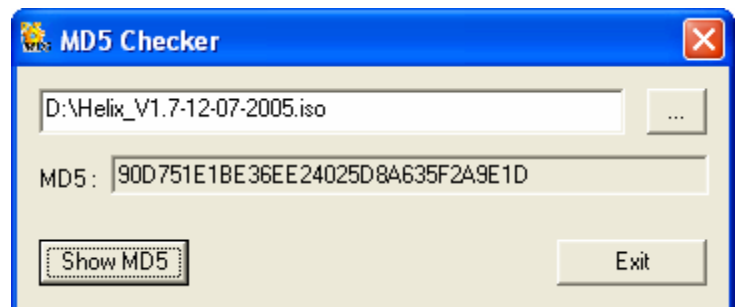
Checking the Download

After it is downloaded, but before it is burned to a CD, you should check that the file has been downloaded properly. On the download page is an MD5 signature of the file. For the Helix 1.7 12-07-2005 release, the MD5 value is 90d751e1be36ee24025d8a635f2a9e1d. You should use a Windows or DOS based MD5 generator to make sure the file you downloaded has the same signature. If even a single bit is different, a different MD5 will be generated. If the values are the same, great, you can now burn it to a CD. If the values are different, you should try downloading the file again. Some users have reported having to download the file several times before the MD5 values matched.

A handy little MD5 tools for Windows is WinMD5 from Softgears. <http://www.softgears.com/WinMD5.html>. It's free, and doesn't have to be installed. Simply unzip and execute the WinMD5.exe file.

In the MD5 windows, you can browse using the "... " button to find where the downloaded file is located.

Click on the "Show MD5" button to generate the MD5 signature for the file. Because of the size of the file, it may take a moment for the value to display.

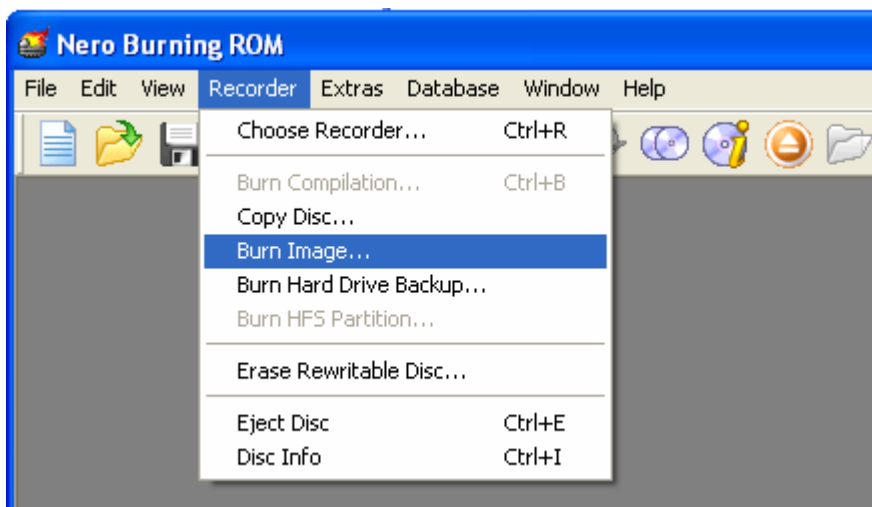


This value should match the MD5 value from the website. If the values are the same, you are now ready to burn the CD.

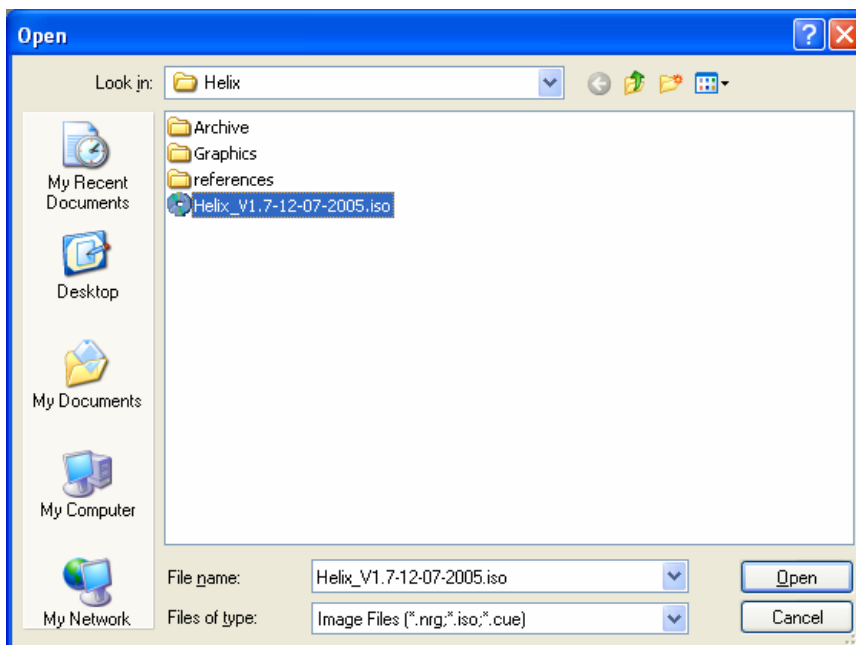
Burning to a CD with Nero

The Helix .ISO file is a copy of the Helix CD. You should not just copy this file to a CD, you have to burn a CD from the image contained inside the .ISO file. To do this, you would normally select an option from your CD burner software called something like “Burn Image...” or “Create CD from Image”.

With Nero Burning ROM, select the “Recorder” from the menu, and then select “Burn Image...”.

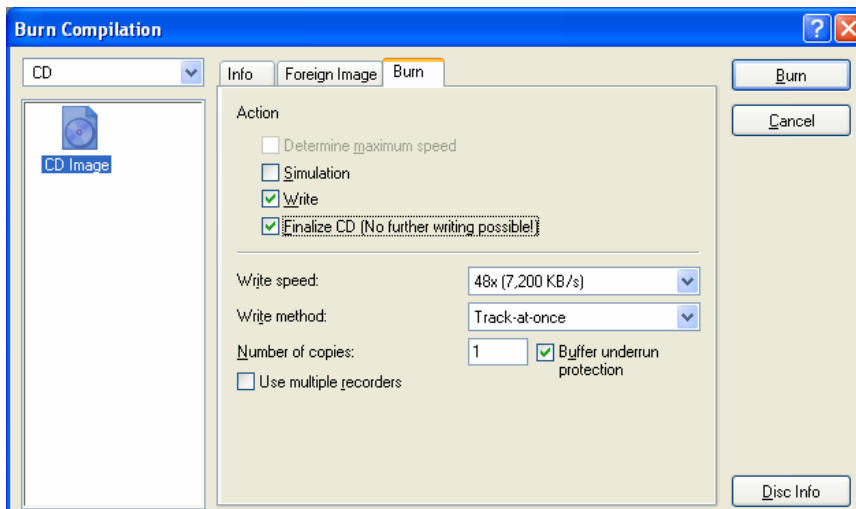


Select the .ISO image that you downloaded, and click “Open”.



At this point, just click “Burn” and Nero will create the Helix CD from the .ISO image. Once the CD is ejected at the end of the burn process, you will have a bootable Helix CD.

For more information on Nero, visit <http://www.nero.com/>.



Burning to a CD with Roxio EasyCD Creator

The Roxio EasyCD Creator can create a CD from an ISO file. The installation process automatically associates .ISO files to the EasyCD Creator. For more information, visit: <http://www.roxio.com>.

From Windows Explorer, right click on the Helix .ISO file that you downloaded, and click Open.

In the CD Creation Setup dialog box, select Disk at Once from the Write Method section for the best results.

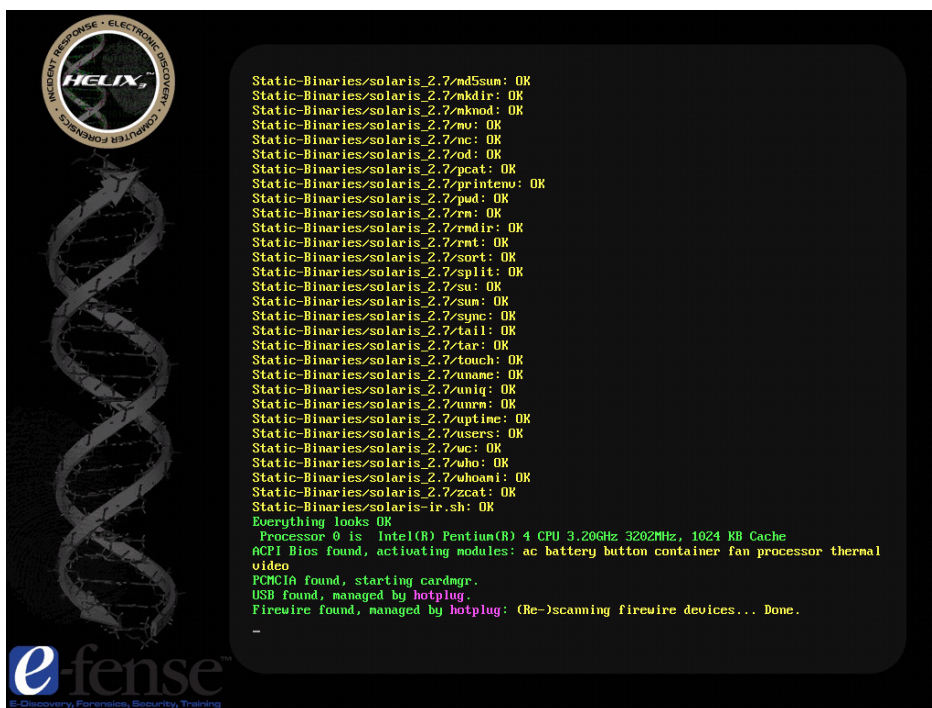
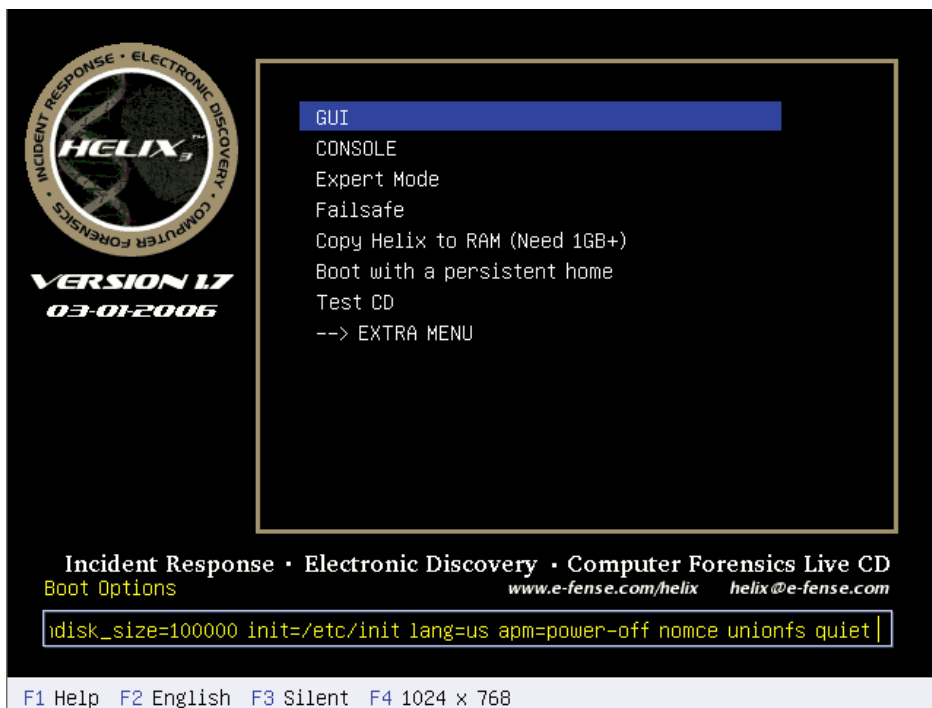
Click OK to write the image to the CD.

Checking the CD

Once the CD is burned, it can be checked again. By place in the CD in a bootable CDROM, once the Helix boot menu appears, you can use the TestCD option to make sure the CD was created properly.

Depending on the speed of the CD, they can take some time to complete.

During the boot process, Helix will check every file to verify it's MD5 value. When it is done checking, it will display the message "Everything looks OK", and then continue booting into the Helix environment.





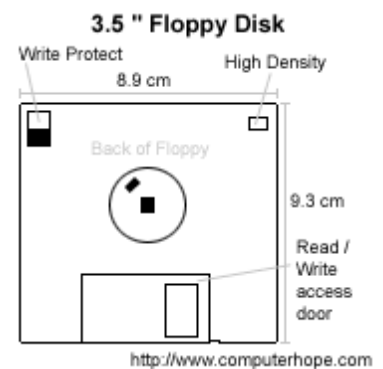
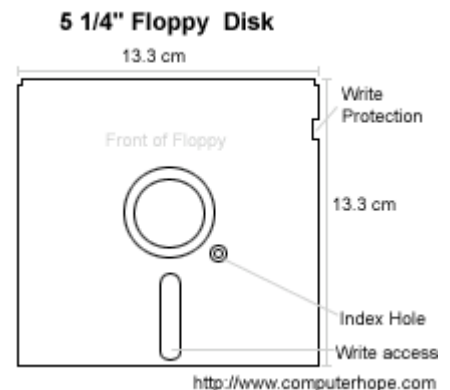
Forensic Topics (Windows Side)

Write Protecting Media

To ensure that digital evidence media is not modified, before it is placed into a system for duplication, it should be set to "Read Only", "Locked" or "Write Protect", to prevent accidental modification. By default, Windows set all devices as read/write, so they can be easily modified. The bootable Linux side of Helix mounts all media as read-only.

Some media has built-in hardware write protect options, which should be used. If the media doesn't have a write protect options, other means should be used (hardware write protect blockers are preferred over software).

5.25 Floppy Disk: 5.25 floppy disks have a small notch. If that notch is open, the disk can be modified. To write-protect the floppy, cover the notch with opaque tape, such as electrical tape, since some drives use mechanical or optical detection of the notch. Image taken from Computer Hope (2006b).



3.25 Floppy Disk: 3.25 floppy disks have a small window with a slider in it. If the slider blocks the hole, the disk can be modified. Moving the slide so that the hole is exposed will write-protect the disk. Some manufacturers will physically remove the slider from software distributions to prevent them from being accidentally overwritten. Image taken from Computer Hope (2006a).

Zip Drives: The Iomega Zip drives have software write-protect option, but the Zip software must be loaded, and it probably modifies the content of the drive. However, the latest 750MB USB drives can not write to 100MB disks, so they are write-protected when used in the 750MB drives. If using a USB Zip drives, it is recommended to use a hardware write-blocker such as Digital Intelligence's UltraBlock Forensic USB Bridge.



MMC/SD (Multimedia Card / Secure Digital): Some MMC/SD cards have a write protect switch on the side. They are write-enabled by default. Image taken from The Living Room (2006).



Erasure Prevention Switch

Sony Memory Stick: The Sony memory stick has a lock switch on the bottom. It is write-enabled by default. Image taken from MemoryStick.com (2006)

Flash Memory Cards: It is recommended to use a hardware write-blocker such as Digital Intelligence's Forensic Card Reader which can read the following multimedia card formats: Compact Flash Card (CFC), MicroDrive (MD), Memory Stick Card (MSC), Memory Stick Pro (MS Pro), Smart Media Card (SMC), xD Card (xD), Secure Digital Card (SDC), and MultiMedia Card (MMC).

USB Thumb Drive: Some USB thumb drives have a write-protect switch built into them. Sometimes they are recessed and difficult to change. Many USB thumb drives so not have a write-protect switch. Image taken from NuLime.com (2006).

USB Devices: It is recommended to use a hardware write-blocker such as Digital Intelligence's UltraBlock Forensic USB Bridge.

IDE / SATA / SCSI: It is recommended to use a hardware write-blocker such as Digital Intelligence's Ultrablock.



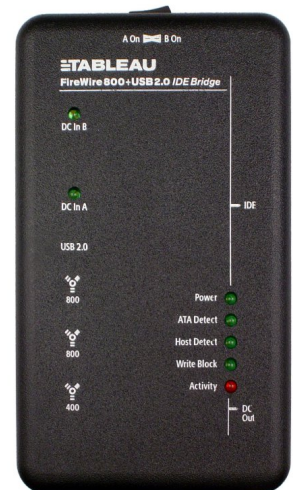
Other devices: If the device doesn't have a built-in write protect switch, then the device should be protected via hardware (preferred) or software. The bootable side of the Helix CD mounts all drives as write-protected by default.

Hardware Write Protection

There are numerous hardware write-blockers available from many sources. These are designed and tested to allow you to mount and access the media without modifying it. There are write-blockers for three primary devices: Hard Drives, Flash Drives, and USB devices. While the three devices illustrated are from Digital Intelligence (<http://www.digitalintelligence.com/>), there are several companies producing similar devices. The software for these devices are updated regularly, users should make sure they are using the most recent version.

Hard Drives: Ultrablock

"The UltraBlock-IDE is a FireWire/USB to Parallel IDE Bridge Board with Forensic Write Protection. The UltraBlock-IDE can be connected to your laptop or desktop using the FireWire-A (400 Mb/s), the new FireWire-B (800 Mb/s), or the USB 1.X/2.0 interface. The UltraBlock-IDE is provided with write protection enabled by default. By connecting a suspect drive to the UltraBlock-IDE, you can be certain that no writes, modifications, or alterations can occur to the attached drive" (Digital Intelligence, 2006b). There are also versions of the Ultrablock available for SCSI and SATA.



Flash Drives: Forensic Card Reader

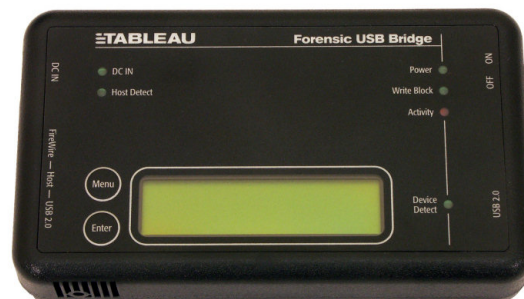
"The UltraBlock Forensic Card Readers are available as a set: One Read-Only and One Read-Write. The Read-Only unit should be used for forensic acquisition of information found on multimedia and memory cards. The Read-Write unit is included to provide the ability to write to memory cards for testing or validation. The Forensic Card Readers can be connected directly to a USB 2.0 (or



USB 1.x) port on your workstation or laptop” (Digital Intelligence, 2006a).

USB Devices: Forensic USB Bridge

“The UltraBlock USB Write Blocker supports USB2.0 High-Speed (480 Mbit/s), USB 1.1 Full-Speed (12 Mbit/s) and Low-Speed (1.2 Mbit/s) devices conforming to the USB Mass Storage "Bulk-only" class specification. The UltraBlock USB Write Blocker works with USB thumb drives, external USB disk drives, even USB-based cameras with card-reader capability.” (Digital Intelligence, 2006c)



Software Write Protection

While not completely tested from a forensic standpoint, Windows XP, service pack 2, Microsoft has created a way to write-protect USB devices (Hurlbut, 2005). This method requires making changes to the registry, however, there are utilities available to allow these changes to be done automatically. For more information on this, see “How to disable the use of USB storage devices” at <http://support.microsoft.com/default.aspx?scid=kb;en-us;823732>

Validating Write Protection Hardware / Software

This process is based on the National Center for Forensic Science (NCFS) 5 step validation process for testing write protection devices (Erickson, 2004). It was originally designed to test the Windows XP SP2 USB software write blocker, but has been adapted to test any hardware and/or software write blockers.

Step #1 – Prepare the media

- a) Attach the storage media you will be testing with to your forensic workstation in write-enabled mode.
- b) Wipe the media - validate that this has been successful.
- c) Format the media with a file format of your choosing.
- d) Copy an amount of data to the media.
- e) Delete a selection of this data from the media.
- f) On the desktop of your forensic workstation create 3 folders. Call these Step-1, Step-2 and Step-5.
- g) Image the media into the Step-1 folder and note the MD5 hash.

Step #2 – Testing the media

- a) Remove and then replace the testing media into your forensic workstation.
- b) Copy some data to the media.
- c) Deleted a selection of this data from the media.
- d) Image the media into the Step-2 folder and note the MD5 hash.
- e) Validate that this hash value is "different" to that produced in Step #1.

Step #3 – Activate the write blocking device

- a) Remove the media from your forensic workstation.
- b) Attach and/or activate the write protection device.
- c) Follow any specific activation procedures for the specific blocker.

Step #4 – Test the write blocking device

- a) Insert the media into your forensic workstation.
- b) Attempt to copy files onto the media.
- c) Attempt to delete files from the media.
- d) Attempt to format the media.

Step #5 – Check for any changes to the media

- a) Image the media into the Step-3 folder and note the MD5 hash.
- b) Validate that this MD5 hash is the "same" as the MD5 hash from Step #2.



Helix Main Screen (Windows Side)

Helix operates in two modes, a Windows side, and a bootable Linux environment. The windows side can be used to perform a preliminary evaluation to see if there is any that warrants further investigation of the system. It can also be used to capture system that can not be turn off or taken offline for the extended periods of time it take to perform a forensic duplication.

Note: When performing a live preview of a system, many of the actions taken can and will modify information on the suspect machine. This method should only be used when the system can not be taken offline.

Running the Windows GUI Interface

If the CD autorun features is enabled (which is the Windows default), a Helix warning window should appear. If autorun is disabled, you can run Helix by double clicking on the helix.exe file on the CD.



The user can select the default language that Helix will use via the drop-down box. English is the default, but French, German, and Italian are also available.

To use Helix, you should first read the warning. As it has been pointed out several times in the manual, using Helix in a live environment will make changes to the system – that is one of the inherent risks in a live-response situation. But remember, just inserting this CD has modified the system – even just leaving the system turned on is modifying the system. So what do you do? It boils down to this – will you lose more evidence by using this tool or by turning off the system? You need to make your decision, and when ready, press the “I Agree” button to continue. Once the user accepts the agreement, the main screen will appear.



Users can select any of these options by clicking on the associated icons.

This Main screen doesn't behave as a standard window – it doesn't show up in the taskbar, and you can not switch to it via the <ALT><TAB> key sequence. Helix does place an icon in the system tray which can be used to access the program. To bring the Helix main screen to the front, you can double-click on the icon, or right-click, and select Restore. Other options on the right-click menu include Minimize and Exit.



The main screen provides examiners with six main options to examine the system under investigation. These options are described below.



Preview System Information

This choice will provide you with the basic information of the system. It includes Operating system version, network information, owner information, and a summary of the drives on the system. In addition, there is a second page that will show a list of running processes.



Acquire a “live” image of a Windows System using dd

This option will allow the investigator to make copies of hard drives, floppy disks, or memory, and store them on local removable media, or over a network.



Incident Response tools for Windows Systems

This option provides access to 20 tools, all of which can be run directly from the CDROM. Once you click the icon, a small triangle will appear, next to the icon. Clicking on this small triangle will provide access to the others pages of tools.



Documents pertaining to Incident Response, Computer Forensics, Computer Security & Computer Crime

The option provides the user with access to some common reference documents in PDF format. The documents include a chain of custody form, preservation of digital evidence information, Linux forensics Guide for beginners, and forensic examination for digital evidence guide. These documents are highly recommended, and the investigator should review them before attempting any forensic examination.



Browse contents of the CD-ROM and Host OS

This is a simple file browser that will provide the investigator with information about the selected file. It will display the filename, created, accessed and modified dates, Attributes, CRC, MD5 and the file size. Due to the nature of the windows operating system, the first time you select a file; it will display the access date of the last access. If you select the same file again, it will display the date and time of the previous access. This is a feature of the windows operating system, and can not be prevented. This is one of the problems with examining a live system – the investigator’s actions may modify the system.



Scan for Pictures from a live system

This tool will allow the investigator to quickly scan the system to see if there are any suspect graphic images on the suspect system. Many different graphic formats are recognized, and displayed as thumbnails.

Menu Bar

In addition to the icons, all the features are directly accessible via the Helix menu bar.

File – Allows the user to exit the Helix application

Quick Launch – Allows the user to launch a command tool or the FTK Imager software

Page – Allows the user to jump directly to any of the utility screens

Help – Displays information about the program, and the license agreement

Note: Since these tools run directly off the CDROM, and most CDROM spin down when not in use, when you click on an icon, it may take a moment for the CDROM to spin up before there is a response from the application.

Note: All the tools run at the same level as the current logged in user. Normal users may have many restrictions on them that prevent some of these tools from running. Accessing the system using the Administrator account will provide the most access.



Preview System Information

The screenshot shows the HELIX v1.7 (3/7/2006) application window. The title bar includes the text "HELIX v1.7 (3/7/2006)" and standard window controls. The menu bar contains "File", "Quick Launch", "Page", and "Help". The main interface has a sidebar on the left with icons for various system components, with the first icon (a computer monitor) highlighted by a red box and an orange arrow. The main area is titled "SYSTEM INFORMATION" and displays the following data:

Operating System:
Windows XP Service Pack 2

Owner Information:
Owner:
Organization:
Admin: No
Admin Rights: Yes

Network Information:
Host: TAL_MC
User: bj_gleason
IP: 192.168.1.100
NIC: 005056c00008
Domain: TAL_MC

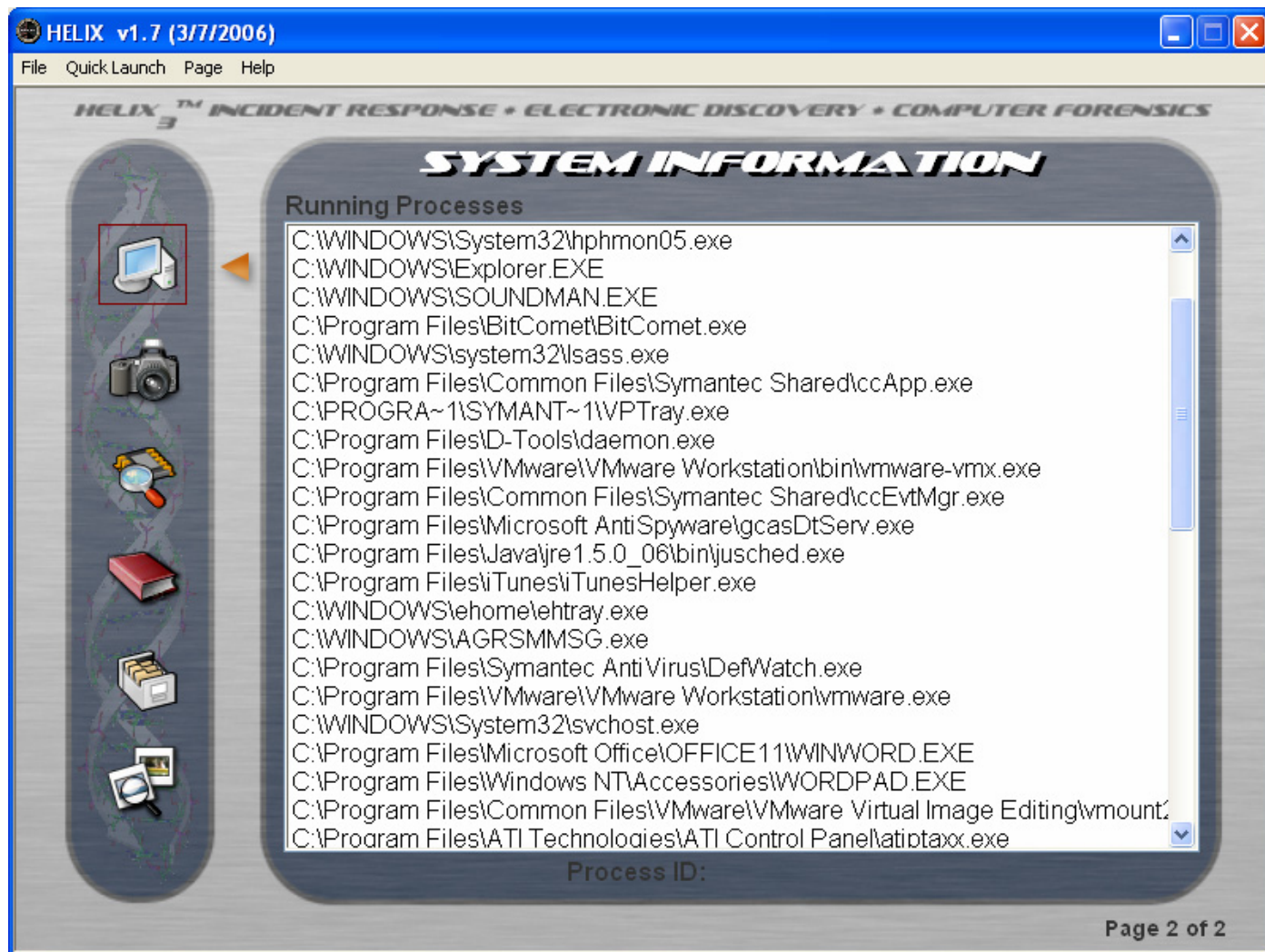
Drive:	Label:	Type:	Size:
C:\	(Logical drive)	ERROR	40000 MB
D:\	(Logical drive)	ERROR	49996.3 MB
E:\	(Logical drive)	NTFS	143890.9 MB
F:\	(Logical drive)	NTFS	199996.6 MB
G:\	(CD/DVD-ROM drive)		
H:\	(CD/DVD-ROM drive)		
I:\	(Logical drive)	CDFS	695.1 MB
J:\	(Removable drive)		
K:\	(Removable drive)		
L:\	(Removable drive)		
M:\	(Removable drive)		

Page 1 of 2

This screen displays some general information about the system being investigated. Some points of interest:

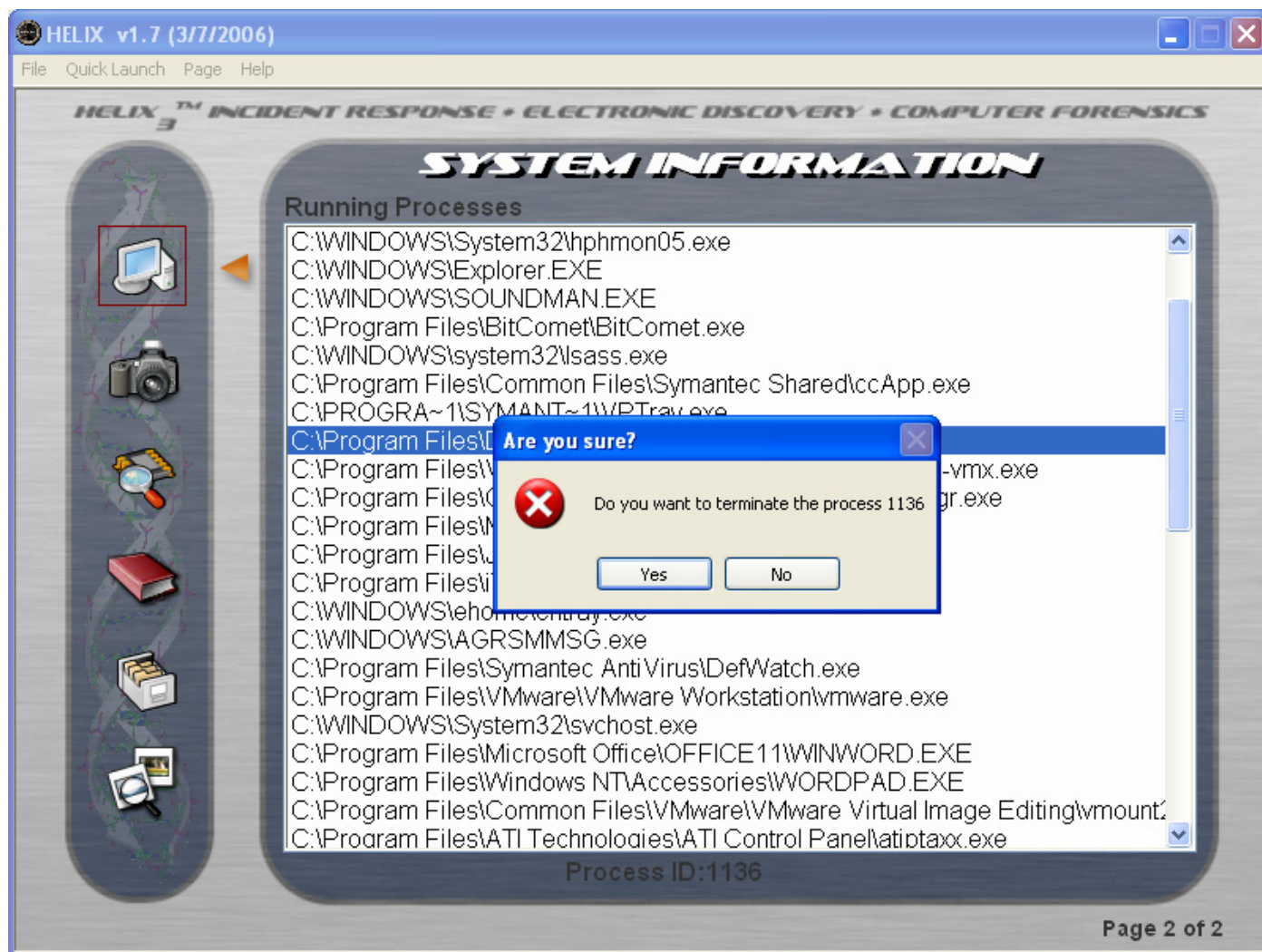
- “Admin:” tells us if the current user is the administrator (good security practice to change the name of the administrator account)
- “Admin Rights” tell us of the current user has administrator privileges.
- “NIC:” is the MAC address of the network card. If this value is “000000000000” it indicates that the network card is in promiscuous mode, and could be capturing all the network traffic on the system.
- “IP:” is the current IP address – this could change if the system is set up for DHCP.
- Drives name listed with no additional information (such as A:\, E:\, and G:\ in the example above) typically indicate removable drives with no media inserted.

Clicking on the small triangle next to the Preview Icon will display the second page of information, which lists the running processes. Clicking the triangle will flip the between the two pages of information.



In addition to displaying all the running processes in memory, double-clicking on any process will provide the user the option to terminate the selected application.

Care should be taken, and the investigator should be sure they are terminating the proper process. Terminating the wrong process could result in system damage and loss of forensic evidence.



FAQ: Why don't we just use the built in "task manager" to display this information? If the system has been hijacked by a rootkit, or some other malicious program, it is possible that the Windows Task Manager has been modified to not display the malicious code. Since Helix is running from the CD, it can not be modified, and should be able to display all the programs currently running on the system.



Acquire a “live” image of a Windows System using dd



There are two tools provided to acquire images of physical memory or disk drives. On the first page, there is a graphical front-end to the command line version of dd, a common disk duplication utility. On the second page, the investigator has access to the FTK Imager from AccessData. The dd utility can capture physical memory and drives, while FTK Imager can only acquire drives. In addition, dd can image over a network, while FTK imager can only image to local devices. Clicking on the small triangle next to the Acquisition Icon will display the FTK Imager. Clicking on the triangle will flip the between the two image acquisition tools.



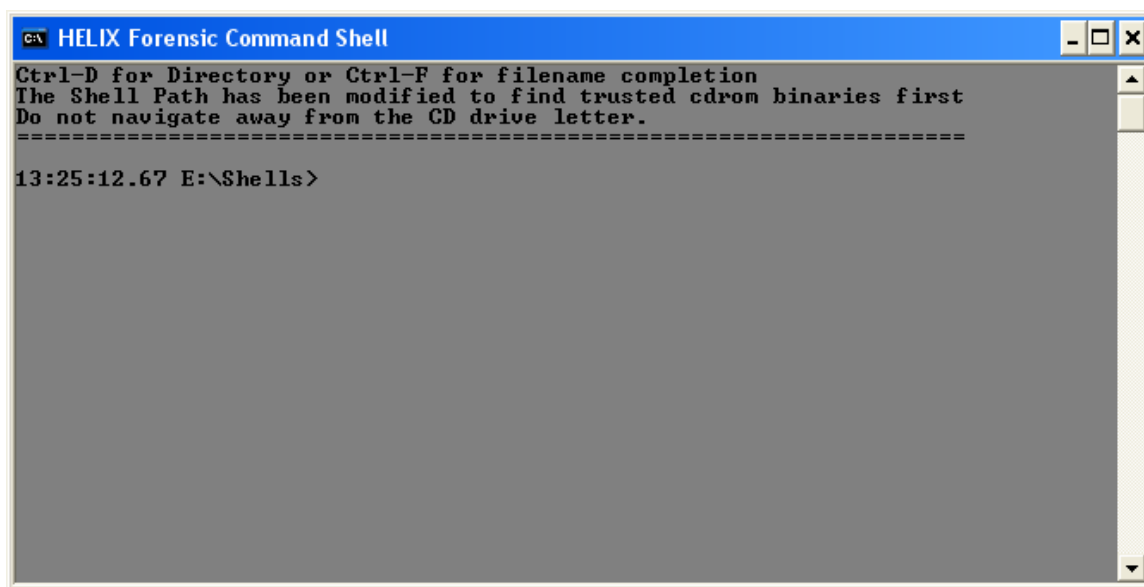
Using dd

The source field includes a drop-down box for the investigator to select any drive in the system. The destination can be a local removable drive, network drive or a netcat listener. The image name is the user chosen name, and the standard extension is “.dd”.

The Options include:

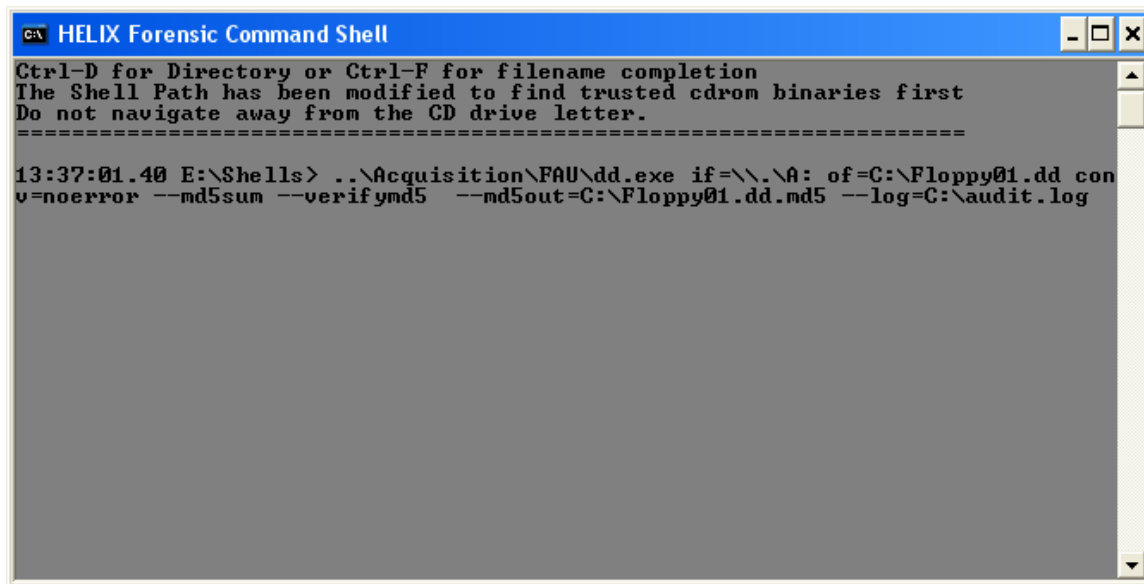
- Attached/Shared: check this option to save the image to a local drive, or a network share.
- NetCat: check this option to transfer the image to a netcat server located on the network. With this option you will need to specify the IP address and port number of the netcat server.
- Split Image: Allows you to split the image into multiple files if the image will exceed the capacity of the storage medium. For example, if you are imaging a 10 gig hard drive, you can split the image so that it will fit on a CDROM, DVD, or FAT 32 file system, which has a 4 gig file size limitation.

Once you enter all the parameters, and press the “Acquire” button, a forensic command shell window will open up. This command shell uses trusted binaries to prevent root kits from tampering with the image being created.



```
HELIX Forensic Command Shell
Ctrl-D for Directory or Ctrl-F for filename completion
The Shell Path has been modified to find trusted cdrom binaries first
Do not navigate away from the CD drive letter.
=====
13:25:12.67 E:\Shells>
```

You can now paste the dd command line into the shell by right clicking and selecting “paste” from the context menu. Press enter to execute the command.

A screenshot of a Windows command prompt window titled "HELIX Forensic Command Shell". The window has a blue title bar with standard minimize, maximize, and close buttons. The command prompt text is as follows:
Ctrl-D for Directory or Ctrl-F for filename completion
The Shell Path has been modified to find trusted cdrom binaries first
Do not navigate away from the CD drive letter.
=====

Once the command is finished, there will be 3 files in the destination directory:

- *filename.dd* – the image of the floppy disk
- *filename.dd.md5* – a file containing the MD5 of the image file.
- *Audit.log* – a file containing the command and the output of the program.



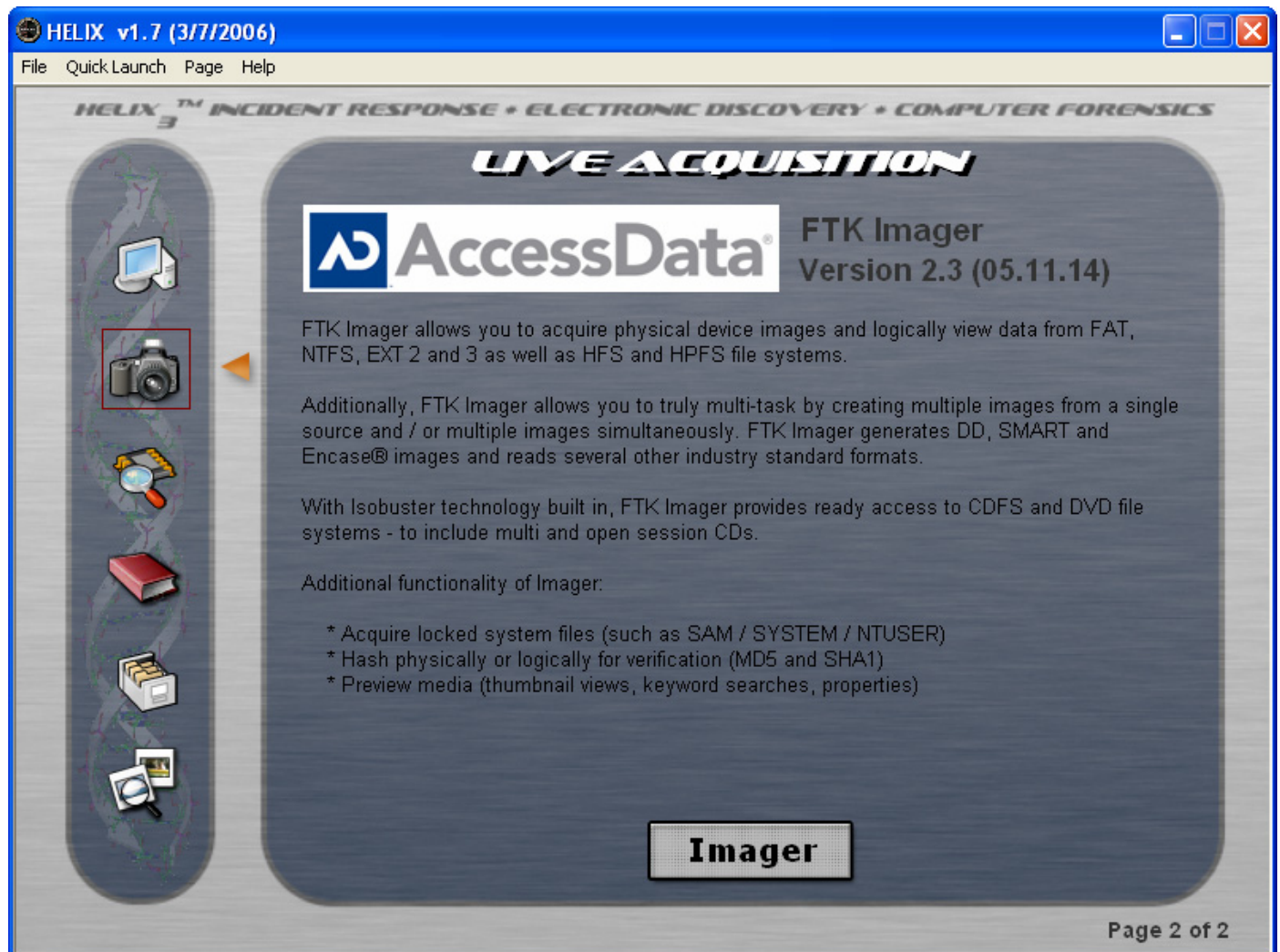
FTK Imager

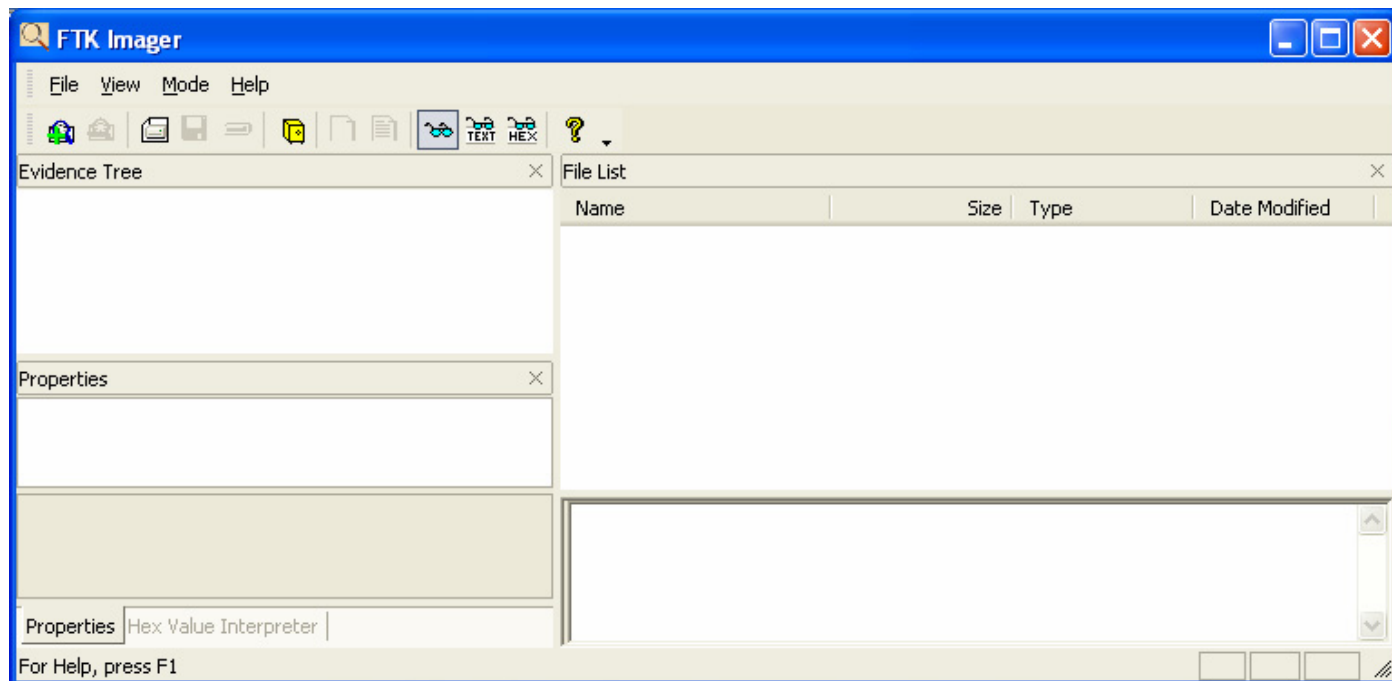
“FTK Imager is a data preview and imaging tool that lets you quickly assess electronic evidence to determine if further analysis with AccessData® Forensic Toolkit® (FTK™) is warranted. FTK Imager can also create perfect copies (forensic images) of computer data without making changes to the original evidence.” (Access Data, 2005)

According to the FTK Image Help File (Access Data, 2005), you can:

- Preview files and folders on local hard drives, floppy diskettes, Zip disks, CDs, and DVDs.
- Create forensic images of local hard drives, floppy diskettes, Zip disks, CDs, and DVDs.
- Preview the contents of forensic images stored on the local machine or on a network drive.
- Export files and folders.
- Generate hash reports for regular files and disk images (including files inside disk images).

To access the FTK Imager, select the second page of the Image Acquisition page. This page will display the release notes for the current version of the tool. Click on the “Imager” to launch the actual application.





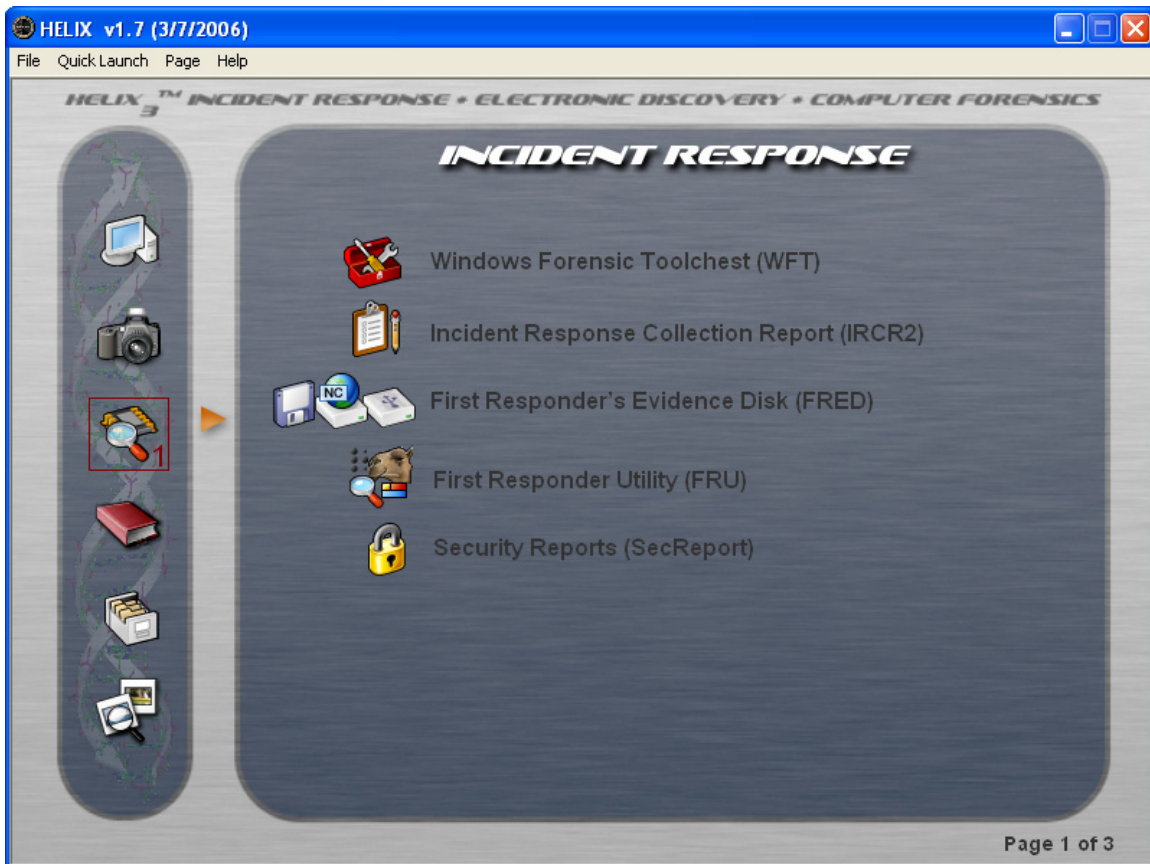
Note: FTK Imager can now also be launched via the Quick Launch menu on the main screen.

The FTK imager is a powerful and flexible tool. It can be used to examine media and images, and extracted deleted files. It has extensive information available via the Help menu or the question mark icon on the toolbar.

To see how to create an image of the floppy disk using FTK imager, see Lab 1b - Create an Image of a suspect Floppy Disk (Windows, FTK Imager).



Incident Response tools for Windows Systems



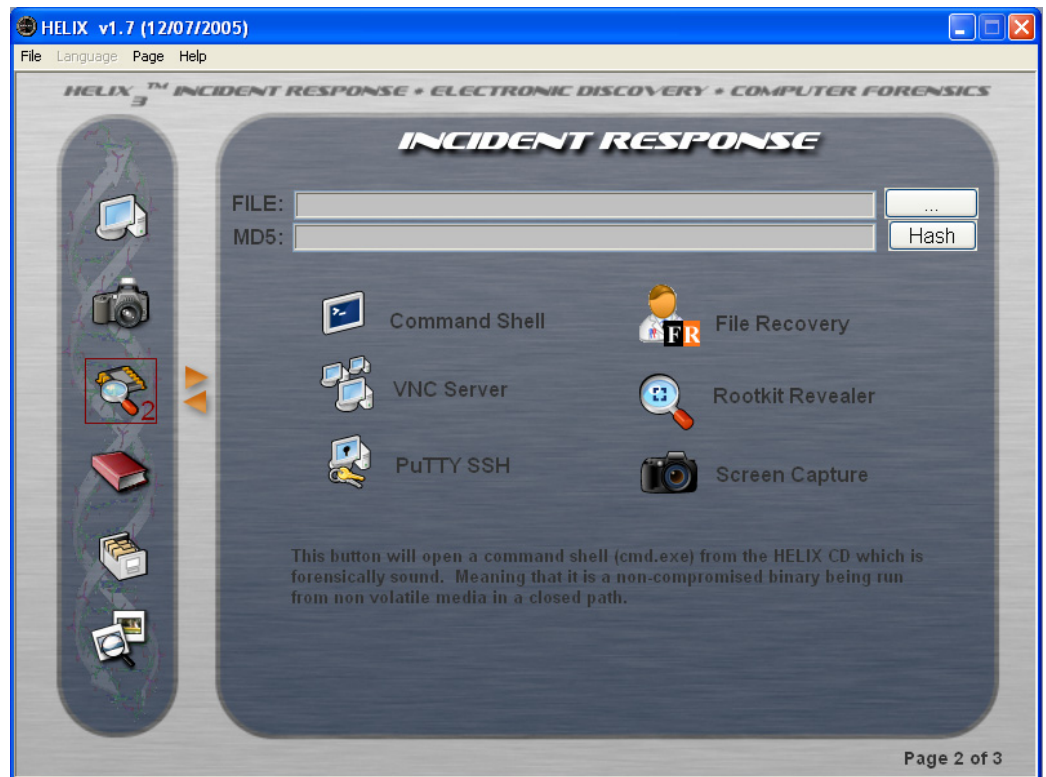
This panel provides the investigator with a number of tools to respond to incidents. There are three pages to this panel, the other pages can be accessed by clicking on the small triangles next to the Incident Response icon in the left tool bar.

The tools include:

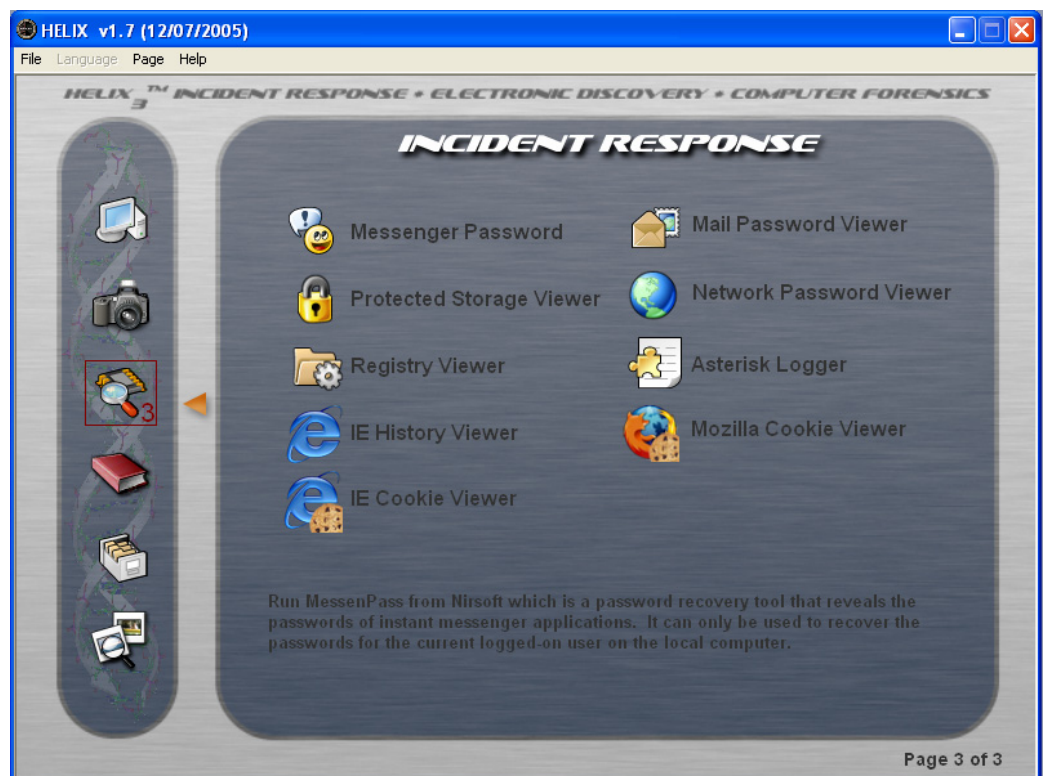
- Windows Forensics Toolchest (WFT)
- Incident Response Collection Report (IRCR2)
- First Responder's Evidence Disk (FRED)
- First Responder Utility (FRU)
- Security Reports (SecReport)
- Md5 Generator
- Command Shell – a forensically sound command shell
- File Recovery – recover deleted files
- Rootkit Revealer – detect the presence of rootkits on the system
- VNC Server
- Putty SSH
- Screen Capture
- Messenger Password
- Mail Password Viewer
- Protected Storage Viewer
- Network Password Viewer

- Registry Viewer
- Asterisk Logger
- IE History Viewer
- IE Cookie Viewer
- Mozilla Cookie Viewer

The 2nd page of Utilities



The 3rd page of Utilities





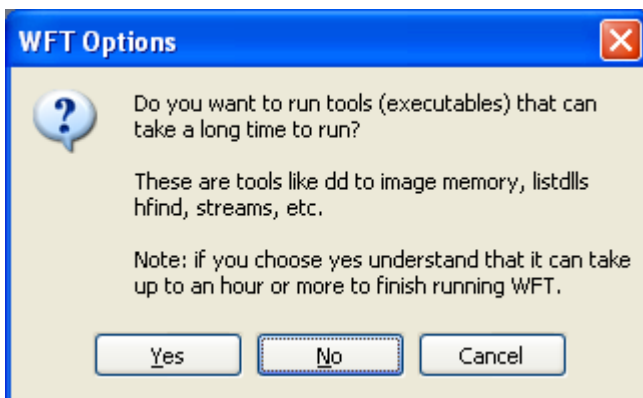
Windows Forensic Toolchest (WFT)

The Windows Forensic Toolchest (WFT) was written by Monty McDougal. It is available from <http://www.foolmoon.net/security/wft/index.html>

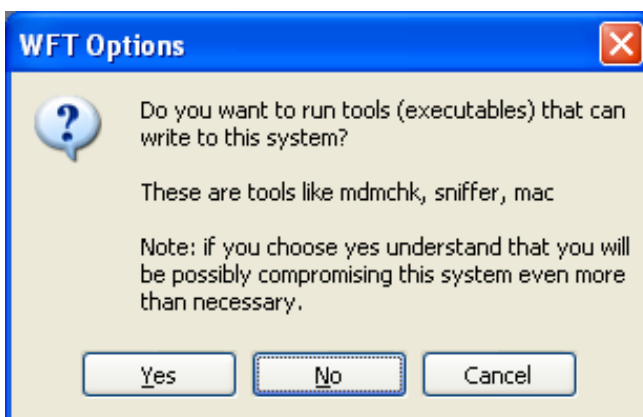
"The Windows Forensic Toolchest (WFT) was written to provide an automated incident response [or even an audit] on a Windows system and collect security-relevant information from the system. It is essentially a forensically enhanced batch processing shell capable of running other security tools and producing HTML based reports in a forensically sound manner. A knowledgeable security person can use it to help look for signs of an incident (when used in conjunction with the appropriate tools). WFT is designed to produce output that is useful to the user, but is also appropriate for use in court proceedings. It provides extensive logging of all its actions along with computing the MD5 checksums along the way to ensure that its output is verifiable. The primary benefit of using WFT to perform incident responses is that it provides a simplified way of scripting such responses using a sound methodology for data collection." (McDougal, 2005)

When the WFT program is started, it will prompt for an output folder. The investigator should point to a folder on removable media (floppy, zip, USB device), or a shared folder on the network to prevent the tool from modifying the suspect drive.

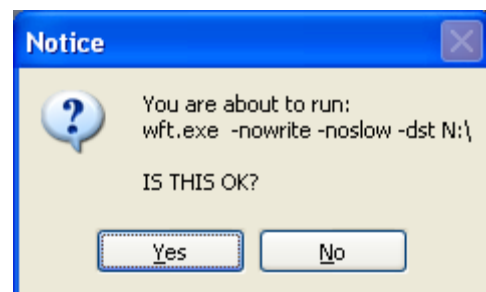
Once the folder is selected, the program will prompt the user if they want to run some detailed collection utilities? These can take up to an hour or more to run, depending on the system of the system, the amount of RAM, and many other factors.



Next the program will ask if the user wants to run some programs that can write information to the suspect's system. These tools can compromise the integrity of the system, so this option should be used with care.



Finally, the program will display the command and ask for conformation.



Once the user clicks “Yes”, a command tool will open, and the collection process will start. Depending on the options selected, this collection process can take anywhere from a few minutes to a few hours.

Once the collection is finished, the user will be returned to the Helix program. If the user examines the destination folder, there is a now a file “index.html”. This file can be examined with a browser of your choice. To prevent additional contamination of the suspect system, it should be viewed on another system.

A sample output is shown below.



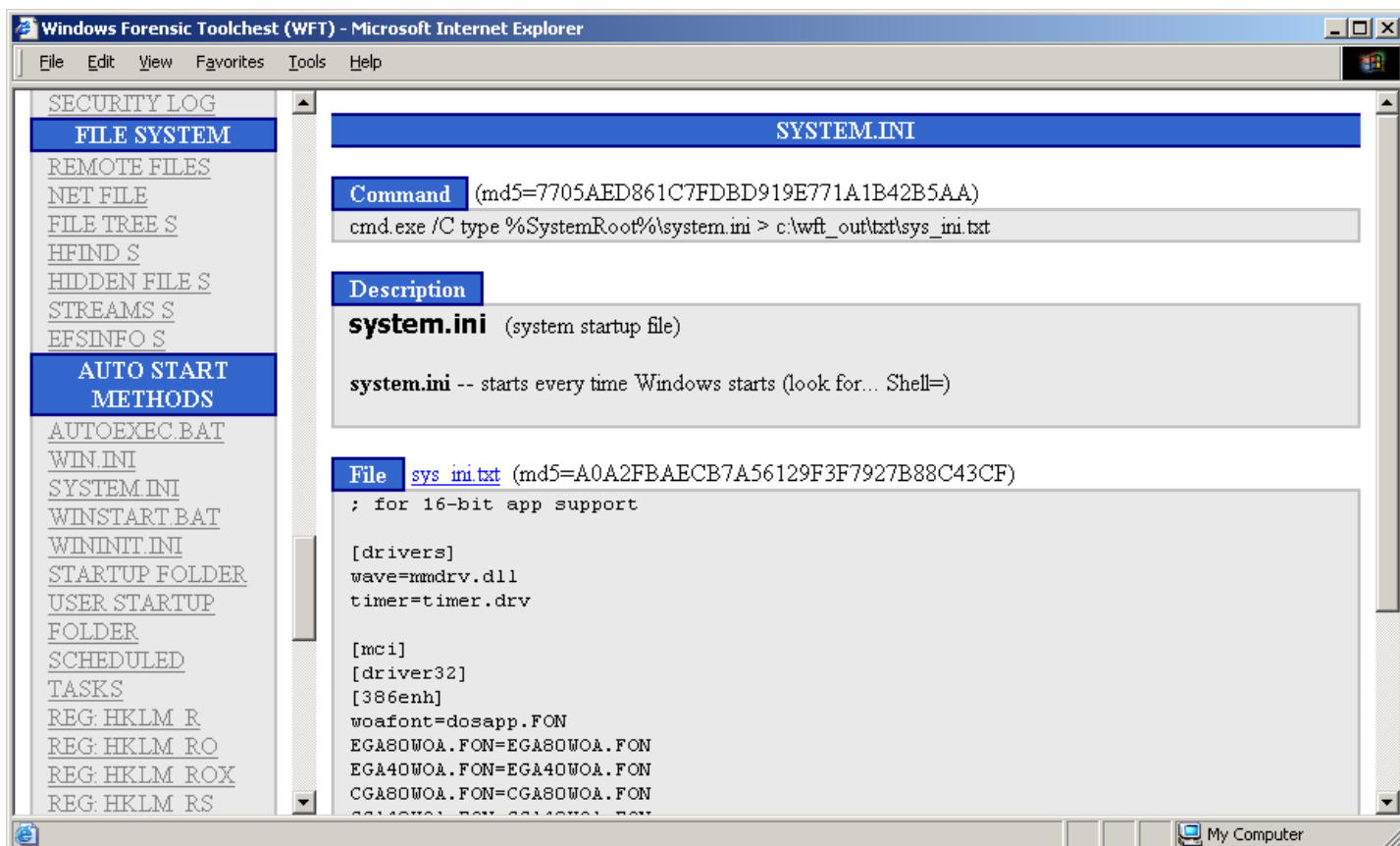
The output can be navigated use the highlight hyperlinks. WFT produces an impressive collection of information from the system. In addition, the LOG tab shows the commands that were executed, and the CONFIG tab shows the WFT configuration file.

From the website: Windows Forensic Toolchest (WFT) was designed to be useful both for a security administrator and as a tool to be used in a court of law. One of the biggest issues involved in a court case is ensuring that you have an adequate record of all the actions that you have taken. It is also necessary to have the appropriate safeguards in place to ensure that the data being presented has not been altered.

WFT seeks to meet both of these requirements. One of the most important features of WFT is the fact that it logs every action it takes as part of running commands.

An investigator using Windows Forensic Toolchest (WFT) would need some level of knowledge to interpret the data produced by running it. If the configuration file is used properly, then WFT is self

documenting to some degree as each HTML report produced will have the Description of the tool as part of it's output. Ultimately, the investigator needs to have a working knowledge of the tools that are being invoked via WFT to be able to interpret its output. WFT's primary benefit to the investigator is its ability to provide a scripted, automated response while promoting forensic integrity and detailed logging.



Windows Forensic Toolchest (WFT) provides output in two data formats. Each of these serves a specific purpose as described below.

The first and more useful format is HTML output. Opening the index.htm file produced by WFT provides an easy to read and easy to navigate interface to the output of the various tools invoked via WFT. Each of the reports produced under WFT includes the MD5 checksum for the binary being run, the exact command line issued to generate the output, a description of the tool, and the output produced by the tool along with the MD5 checksum associated with the output. The HTML reports are designed to be self-documenting via the text provided in the configuration file.

The second type of output produced by WFT is the raw text output from the tools. This format allows the viewer to see the output of the individual command exactly as it was produced. It is generally a bad idea to, in any way, manipulate data being used as evidence in a court of law. WFT seeks to preserve the original data while providing a user-friendlier HTML version for viewing. The MD5 checksums produced for each of the output files during collection provides a safeguard to ensure the output can be verified at a later date.



Incident Response Collection Report (IRCR2)

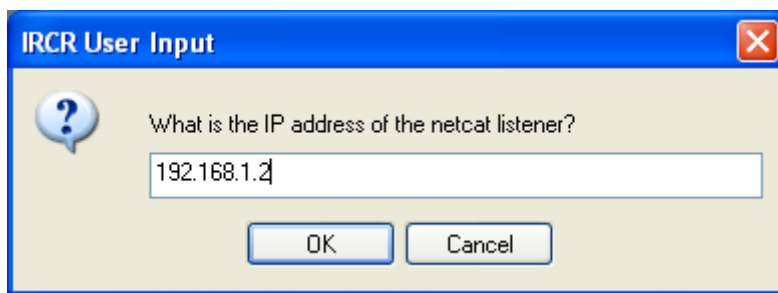
Written by John McLeod. Available from <http://ircr.tripod.com/>

“The Incident Response Collection Report is a script to call a collection of tools that gathers and/or analyzes data on a Microsoft Windows system. You can think of this as a snapshot of the system in the past. Most of the tools are oriented towards data collection rather than analysis. The idea of IRCR is that anyone could run the tool and send the output to a skilled computer security professional for further analysis.” (McLeod, 2005)

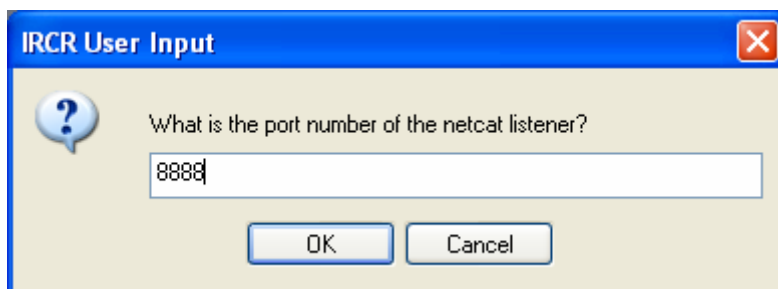
To prevent the suspect system from being modified, this tool sends the output to a listener system that is connected via a network connection. On the listener system, you need to run netcat. For this example, we have executed the command on a machine with the IP address of 192.168.1.2:

```
nc -l -p 8888 > IRCR2OutputReport.txt
```

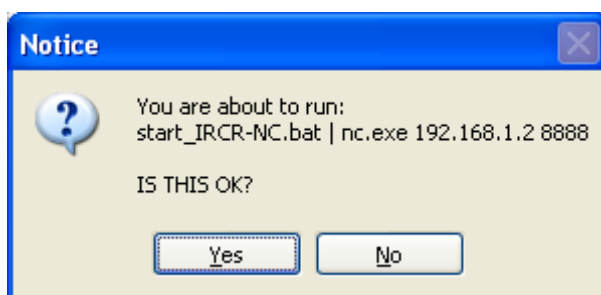
On the suspect system, when the IRCR2 program is started, it will prompt the user for the address of the listener system.



Next, it will ask for the port of the listener.



Finally, it will show the command and ask for confirmation.



Depending on the speed of the system and the speed of the network, this program can take a while to run. It is not unusual for the program to generate several errors. Once the program is finished, the output file will contain a detailed report of the suspect system.

Sample Output File

```
=====
                        Incident Response Collection Report
=====

      Name:
Computer Name:  tal_mc
      OS:  Microsoft Windows XP [Version 5.1.2600]

-----

START --  Time:  21:30:47.45  Date: Mon 12/12/2005
-----

21:30:47.63
Command:  AT
AT Schedule List
OUTPUT:
There are no entries in the list.

-----

21:30:47.76
Command:  doskey /history
MS-DOS history list
OUTPUT:

-----

21:30:47.87
Command:  ipconfig /all
Displays configuration information
OUTPUT:

Windows IP Configuration
    Host Name . . . . . : tal_mc
    Primary Dns Suffix . . . . . :
    Node Type . . . . . : Mixed
    IP Routing Enabled. . . . . : No
    WINS Proxy Enabled. . . . . : No

Ethernet adapter VMware Network Adapter VMnet8:
    Connection-specific DNS Suffix  . :
    Description . . . . . : VMware Virtual Ethernet Adapter for VMnet8
    Physical Address. . . . . : 00-50-56-C0-00-08
    Dhcp Enabled. . . . . : No
    IP Address. . . . . : 192.168.95.1
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . :

-----

21:30:48.15
Command:  MEM.exe /d
Displays memory usage
OUTPUT:

  Address      Name      Size Type
  -----
  000000      -----
  000000      000400      Interrupt Vector
  000400      000100      ROM Communication Area
  000500      000200      DOS Communication Area

  000700      IO      000370      System Data
                        CON      System Device Driver
                        AUX      System Device Driver
                        PRN      System Device Driver
```

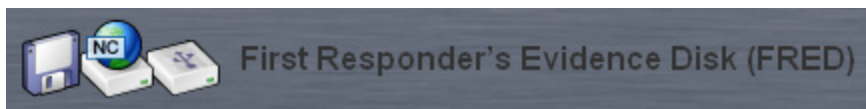


First Responder's Evidence Disk (FRED)

Written by Special Agent Jesse Kornblum of the Special Investigations Office of the United States Air Force. For more information on his work, see <http://research.jessekornblum.com/>

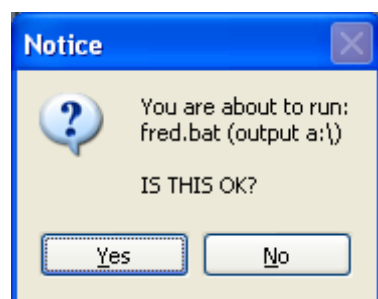
FRED is the First Responder's Evidence Disk. This MS/DOS batch file will collect a large amount of information from the system and store it in a text file.

There are 3 icons available for FRED that determine where the output file will be located. The three icons indicate: Floppy, Netcat and Other Storage Device.



Output to Floppy

If the suspect system has a floppy drive, the first icon will write the information to a floppy disk.



If the user clicks "Yes", two files will be created on the floppy disk. The output of the report will be in a:\audit.txt, and the MD5 signature of the audit.txt file will be in the a:\audit.MD5.

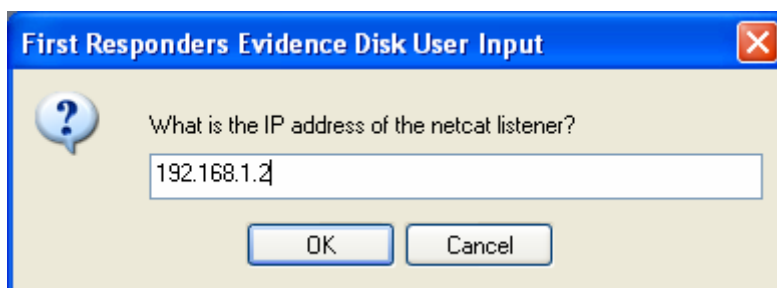


Output via NetCat

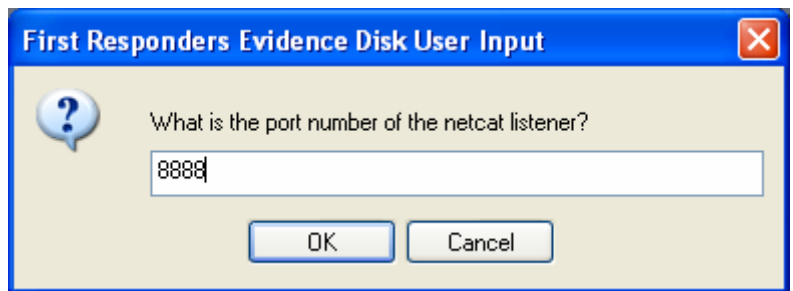
The second icon will transmitted the output via netcat to another system. This is useful if the system is connected to a network, and doesn't have a floppy disk. On the listener system, you need to run netcat. For this example, we have executed the command on a machine with the IP address of 192.168.1.2:

```
nc -l -p 8888 > FREDOutputReport.txt
```

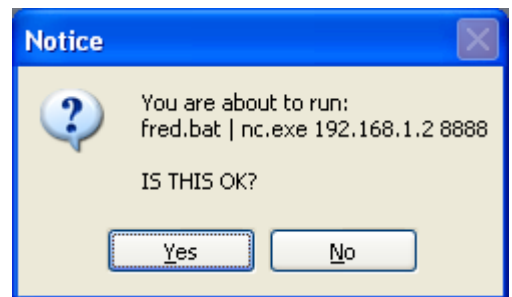
On the suspect system, when the F.R.E.D. program is started, it will prompt the user for the address of the listener system.



Next, it will ask for the port of the listener.



Finally, it will show the command and ask for confirmation.



Once the program is finished, press <CTRL>-C on the listener system. The output file will contain a detailed report of the suspect system.

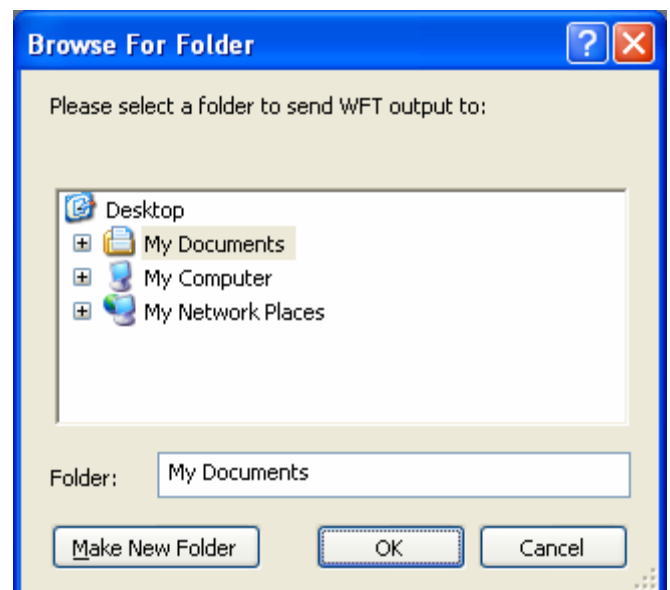
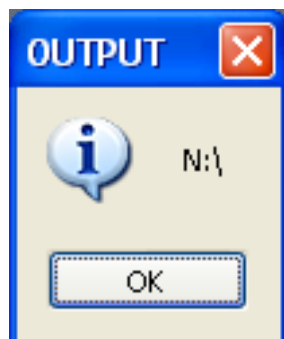
Unlike the other two options, this option will not automatically create a MD5 file for the output. You should now run the md5sum on the acquired audit log, and save that number. Do not modify the original file, since that will change the MD5 signature. It is recommended that the MD5 is written down on the evidence tag for the floppy.



Output to Other Storage Device

Finally, the third icon will allow the user to select the output folder. It is recommended that the output be send to a removable drive, or a network share. The output should not be written to the suspect's system, as this can compromise the integrity of the system.

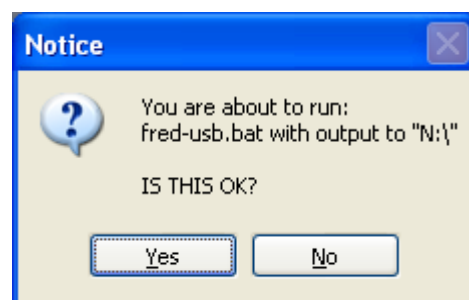
The program will confirm the output directory.



The program will then confirm the command.

If the user clicks “Yes”, two files will be created in the destination folder. The output of the report will be in audit.txt, and the MD5 signature of the audit.txt file will be in the audit.MD5.

Sample Output File



```
FRED v1.4 - 6 October 2005 [modified for HELIX 10/2005]
=====
START TIME
=====
Time:  21:52:54.84  Date: Mon 12/12/2005
=====
PSINFO
=====
System information for \\TAL_MC:
Uptime:          12 days 23 hours 21 minutes 22 seconds
Kernel version:   Microsoft Windows XP, Multiprocessor Free
Product type:     Professional
Product version:  5.1
Service pack:     1a
Kernel build number: 2600
Registered organization:
Registered owner:
Install date:     12/22/2004, 6:54:16 AM
Activation status: Activated
IE version:       6.0000
System root:      C:\WINDOWS
Processors:       2
Processor speed:  3.2 GHz
Processor type:   Intel(R) Pentium(R) 4 CPU
Physical memory:  1536 MB
Video driver:     Intel(R) 82845G/GL/GE/PE/GV Graphics Controller

=====
NET ACCOUNTS
=====
Force user logoff how long after time expires?:  Never
Minimum password age (days):                   0
Maximum password age (days):                   42
Minimum password length:                        0
Length of password history maintained:           None
Lockout threshold:                             Never
Lockout duration (minutes):                     30
Lockout observation window (minutes):            30
Computer role:                                  WORKSTATION
The command completed successfully.

=====
NET SHARE
=====

Share name  Resource                                Remark
-----
IPC$        Remote IPC
D$          D:\ Default share
C$          C:\ Default share
F$          F:\ Default share
ADMIN$ C:\WINDOWS Remote Admin
E$          E:\ Default share
```



First Responder Utility (FRU)

Written by Harlan Carvey. Available from <http://www.windows-ir.com/tools.html>

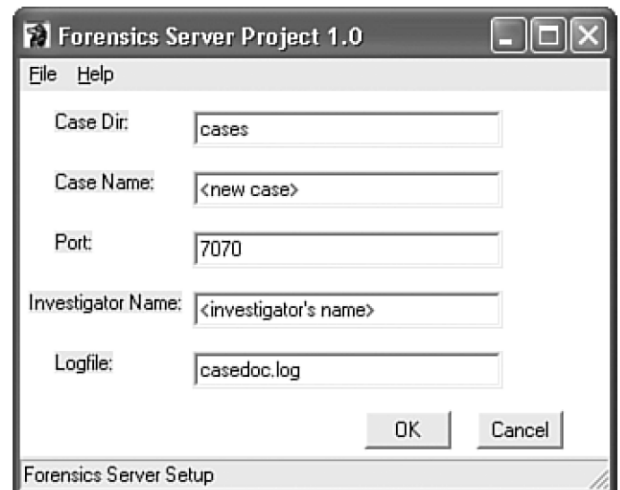
"The First Responder Utility (FRU) is used by a first responder to retrieve volatile data from "victim" systems. The current version of the FRU is a CLI (command line interface) tool called FRUC. The FRUC operates using a combination of an INI file and command line options." (Carvey, 2005a)

To prevent the suspect system from being modified, this tool sends the output to a Forensic Server Project (FSP) system that is connected via a network connection. The FSP server is available on the Helix CD in the \\IR\FSP directory in the fspc.zip file.

Once FSP has been installed on the investigator's system, the command:

```
C:\Per1\FSP>fsp.pl
```

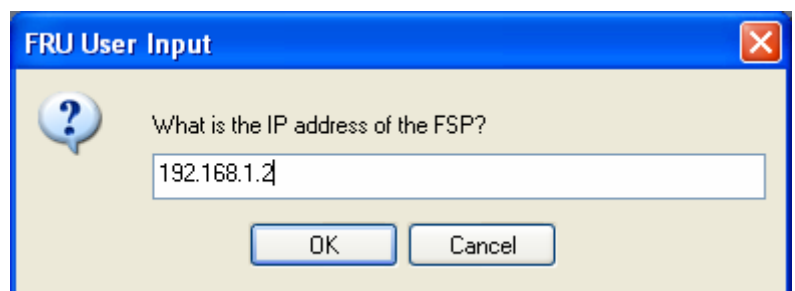
Will start the FSP server. It will display a configuration menu (Carvey, 2005b).



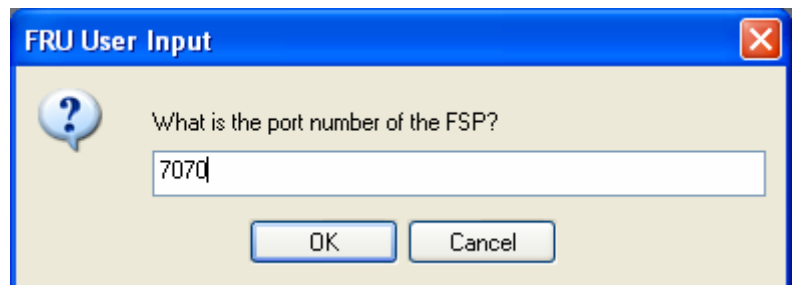
For more information on the FSP, see Carvey, H. A. (2005). *Windows forensics and incident recovery*. Boston: Addison-Wesley. Chapter 8 of the book, which deals with the FSP, is available as a sample chapter from:

http://awprofessional.com/content/images/0321200985/samplechapter/carvey_ch08.pdf

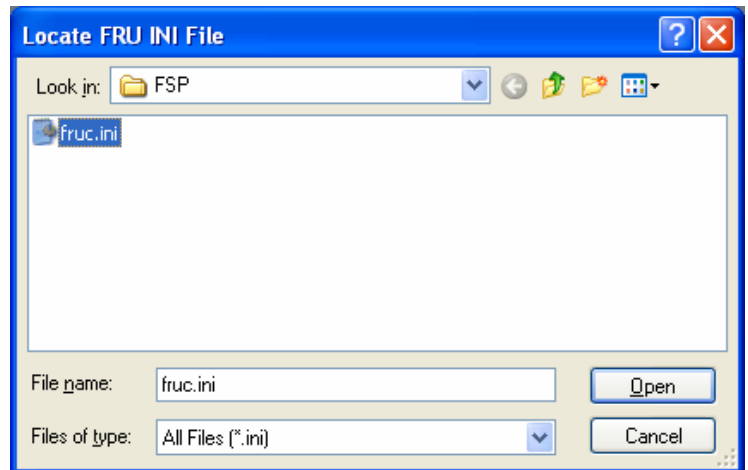
On the suspect system, when the FRU program is started, it will prompt the user for the address of the listener system.



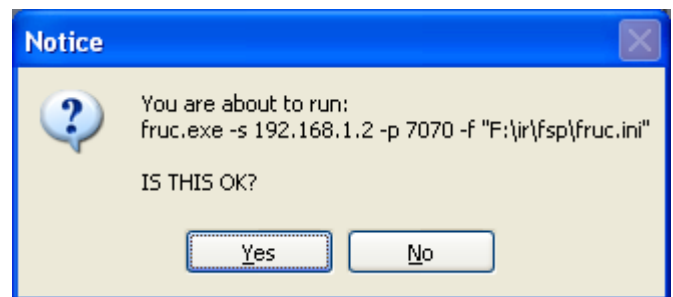
Next, it will ask for the port of the listener. It is important that this port number matches the port number on the FSP server.



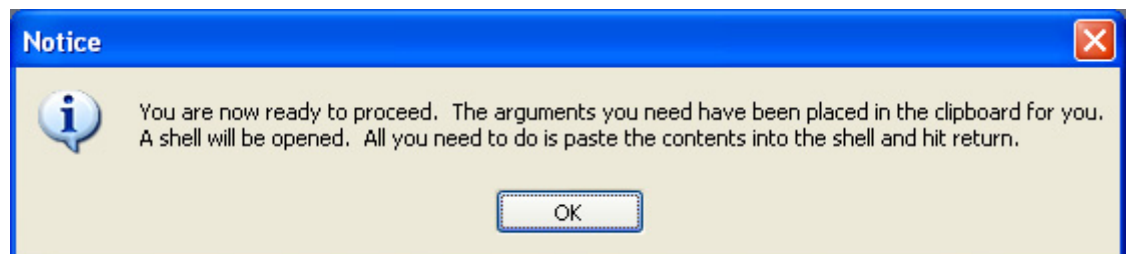
It will then ask for the location of the fruc.ini file. The user should select the default file, unless they have created their own fruc.ini file.



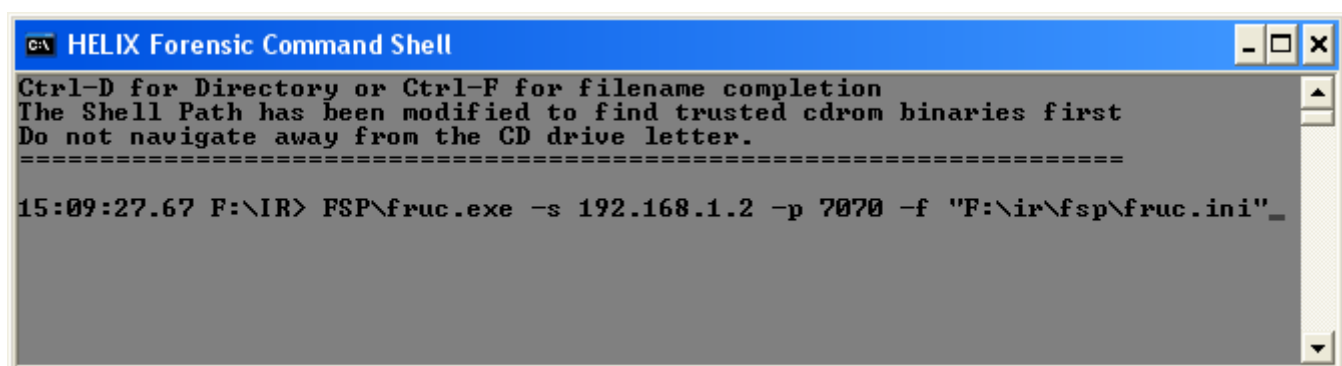
Finally, it will show the command and ask for confirmation.



Once the user clicks "Yes", the command will be placed into the clipboard.



When the command shell opens, the user should right-click inside of it, and select Paste to insert the command into the command shell.



Pressing Enter will execute the command.

The information on the suspect's system will now be transferred to the Forensics Server Project System.



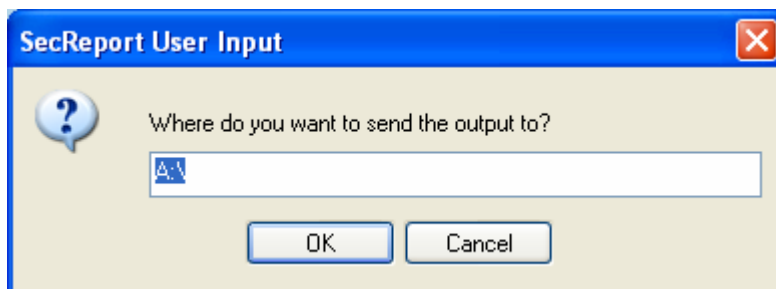
SecReport

SecReport is a freeware tool available from <http://members.verizon.net/~vze3vkmg/index.htm>.

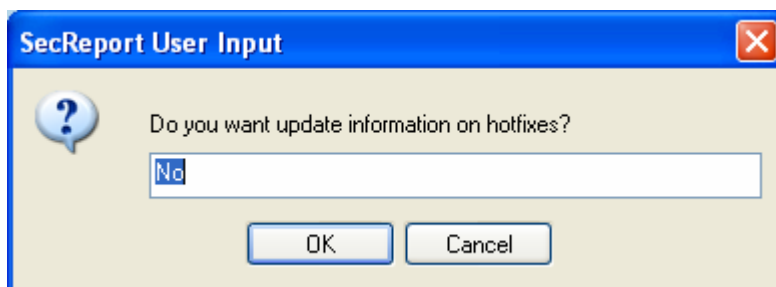
"It is a small suite of two command-line tools for collecting security-related information from Windows-based system (*SecReport*) and comparing any two reports either from any two systems or from same system after some time (*Delta*). I use these tools to quickly assess level of securing of Windows system and to compare results to baseline. The tools are useful both in daily security administration and during incident response - for fast collection of information. Tools do not need to be installed on system and can be run directly from hard or CD-R disk or network drive (mapped or UNC). Format of reports - XML. Reports can be viewed with IE 6.0 browser. MD5 hash file for report automatically created." (*SecReport*, 2005)

Supported platforms: Windows 2000, XP, 2003 - full support; NT4 (SP6 or later) - limited support

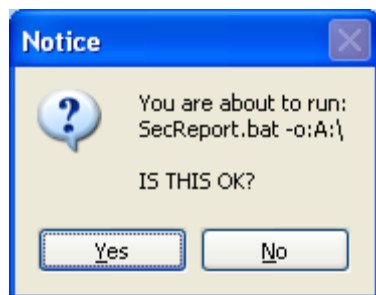
Helix provides the investigator with a graphical front end for the application. Clicking on the SecReport icon generate the following window:



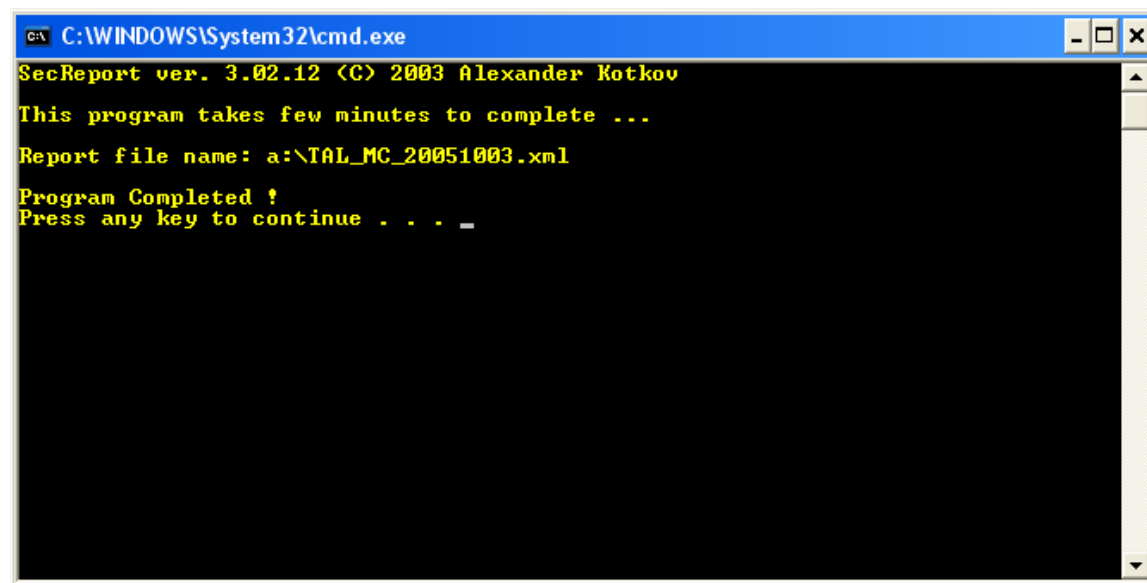
The output should be directed to removable media to prevent contamination of the suspect's system. Enter the drive name and click OK.



Information on hotfixes requires a fast connection to the Internet. The default is No. Click OK to continue. The confirmation prompt will show the command and its parameters.



Clicking YES will open a command shell, which will execute the command. After a few minutes, the program will complete and prompt the user to press any key to continue.



Pressing any key will close the window. The investigator will find two output files at the specified location. `securityreport.xsl` is the stylesheet for the report, and `machinename_date.xml` file. Double clicking on the .xml file will open the report in Internet Explorer.

This report goes on for several pages, detailing the following information:

- Network Configuration
- Audit Policy
- Event Log Configuration
- Services
- Applications
- Hotfixes
- Ports Open
- Page File Settings
- Hardware
- Processors
- Fixed Disks
- Mixed Checkpoints

Sample Output

A:\TAL_MC_20051003.xml - Microsoft Internet Explorer

FileEditViewFavoritesToolsHelp

Back

Search

Favorites

Media

AddressA:\TAL_MC_20051003.xmlGoLinks

Googlevze3vkmgSearch215 blockedCheckAutoLinkAutoFillOptionsvze3vkmg

Security Report TAL_MC

Hostname: TAL_MC

Date and time of report: 2005-10-03, 11:12, (GMT+09:00)

Operating System: Microsoft Windows XP Professional 5.1.2600

Service Pack: 1.0

Server Domain: MLC

Server Role: Standalone Workstation

IE Version: 6.0.2800.1106

Media Player Version: 8.0.0.4490

WSH Version: 5.6

Network Configuration

NIC Brand and Model: Intel(R) PRO/1000 MTW Network Connection - Packet Scheduler Miniport

IP Address: 192.168.1.100Subnet Mask: 255.255.255.0

Gateway: 192.168.1.1

DNS Server: 144.0.0.44

DNS Server: 143.0.0.44

DNS Server: 143.0.0.44

MAC Address: 23EB1F9623EB

Audit Policy

Policy	Security setting
Account Logon	No
Account Management	No
Directory Service Access	No
Logon	No
Object Access	No
Policy Change	No
Privilege Use	No
Process Tracking	No
System	No

Event Log configuration

Log Name	Max Size (KB)	Overwrite Old Events	Overwrite Policy
Application	512	7	OutDated
Security	512	7	OutDated
System	512	7	OutDated

Services

Total number of services: 94; Number of Running services: 49; Number of Automatic services: 42; Number of Manual services: 48

Service	Start Type	Status	Service full name	Account
---------	------------	--------	-------------------	---------

DoneMy Computer

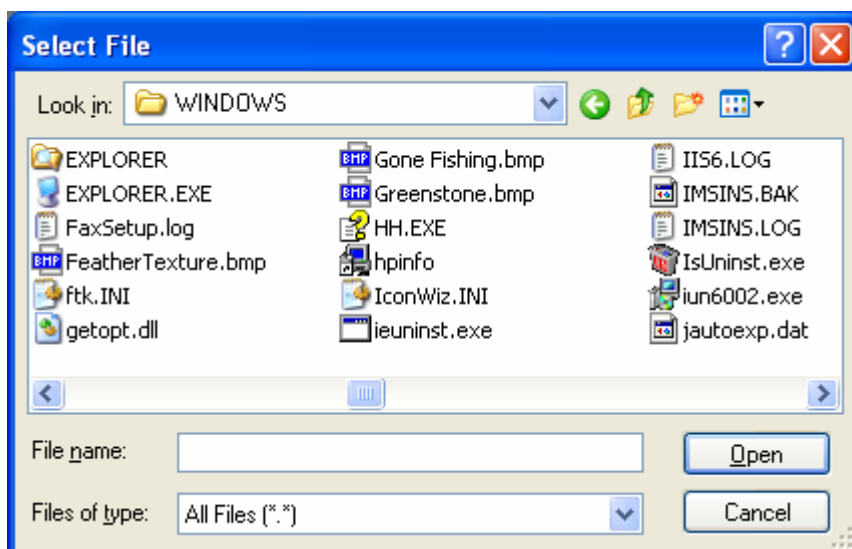
5f4dce3b5
ac765d41
d8327deb
882cf99

Md5 Generator

On page 2 of the Incident Response tools, you will see an input box that will allow you to generate the MD5 signature of any file.



Start by pressing the button "...". This will bring up a file manager that you can use to select a file.



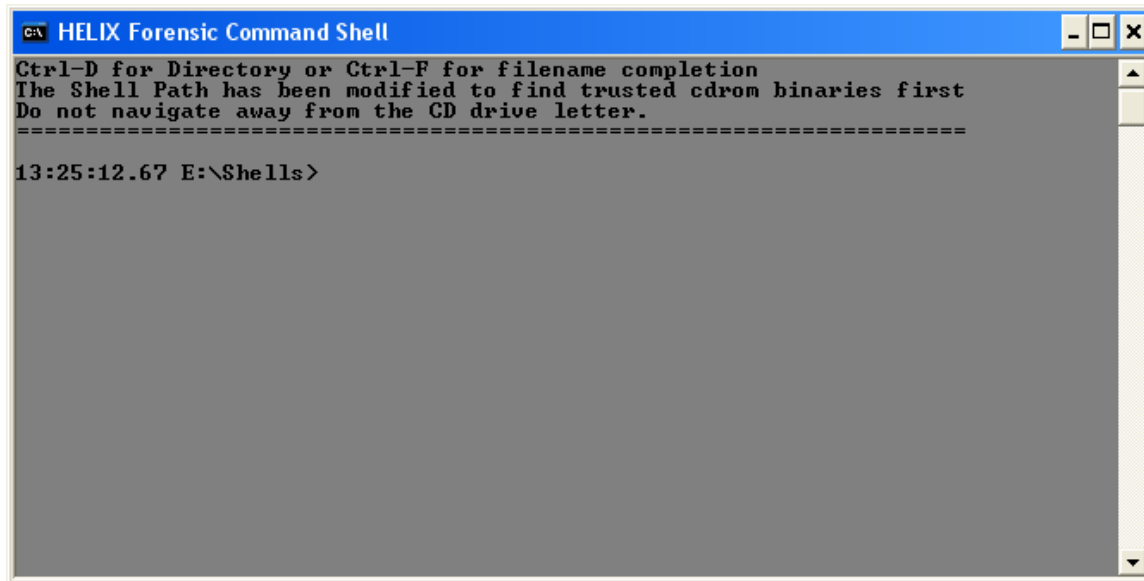
Select a file, and once it has been listed in the “FILE:” textbox, the user can click on the “HASH” button to generate the MD5 of the file.





Command Shell

This is a forensically sound command shell, which means runs only trusted, non-compromised, binaries that are included on the CD.



```
C:\ HELIX Forensic Command Shell
Ctrl-D for Directory or Ctrl-F for filename completion
The Shell Path has been modified to find trusted cdrom binaries first
Do not navigate away from the CD drive letter.
=====
13:25:12.67 E:\Shells>
```

The Helix GUI will autodetect and run the appropriate command shell for the OS that is in use.

All the standard commands are available, as well as access to the command line versions of many of the forensics tools included on the CD. The path command will show all the directories that are searched to find the command.

Since several directories can contain commands with the same name, if the user wants a specific command, they should specify the entire path to the specific command.

```
22:32:49.43 I:\IR> path
PATH=I:\IR\FAU\;I:\IR\Cygwin\;I:\IR\bin\;I:\IR\WFT;I:\IR\IRCR\;I:\IR\unxu
tils\;I:\IR\sysinternals\;I:\IR\microsoft\;I:\IR\systemtools\;I:\IR\ntsec
urity\;I:\IR\perl\;I:\IR\Foundstone\;I:\IR\2k\;I:\IR\2k3\;I:\IR\FSP\;I:\I
R\nt\;I:\IR\xp\;I:\IR\shells\;I:\IR\nirsoft\;I:\IR\windbg\

22:32:53.43 I:\IR>
```



Rootkit Revealer

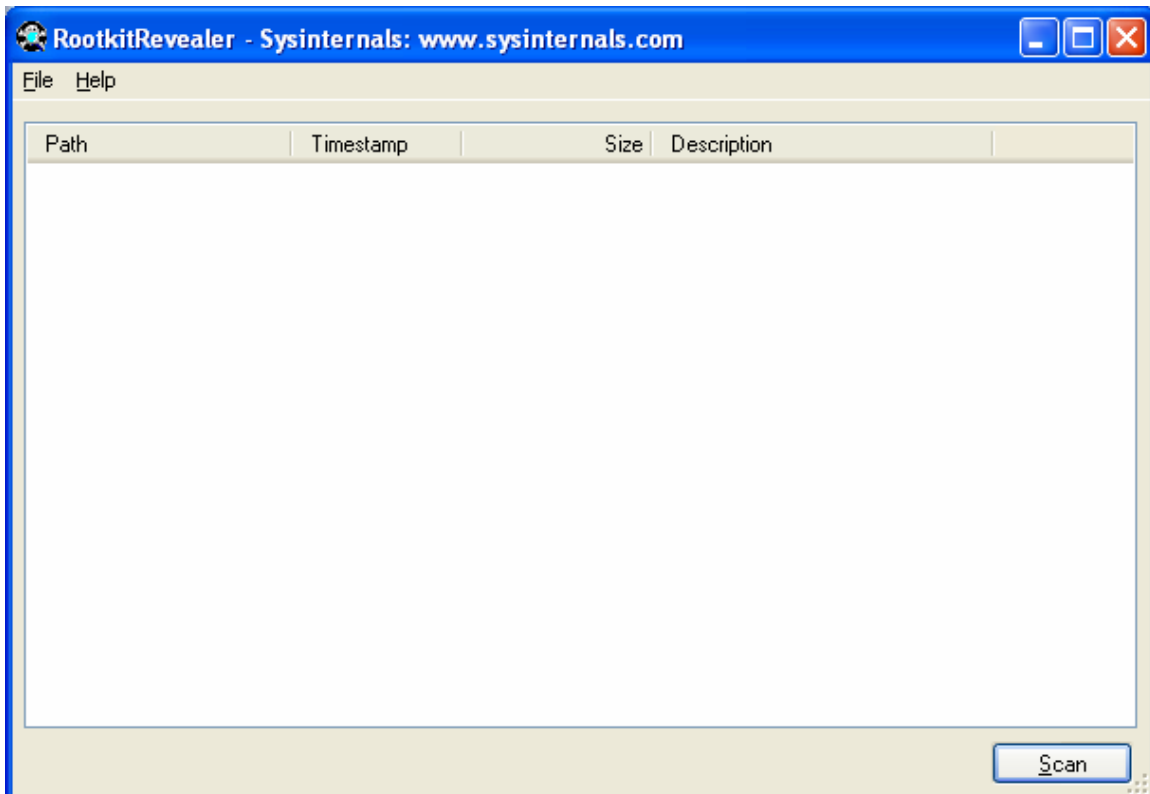
This is a freeware tool from SysInternals

(<http://www.sysinternals.com/Utilities/RootkitRevealer.html>). According to the website, "It runs on Windows NT 4 and higher and its output lists Registry and file system API discrepancies that may indicate the presence of a user-mode or kernel-mode rootkit. RootkitRevealer successfully detects all persistent rootkits published at www.rootkit.com, including AFX, Vanquish and HackerDefender."

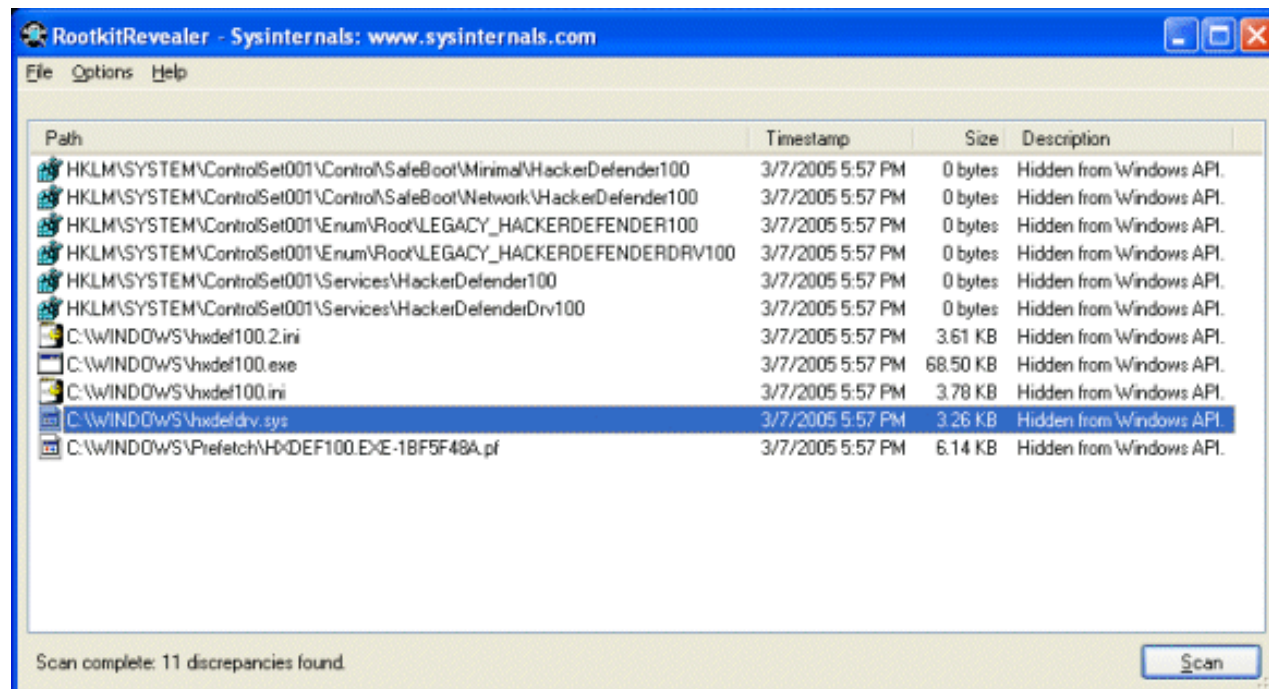
What is a rootkit? It is a series of malware applications that replace the standard windows utilities with Trojan horse programs, in an attempt to take over your system. This rootkits modify the operating system so that it can successfully hide and avoid traditional means of detection. For example, it may modify the Windows Explorer and DIR commands so the user will not be able to see the directory the rootkit is installed in. In addition, the rootkits open up backdoors to the system to allow the remote control of the system for sending out spam, launching denial-of-service attacks, or for pirating software.

For more information on rootkits, see www.rootkit.com and Microsoft's page on rootkit research research.microsoft.com/rootkit/.

To run the application, click on the rootkit revealer icon. At the confirmation window, click Yes to run the program. The main scanning windows will appear.



As with many other tools, this program will only run at the level of the currently logged in user. It would be best to run this as the system administrator for the most accurate results. Below is an example of the program detecting the HackerDefender rootkit (from the Sysinternals website).



When the scan is completed, the output can be saved to a file using the File / Save as option. To interpret the output, the following information is taken from the sysinternals website.

Hidden from Windows API.

These discrepancies are the ones exhibited by most rootkits, however, if you haven't checked the Hide NTFS metadata files you should expect to see a number of such entries on any NTFS volume since NTFS hides its metadata files, such as \$MFT and \$Secure, from the Windows API. The metadata files present on NTFS volumes varies by version of NTFS and the NTFS features that have been enabled on the volume. There are also antivirus products, such as Kaspersky Antivirus, that use rootkit techniques to hide data they store in NTFS alternate data streams. If you are running such a virus scanner you'll see a Hidden from Windows API discrepancy for an alternate data stream on every NTFS file. RootkitRevealer does not support output filters because rootkits can take advantage of any filtering. Finally, if a file is deleted during a scan you may also see this discrepancy.

This is a list of NTFS metadata files defined as of Windows Server 2003:

- \$AttrDef
- \$BadClus
- \$BadClus:\$Bad
- \$BitMap
- \$Boot
- \$LogFile
- \$Mft
- \$MftMirr

- \$Secure
- \$UpCase
- \$Volume
- \$Extend
- \$Extend\\$\Reparse
- \$Extend\\$\ObjId
- \$Extend\\$\UsnJrnl
- \$Extend\\$\UsnJrnl:\$Max
- \$Extend\\$\Quota

Access is Denied.

RootkitRevealer should never report this discrepancy since it uses mechanisms that allow it to access any file, directory, or registry key on a system.

Visible in Windows API, directory index, but not in MFT.

Visible in Windows API, but not in MFT or directory index.

Visible in Windows API, MFT, but not in directory index.

Visible in directory index, but not Windows API or MFT.

A file system scan consists of three components: the Windows API, the NTFS Master File Table (MFT), and the NTFS on-disk directory index structures. These discrepancies indicate that a file appears in only one or two of the scans. A common reason is that a file is either created or deleted during the scans. This is an example of RootkitRevealer's discrepancy report for a file created during the scanning:

C:\newfile.txt

3/1/2005 5:26 PM

8 bytes

Visible in Windows API, but not in MFT or directory index.

Windows API length not consistent with raw hive data.

Rootkits can attempt to hide themselves by misrepresenting the size of a Registry value so that its contents aren't visible to the Windows API. You should examine any such discrepancy, though it may also appear as a result of Registry values that change during a scan.

Type mismatch between Windows API and raw hive data.

Registry values have a type, such as DWORD and REG_SZ, and this discrepancy notes that the type of a value as reported through the Windows API differs from that of the raw hive data. A rootkit can mask its data by storing it as a REG_BINARY value, for example, and making the Windows API believe it to be a REG_SZ value; if it stores a 0 at the start of the data the Windows API will not be able to access subsequent data.

Key name contains embedded nulls.

The Windows API treats key names as null-terminated strings whereas the kernel treats them as counted strings. Thus, it is possible to create Registry keys that are visible to the operating system, yet only partially visible to Registry tools like Regedit. The Reghide sample code at Sysinternals demonstrates this technique, which is used by both malware and rootkits to hide Registry data.

Data mismatch between Windows API and raw hive data.

This discrepancy will occur if a Registry value is updated while the Registry scan is in progress. Values that change frequently include timestamps such as the Microsoft SQL Server uptime value, shown below, and virus scanner "last

scan" values. You should investigate any reported value to ensure that its a valid application or system Registry value.

HKLM\SOFTWARE\Microsoft\Microsoft SQL Server\RECOVERYMANAGER\MSSQLServer\uptime_time_utc
3/1/2005 4:33 PM
8 bytes

This tool will only help find rootkits, and will not remove them. Depending on the nature of the investigation, the detection of the rootkit needs to be documented, and the system preserved for further investigation. If the investigator believes a rootkit has been found, and the rootkit needs to be removed from the production system, there are typically only two ways to remove the rootkit. The first is to search the web to find removal instructions, and the second is to reformat the entire system a reinstall windows from a trusted source.



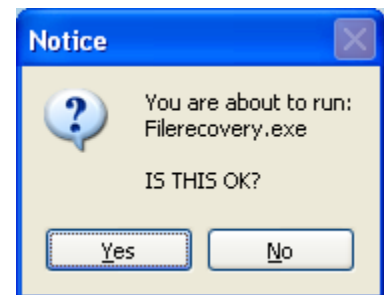
File Recovery

This button launches PC Inspector File Recovery from http://www.pcinspector.de/file_recovery/UK/welcome.htm. This freeware utility can be used to detect and recover deleted files. It supports file recovery from FAT 12/16/32 and NTFS file systems. [Developer's note: version 4.0 is now available].

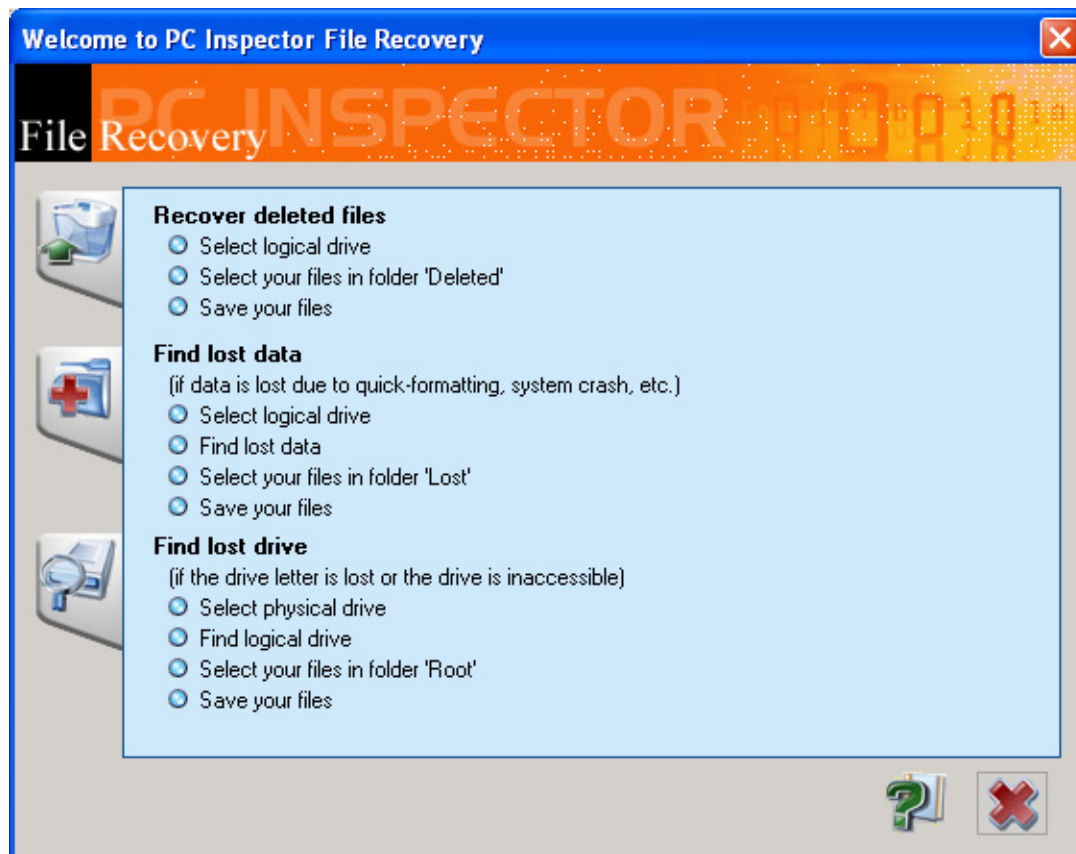
According to the website, it can find partitions automatically, even if the boot sector or FAT has been erased or damaged; Recovers files with the original time and date stamp; Supports the saving of recovered files on network drives; Recovers files, even when a header entry is no longer available. Competition products cannot recover such files. The "Special Recovery Function" supports the following disk formats: ARJ AVI BMP CDR DOC DXF DBF XLS EXE GIF HLP HTML HTM JPG LZH MID MOV MP3 PDF PNG RTF TAR TIF WAV ZIP.

If the hard disk is no longer recognized by the BIOS, or is having mechanical problems (such as grinding sounds), this program will not be able to help.

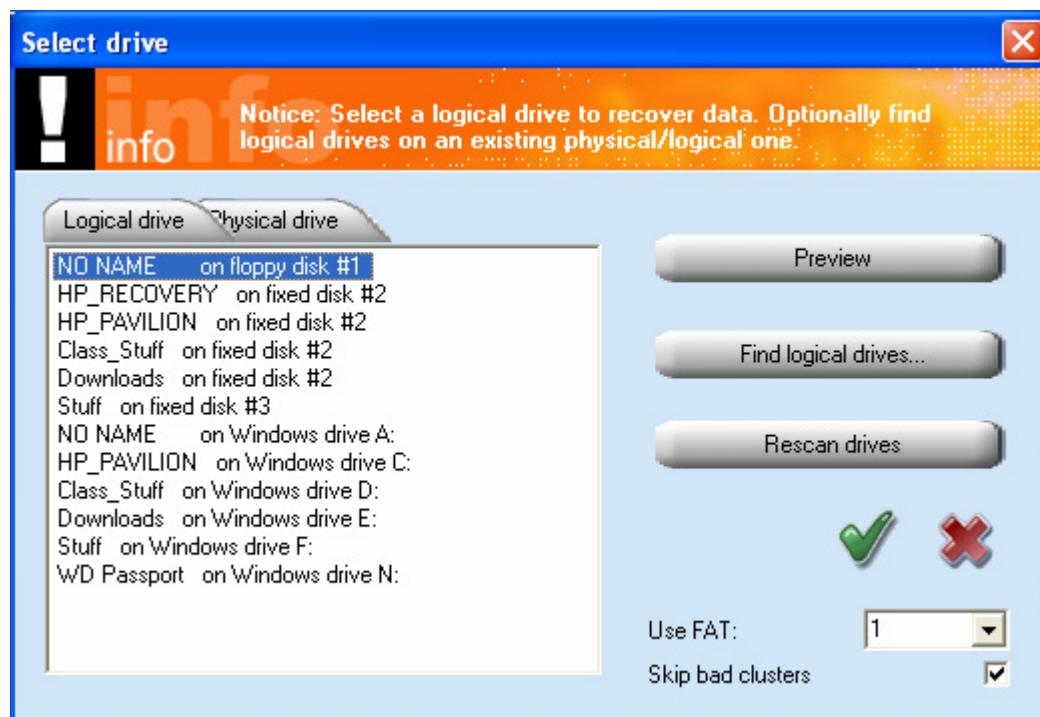
To use the program, click on the file recovery icon, and answer yes to the confirmation dialogue. The main program will start, and open up a file recovery wizard. The program allows the investigator to select the language of their choice.



The main window will appear, giving several options.



Clicking on any of the options will scan the system and present a list of recognized drives.

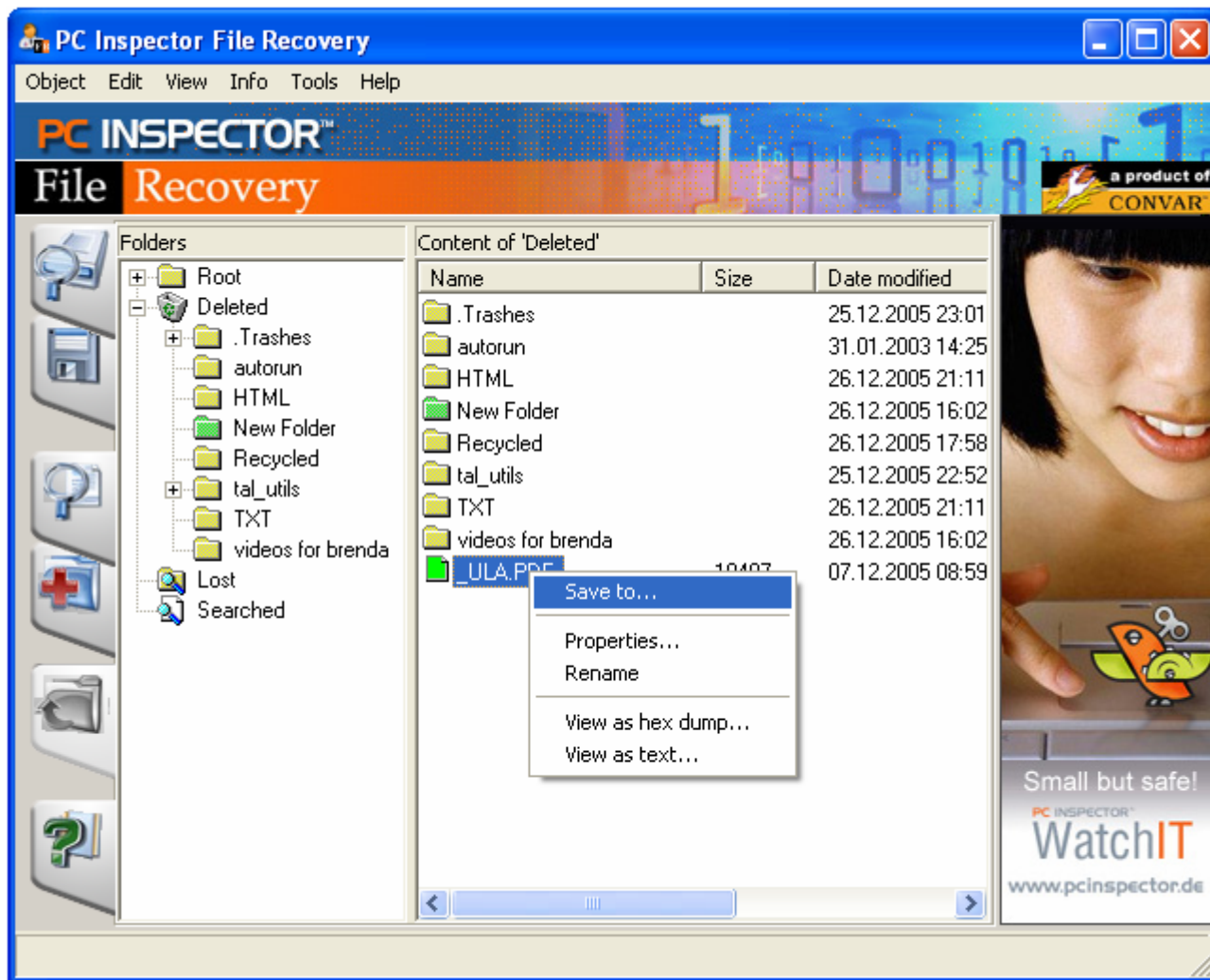


Select the drive to examine. To continue, select the green checkmark icon. Each option will provide different methods on how to recover data.

In the recovered deleted files options, the program will display a windows explorer-like interface.



In this screen, in the deleted folders, we see that we can recover the _ULA.PDF file. To recover the file, right click on the filename and select "Save To..."



There are also other options, such displaying the properties, renaming the file, and viewing it either as a hex dump or as a text file.



VNC Server

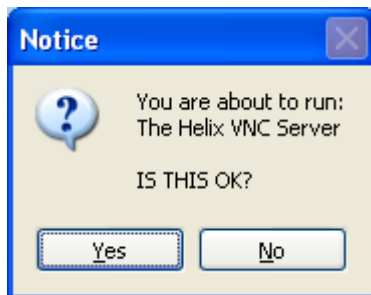
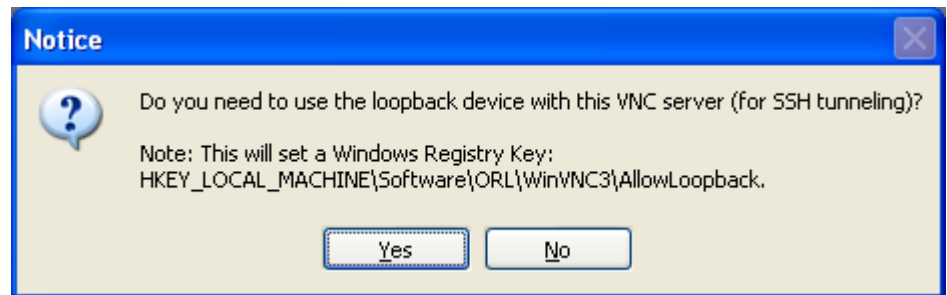
<http://www.realvnc.com/>

From the website: VNC stands for Virtual Network Computing. It is remote control software which allows you to view and interact with one computer (the "server") using a simple program (the "viewer") on another computer anywhere on the Internet. The two computers don't even have to be the same type, so for example you can use VNC to view an office Linux machine on your Windows PC at home.

To use VNC, click on the icon next to VNC server.

This option allows VNC to modify the registry so it can use the PuTTY SSH to provided encrypted communications.

After you make your decision, it will provide a confirmation prompt:



Click YES to continue. WinVNC will open a properties box. For the most part, you can leave it as it is, with one exception. You must enter a password in the password dialog box. VNC server will not accept incoming connections without a password.

To access this system from another location, you can use a VNC viewer, or a web browser. To use a web browser (from the Real VNC website): The VNC servers also contain a small web server. If you connect to this with a web browser, you can download the Java version of the viewer, and use this to view the server. You can then see your desktop from any Java-capable browser, unless you are using a proxy to connect to the web. The server listens for HTTP connections on port 5800+display number. So to view display 2 on machine 'snoopy', you would point your web browser at:

<http://snoopy:5802/>

The applet will prompt you for your password,



and should then display the desktop.

From the viewer, you should now have full control of the system that the server is running on.

This is useful if the system you are examining and the system you are using to collect the data are too far apart work on them at the same time.



PuTTY SSH

Written and maintained primarily by Simon Tatham. It is available from <http://www.chiark.greenend.org.uk/~sgtatham/putty/>

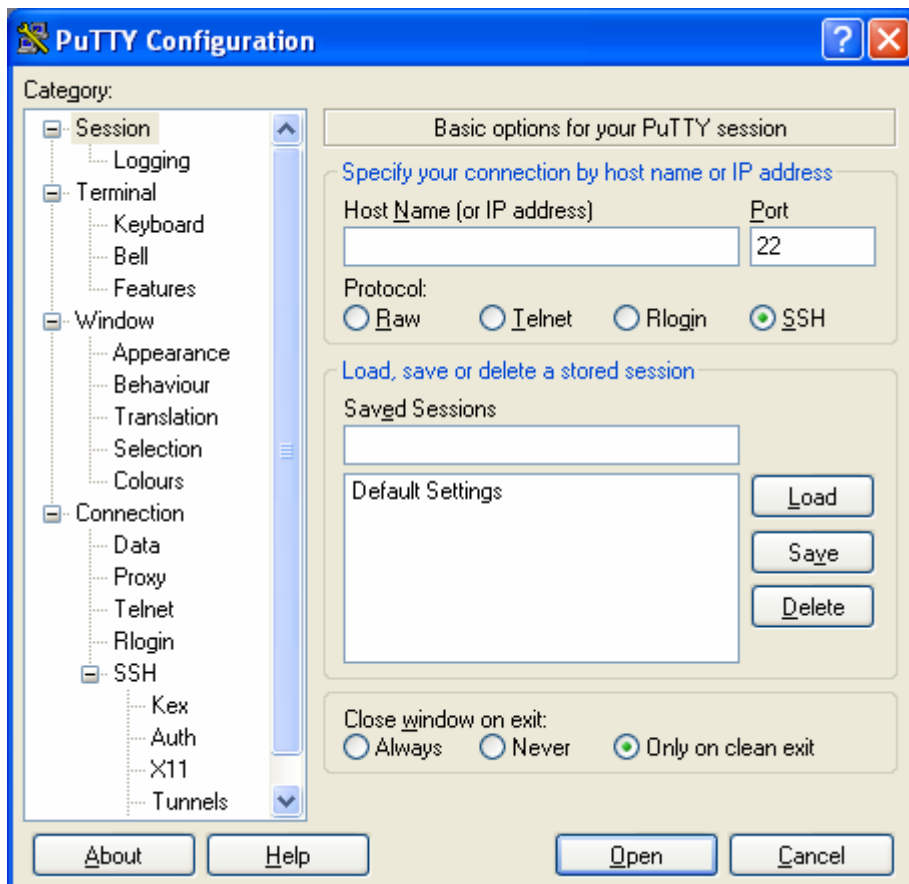
From the website: PuTTY is a free implementation of Telnet and SSH for Win32 and Unix platforms, along with an xterm terminal emulator.

This tool allows the user to remotely logon to a remote system and issue commands. This can be used to login into a remote system and run a netcat listener. The remote system must have a SSH server up and running.

Once selected, the program will display a confirmation.



Clicking on “Yes” will launch the PuTTY SSH program and display the configuration window.



For normal operations, the user should only have to enter the Host name or IP address, and click “Open”.



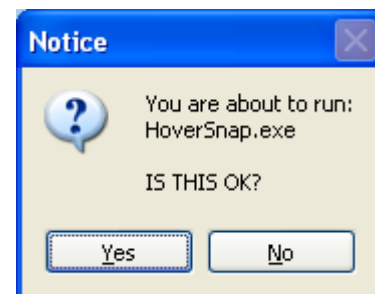
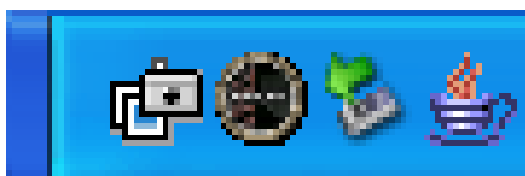
Screen Capture

<http://www.hoverdesk.net/freeware.htm>

From the website: HoverSnap is a free handy snapshot tool with jpg, png, bmp and gif support. HoverSnap can take snapshots of the full screen, active window or a selected area. It can even capture layered windows (alphablended ones under 2K / XP). You can even FTP upload your screenshots. In addition, you can set up the capture folder / filename and format, reduce the capture size, and auto-generate filename option will add the time stamp (date/time) to your filename in order to be able to take several captures without having to change the filename.

When you select the HoverSnap icon, it will present a confirmation prompt. When the user clicks “Yes”, there will be a HoverSnap icon in the system tray.

Clicking on this icon will display the configuration screen.



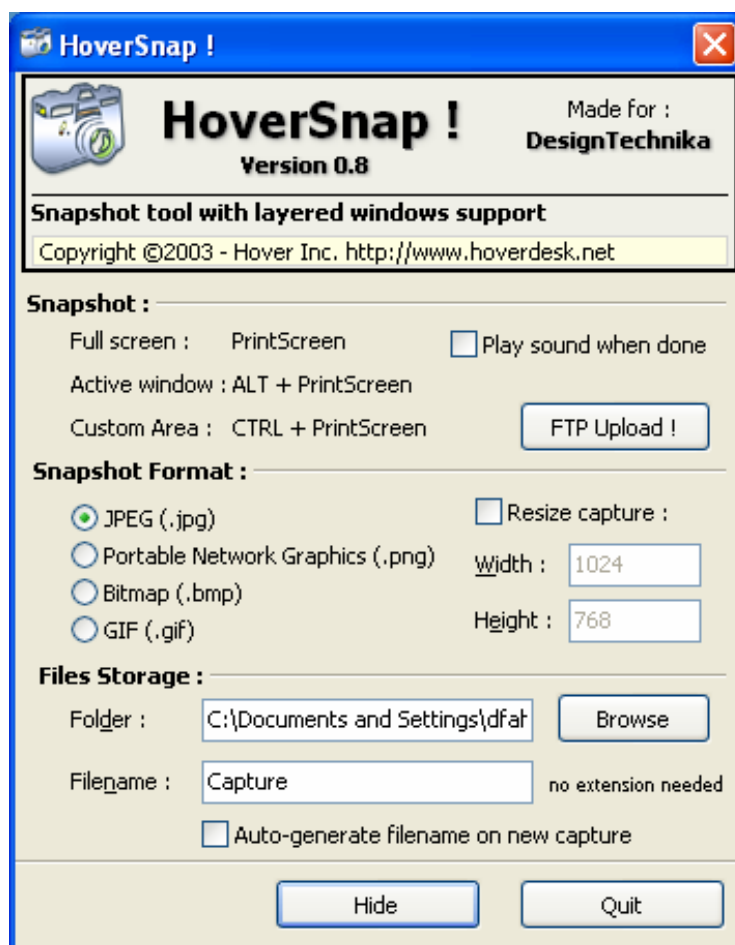
It is recommended that you change the destination folder to your removable evidence collection drive. In addition, checking “Auto-generate filename on new capture” option will automatically create filenames that start with the name in the filename box and automatically add a datetime stamp to the filename. Here is a sample auto-generated filename:

Capture12-12-2005-11.15.55 PM.png

To capture the full screen, the user presses the PrintScreen button. To capture the active window, press ALT+PrintScreen, and to select a custom area, press CTRL+PrintScreen.

With CTRL+PrintScreen, the cursor will change to a crosshair. Move the cursor to the upper left corner, then click and hold the left-mouse button and drag the cursor to the lower right corner. Release the mouse button to take the picture.

Once you have finished the screen captures, you should generate to the MD5 of the screen to ensure they are not modified.





Messenger Password

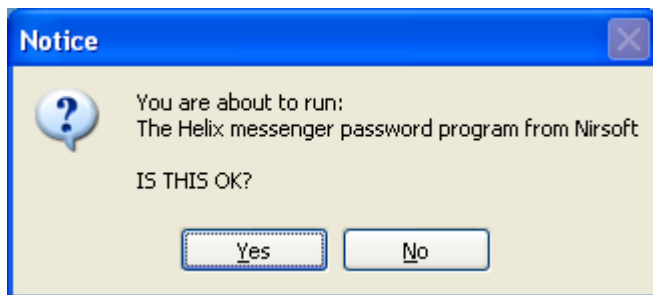
<http://www.nirsoft.net/utis/mspass.html>

From the website: MessenPass is a password recovery tool that reveals the passwords of the following instant messenger applications:

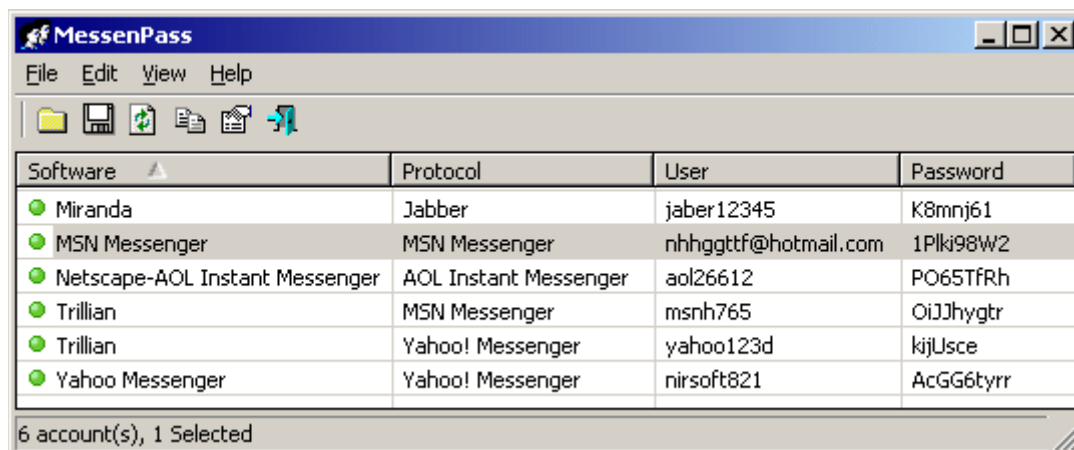
- MSN Messenger
- Windows Messenger (In Windows XP)
- Yahoo Messenger (Versions 5.x and 6.x)
- ICQ Lite 4.x/2003
- AOL Instant Messenger (only older versions, the password in newer versions of AIM cannot be recovered)
- AOL Instant Messenger/Netscape 7
- Trillian
- Miranda
- GAIM

MessenPass can only be used to recover the passwords for the current logged-on user on your local computer. You cannot use it for grabbing the passwords of other users.

When the user clicks on the icon, Helix presents a confirmation message.



Once the user clicks “Yes”, the program will automatically start and display any passwords it can find.



From the website: When you run MessenPass, it automatically detects the Instant Messenger applications installed on your computer, decrypts the passwords they stores, and displays all user name/password pairs that it found in the main window of MessenPass. If from some reason, MessenPass fails to locate the installed Instant Messenger application, you can try to manually select the right folder of your IM application by using 'Select Folders' option (from the File menu). On the main window of MessenPass, you can select one or more password items, and then copy them to the clipboard in tab-delimited format (you can paste this format into Excel or Open-Office Spreadsheet), or save them into text/html files.



Mail Password Viewer

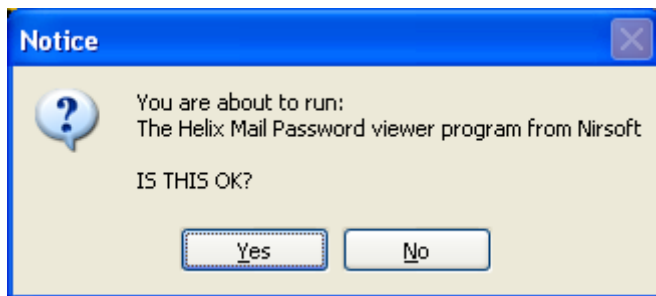
<http://www.nirsoft.net/utils/mailpv.html>

From the website: Mail PassView is a small password-recovery tool that reveals the passwords and other account details for the following email clients:

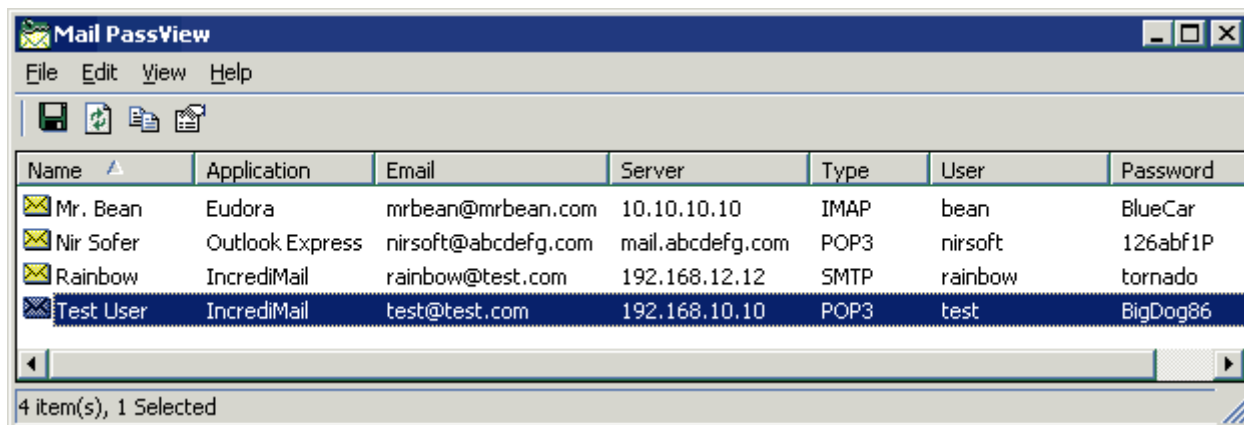
- Outlook Express
- Microsoft Outlook 2000 (POP3 and SMTP Accounts only)
- Microsoft Outlook 2002/2003 (POP3, IMAP, HTTP and SMTP Accounts)
- IncrediMail
- Eudora
- Netscape 6.x/7.x
- Mozilla Thunderbird
- Group Mail Free
- Yahoo! Mail - If the password is saved in Yahoo! Messenger application.
- Hotmail/MSN mail - If the password is saved in MSN Messenger application.
- Gmail - If the password is saved by Gmail Notifier application.

For each email account, the following fields are displayed: Account Name, Application, Email, Server, Server Type (POP3/IMAP/SMTP), User Name, and the Password.

When the user clicks on the icon, Helix presents a confirmation message.



Once the user clicks “Yes”, the program will automatically start and display any passwords it can find.





Protect Storage Viewer

<http://www.nirsoft.net/utis/pspv.html>

From the website: Protected Storage PassView is a small utility that reveals the passwords stored on your computer by Internet Explorer, Outlook Express and MSN Explorer. The passwords are revealed by reading the information from the Protected Storage.

Starting from version 1.60, this utility reveals all AutoComplete strings stored in Internet Explorer, not only the AutoComplete password, as in the previous versions.

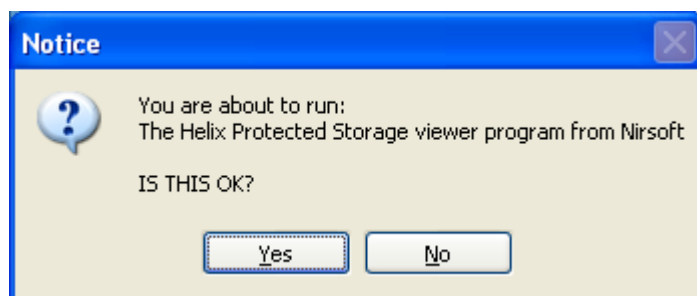
This utility can show 4 types of passwords:

1. **Outlook passwords:** When you create a mail account in Outlook Express or a POP3 account in Microsoft Outlook, and you choose the "Remember password" option in the account properties, the password is saved in the Protected Storage, and this utility can instantly reveal it.
Be aware that if delete an existing Outlook Express account, the password won't be removed from the Protected Storage. In such a case, the utility won't be able to obtain the user-name of the deleted account, and only the password will be shown.
Starting from version 1.50, the passwords of Outlook Express identities are also displayed.
2. **AutoComplete passwords in Internet Explorer:** Many Web sites provides you a logon screen with user-name and password fields. When you log into the Web site, Internet Explorer may ask you if you want to remember the password for the next time that you log into this Web site. If choose to remember the password, the user-name and the password are saved in the Protected Storage, and thus they can be revealed by Protected Storage PassView.
In some circumstances, multiple pairs of user-name and passwords are stored for the same logon window. In such case, the additional passwords will be displayed as sub-items of the first user-password pair. In sub-items, the resource name is displayed as 3 dots ('...')
3. **Password-protected sites in Internet Explorer:** Some Web sites allows you to log on by using "basic authentication" or "challenge/response" authentication. When you enter the Web site, Internet Explorer displays a special logon dialog-box and asks you to enter your user-name and password. Internet Explorer also gives you the option to save the user-name/password pair for the next time you log-on. If you choose to save the logon data, the user-name and the password are saved in the Protected Storage, and thus they can be revealed by Protected Storage PassView.
In this category, you can also find the passwords of FTP servers.
4. **MSN Explorer Passwords:**
The MSN Explorer browser stores 2 types of passwords in the Protected Storage:
 - o Sign-up passwords
 - o AutoComplete passwords

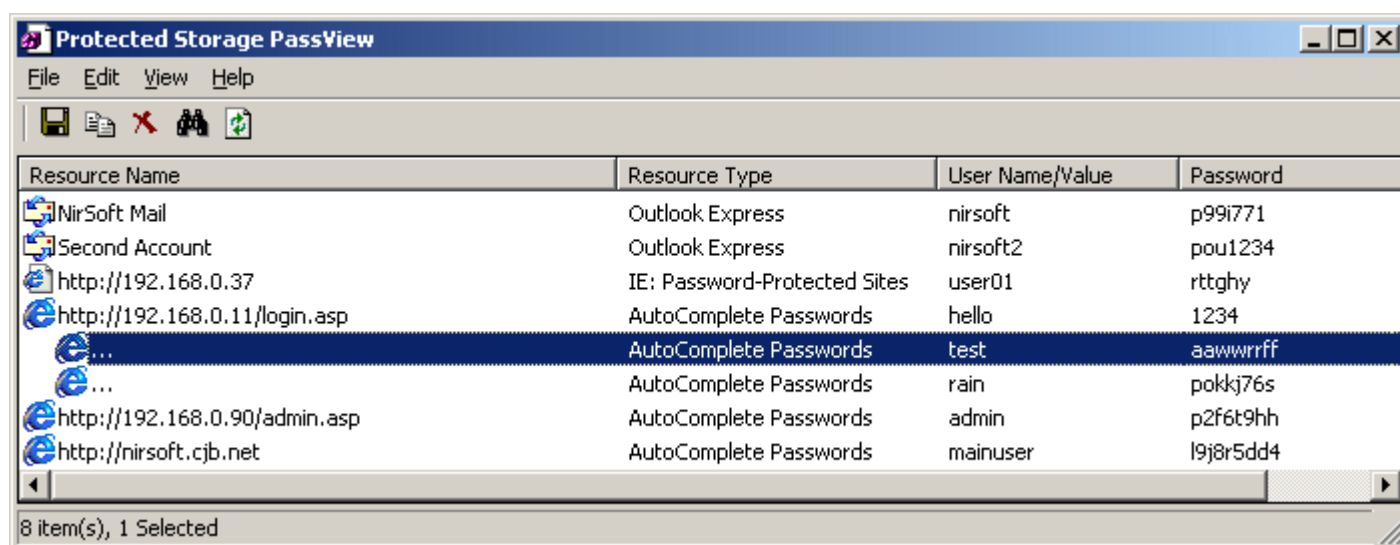
By default, this utility shows all 4 types of passwords. You can select to show or hide a specific type of password, by choosing the right password type from the View menu.

This utility can only show the passwords of the current logged-on user. it cannot reveal the passwords of other users.

When the user clicks on the icon, Helix presents a confirmation message.



Once the user clicks “Yes”, the program will automatically start and display any passwords it can find.



The Protected Storage information is saved in a special location in the Registry. The base key of the Protected Storage is located under the following key:

"HKEY_CURRENT_USER\Software\Microsoft\Protected Storage System Provider"

You can browse the above key in the Registry Editor (RegEdit), but you won't be able to watch the passwords, because they are encrypted. Also, some passwords data are hidden by the operating system.



Network Password Viewer

http://www.nirsoft.net/utils/network_password_recovery.html

From the website: When you connect to a network share on your LAN or to your .NET Passport account, Windows XP allows you to save your password in order to use it in each time that you connect the remote server. This utility recovers all network passwords stored on your system for the current logged-on user.

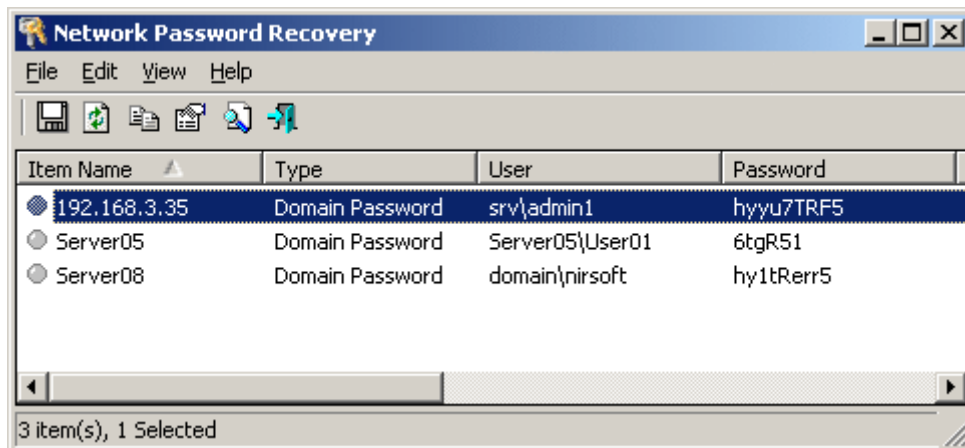
Which passwords this utility can recover ?

- Login passwords of remote computers on your LAN.
- Passwords of mail accounts on exchange server (stored by Outlook 2003)
- Password of MSN Messenger account (Only until version 7.0, for Newer versions - Use MessenPass)

When the user clicks on the icon, Helix presents a confirmation message.



Once the user clicks “Yes”, the program will automatically start and display any passwords it can find.



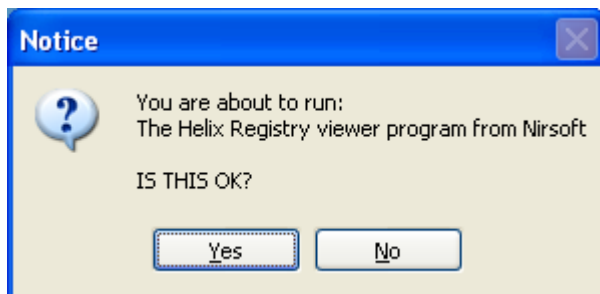


Registry Viewer

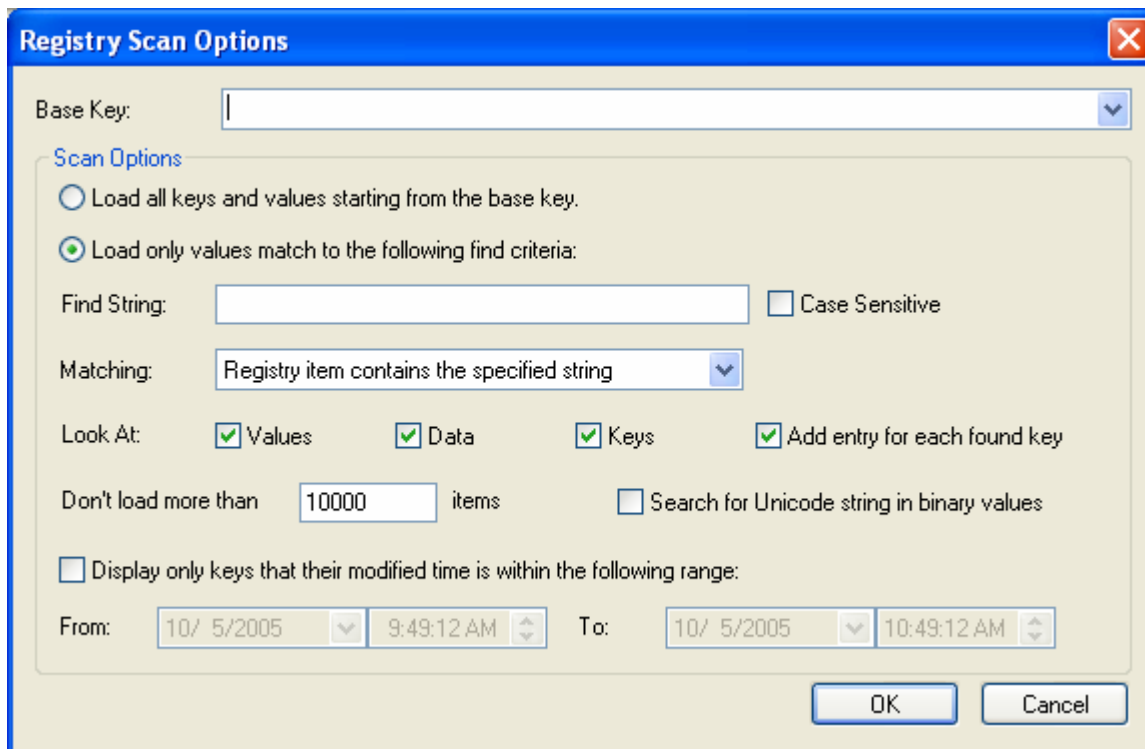
<http://www.nirsoft.net/utils/regscanner.html>

From the website: RegScanner is a small utility that allows you to scan the Registry, find the desired Registry values that match to the specified search criteria, and display them in one list. After finding the Registry values, you can easily jump to the right value in RegEdit, simply by double-clicking the desired Registry item.

When the user clicks on the icon, Helix presents a confirmation message.



Once the user clicks “Yes”, the program will automatically start and display registry scan options page. This can be used to limit the searches, which can greatly speed up the process.



Once the user clicks “OK”, the scanner will display registry keys matching their options.

RegScanner			
File Edit View Help			
Registry Key	Name	Type	Data
HKCU\Software\NirSoft\pspv	ShowMsnExplorer	REG_DWORD	0x00000001 (1)
HKLM\SOFTWARE\Microsoft\Internet Explorer\AdvancedOpti...	Text	REG_SZ	Show Internet E...
HKLM\SOFTWARE\Microsoft\Internet Explorer\AdvancedOpti...	Text	REG_SZ	Automatically che...
HKLM\SOFTWARE\Microsoft\Internet Explorer\AdvancedOpti...	Text	REG_SZ	Enable Install On...
HKLM\SOFTWARE\Microsoft\IE Setup\Options	UninstallDir	REG_SZ	F:\Program Files\...
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninst...	UninstallString	REG_SZ	F:\WINNT\System...
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninst...	UninstallString	REG_SZ	F:\Program Files\...
HKLM\SOFTWARE\JavaSoft\Java Plug-in\1.4.0_01	UseJava2IEExplorer	REG_DWORD	0x00000000 (0)
HKLM\SOFTWARE\JavaSoft\Java Plug-in\1.4.2_05	UseJava2IEExplorer	REG_DWORD	0x00000001 (1)
HKCU\Software\Microsoft\Windows\CurrentVersion\Applets\	User Shell Folders	REG_SZ	My Computer\HK...
939 item(s), 1 Selected			



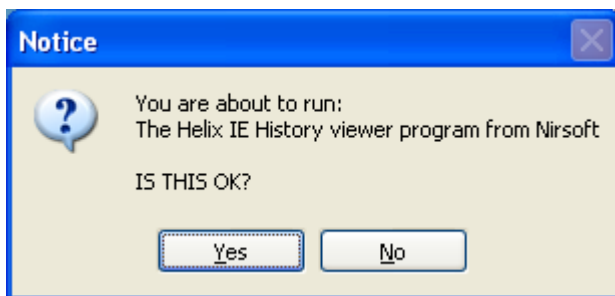
IE History Viewer

<http://www.nirsoft.net/utis/iehv.html>

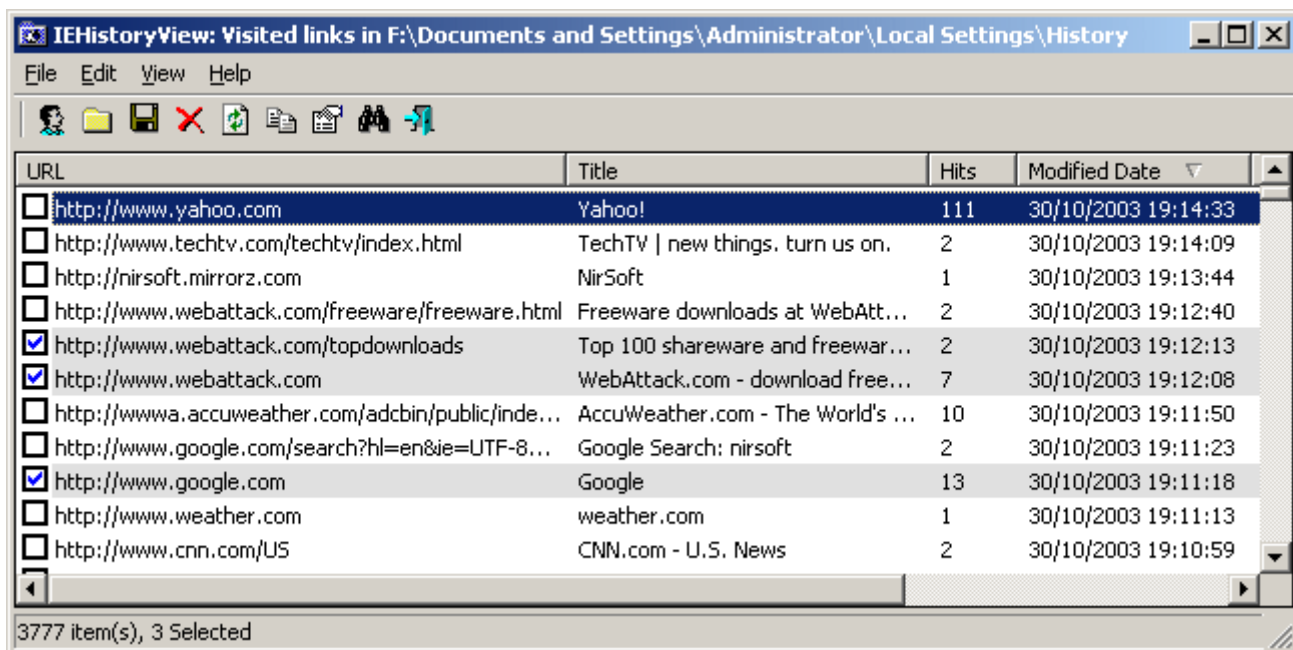
From the website: Each time that you type a URL in the address bar or click on a link in Internet Explorer browser, the URL address is automatically added to the history index file. When you type a sequence of characters in the address bar, Internet Explorer automatically suggests you all URLs that begins with characters sequence that you typed (unless AutoComplete feature for Web addresses is turned off). However, Internet Explorer doesn't allow you to view and edit the entire URL list that it stores inside the history file.

This utility reads all information from the history file on your computer, and displays the list of all URLs that you have visited in the last few days. It also allows you to select one or more URL addresses, and then remove them from the history file or save them into text, HTML or XML file. In addition, you are allowed to view the visited URL list of other user profiles on your computer, and even access the visited URL list on a remote computer, as long as you have permission to access the history folder.

When the user clicks on the icon, Helix presents a confirmation message.



Once the user clicks “Yes”, the program will automatically display the URL history.





Asterisk Logger

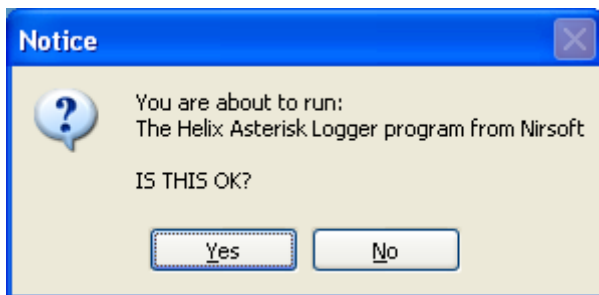
<http://www.nirsoft.net/>

From the website: Many applications, like CuteFTP, CoffeeCup Free FTP, VNC, IncrediMail, Outlook Express, and others, allows you to type a password for using it in the application. The typed password is not displayed on the screen, and instead of the real password, you see a sequence of asterisk ('****') characters. This utility can reveal the passwords stored behind the asterisks in standard password text-boxes.

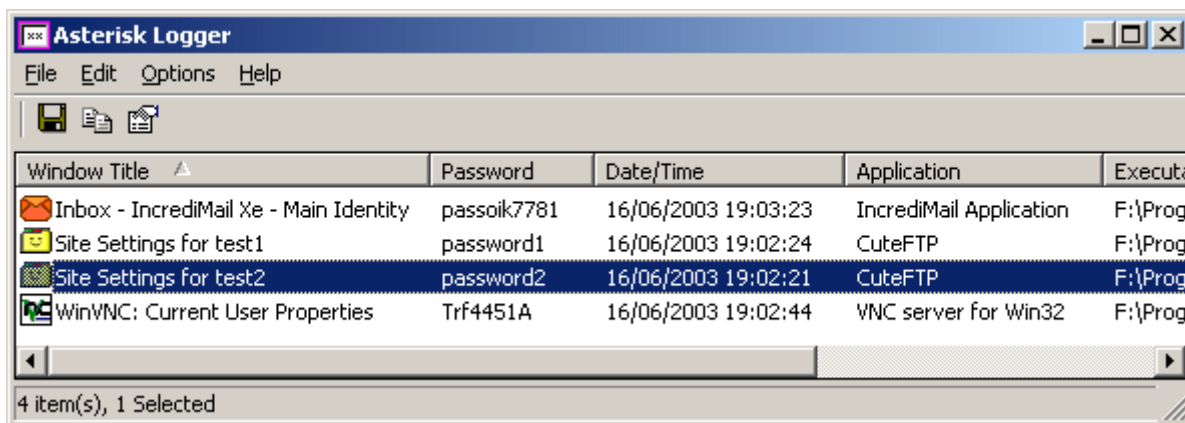
Asterisk Logger is a successor of AsterWin utility. It reveals the asterisk passwords in the same way as AsterWin utility, but it has some advantages over the previous utility:

- You don't have to press a button in order to reveal the asterisk passwords. Whenever a new window containing a password box is opened, Asterisk Logger automatically reveals the password inside the password-box, and add a record to passwords list in the main window of Asterisk Logger.
- Asterisk Logger displays additional information about the revealed password: The date/time that the password was revealed, the name of the application that contains the revealed password box, and the executable file of the application.
- Asterisk Logger allows you the save the passwords to HTML file and to 3 types of text files.

When the user clicks on the icon, Helix presents a confirmation message.



Once the user clicks “Yes”, the program will automatically start and display any passwords it can find.





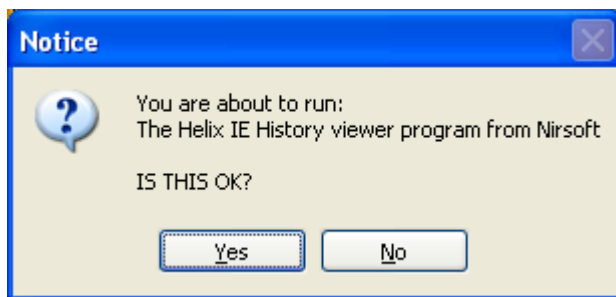
IE Cookie Viewer

<http://www.nirsoft.net/utis/iecookies.html>

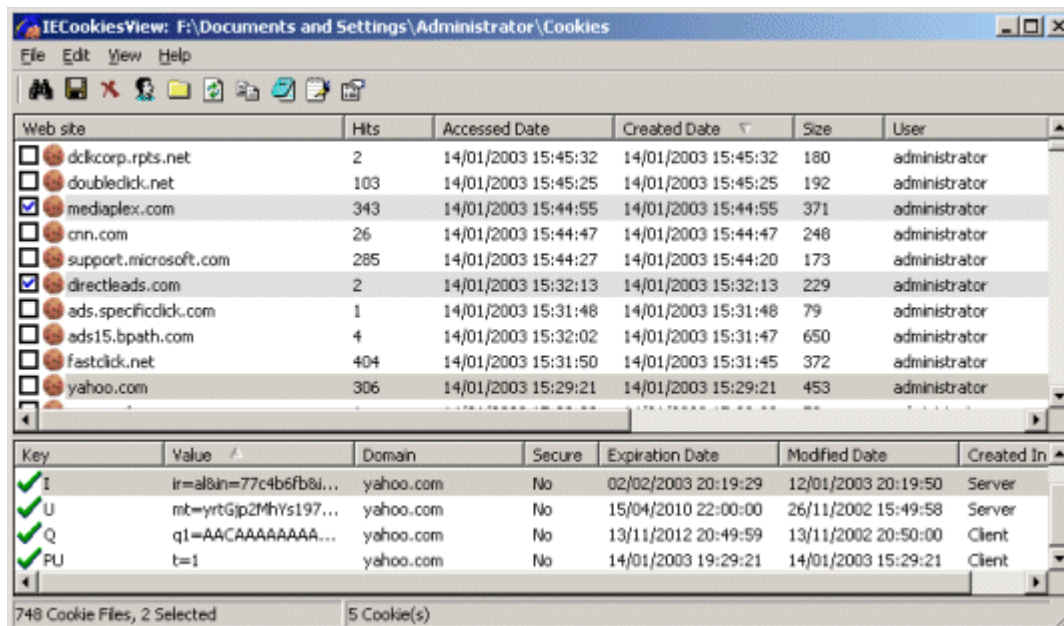
From the website: IECookiesView is a small utility that displays the details of all cookies that Internet Explorer stores on your computer. In addition, It allows you to do the following actions:

- Sort the cookies list by any column you want, by clicking the column header. A second click sorts the column in descending order.
- Find a cookie in the list by specifying the name of the Web site.
- Select and delete the unwanted cookies.
- Save the cookies to a readable text file.
- Copy cookie information into the clipboard.
- Automatically refresh the cookies list when a Web site sends you a cookie.
- Display the cookies of other users and from other computers.

When the user clicks on the icon, Helix presents a confirmation message.



Once the user clicks “Yes”, the program will automatically display the cookies on the system.





Mozilla Cookie Viewer

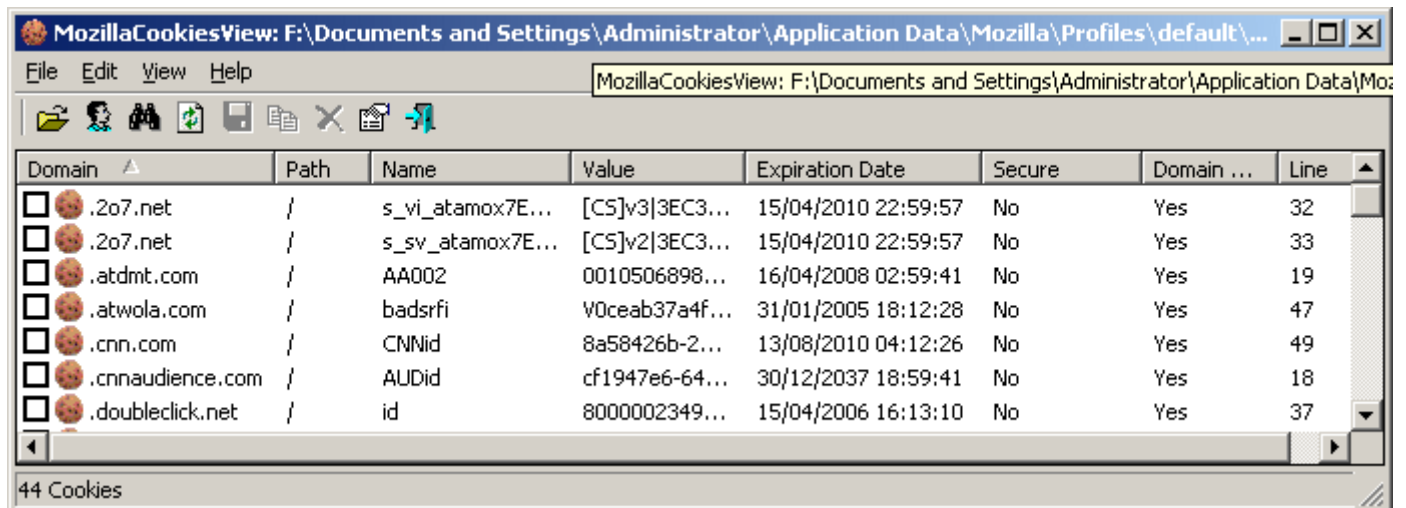
<http://www.nirsoft.net/utils/mzcv.html>

From the website: MozillaCookiesView is an alternative to the standard 'Cookie Manager' provided by Netscape and Mozilla browsers. It displays the details of all cookies stored inside the cookies file (cookies.txt) in one table, and allows you to save the cookies list into text, HTML or XML file, delete unwanted cookies, and backup/restore the cookies file.

When the user clicks on the icon, Helix presents a confirmation message.



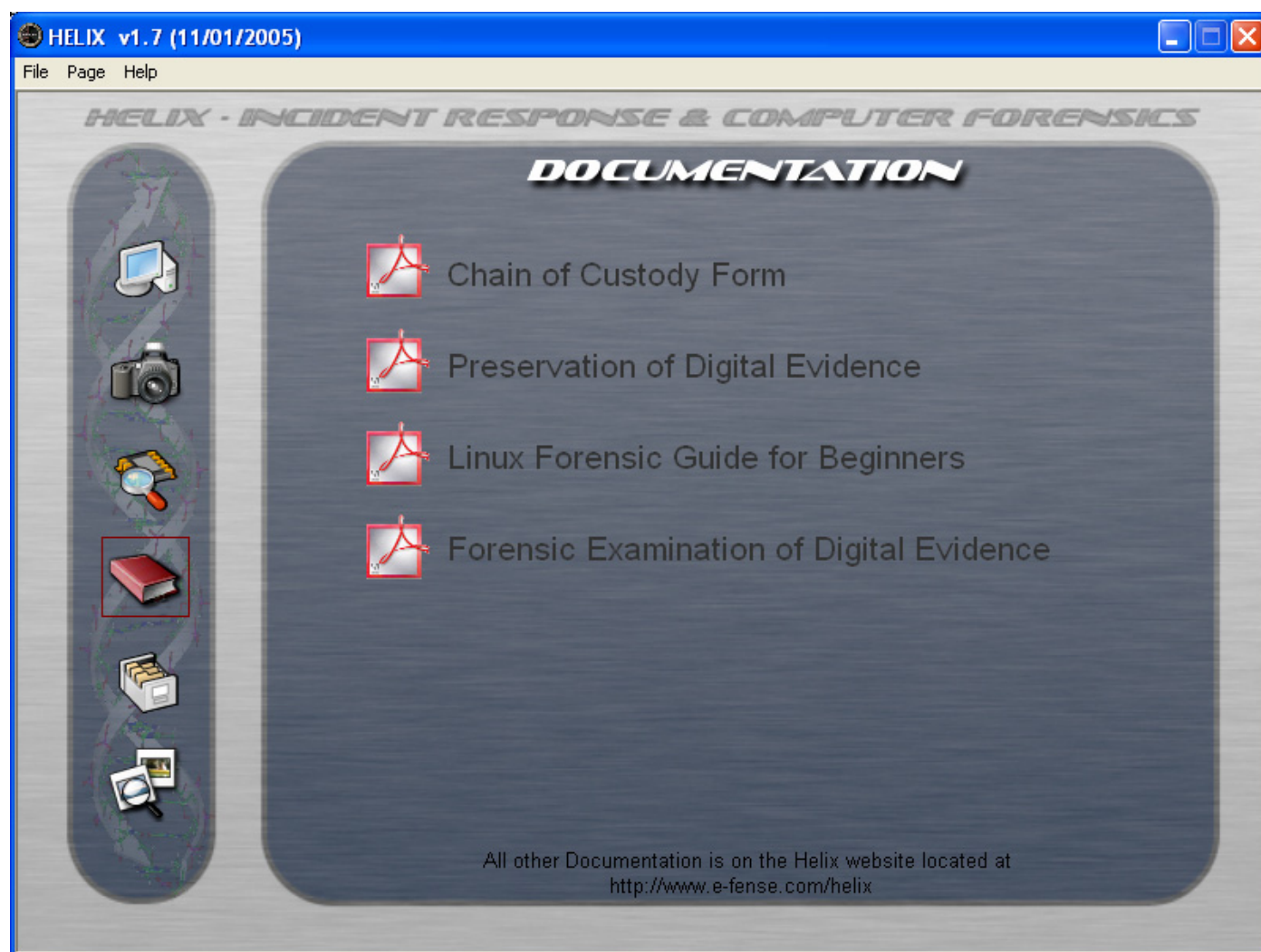
Once the user clicks “Yes”, the program will automatically display the cookies on the system.





Documents pertaining to Incident Response, Computer Forensics, Computer Security & Computer Crime

This section provides the user with access to some common reference documents in PDF format. The documents include a chain of custody form, preservation of digital evidence information, Linux forensics Guide for beginners, and forensic examination for digital evidence guide. These documents are highly recommended, and the investigator should review them before attempting any forensic examination.



These documents can be accessed by clicking on its respective icon.



Chain of Custody

This is a sample chain of custody form used during forensic investigations by e-fense.inc. There should be a separate chain of custody form created for each piece of evidence collected.



ELECTRONIC EVIDENCE CHAIN OF CUSTODY FORM

Case No:

Page: of:

ELECTRONIC MEDIA/COMPUTER DETAILS

Name No: <input type="text"/>	Description: <input type="text"/>		
Manufacturer: <input type="text"/>	Model No: <input type="text"/>	Serial No: <input type="text"/>	

IMAGE DETAILS

Date/Time: <input type="text"/>	Created By: <input type="text"/>	Method Used: <input type="text"/>	Image Name: <input type="text"/>	Segment: <input type="text"/>
Storage Drive: <input type="text"/>	HASH: <input type="text"/>			

CHAIN OF CUSTODY

Tracking No.	Date/Time:	FROM:	TO:	Reason:
	Date:	Name/Org:	Name/Org:	
	Time:	Signature:	Signature:	
	Date:	Name/Org:	Name/Org:	
	Time:	Signature:	Signature:	
	Date:	Name/Org:	Name/Org:	
	Time:	Signature:	Signature:	
	Date:	Name/Org:	Name/Org:	
	Time:	Signature:	Signature:	
	Date:	Name/Org:	Name/Org:	
	Time:	Signature:	Signature:	
	Date:	Name/Org:	Name/Org:	
	Time:	Signature:	Signature:	



Preservation of Digital Evidence

The full title of this paper is “Preservation of Fragile Digital Evidence by First Responders”, and was written by Special Agent Jesse Kornblum, Air Force Office of Special Investigations in 2002.

From the Introduction: The nature of computer based evidence makes it inherently fragile. Data can be erased or changed without a trace, impeding an investigator’s job to find the truth. The efforts of first responders are critical to ensure that the evidence is gathered and preserved in a simple, secure, and forensically sound manner. This paper describes the challenges first responders face and some strategies for dealing with them. As an example, the paper also details a sample tool for first responders to incidents on Windows based computers.

Preservation of Fragile Digital Evidence by First Responders

Special Agent Jesse Kornblum
Air Force Office of Special Investigations
jesse.kornblum@ogn.af.mil
8 August 2002
Digital Forensics Research Workshop



Introduction

The nature of computer based evidence makes it inherently fragile. Data can be erased or changed without a trace, impeding an investigator’s job to find the truth. The efforts of first responders are critical to ensure that the evidence is gathered and preserved in a simple, secure, and forensically sound manner. This paper describes the challenges first responders face and some strategies for dealing with them. As an example, the paper also details a sample tool for first responders to incidents on Windows based computers.

This paper also describes the creation of F.R.E.D, the First Responder’s Evidence Disk, which is included on the Helix disk. While F.R.E.D has been updated significantly since this paper was originally published, this paper provides the basic details on how to preserve as much evidence as possible while disturbing as little as possible.



Linux Forensic Guide for Beginners

One of the first and most extensive guides on using Linux for forensic analysis, “The Law Enforcement and Forensic Examiner Introduction to Linux: A Beginner’s Guide” by Barry J. Grundy, Special Agent of the NASA Office of Inspector General, Computer Crimes Division. First written in 1998, the most recent revision was in 2004.

While this is not Helix specific, it provides a lot of background information for those investigators who are willing to boot into the Helix bootable Linux environment.

The Law Enforcement and Forensic Examiner Introduction to Linux A Beginner's Guide



From the foreword: This purpose of this document is to provide an introduction to the GNU/Linux (Linux) operating system as a forensic tool for computer crime investigators. There are better books written on the subject of Linux (by better qualified professionals), but my hope here is to provide a single document that allows a user to sit at the shell prompt (command prompt) for the first time and not be overwhelmed by a 700-page book.

Tools available to investigators for forensic analysis are presented with practical exercises. This is by no means meant to be the definitive “how-to” on forensic methods using Linux. Rather, it is a starting point for those who are interested in pursuing the self-education needed to become proficient in the use of Linux as an investigative tool. Not all of the commands offered here will work in all situations, but by describing the basic commands available to an investigator I hope to “start the ball rolling”. I will present the commands, the reader needs to follow-up on the more advanced options and uses. Knowing how these commands work is every bit as important as knowing what to type at the prompt. If you are even an intermediate Linux user, then much of what is contained in these pages will be review. Still, I hope you find some of it useful.

Over the past couple of years I have repeatedly heard from colleagues that have tried Linux by installing it, and then proceeded to sit back and wonder “what next?” You have a copy of this introduction. Now download the exercises and drive on.

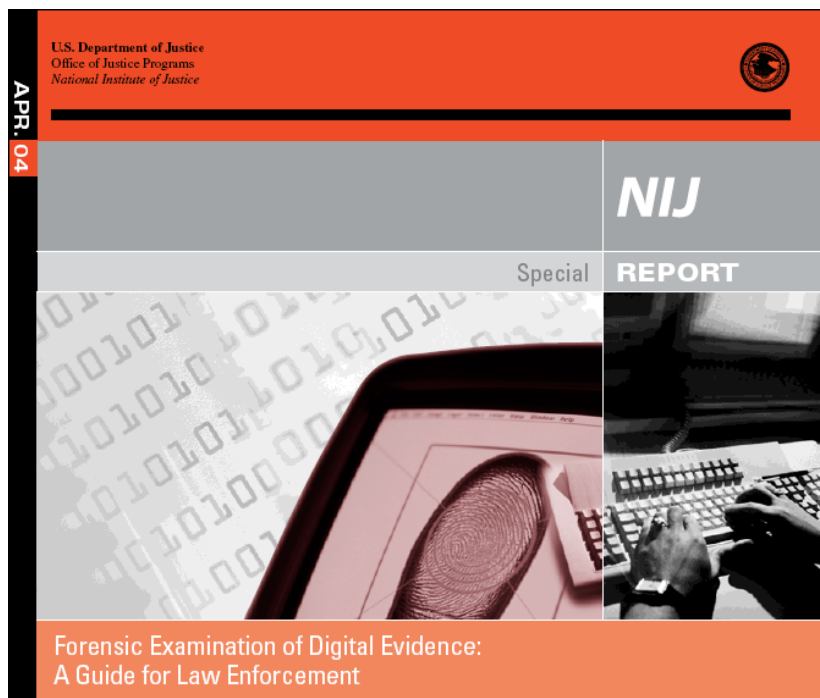


Forensic Examination of Digital Evidence

Published in April 2004 by the U.S. Department of Justice, Office of Justice Programs, National Institute of Justice, this special report “Forensic Examination of Digital Evidence: A Guide for Law Enforcement”, provides the digital forensic investigator a detailed guide on how to collect, process, and document digital evidence.

From the foreword: To assist law enforcement agencies and prosecutorial offices, a series of guides dealing with digital evidence has been selected to address the complete investigation process. This process expands from the crime scene through analysis and finally into the courtroom. The guides summarize information from a select group of practitioners who are knowledgeable about the subject matter. These groups are more commonly known as technical working groups. This guide is the second in a series.

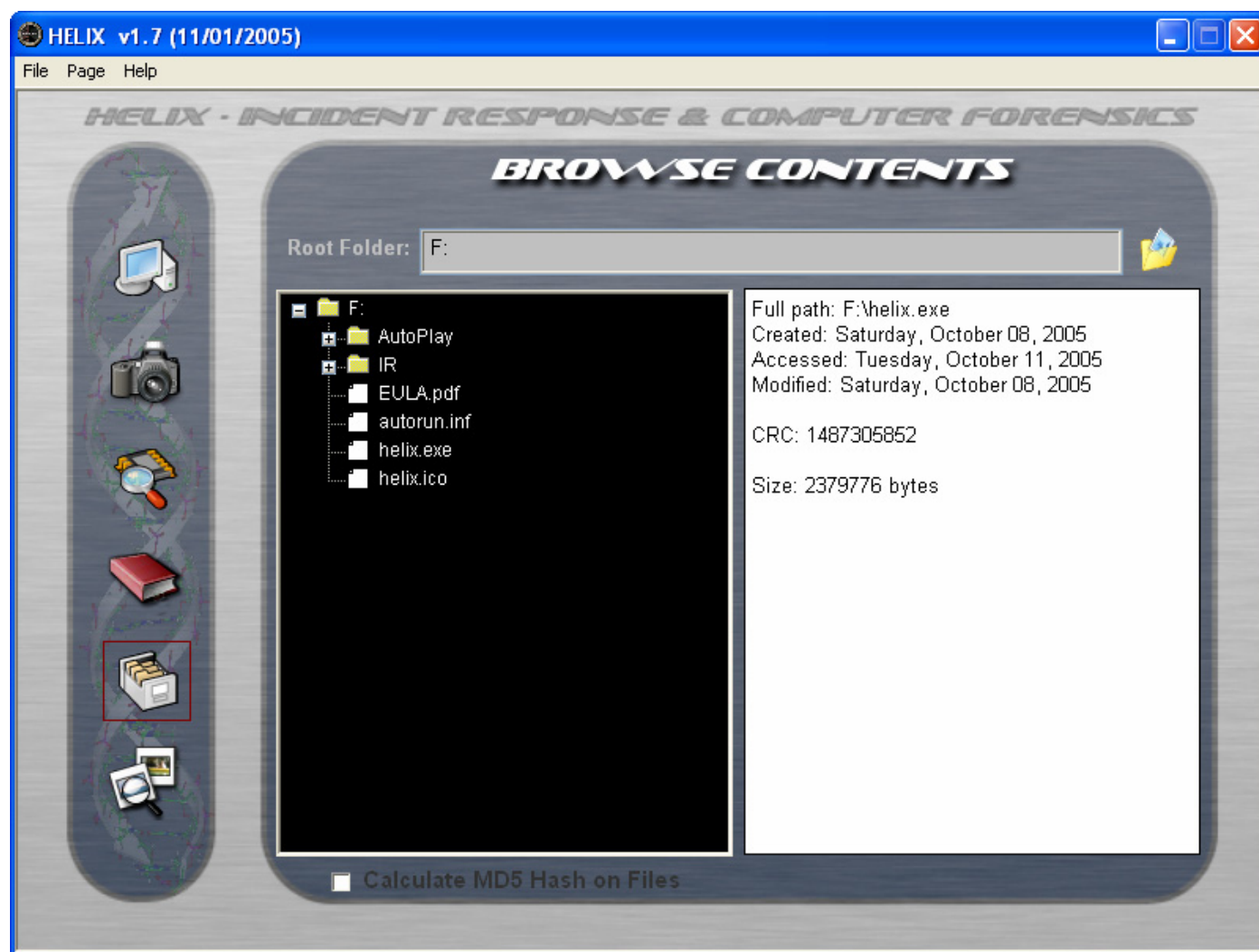
The first guide, Electronic Crime Scene Investigation: A Guide for First Responders, is available through the National Institute of Justice Web site at <http://www.ojp.usdoj.gov/nij/pubs-sum/187736.htm>.





Browse contents of the CD-ROM and Host

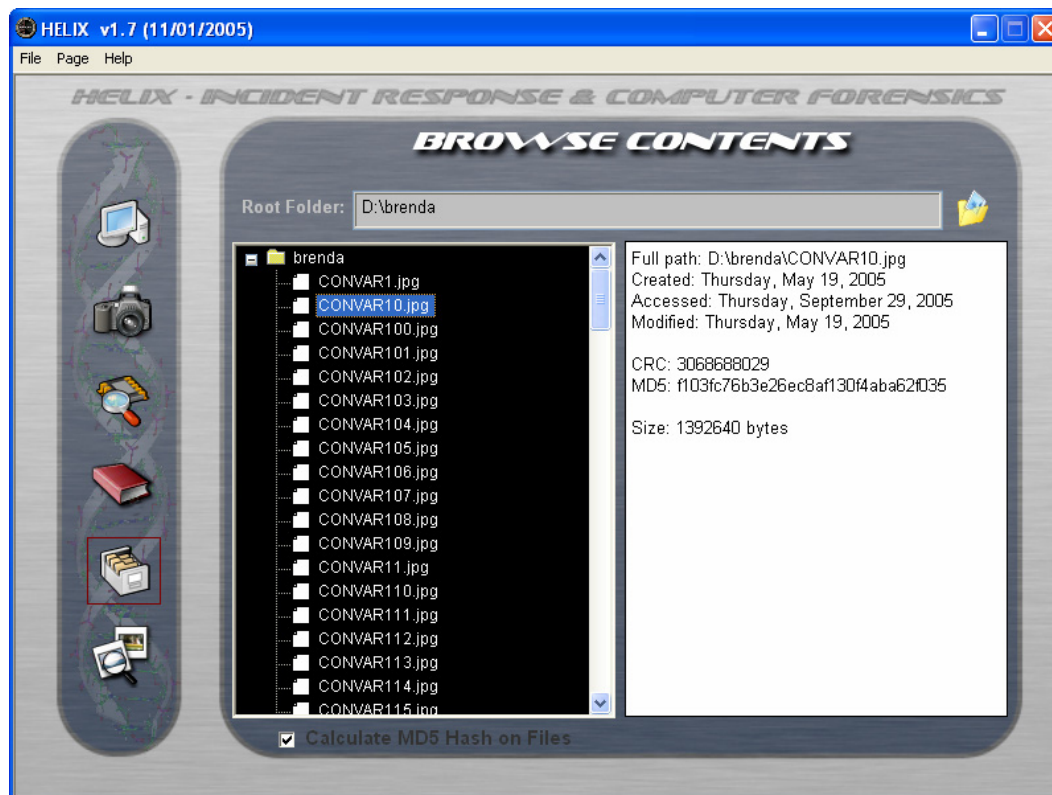
This is a simple file browser that will provide the investigator with information about the selected file. It will display the filename, created, accessed and modified dates, Attributes, CRC, MD5 and the file size.



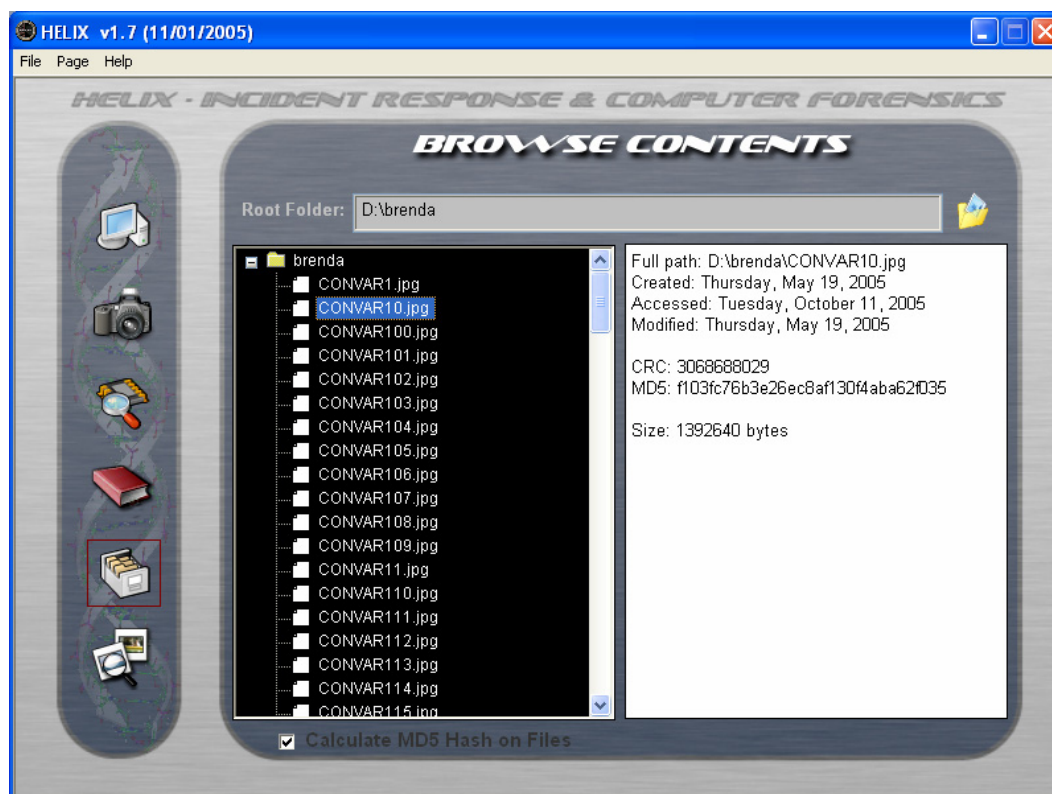
If the “Calculate MD5 hash on files” is selected, the MD5 of the hash will also be displayed with the rest of the file information. It is turned off by default, since it can sometimes take a while to generate the MD5 for very large files.

Due to the nature of the windows operating system, the first time you select a file (on any read/write media, such as a hard drive) it will display the access date of the last access. If you select the same file again, it will display the date and time of the previous access. This is a feature of the windows operating system, and can not be easily prevented. This is one of the problems with examining a live system – the investigator’s actions may modify the system.

Here is an example:



In this list of recovered files, we see that the accessed date on CONVAR10.jpg is Thursday, September 29, 2005. If another file is selected, and then CONVAR10.jpg is selected again, we will see that the accessed data has changed to the today's date. Everything else has remained the same.





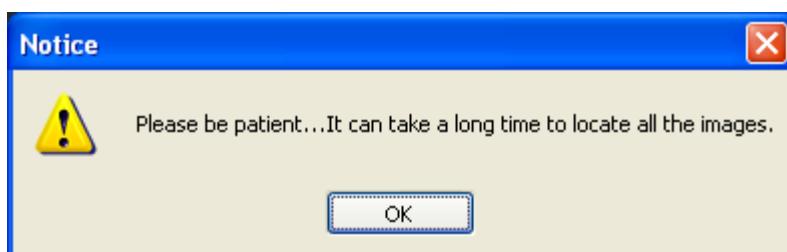
Scan for Pictures from a live system

This tool will allow the investigator to quickly scan the system to see if there are any suspicious graphic images on the suspect system. Many different graphic formats are recognized, and displayed as thumbnails. This feature was added to support “Knock and Talks.” This allows a parole officer to preview a system for graphic images that may violate a parole.

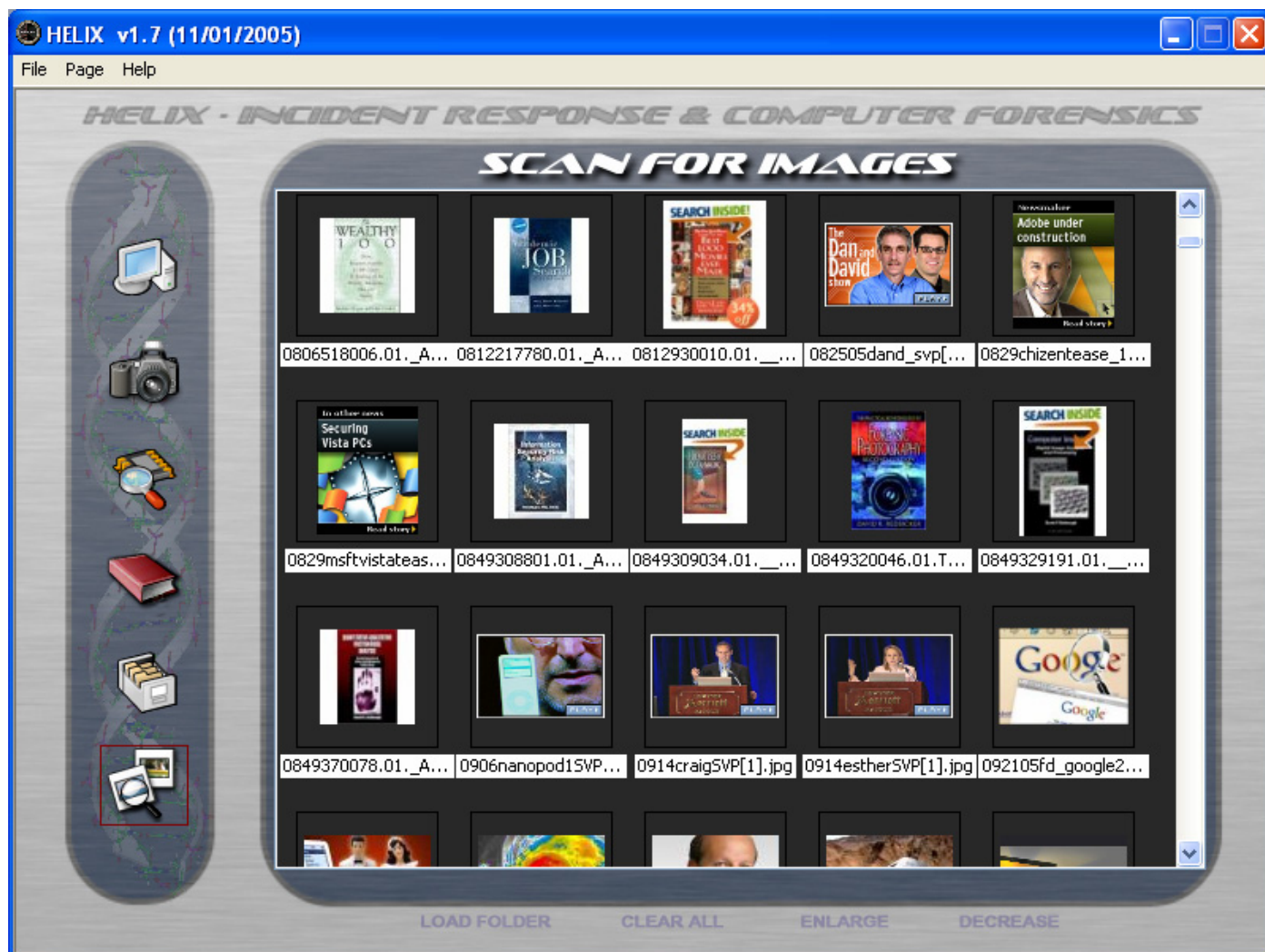
When the scan for pictures icon is selected, this window appears:



The investigator should select “Load Folder” and select the drive they wish to examine. Be aware that depending on the size of the hard drive, the amount of memory, and the speed of the system, this can take a while. A reminder windows pops up to inform the investigator. Scanning will not begin until the “OK” button is pressed.



Investigators will need to examine each drive letter separately.



Double-clicking on any thumbnail will open the image in the local viewer. You can enlarge or decrease the size of the thumbnails by clicking “Enlarge” or “Decrease”. Be advised that this will increase or decrease the size of all the thumbnails, and may take a few moments to complete, depending on the number of thumbnails.

Also be aware that this application will change the last access time on just about every file on the system, since it examines the file headers to determine if the file is a graphic.

Note: The scan image utility does not currently find .gif files due to an old license restriction, but will be updated soon.

Exiting Helix

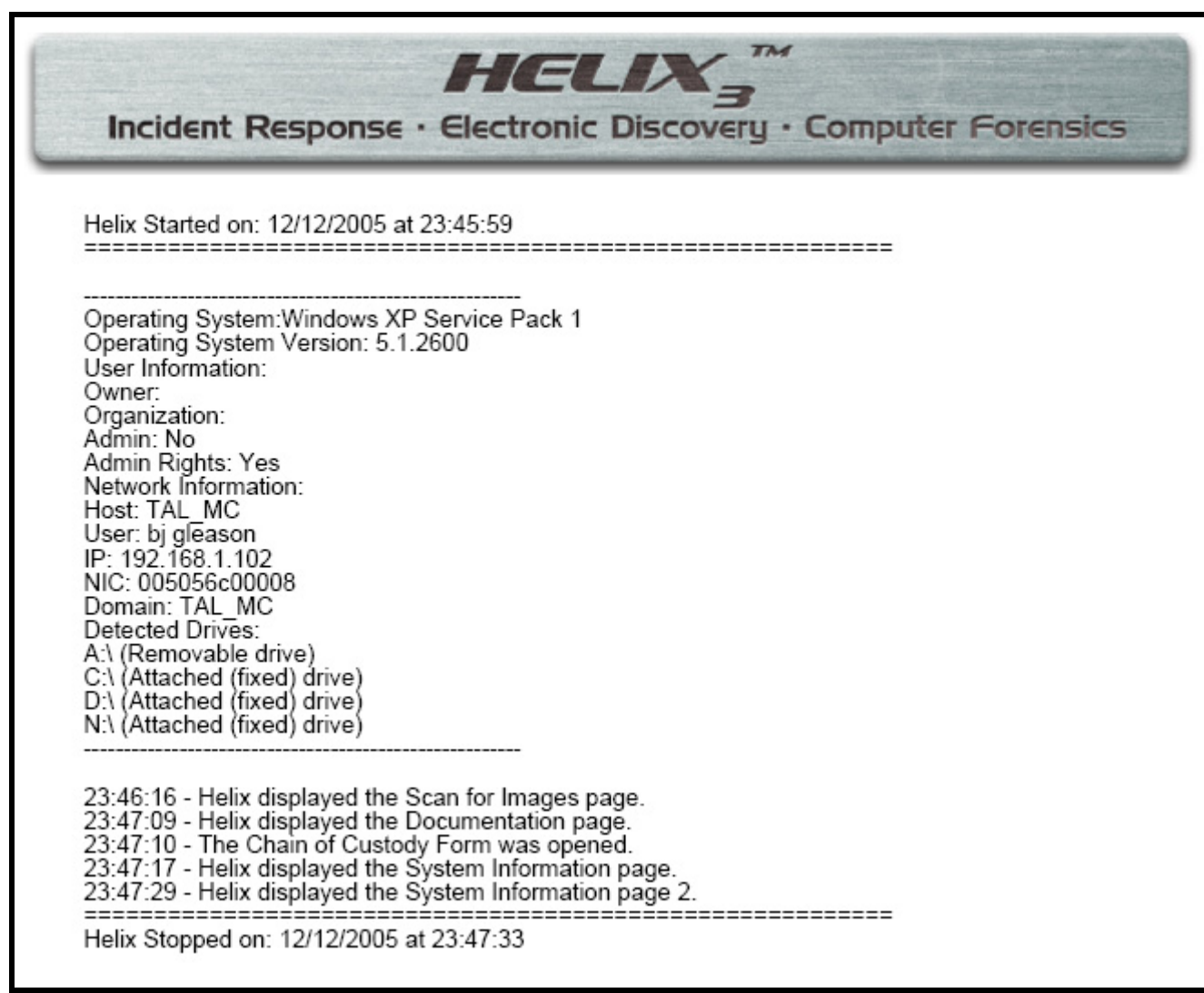
There are several ways to exit the Helix application.

1. File / Exit for the menu bar – this will prompt to save a PDF of your transactions
2. Click the close windows button - this will prompt to save a PDF of your transactions
3. Right-click on the Helix icon in the system tray – this will **NOT** save your transactions.

Note that the first two ways to exit will save a copy of all your transactions, while exiting from the system tray icon will not.

If you chose to save the output, you will be prompted where to save the file. It should be saved on a network share or a removable evidence collection drive to prevent any contamination of the suspect computer. The default filename is Helix_Audit_Log.pdf.

Sample Output





Helix from the Command Line (Windows Side)

While the graphical user interface for Helix on the Windows side make it very easy to run many of the tools, there are some who argue that the GUI tramples too much memory, and therefore contaminates the crime scene. It is possible to run many of the Windows tools (but not all of them) from a command line.

Note: When performing a live preview of a system, many of the actions taken can and will modify information on the suspect machine. This method should only be used when the system can not be taken offline.

Not Starting the GUI

The Helix GUI is configured to start automatically via the autorun.inf file on the CD. To prevent the GUI from starting automatically, there are several techniques that can be used:

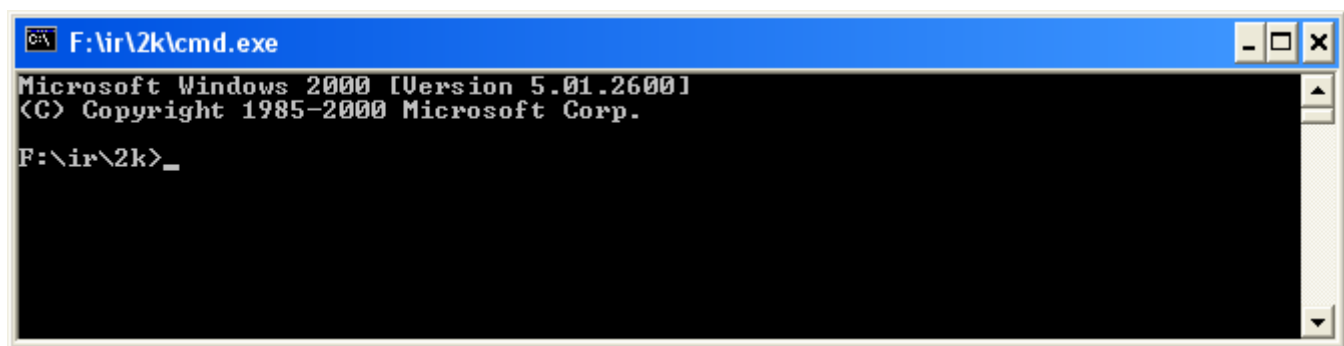
1. Hold down the SHIFT key when the Helix CD is inserted to the system.
2. Disable the autorun on the target system
3. Remaster the Helix CD, and remove the autorun.inf file.

Starting a command shell

The command shell you need to start depends on the host operating system. The command shells are located in the `\ir\` directory of the CD.

Windows NT	<code>\ir\nt</code>
Windows 2000	<code>\ir\2k</code>
Windows XP	<code>\ir\xp</code>
Windows 2003	<code>\ir\2k3</code>

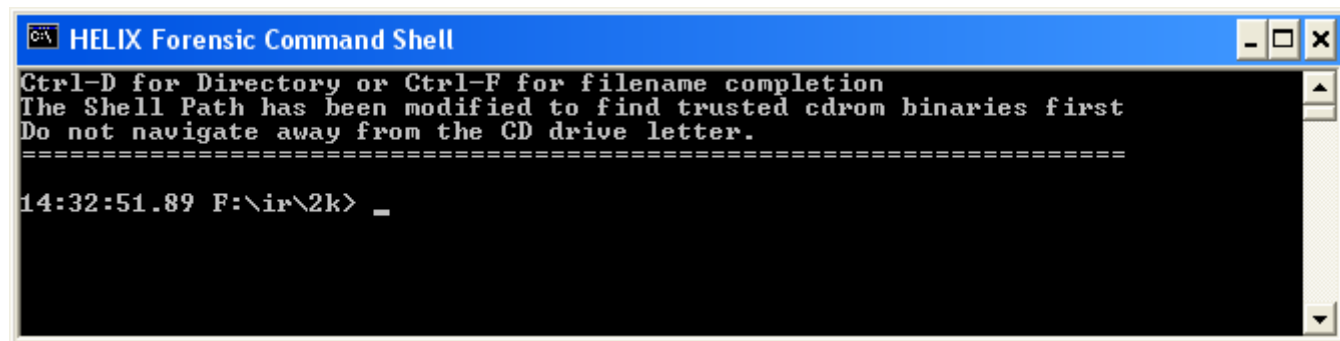
Go to the directory for your specific OS, and execute the CMD.EXE. This is a static binary on a CD ROM, so it will not be compromised by any malware running on the system.



Once the command shell is running, you need to execute an environment configuration file, to set the path and environment variables to point to utilities on the CD, and not on the target system.

Inside the command shell, execute the command: `cmdenv.bat`

This will reset the system path to point to the CD, along with adding paths for the various forensic tools.



You should now be running in the same environment as if you selected the “Command Shell” option from the second page of the Incident Response menu in the Helix GUI.

Tools available from the Command Line

The follow is a list of some of the tools available from the command line in Windows. Some of these tools launch a GUI for the application; others are pure command line tools.

Some of these tools are very powerful, and can be very destructive, so be very careful when using them.

2hash-v0-2w9x	MD5 & SHA1 parallel hashing (Windows 9x)
2hash-v0-2w9p	MD5 & SHA1 parallel hashing (Windows XP)
AccessEnum	full view of your file system and Registry security settings
AFind	NTFS Last Access Time Finder
Attacker	A TCP/UDP port listener
Audited	NTFS SACL Reporter - Finds audited files
Autoruns	Displays what programs are configured to run during at boot or login
Autorunsc	Command line version of Autoruns
Bintext	A file text scanner
Bopping	Back Orifice Pinger
browselist	lists computer names, and the roles they play in the network
CIScan	Identify potentially vulnerable Cisco devices
cmdline	Shows processes, process IDs, paths, and commandline parameters
cryptcat	netcat enhanced with twofish encryption
DACLchk	Dumps any ACL that has Denied and Allowed ACE's
Datetime	System Date and Time
dd	Unix disk duplicator
DiskView	shows you a graphical map of your disk
DSScan	scan multiple IP ranges to detect vulnerable systems
dumpusers	dump account names and information
EFSDUMP	Dump information about Win2K encrypted files
efsview	lists the users who decryption keys or recovery keys for an EFS files
etherchange	change the Ethernet address of the network adapters in Windows

filehasher	calculates the MD5 or SHA hash for a file
Filemon	monitors and displays file system activity on a system in real-time
FileStat	Dumps NTFS security, file, and stream attributes
Filewatch	Monitor specific files
foremost	Carve files based on header and footer
FPipe	TCP/UDP port redirector
Fport	TCP/IP Process to Port Mapper
fred	First Responder's Evidence Disk (FRED)
fruc	First Responder Utility (FRU)
galleta	Cookie analyzer for Internet Explorer
gplist	lists information about the applied Group Policies
gsd	Displays the DACL of any Windows NT service
Handle	GUI-based DLL and handle viewer
HFind	Hidden file finder with last access times
Hunt	SMB share enumerator and admin finder
iplist	Enumerates the ip's of the computer
ircr	Incident Response Collection Report (IRCR2)
Listdlls	List all the DLLs that are currently loaded
listmodules	lists the modules (EXE's and DLL's) that are loaded into a process
Livekd	Use Microsoft kernel debuggers to examine a live system.
Ins	searches for NTFS streams
LogonSessions	List active logon sessions
lsadump2	Dump LSA secrets
macmatch	search for files by their MAC times without changing them
md5deep	Recursive MD5 sum with db lookups.
md5sum	MD5 generator
MessengerScan	Scan systems for MS Messenger vulnerability
nbname	Decodes and displays NetBIOS Name traffic
nc	Netcat
NTFSINFO	Displays information about NTFS volumes
NTLast	Retrieve login information
openports	View all open TCP and UDP ports
pasco	Forensic tool for Internet Explorer Analysis
pdd	Imaging tool for forensic analysis of Palm OS platform devices
periscope	PE file inspection tool
pmdump	dump the memory contents of a process to a file
Procexp	Lists files, registry keys and other objects processes have open
procinterrogate	Displays process list, associated dlls, md5 sum of each dll.
promiscdetect	checks if your network adapter(s) is running in promiscuous mode
Psexec	Execute processes remotely
Psfile	See what files are opened remotely
Psgetsid	display the SID of a computer or a user
Psinfo	gathers key information about the local or remote Windows system
Pskill	Terminate local or remote processes
Pslist	Show information about processes and threads
Psloggedon	Show users logged on to a system
Psloglist	command-line event-log viewer
Pspasswd	change an account password on the local or remote systems
Psservice	service viewer and controller
Psshutdown	command-line shutdown utility

Pssuspend	Suspend and resume processes
pstoreview	lists the contents of the Protected Storage
Psuptime	shows you how long a system has been running since its last reboot
pwdump2	Dump the SAM database
PwDump3e	Obtain LM password hashes from a server
reg	Command-line registry manipulation utility
Regmon	show you which applications are accessing your Registry
rifiuti	"Recycle BIN" analyzer.
rmtshare	command-line utility that allows you to set up or delete shares remotely
secreport	Security Reports (SecReport)
ServiceList	list running services on a system
Servicelist	utility for querying service status from a workstation or server
SFind	Alternate Data Stream Finder
sha1deep	Recursive sha1 sum with db lookups.
sha256deep	Recursive sha256 sum with db lookups.
Showin	Display specific window information
sid2user	Convert SID to User ID
SI	command-line port scanner
Streams	Displays which NTFS files have streams associated with them
strings	Search for ANSI and UNICODE strings in binary images
tcpvcon	command-line version of TCPView
Tcpview	View all open TCP and UDP endpoints
tigerdeep	Recursive tiger sum with db lookups.
Trout	Traceroute and Whois program
user2sid	Convert User ID to SID
UserDump	Command line tool to dump basic user info
volume_dump	Displays information on the logical volumes on a system
wft	Windows Forensics Toolchest (WFT)
whirlpooldeep	Recursive whirlpool sum with db lookups.
winfo	queries the host for information made available by a NULL session
winrelay	TCP/UDP forwarder/redirector that works with both IPv4 and IPv6
wipe	Secure file deletion



Bootable Helix (Linux Side)

One of the greatest benefits of Helix is the bootable Linux forensic environment.

Linux is a computer operating system and its kernel. Designed by Linux Torvalds while still in college, Linux is one of the most prominent examples of free software and of open-source development: unlike proprietary operating systems such as Windows and Mac OS, all of its underlying source code is available to the public for anyone to freely use, modify, improve, and redistribute (Wikipedia, 2006b).

Helix is based on a version of Linux called Knoppix. Developed by Klaus Knopper, Knoppix is a version of Linux that runs entirely off a CD (Wikipedia, 2006a). This is called a LiveCD. In 2003, Drew Fahey of e-fense.com modified Knoppix so that it could be used for digital investigations and forensic analysis. Helix, like Knoppix, will boot into a self-contained Linux environment. This environment has been tweaked for forensic purposes. While there are many distributions of Knoppix such as Knoppix-STD, Helix only concentrates on Incident Response and Forensics.



Helix will boot on all x86 architectures which make up a majority of the computers in the world. It is for this reason that Helix for the immediate future will remain on a CDROM. Almost every computer in the world has a CDROM, but most do not have DVD's, etc. While derivatives like Helix USB will be forthcoming, it is most stable on the CD platform.

Learning more about Linux

Linux is a rich, complex, operating system. This segment of the manual only covers the Forensic and Incident response tools included with the Helix environment. There are numerous books, magazines, and tutorials available to help you learn Linux. While it is not Windows, it is similar in many ways, and should be all that long before the user becomes comfortable working with it.

For more information on Linux, take a look at some of these websites:

<http://www.linux.org> – All about Linux

<http://www.us.debian.org/> - the base installation that Helix is based on.



Forensic Topics (Linux Side)

Write Protecting Media

To ensure that digital evidence media is not modified, before it is placed into a system for duplication, it should be set to "Read Only", "Locked" or "Write Protect", to prevent accidental modification. By default, Helix set all devices as read only, so they can not be easily modified.

However, it is still recommended to hardware write protect digital media whenever possible. See the ***Forensic Topics (Windows Side)*** for a complete list of media and the various ways to write protect them.

Setting a USB device to Read/Write

If you are using the Linux side of Helix to search for all collect evidence, or imaging a drive, you may want to save some files to the a USB thumb drive or flash drive. When the USB device is inserted, Helix should recognize it and put an icon for it on the desktop. However, in keeping with the Helix philosophy, the device will be read-only. USB devices are normally mounted to the /media/sda or /media/sda1 mount point. To change the device so that data can be save to it, open a command shell and execute the following commands:

```
umount /media/sda1
```

This makes sure that the drive is unmounted. It may generate an error if the drive is not mounted, but that is ok.

```
mount -o rw /dev/sda1 /media/sda1
```

This will mount the drive as read/write, and you can now you can write to it.

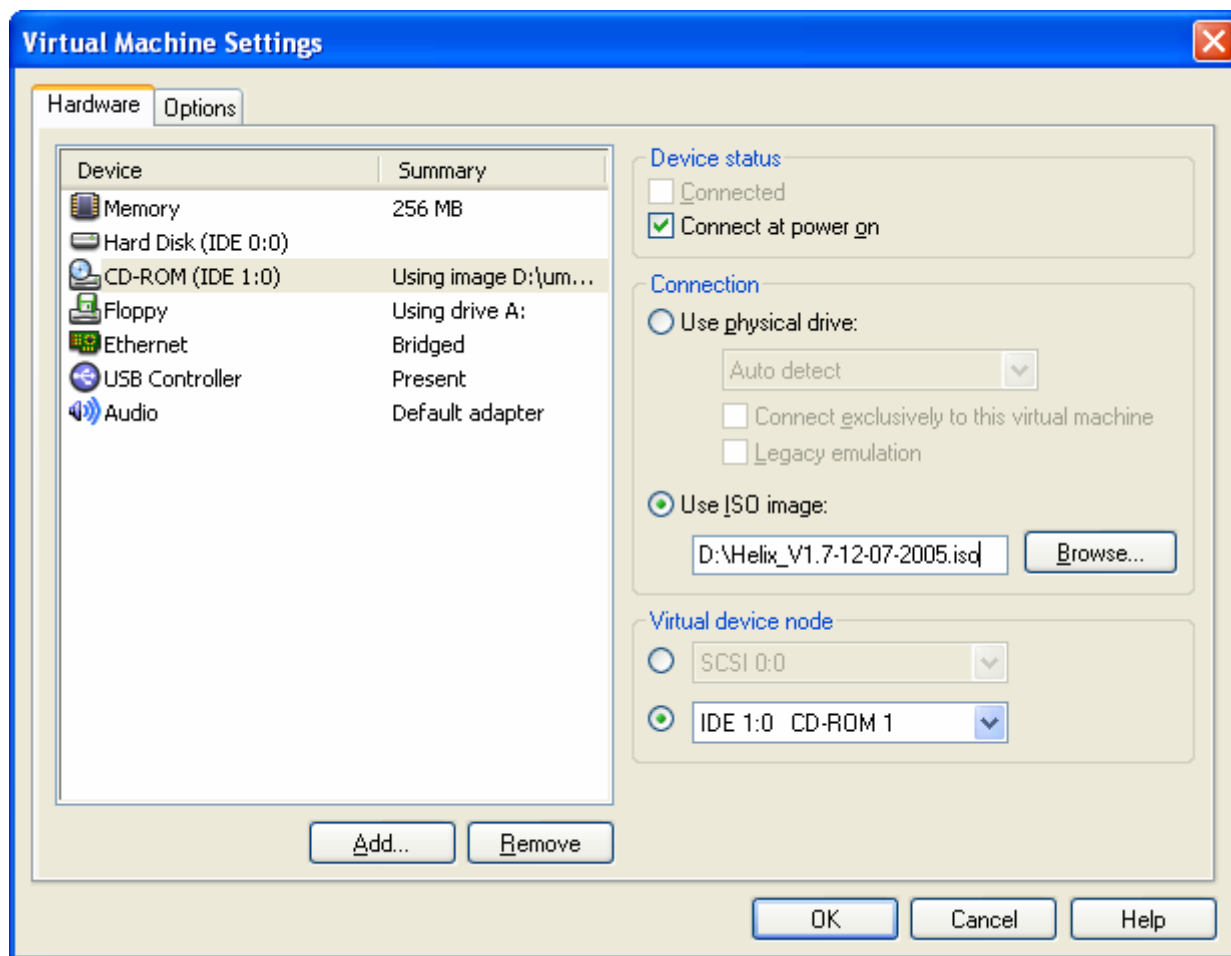
```
umount /media/sda1
```

Finally, this will unmount the drive, so it can be removed.

Using Helix in VMWare

If you have access to VMWare (<http://www.vmware.com/>), you can "boot" Helix in the Virtual machine. This will give you an opportunity to try the Linux side of Helix without having to reboot you physical machine.

To use Helix in VMWare, create a Virtual Machine. Here are some sample settings, but these can be modified to work better with your own system configuration.



To boot Helix in the Virtual CD-ROM, you can use the physical drive, but the process is faster using the downloaded ISO image. As shown in the example above, the CD-ROM is configured to use the ISO image. When the virtual machine boots, it will boot from the virtual CD-ROM, and Helix will start up.

Of course, there are some limitations, and complexities of using Helix this way. To really learn Helix you need to load the Helix CD into a real machine, and boot it up.

The Helix Filesystem

Obviously Helix is run from a CDROM which uses the ISO9660 standard for its file system. This has its benefits and drawbacks. The largest benefit is that files on the CDROM cannot be changed making it a permanent storage and security solution. You can use the CD in an incident response role and not have to worry about your files being altered. However this also is the largest drawback in that you cannot change any of the files once they are in place.

Why would you want to change a file? Well the biggest reason is to updated configuration settings. So how do you do this on a write once medium like a CDROM? There are actually two ways to do this, the old way and the new way. The old way consisted of linking the files that needed to be changed into a specific area that was stored into a RAM disk. This worked but was a poor solution do to the amount of files that would need to be linked and the fact that the files

would have to be linked prior to burning the iso file to a CDROM. The new way involves using a file system overlay called Unionfs.

Unionfs

Unionfs merges directories into a single unified view. A collection of merged directories is called a union, and each physical directory is called a branch. Each branch is assigned precedence and a branch with a higher precedence overrides a branch with a lower precedence. Unionfs operates only on directories as opposed to devices (like cowloop). If a directory exists in two underlying branches, the contents and attributes of the Unionfs directory are the combination of the two lower directories. Unionfs automatically removes any duplicate directory entries.

Copy-On-Write Unions

Unionfs can mix read-only and read-write branches. In this case, the union as a whole is read-write, and Unionfs uses copy-on-write semantics to give the illusion that you can modify files and directories on read-only branches.

If the CD-ROM is mounted on /mnt/cdrom, and an empty directory is created in /tmp/cd, then Unionfs can be mounted as follows:

EXAMPLE – Copy-On-Write Union

```
# mount -t unionfs -o dirs=/tmp/cd,/mnt/cdrom none /mnt/cdrom-rw
```

When viewed through /mnt/cdrom-rw, it appears as though you can write to the CD-ROM, but all writes actually will take place in /tmp/cd. Writing to read-only branches results in an operation called a copyup. When a read-only file is opened for writing, the file is copied over to a higher-priority branch. If required, Unionfs automatically creates any needed parent directory hierarchy. With a simple modification, Helix uses Unionfs as an overlay mount. Meaning that, the Helix CDROM becomes writable by replacing /HELIX/ /tmp/HELIX with the unified view:

EXAMPLE – Copy-On-Write Union with Overlay

```
# mount -t unionfs -o dirs=/home/cpw/linux:/usr/src/linux=ro \  
> none /home/cpw/linux
```

Helix automatically uses Unionfs and creates the union upon bootup and initialization. You will be able to “write” to all of the directories. This makes installing software or updating configuration files very simple. You can also control the behavior of the Unionfs by using the following boot options:

unionfs - allows you to make changes to the running system by storing changes in ram

unionro - allows you to restore changes to the running system from the supplied partition or loopback filesystem. This partition or loopback filesystem will be mounted read only so it will not allow more changes to it

unionrw - allows you to make changes to the running system by storing changes in the supplied partition or loopback filesystem

Raid Essentials

Although RAIDs can be the hardest devices to image, especially the proprietary kind from Dell and Compaq, Helix provides a fairly simple solution. Helix can see most hardware RAIDs as the RAID card initializes the RAID before Helix even boots. Helix also has many RAID drivers for both software and hardware RAID.

Depending on the actual RAID device, Helix may not place the RAID in the /mnt directory like other devices, but that does not mean Helix does not see the RAID. For instance, to identify a Compaq Raid Devices, do a “dmesg” and look for “cpqarray.” If you see that you should see the devices that the Compaq RAID sees. The device should show partitions as:

Example – Compaq Array Partitions

```
ida/c0d0: p1 p2 p3
so partitions will be:
    /dev/ida/c0d0p1
    /dev/ida/c0d0p2
    /dev/ida/c0d0p3
```

Helix has many built in RAID drivers in the kernel and many more as loadable modules. If for some reason Helix does not see the RAID you will have to try and load the appropriate modules by:

Example – Loading a Kernel Module for an Adaptec 2120 RAID

```
# modprobe aacraid
```

This will load the aacraid module into the running kernel so that you can access the Adaptec RAID.

Understanding dd

dd has an interesting history. The most interesting is what dd stands for; most people assume dd stands for “device dump,” or “device-to-device,” or “data dump.” Some think it stands for “copy and convert” but that it was renamed to dd because the letters “cc” were reserved for the C compiler. The most interesting definition is that dd stands for “death and destruction” for what happens if you mess up the options; which is most definitely true. In actuality dd stands for “data definition,” if it can be said to stand for anything at all. The reason is that it was derived from the IBM OS/360 JCL (Job Control Language) command of the same name. IBM System/360 JCL had an elaborate dd “Dataset Definition” specification suitable for copying block-oriented I/O devices.

The dd command is used in computer forensics to perform a physical backup of hardware device media. What makes the dd command special is that it has special flags that make it suitable for imaging block-oriented devices such as tapes. dd is capable of addressing these block devices sequentially. In order to proceed, it is very important to understand the basic syntax of the dd command:

DD – Understanding Syntax

dd if=**source** of=**destination**

Where:

if= input file, or device you are copying (a hard disk, tape, etc.)

source = source of image

of= output file, or copy of image

destination = where you want to place the copy

For example:

if the device to be imaged is /dev/hda, the following would produce an exact copy with the name of 'ForensicCopy.img':

```
dd if=/dev/hda of=/mnt/hdd1/ForensicCopy.img
```

As mentioned earlier, dd is very useful when copying and/or restoring block-oriented devices such as tapes. Some of the options available to dd which make it very useful are:

bs = block size

ibs = input block size

obs = output block size

count = number of blocks to copy

skip = number of blocks to skip at start of input

seek = number of blocks to skip at start of output

conv = conversion

These options are extremely useful in many instances. For example, if you wanted to just acquire the Master Boot Record (MBR) from a hard drive, you would need to obtain the first 512 bytes from the hard drives partition table. In order to do this you would need to pass some options to dd to only grab the first 512 bytes, otherwise dd would acquire the entire hard drive. So to accomplish this you would type in:

Example – Acquiring the MBR using DD

```
dd if=/dev/hda of=/mnt/hdd1/MBR.img bs=512 count=1
```

Another example of using dd is to use it to split up a large image into much smaller images. This is a long way of accomplishing this as you would normally use the split utility, but this serves as just an example of the power of dd. For our example let's assume we have a 4GB device and we want to split the image up into four 1GB files.

Example – Splitting an image file using DD

Using dd with the flags below will create four images each 1GB in size.

```
dd if=/dev/hda count=1000000 of=image1  
dd if=/dev/hda count=1000000 skip=1000000 of=image2  
dd if=/dev/hda count=1000000 skip=2000000 of=image3  
dd if=/dev/hda count=1000000 skip=3000000 of=image4
```

Now each image is 1GB in size rather than the original 4GB. The first thing you should notice is that the first command takes 1GB (count=1000000) and copies it, naming the copy 'image1.' The second command *skips* the first 1GB (skip=1000000) and then copies the next 1GB (count=1000000), naming this image 'image2' and so on. This is the purpose of the 'count' and 'skip' flags.

Traditional Acquisition (Dead Imaging)

The process by which an image is acquired when the hard drive has been powered down is also known as dead imaging. This is the best method for obtaining the most forensically sound image. It is also the method that law enforcement uses as their primary practice.

There are many ways you can accomplish this task in a lab environment, but one of the easiest methods to use Helix. Helix is valuable in many ways but mostly for the ability to quickly image systems that utilize RAID devices. It is far more economical to image a RAID device at the logical level than to individually image each drive and try to rebuild the RAID device later on.

Boot the Helix CD in the system to be imaged (evidence system). You may need to boot into failsafe mode in order to be operational. There are some instances with proprietary raids like the Compaq Proliants using the SMART-2/P Raid controller in which Helix will not boot normally it will just hang during the auto detection phase.

Once Helix has booted, you have several options that you can use to image the system. You will need to make a decision whether to image the entire drive (physical) or the individual partitions (logical). In either case, your image will contain deleted files, slack space, and unallocated space. If you choose a logical image, the only thing you will be missing is the MBR and swap space if you forget to image it. Currently Autopsy cannot parse a physical image. Autopsy needs logical images but you can extract those from the physical image later on so make a physical image.

Imaging to a Netcat/Cryptcat Listener

Netcat is the preferred tool (over Samba) for imaging across a network due to a lower overhead.

Netcat is a networking utility which reads and writes data across network connections using the TCP/IP protocol. Cryptcat is the standard Netcat enhanced with twofish encryption. Helix does not use the default secret key (metallica), the key has been changed for Helix. You can change the key by using the -k option.

You still need to mount your collection/harvest drive which is the same method as above. The next step would be to set up a Netcat listener on the forensics server depending on what you want to collect. If you want to acquire the entire physical drive:

Example – Using a Netcat/Cryptcat server

Issue the following command the Forensics Server:
`nc -v -n -l -p 8888 -O myimage.img`

On the Helix system you can then run dd:
`dd if=/dev/<device> | nc <IP> 8888`

Imaging to a Samba Server

The first thing you need to do after booting the system to be imaged with Helix is set up a Samba server on your Acquisition system. Samba is very easy to implement and but uses additional bandwidth. Netcat is generally the best method to use. Samba is configured by the smb.conf file located in /etc/samba. See Appendix 1 for an example of a working Samba Forensics Server smb.conf file.

There are a few tricks that you must be aware of with a Samba Forensic Server. First you must mount the drive you want to use as your collection/harvest drive and that drive must be writable. The easiest way to accomplish this is by the following mount command:

Example – Mounting a device for use as a Samba share

```
mount -o rw,umask=000 /dev/hd?? /mnt/images
```

where:

?? = linux device that is your collection/harvest drive i.e. /dev/hdd1

/mnt/images = mount point of your collection/harvest drive.

Start your Samba server after you have mounted your collection/harvest drive by simply typing:
`service smb start` OR `service smb restart`

You are now ready to use the Samba share as your image destination. The only step that remains is to mount the Samba share on the Helix system. To mount the Samba share (*GRAB WILL DO THIS FOR YOU.*):

Example – Mounting a Samba or Windows share within Helix

```
mount -wt smbfs -o username=username,password=password //<IP>/<share> /<mount_point>
```

where:

- wt smbfs = mount read/write and set file system type to samba fs*
- o username=username,password=password = set username and password*
- //<IP> = netbios flag and IP address of the Samba/Windows System*
- /<share> = share name you want to mount*
- /<mount_point> = place you want to mount the share drive to*

** You can also add dmask=0777,fmask=0777 to the options for read/write*

The nice thing about setting up a Samba Forensics Server and imaging to it is that you can direct all of your commands to the local file system as that is where your Samba/Windows share will be located. It will appear as if you are writing to another directory on your local system when in fact the data will be traversing over the network.

EXAMPLE – Imaging to a samba share

```
dcfldd if=<device> | tee >(sha1sum > /<mount dir>/<case#>/<filename.sha1 ) | split -a 3 -d -b 1436m - /<mount dir>/<case#>/<filename_img.>
```

According to Drew Fahey of e-fense.com, this is the best command line to use to share a drive over samba:

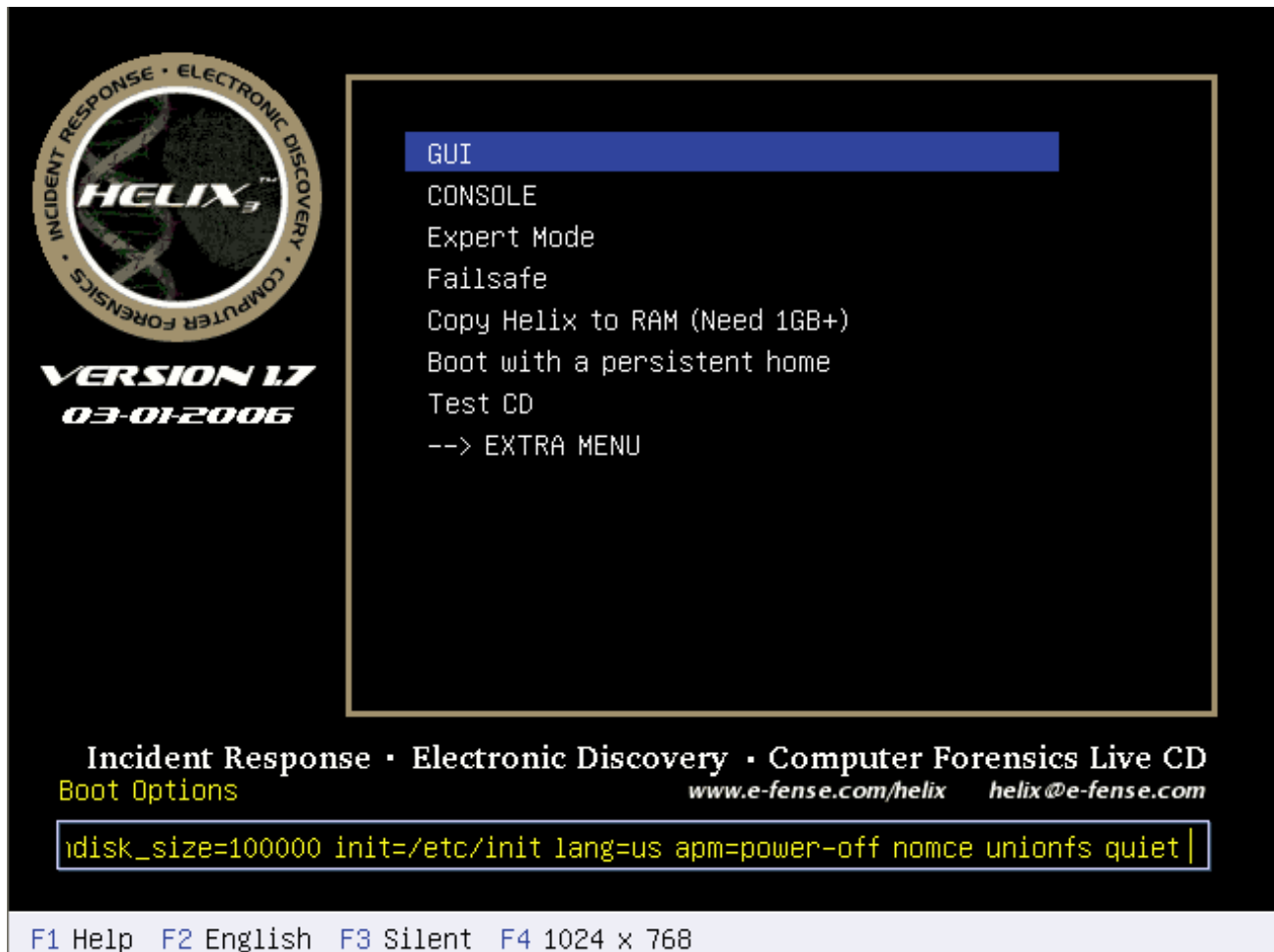
```
mount -t ntfs -o umask=000,noauto,user,uid=500,gid=100,ro,loop,noexec,noatime,show_sys_files=true /ntfs.img /mnt/hack/ntfs
```




Bootable Basics

The first step you must accomplish to boot into Helix is to make sure that your BIOS is setup to boot from the CDROM before any other device. If your BIOS does not support booting from a CDROM, then you must resort to booting from a floppy disk (time to upgrade your system).

All that is required to boot Helix is to place the Helix CD into the CDROM drive and reboot/turn on the computer. When the system passes the POST, you should see the screen:



As you can see you are presented with a graphical boot menu courtesy of Grub (Grand Unified Bootloader). You can choose the option that is best for you and your environment. The initial default setting should work for most people; however there are occasions when that won't work. Some laptops and other hardware do not like some of the standard options like using DMA on all devices. So you must choose the option to turn off DMA. Some of the other popular options that are available are using failsafe mode, and console mode. See Appendix A for a list of all the boot methods. Helix is currently using the 2.6.14 kernel.

While there are many options pre-determined for you in the boot screen, it may sometimes be necessary to add or delete extra commands. In order to do this within grub just type in or delete the commands you want by simply typing. The boot options will change as you type. As an example below, the quiet and BOOT_IMAGE=helix option have been deleted simply by pressing the backspace key.



F1 – Help and Cheat Codes

The F1 key will display the help screen, and provide the user with the “Cheat Codes”, the boot options to help configure Helix to run properly on the target system.

Navigating the Help System

The boot loader online help is context sensitive. It gives information about the selected menu item or, if you are editing boot options, it tries to look up information about the option in which the cursor is positioned.

Navigation Keys

Up Arrow	highlight previous link
Down Arrow	highlight next link
Left Arrow, Backspace	return to previous topic
Right Arrow, Enter, Space	follow link
Page Up	scroll up one page
Page Down	scroll down one page
Home	go to page start
End	go to page end
Esc	leave help

Boot Options (aka Cheat Codes)

About	short introduction to Helix.
ACPI	advanced configuration and power interface
APM	toggle power management
Clock	Clock options
Debug	Settings to debug your Helix Live CD
Expert	Interactive setup for experts
Failsafe	Boot with (almost) no HW-detection.
Framebuffer	Use the Framebuffer for graphics
FromHD	Boot from previously copied CD-Image

/home	Mount loopback file.
Hostname	Use a different hostname instead
HRate	Use specified horizontal refresh rate for X.
IDE DMA	Enable/Disable DMA for IDE-Drives
Keyboard	Use different keyboard (text/X)
Language	specify language/keyboard.
Main	Search for Helix mainmodules
Mem	Specify Memory size in MByte.
Memtest	Run the memtest86 utility.
No	Skip specified parts of HW-detection.
PCI	some PIC settings.
Runlevel	Runlevel 1, load Helix base, Textmode only
Splash	influence the behavior of the splash screen.
TestCD	Check CD data integrity and md5sums
ToHD	Copy CD to HD partition and run from there
ToRAM	Copy CD to RAM and run from there
VGA	Framebuffer settings.
VRate	Use specified vertical refresh rate for X.
WMScreen	Use specified Screen resolution for X.
Xmodule	Use specified X Window System driver.

About

Helix is a Live CD distribution with an emphasis on forensics and incident response. It boots from the CD media, while not touching the contents of your harddisk. Helix is based on Knoppix but has been significantly altered to prevent changes to data. Helix has a large number of users and is used for forensic training. For more information, please take a look at <http://www.e-fense.com/helix>

ACPI

ACPI (Advanced Configuration and Power Interface) is a standard that defines power and configuration management interfaces between an operating system and the BIOS. By default, acpi is switched on when a BIOS is detected that is newer than from year 2000. There are several commonly used parameters to control the behavior of ACPI:

pci=noacpi	do not use ACPI to route PCI interrupts
acpi=oldboot	only the parts of ACPI that are relevant for booting remain activated
acpi=off	switch off ACPI completely
acpi=force	switch on ACPI even if your BIOS is dated before 2000

Especially on new computers, it replaces the old apm system.

APM

APM is one of the two power management strategies used on current computers. It is mainly used with laptops for functions like "suspend to disk", but it may also be responsible for switching off the computer after power down. APM relies on a correct working BIOS. If the BIOS is broken, APM may have only limited use or even prevent the computer from working. Therefore, it may be switched off with the parameter

`apm=off` switch off APM completely

Some very new computers may take more advantage from the newer ACPI.

Clock

Use the hardware clock as the GMT time

`gmt`

Debug

Sometimes your Helix Live CD doesn't work exactly as hoped for. Here are a few options in order of usefulness:

Failsafe	Try to use conservative defaults
<code>vga=normal</code>	Don't use the framebuffer, disables bootsplash
<code>xmodule=vesa</code>	Load the default X driver, don't autodetect
<code>debug=on</code>	Don't reboot the Live CD after exiting, open a shell

Expert

Expert Settings. Use

`expert`

to enable this modus.

Failsafe

Boot Helix with (almost) no HW-detection. Use

`failsafe`

to enable this modus.

Framebuffer

Some Framebuffer settings:

fb1280x1024	used fixed framebuffer for graphics
fb1024x768	used fixed framebuffer for graphics
fb800x600	used fixed framebuffer for graphics

fromHD

Use the command

fromhd=/dev/hda1 (hda2,hda3,...)

to boot from previously copied CD image to this partition.

home

Mount Helix homedir.

home=/dev/sda1	Mount loopback file (helix.img) as /home/morph.
home=scan	Automatic search for helix homedir image.

hostname

Modify the default hostname, which is "Helix"

hostname=mybox Set the hostname to "mybox"

HRate (Xhrefresh)

Horizontal refresh rate - You can set your horizontal refresh rate with:

xhrefresh=80 (or hsync=80) Use 80 kHz horizontal refresh rate for X

ide

IDE is, unlike SCSI, commonly used in most desktop workstations. To circumvent some hardware problems that occur with IDE systems, use the kernel parameter:

ide=nodma switch off dma for IDE drives

keyboard

Keyboard settings:

keyboard=us xkeyboard=us

Use different keyboard (tet/X)

lang

Specifies a language for your keyboard. If available, Helix sets the correct locale for your language. Possible settings are:

lang=be	Belgian	lang=fi	Finnish	lang=pl	Polish
lang=bg	Bulgarian	lang=fr	French	lang=ru	Russian
lang=ch	Swiss	lang=gl	Galician	lang=sf	Swiss French
lang=cn	Chinese	lang=he	Hebrew	lang=sk	Slovak
lang=cz	Czech	lang=it	Italian	lang=sl	Slovenian
lang=de	German	lang=ja	Japanese	lang=tr	Turkish
lang=da	Danish	lang=lv	Latvian	lang=tw	Taiwanese
lang=el	Greek	lang=lt	Lithuanian	lang=uk	British
lang=es	Spanish	lang=nl	Dutch	lang=us	US (default)

main

Scan for mainmodules on this partition

main=partitionname

mem

Specify Memory size in MByte. Some systems do not report the proper memory size to the linux-kernel, which may cause the error "Panic: cannot mount root file system", and then the system hangs. The mem options allows you to specify the proper amount of memory.

mem=128M

The "M" must be capitalized.

memtest

Check the RAM of your system, doesn't boot Helix

memtest

no

Skip specified parts of HW-detection. Available options are:

noapic	turns APIC off.
noagp	turns AGP off.

noapm	turns APM off.
noacpi	turns ACPI off.
noaudio	turns AUDIO off.
noddc	turns DDC off.
nodma	turns DMA off.
nofirewire	turns FIREWIRE off.
noisapnpbios	turns ISAPNPBIOS off.
nomce	Disable Machine Check Exception
nopcmcia	turns PCMCIA off.
noscsi	turns SCSI off.
noswap	turns SWAP off.
nousb	turns USB off.
nonvidia	turns off the proprietary NVidia driver minimodule (if available)

To turn almost everything off see failsafe.

PCI

Some PCI settings:

pci=irqmask=0x0e98	Try this, if PS/2 mouse doesn't work.
pci=bios	Workaround for bad PCI controllers.

Runlevel

Boot only the Helix base, don't load any modules. Useful for debugging.

1

Splash

The splash screen is the picture shown during system start-up.

splash=0

The splash screen is switched off. This may be useful with very old monitors or if some error occurs.

splash=verbose

Activates splash, kernel and boot messages are still shown.

splash=silent

Activates splash, but no messages. Instead a progress bar is drawn.

testCD

To verify the proper operation of the Helix CD, you can test the CD. If the CD seems to make a lot of noise, or generates many errors, or programs seem to crash constantly, it is possible that the image you downloaded is corrupt, or your CD media is bad. Add the command line option:

`testcd`

to verify the CD data integrity and md5sums of the files.

toHD

Use the command

`tohd=/dev/hda1 (hda2,hda3,...)`

to copy the whole CD to specified partition and boot from there.

toRAM

Use the command

`toram`

to copy the whole CD to RAM and boot from there.

vga

VGA Framebuffer setting.

<code>vga=normal</code>	No framebuffer, but X.
<code>vga=785</code>	640x480 framebuffer.
<code>vga=788</code>	800x600 framebuffer.
<code>vga=791</code>	1024x786 framebuffer.
<code>vga=794</code>	1280x1024 framebuffer.

VRate (Xvrefresh)

Vertical refresh rate. You can set your vertical refresh rate with:

`xvrefresh=60 (or vsync=60)` Use 60 Hz vertical refresh rate for X.

WM screen

Sets the Screen Resolution for X (for your window manager).

screen=1280x1024	to use for a resolution of 1280x1024
screen=1024x768	to use for a resolution of 1024x768

Xmodule

Its possible to use different modules, also possiible to combine them:

xmodule=ati	uses ati module.
xmodule=fbdev	uses bdev module.
xmodule=i810	uses i810 sound module (intel compatible).
xmodule=mga	uses mga module.
xmodule=nv	uses NVidia module.
xmodule=radeon	uses Radeon module.
xmodule=savage	uses Savage module.
xmodule=s3	uses SiS module.
xmodule=svga	uses SVGA module.

Default Options for the different boot modes

The default Helix command line for the GUI mode is:

```
ramdisk_size=100000 init=/etc/init lang=us apm=power-off nomce unionfs quiet
```

The default Helix command line for the CONSOLE mode is:

```
ramdisk_size=100000 init=/etc/init lang=us apm=power-off nomce unionfs quiet 2
```

The default Helix command line for the Expert mode:

```
BOOT_IMAGE=expert ramdisk_size=100000 init=/etc/init lang=us apm=power-off nomce  
nodma
```

The default Helix command line for the Failsafe mode is:

```
ramdisk_size=100000 init=/etc/init lang=us vga=normal nosound noapic  
noscsi nodma noapm nousb nopcmcia nofireware noagp nomce nodhcp xmodule=vesa
```

The default Helix command line for the Copy Helix to RAM (Need 1GB+) is:

```
ramdisk_size=100000 init=/etc/init lang=us apm=power-off toram
```

The default Helix command line for the Boot using persistent home is:

```
ramdisk_size=100000 init=/etc/init lang=us noapic apm=power-off home=scan
```

The default Helix command line for the TestCD mode is:

```
ramdisk_size=100000 init=/etc/init lang=us nomce quiet testcd
```

The default Helix command line for the Framebuffer Mode 1280x1024 mode is:

```
ramdisk_size=100000 init=/etc/init lang=us noapic apm=power-off vga=794 xmodule=fbdev  
nomce quiet
```

The default Helix command line for the Framebuffer Mode 1024x768 mode is:

```
ramdisk_size=100000 init=/etc/init lang=us noapic apm=power-off vga=791 xmodule=fbdev  
nomce quiet
```

The default Helix command line for the Framebuffer Mode 800x600 mode is:

```
ramdisk_size=100000 init=/etc/init lang=us noapic apm=power-off vga=788 xmodule=fbdev  
nomce quiet
```

The default Helix command line for the ACPI on – DAM on – FB off mode is:

```
ramdisk_size=100000 init=/etc/init lang=us noapic apm=power-off vga=normal nomce quiet
```

F2 – Language and Keyboard Layout Selection

The F2 key will change language and keyboard layout the boot loader uses. Some of the currently available languages include: English, Spanish, French, Italian, German, Japanese and Russian.

F3 – Splash Mode Selection

Lets you change the splash screen mode. You can use the splash kernel option directly, if you prefer.

Native	turns the splash screen off (same as splash=0)
Verbose	shows nice picture and kernel and boot messages
Silent	suppresses all kernel and boot messages and shows a progress bar

F4 – Screen Resolution

Text Mode, 640x480, 800x600, 1024x768, 1280x1024, 1600x1200

Once you select your boot options and hit the ENTER key, you will see the Helix startup screen. This screen will show you boot progress.



HELIX
3

Incident Response, Electronic Discovery, Computer Forensics

```
HELIX IMAGE is a squashfs compressed image.
Total memory found: 254568 kB
Creating /ramdisk (dynamic size=188384k) on shared memory...Done.
Creating directories and symlinks on ramdisk...Done.
Starting the init process.
INIT: version 2.86 booting
Running the 2.6.14-9 Linux Kernel.
Processor 0 is Intel(R) Pentium(R) 4 CPU 3.20GHz 3202MHz, 1024 KB Cache
Enabling udev.
Waiting for usbcore to load. Done.
Loading ACPI Bios modules: ac battery button container fan processor thermal video
Autoconfiguring devices... Done.
Detected Mouse: ImPS/2 Generic Wheel Mouse Device: /dev/psaux
Detected Soundcard: ENS1371 - Ensoniq AudioPCI Driver=snd_ens1371
Video is VMware Inc VMware SVGA III PCI Display Adapter, using Xorg(vmware) Server
Monitor is Generic Monitor, H:28.0-96.0kHz, V:50.0-75.0Hz
Using Modes "1024x768" "800x600" "640x480"
Starting FC detection on vt10.
Scanning for Harddisk partitions and creating /etc/fstab... Done.
Network device eth0 detected, Obtaining a DHCP IP address... Done
Enabling unionfs support...
The Union /ramdisk/var/tmp/union/ is now enabled.
INIT: Entering runlevel: 5
```



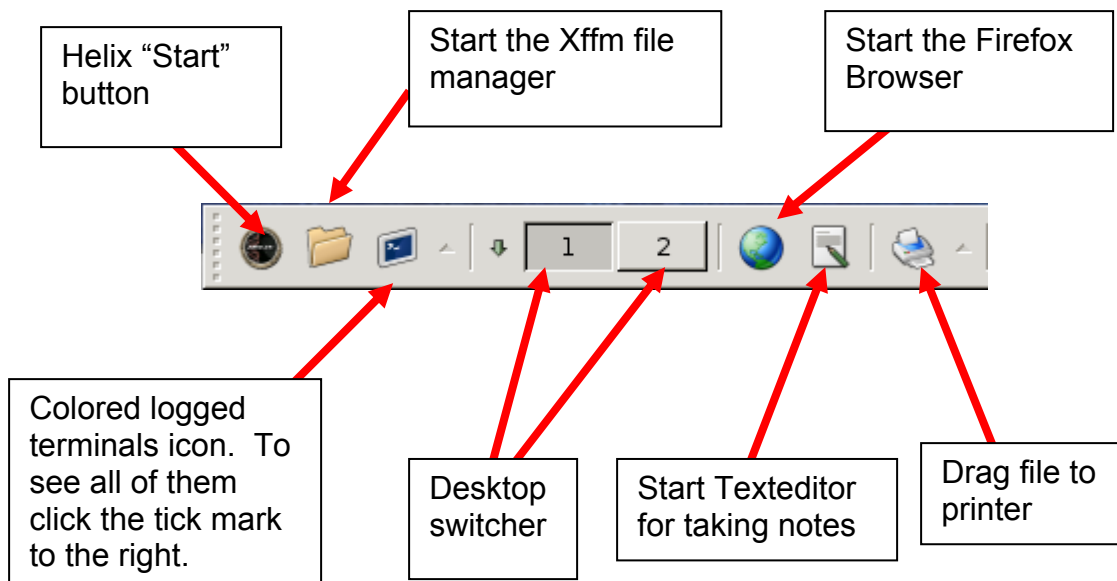
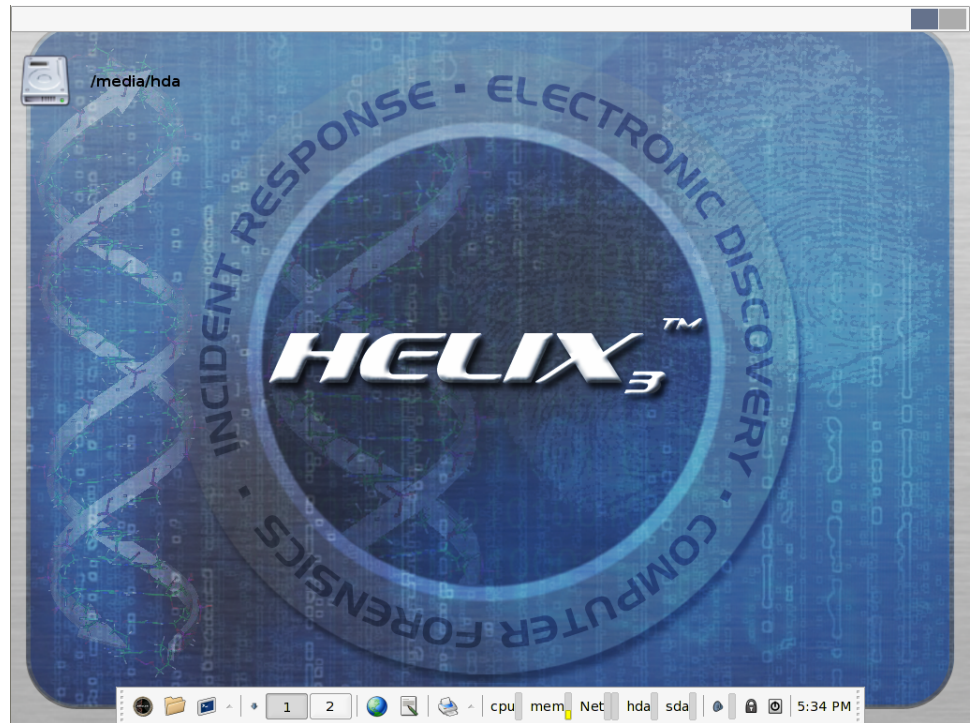
You will be able to see the devices that Helix finds as well as provides you with kernel information, which could be important if there are problems in loading Helix.

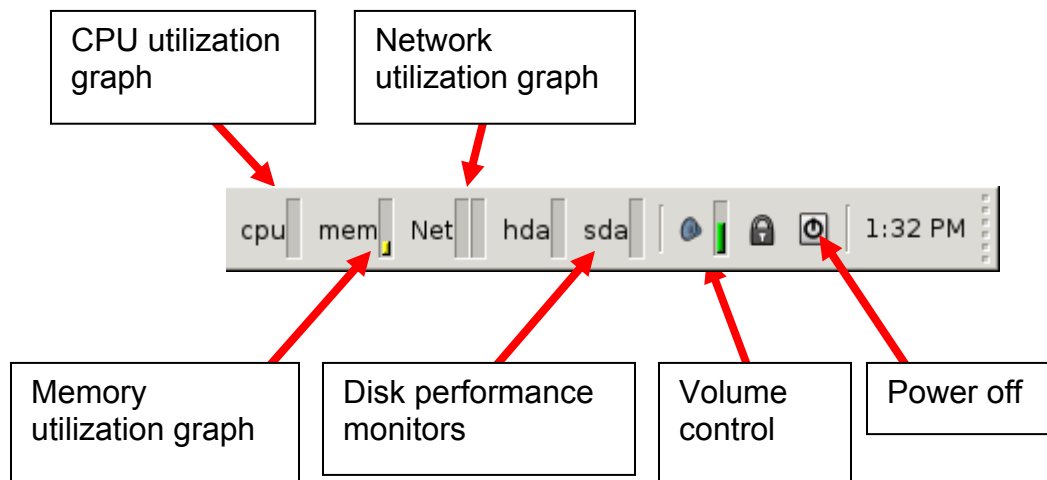


Helix User Interface

Once Helix finishes the boot process, X Windows will automatically start and present you with the Helix desktop. Helix uses the Xfce (<http://www.xfce.org/>) desktop environment as it is extremely light weight and very versatile. All of the other former Desktop environments such as KDE, Fluxbox, larswm, etc have been removed from Helix entirely.

Much of what you will need from Helix is available via the Helix Start Button and the taskbar at the bottom of the screen.





The Helix desktop environment is based on Xfce (<http://www.xfce.org/>). The name "Xfce" originally stood for "XForms Common Environment", but since then, Xfce was rewritten twice and doesn't use XForms toolkit any more. The name survived, but it is no longer capitalized as "XFce", but "Xfce". The developers' current stance is that the acronym doesn't stand for anything any more (Wikipedia, 2006c).

The following is taken from <http://www.xfce.org/documentation/docs-4.2/xfce4-use.html>

The Desktop

The Xfce 4 Desktop Environment is not a single entity that provides all functionality, but rather it tries to adhere to the old UNIX tradition of small tools that do one job and do it best.

Taskbar

At the top of the screen you will see the taskbar. It shows the applications running on the current workspace. You can focus the application by clicking on the button in the taskbar. Clicking again will hide the application. If you use the right mouse button, a menu will appear, allowing you to perform several actions on the application window.

The taskbar can optionally contain a graphical pager showing a miniature view of all your workspaces and a notification area or system tray.

Panel

At the bottom of the screen is the Xfce4 panel. It allows you to run applications and also contains a graphical pager, a clock and a mail checker. Some items have an associated panel menu that gives access to more applications. Panel menus are opened by pressing the small arrow buttons next to the panel item.

Changing the content of the panel and the properties of the items is done by using the right mouse button. Both the panel items and the panel move handles have a right-click mouse menu from where you can change the panel configuration.

Desktop Manager

The desktop manager provides the desktop background image and two menus when you click on the desktop background.

The right mouse button opens a menu that allows you to start applications. Look at the manual to find out how to change the menu contents.

The middle mouse button (or Shift + left click) opens a list of all applications that are currently running. You can activate an application by clicking on its menu entry.

Window Manager

The window manager is responsible for placing the windows on the screen and provides the window borders and decorations. It allows you to move windows around by dragging the titlebar and provides title bar buttons, for example to close, minimize or maximize a window. Look at the manual for a full explanation of the window manager.

Settings Manager

The settings manager runs in the background and makes sure that all Xfce 4 applications update their settings when the user changes something in the settings manager dialog (see following section) and takes care of reading the configuration from disk at startup. Look at the Settings Manager and Settings Plugins manuals for a full explanation of the settings manager.

Common Tasks

This section will explain how to perform several common tasks to quickly get you started working with Xfce 4. Because that is what Xfce 4 is designed for, to allow you to get work done.

Running programs

Xfce 4 panel

The panel is designed to allow quick access to the most frequently used applications by putting them on the main panel. Less often used applications can be put in a panel menu.

Desktop menu

Another method for starting applications is from the desktop mouse menu. Read the Desktop Manager manual for information on how to change the menu contents.

Run dialog

If you know the name of a program and it is not on the panel or in the desktop menu you can use the run dialog. To open the dialog type Alt+F2 or choose the Run program... option from the desktop menu.

The dialog will remember the 10 last commands that were executed successfully.

Managing windows and workspaces

Basic window operations

You can move windows around the screen by dragging their title bar. A window can be closed, hidden, maximized, shaded and made sticky — this means it will show up on all workspaces — by using the title bar buttons.

Right clicking on the title bar will open a menu that gives access to all window operations.

Shading a window, which means collapsing it to only show the title bar, can also be accomplished by using the mouse wheel over the title bar. Mouse wheel up is shade, mouse wheel down is unshade.

If you want maximized windows to not cover the entire screen you can set workspace margins from the settings manager dialog (see below).

Application management

To find out what applications are currently running you can look at the taskbar. Clicking on a button in the taskbar will focus the associated application. Clicking again will hide it.

When you click with the middle mouse button on the desktop background a list of windows is shown, ordered by workspace. You can activate the application or change workspaces by choosing the appropriate menu entry.

The xfce4-iconbox application can also be used to keep track of running applications.

Workspaces

You can change workspaces by clicking on them in the graphical pager, either on the taskbar or on the panel. Pressing Ctrl+Alt+LeftArrow or Ctrl+Alt+RightArrow will cycle through the workspaces. Using the mousewheel over the pager or the desktop background has the same effect.

To add or remove workspaces you can use the middle click desktop menu or the settings dialog (see below).

Using the settings manager dialog

The settings manager dialog provides access to the global preferences of many Xfce 4 applications. You can run it by pressing its launcher on the panel, from the desktop mouse menu or by running `xfce-setting-show`.

Dialogs to change many aspects of the Xfce 4 desktop environment are available. See the separate manuals of the Xfce 4 components for more information. It may be interesting to have a quick look at all the dialogs to find out what options are available that allow you to create the best possible working environment.

The File Manager

The following is taken from <http://www.xfce.org/documentation/docs-4.2/xffm.html>

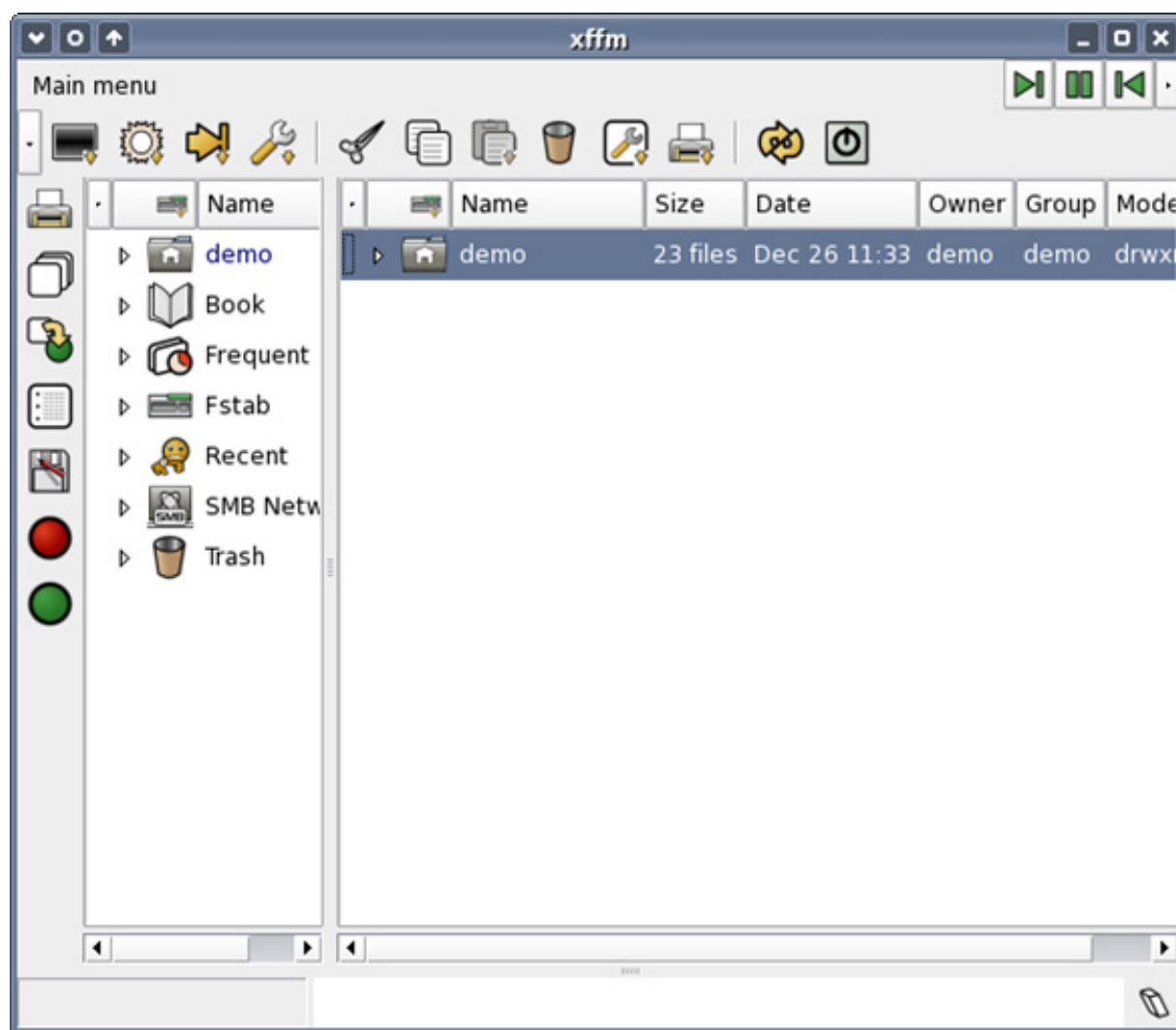
This file manager is treeview designed. Each main branch of the tree is a separate plugin which need not be loaded if not requested. With this file manager you may launch programs, examine contents of directories, manage the contents of trash bins, examine SMB (Wind*ws) network, keep bookmarks, view differences between files, find files, password encrypt files, and move, copy, rename, duplicate, delete or symlink files. You can also create, examine or extract directories to and from compressed tar files, iso file systems, and burn CD rom images. Mounting and unmounting remote SMB shares, local filesystems or removable media is also available with a double click.

Besides the above, the Xfce fast file manager has a sophisticated DBH-based mechanism to keep score on frequently used programs, visited sites, and a tab completion system which shows you the options instead of having you guess what they may be. You can rename files, change user or group information or file protection by simple select and edit the field. You can use drag and drop or cut and paste to move or copy files with other filemanagers or to download and upload files from remote SMB servers.

Getting Started

You will usually start the filemanager by selecting the corresponding entry from the Xfce-panel, the Xfce-desktop menu or by typing in a directory (absolute path or relative to homedir) at the xfrun4 prompt dialog. You can also type xffm at a terminal window or xfrun4 prompt.

When you start the filemanager for the first time you will a window on your screen, looking like this:



File manager root branches

Currently there are the following root level branches. You can any combination of them in either window pane.

Xftree - Local files

The local files branch is the traditional tree where files on the local computer are displayed. The tree can be opened to any level of nesting, and the top level can be relocatable to any directory on the local computer. To invoke the filemanager with only the local files branch activated, use **xftree4** as the command line.

Xfsamba - SMB Network

The SMB network branch is the way to navigate through a SMB network using the samba suite programs. To invoke the filemanager with only the SMB network branch active, use **xfsamba4** as the command line.

Xfbook - Bookmarks

The bookmarks branch is a way to create virtual directories with local files and remote SMB network files or shares. Multiple bookmark configurations can be used and toggled using ctrl-B. To invoke the filemanager with only the bookmark branch active, use **xfbook4** as the command line.

Xfglob - Find results

The find results branch is where the results of find queries are displayed. Full filemanager operations are enabled on the results. To invoke the filemanager with only the find branch active, use **xfglob4** as the command line.

Xffrequent - Frequent files

The frequent files branch contains a tree structure with those files or directories which are frequently accessed via the filemanager. The default frequency threshold is set at 13 hits, but may be changed by means of the main menu. To invoke the filemanager with only the recent branch active, use **xfapps4** as the command line.

Xfrecent - Recent files

The recent files branch contains a tree structure with those files or directories which have been accessed recently via the filemanager. The default recent threshold is set at 3 days, but may be changed by means of the main menu. To invoke the filemanager with only the recent branch active, use **xfapps4** as the command line.

Xffstab - Fstab mount points

The fstab branch is a alternate way of viewing the filesystem, where the physical devices are listed by mount point. This enables easy mount/unmount operations with the keyboard RIGHT and LEFT cursor, mouse double-click, or menu selection. To invoke the file manager with only the fstab branch active, use **xffstab4** as the command line.

Xftrash - Trashcan

The trashcan branch is a collection of trash bins. These may include Xffm wastebaskets or GNOME and KDE trash bins. This branch is a means of managing trash which is generated in different parts of the filesystem. You can collect trash bins belonging to other users among other functions available. To invoke the file manager with only the trash branch active, use **xftrash4** as the command line.

The menus

The key to working with the filemanager is understanding the menus. There are exactly two menus to deal with: the main and the popup. Since this is a keyboard friendly filemanager, to see what keyboard shortcuts are available, you should examine the menus. All the toolbar and sidebar buttons also have a corresponding menu element.

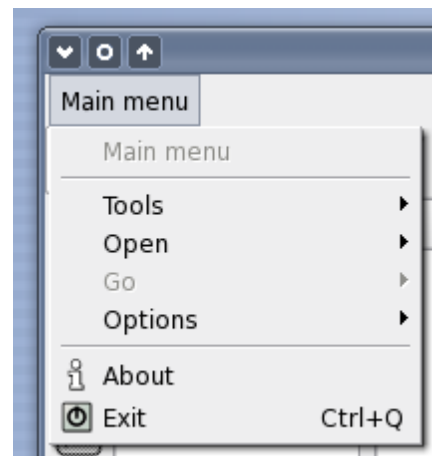
The main menu

In the figure is the main menu. This can be called by right or left clicking on the main menu bar, or by pressing function key F10.

The main menu consists of four submenus:

- *Tools*
- *Open*
- *Go*
- *Options*

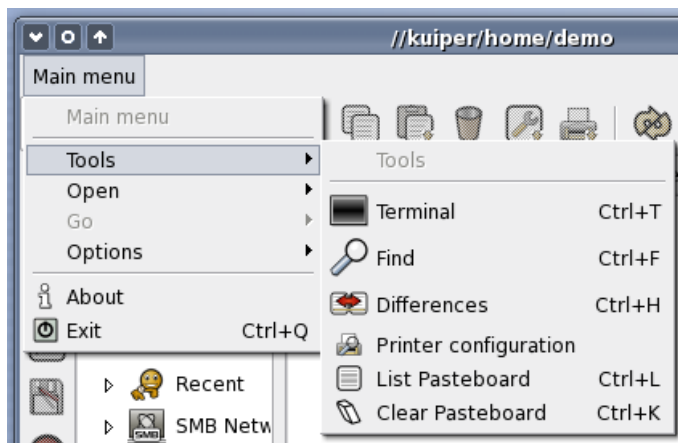
The *Go* menu may be greyed out if there is ambiguity as to which window pane the functions should apply to. If you have anything selected, or only have one window pane visible, there is no ambiguity.



The Tools Menu

The *tools* menu can be displayed from the main menu or by using F3, and has the following entries:

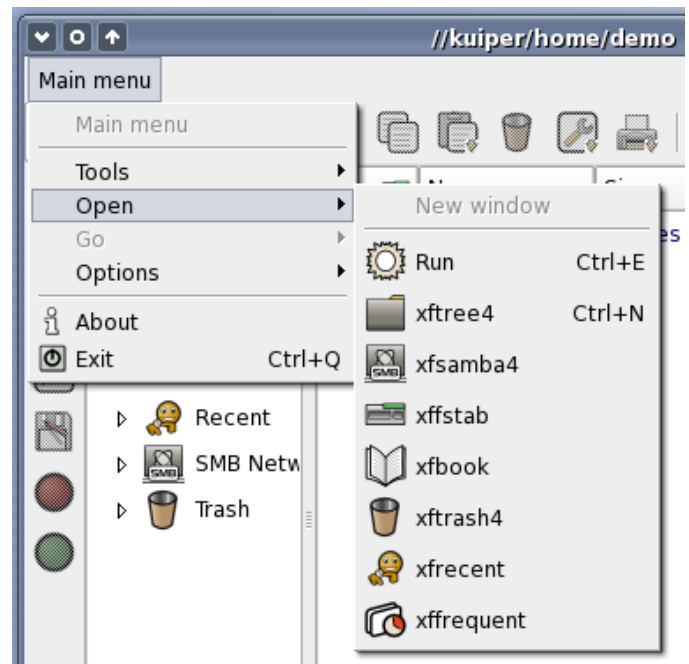
- *Terminal*: Opens a terminal in the currently selected directory. The terminal which is opened is determined in order of preference:
 - TERMCMD set from **xfce-setting-show**
 - **xfce4-terminal**
 - **xterm**
- *Find*: Opens a find dialog window. Results are displayed in a find results branch of the filemanager.
- *Differences*: Opens a difference window between two selected files. If no files are selected, you can drag and drop them in later.
- *Printer configuration*: opens a dialog to configure your printers.
- *List pasteboard*: outputs the content of the current pasteboard to the diagnostics window.
- *Clear pasteboard*: Clears the contents of the current pasteboard (the filesystem is untouched by this operation).



The Open Menu

The *open* menu can be displayed from the main menu or by using F4, and has the following entries:

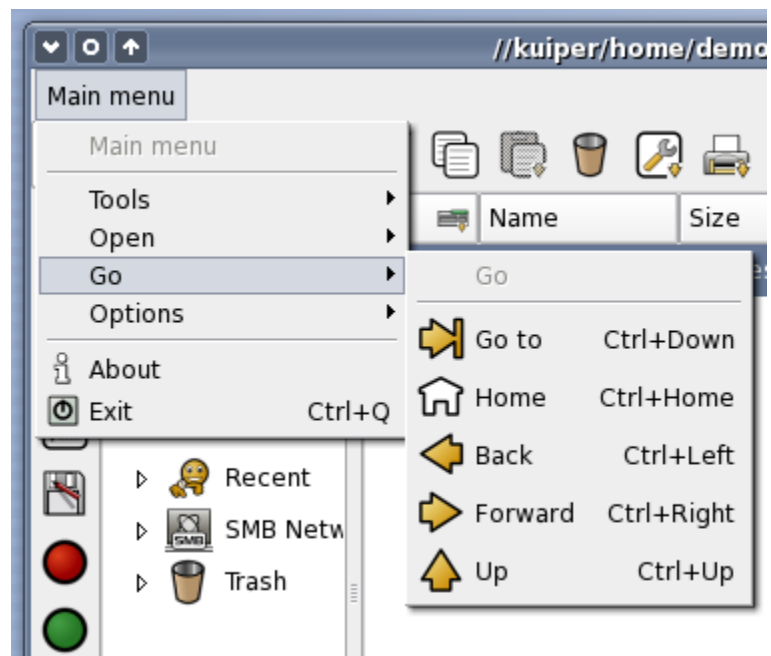
- *Run*: Queries for a program to be run.
- *xfree4*: Queries for a directory path and opens a new filemanager window there. Equivalent to executing **xfree4** **directory_path** from a command line. Absolute path or relative path (to homedir) is acceptable.
- *xfsamba4*: Equivalent to executing **xfsamba4** from a command line.
- *xffstab*: Equivalent to executing **xffstab4** from a command line.
- *xfbook*: Queries for a bookmarks file and opens the filemanager there. Equivalent to executing **xfbook4** **bookname** from a command line.
- *xftrash4*: Equivalent to executing **xftrash4** from a command line.
- *xfrecent*: Equivalent to executing **xfrecent4** from a command line.
- *xffrequent*: Equivalent to executing **xffrequent4** from a command line.



The go menu

The *go* menu can be displayed from the main menu or by using F5, has the following entries:

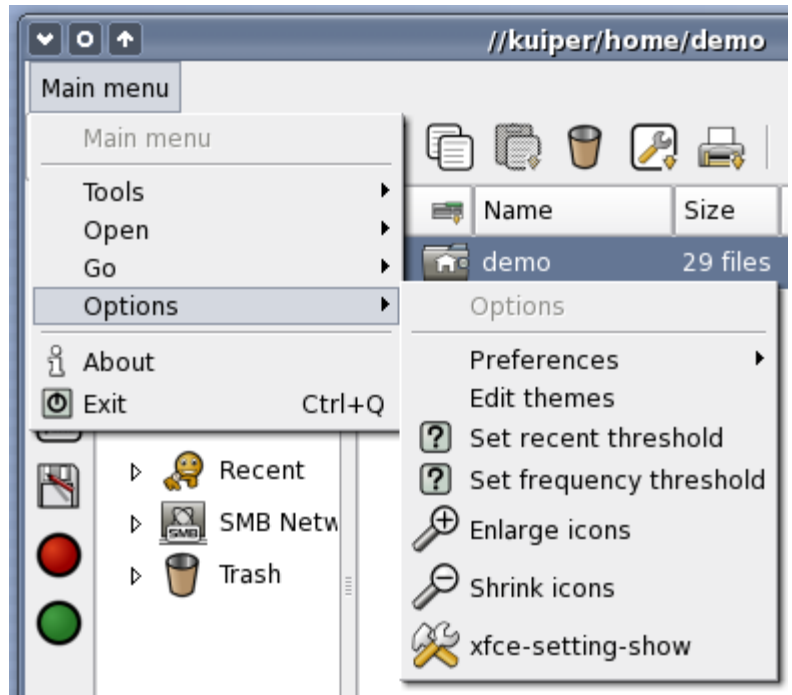
- *Go to*: Opens a query where you can specify where you want to go to. Paths preceded by double slash (//) are interpreted as remote SMB servers.
- *Home*: Go to your home directory, or to XFFM_HOME if defined with the xfce-mcs-manager.
- *Back*: Goes to the previous location.
- *Forward*: Goes forward (after a go-back, of course).
- *Up*: Goes up in the directory file structure.



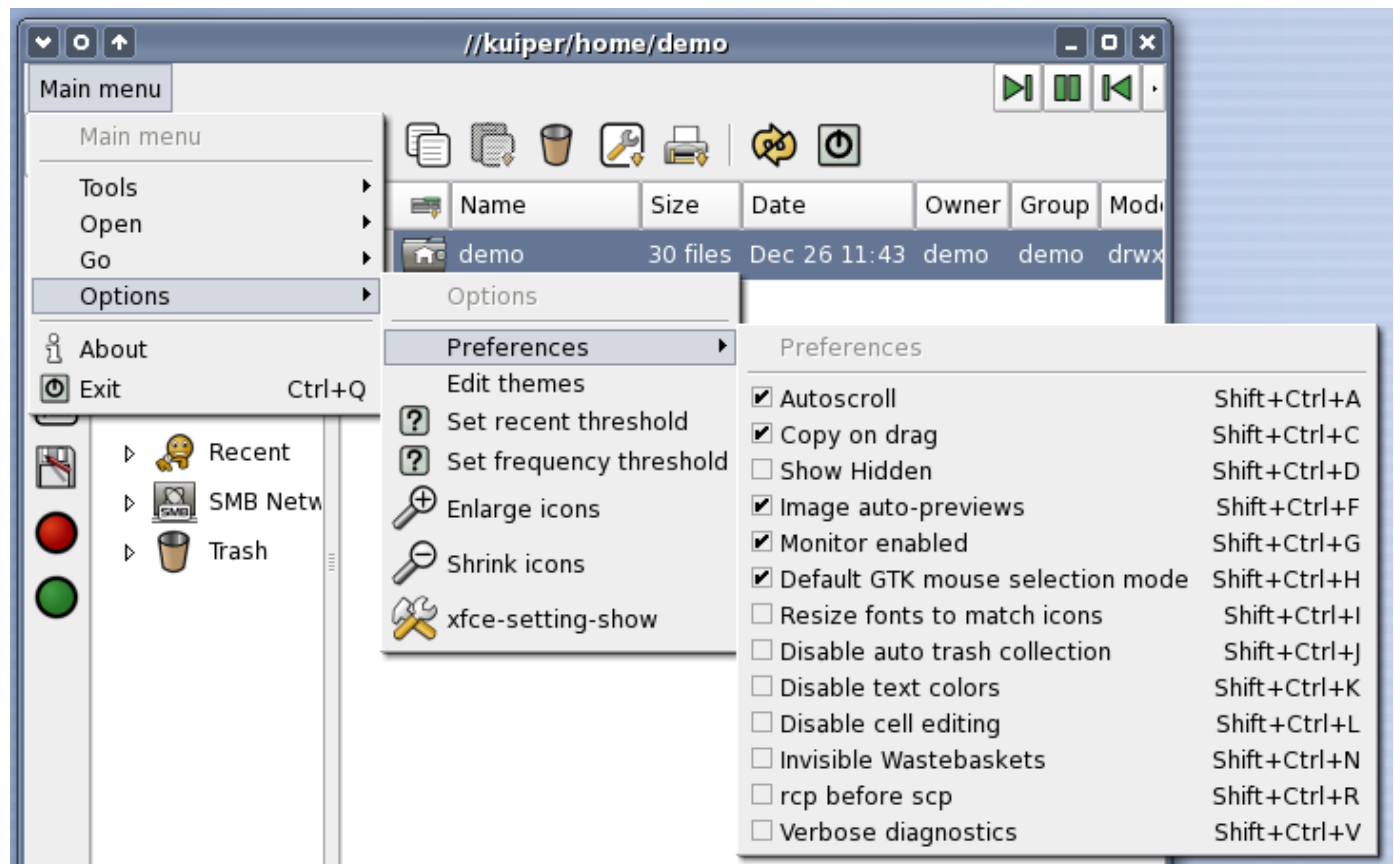
The options menu

The *options* menu can be displayed from the main menu or by using F6, has the following entries:

- *Preferences*: opens the preferences submenu.
- *Edit themes*: runs the `xfmimedit` program which allows you to customize the icon settings.
- *Set frequency threshold*: allows you to change the frequency threshold from its default value of 13 hits.
- *Set recent threshold*: allows you to change the recent threshold from its default value of 3 days.
- *Enlarge icons*: enlarges icons.
- *Shrink icons*: shrinks icons.
- *xfce-setting-show*: Launches the `xfce-setting-show` program which allows you to move the mcs manager settings.



The preferences menu



The *preferences* submenu can be displayed from the main menu or by using F7, and has the following checkboxes:

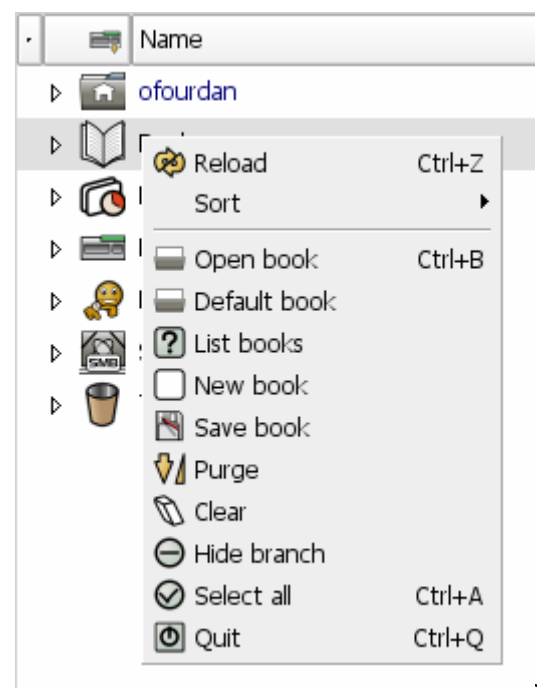
- *Autoscroll*: This makes the treeview scroll automatically when you open a folder.
- *Text headers*: This option is no longer available.
- *Copy on drag*: If this is checked, the default drag-n-drop action will be to copy, if unchecked, the default is to move.
- *Show hidden*: Controls whether hidden files are shown or not.
- *Image auto-previews*: Controls whether previews of graphic files are automatically generated on opening folders.
- *Monitor enabled*: Whether changes in the filesystem should be monitored to do automatic updates.
- *Default GTK mouse selection*: Whether to use the default GTK treeview selection, or the custom mouse selection introduced in 4.0.x.
- *Resize fonts to match icons*: Whether font sizes should be scaled up or down when icon size changes.
- *Disable autotrash collection*: By default, trash is automatically collected in the trash branch. If you prefer to manually collect trash, turn this off.
- *Disable text colors*: Use plain black and white for listings instead of funky colors.
- *Disable cell editing*: Disallow renaming, user/group changes, or mode modifications by direct inline editing.
- *Invisible wastebaskets*: If you do not like to see whether trash exists for directory when you open the folder, check this item.
- *rcp before scp*: When a xffm receives a drop from another xffm window on the same display, but running on a different host, should the files be copied by rcp or scp? In secure cluster configurations this should be rcp, but otherwise scp.
- *Verbose diagnostics*: If this is checked, the amount of processing information which appears in the diagnostics window will be enhanced.

The popup menus

The popup menu is dynamically configured, depending on what is selected when the popup appears. In the following paragraphs we shall examine the most common scenarios. The popup menu can be shown by right clicking with the mouse or pressing F9

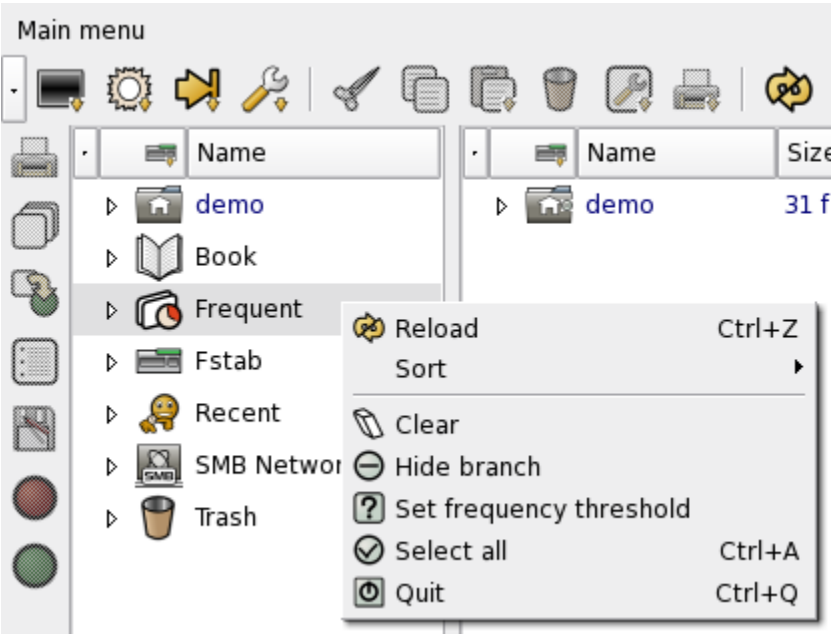
The book popup menu

Aside from the normal operations, from this popup menu you can also open a named book, open the default book, list all named books, create a new book, and save the current book with a new name.



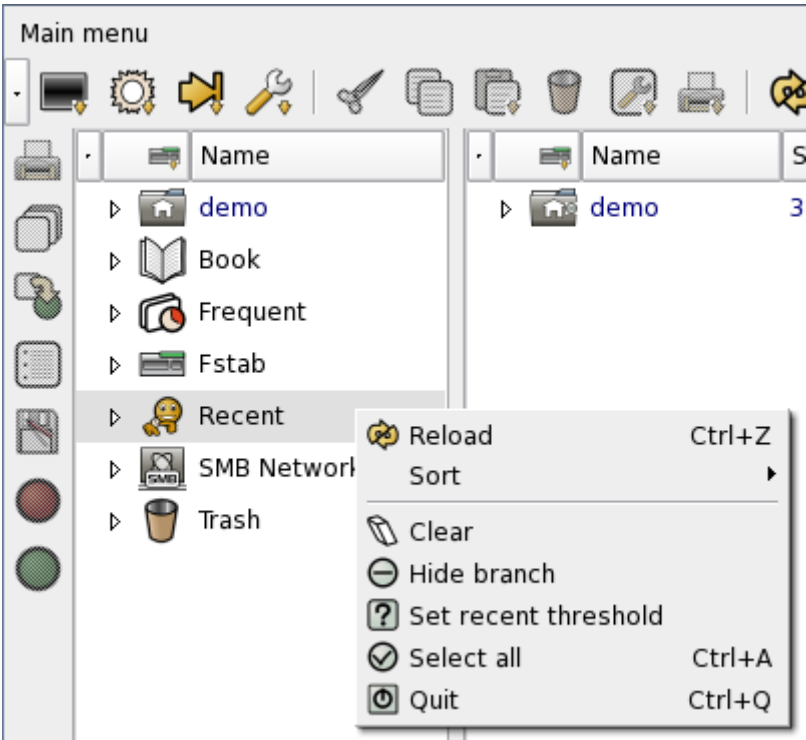
The frequent popup menu

Aside from the normal operations, from this popup menu you can reset the frequency threshold from the default value of 13 hits.



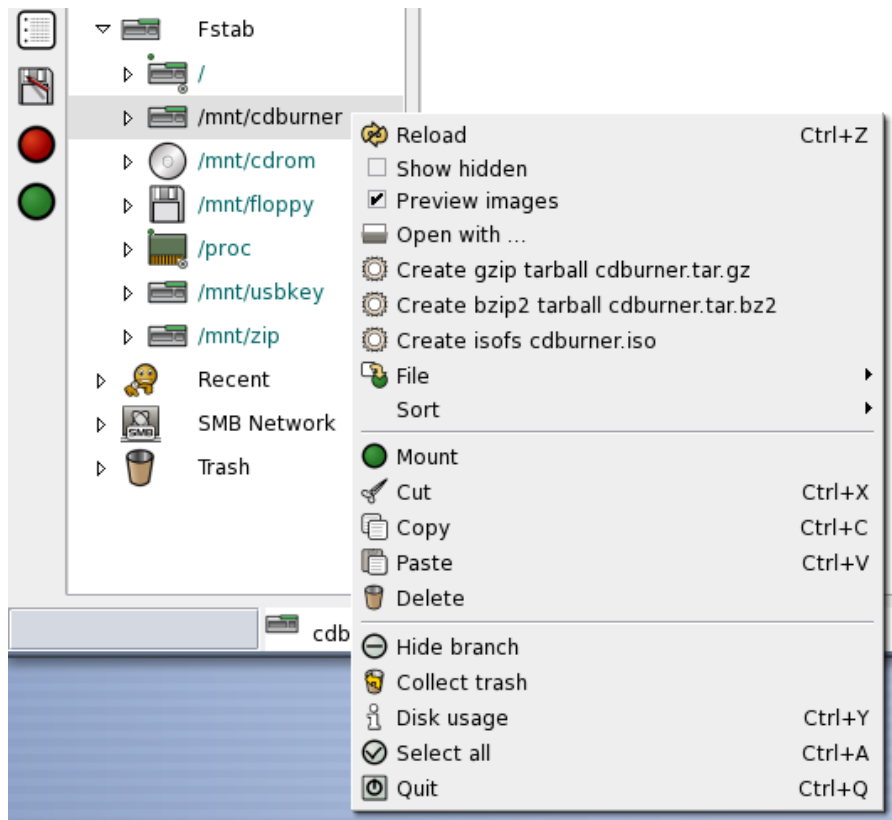
The recent popup menu

Aside from the normal operations, from this popup menu you can reset the recent threshold from the default value of 3 days.



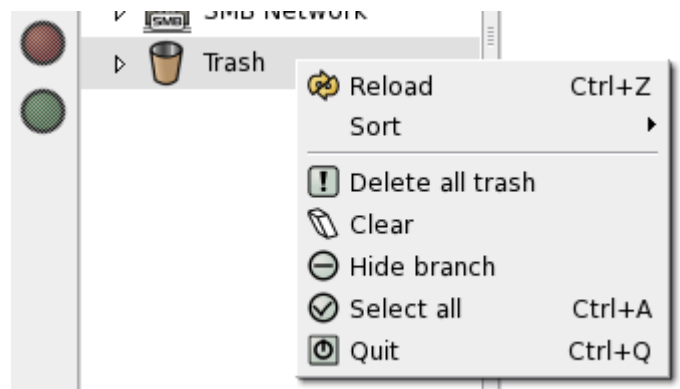
The fstab popup menu (mount/unmount)

Aside from the normal operations, from this popup menu you can either mount or unmount volumes which are listed in the fstab file information (they may be SMB shares, NFS volumes or local filesystems).



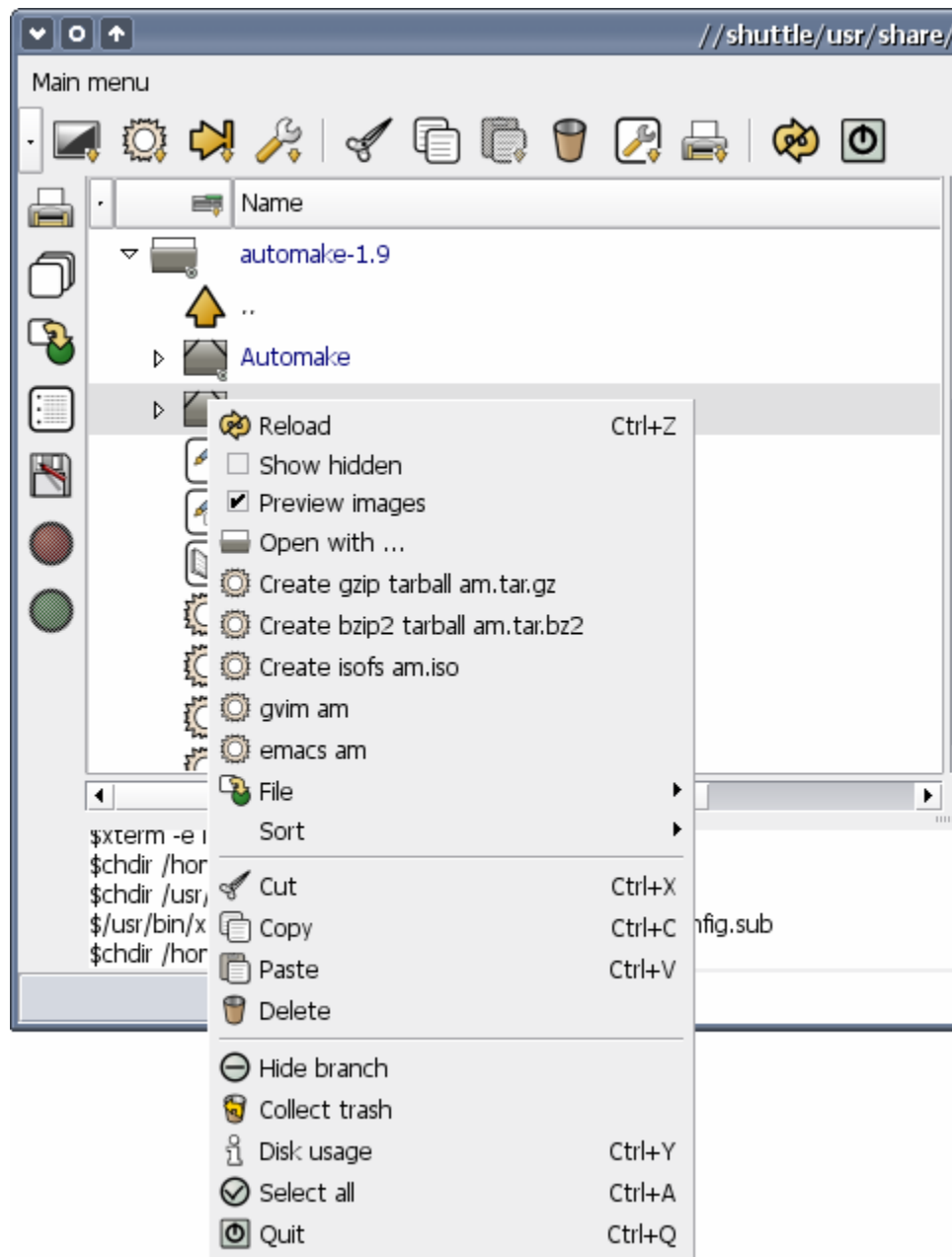
The trash popup menu

Aside from the normal operations, from this popup menu you can permanently delete all the collected trash from the filesystem. You can also clear the contents of the trash, in which case you would have to collect trash from the directory popup to make it appear again.



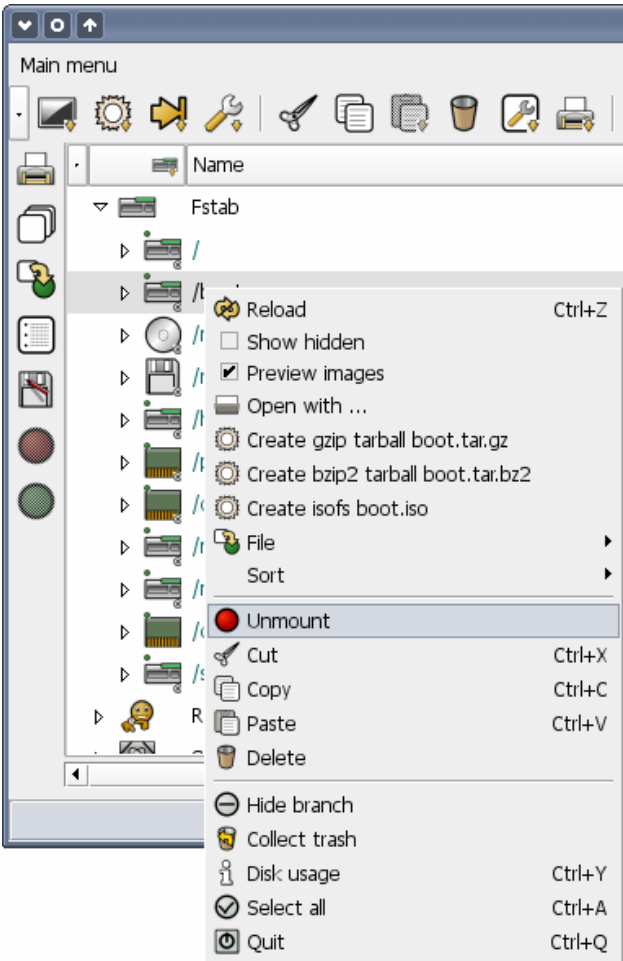
The directory popup menu

Aside from the normal operations, from this popup menu you can create gzipped or bziped tarballs. You can also create iso filesystem files to directly burn CD-RW volumes.



The directory popup menu (unmount)

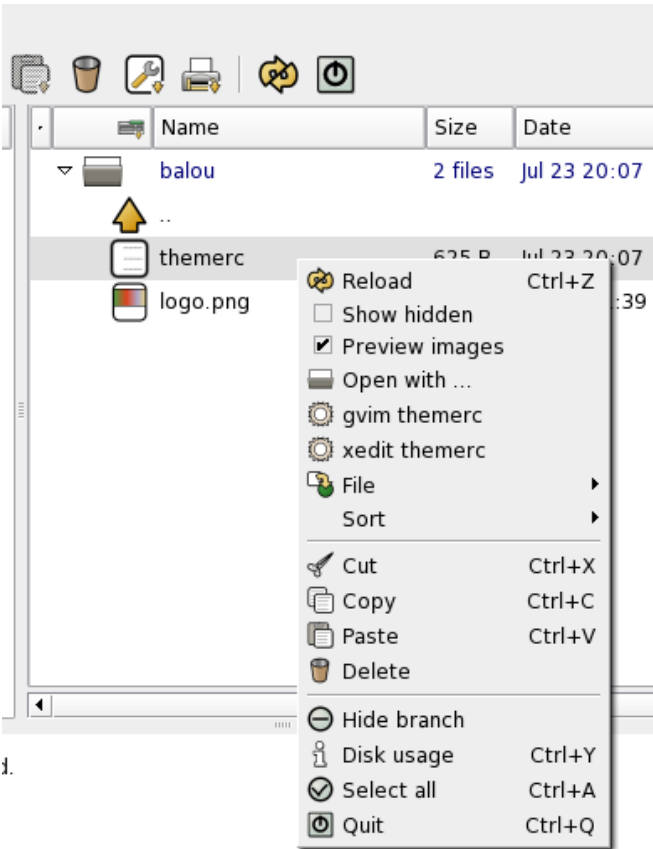
In the case where a directory is also listed in the fstab file as a mount point, you can mount/unmount volumes from this popup.



The file popup menu

The file popup has all the operations normally performed on files. Depending on the mimetype of the selected file, you may get several options with which to open the file. These options are constructed from the system wide mimetype applications, the user mimetype applications (constructed by clicking *remember* when the *open with* function is used), and the last application used to open the file (whether *remember* was checked or not). Thus in the above figure you can observe that the TeX file selected has several option with which to open it with.

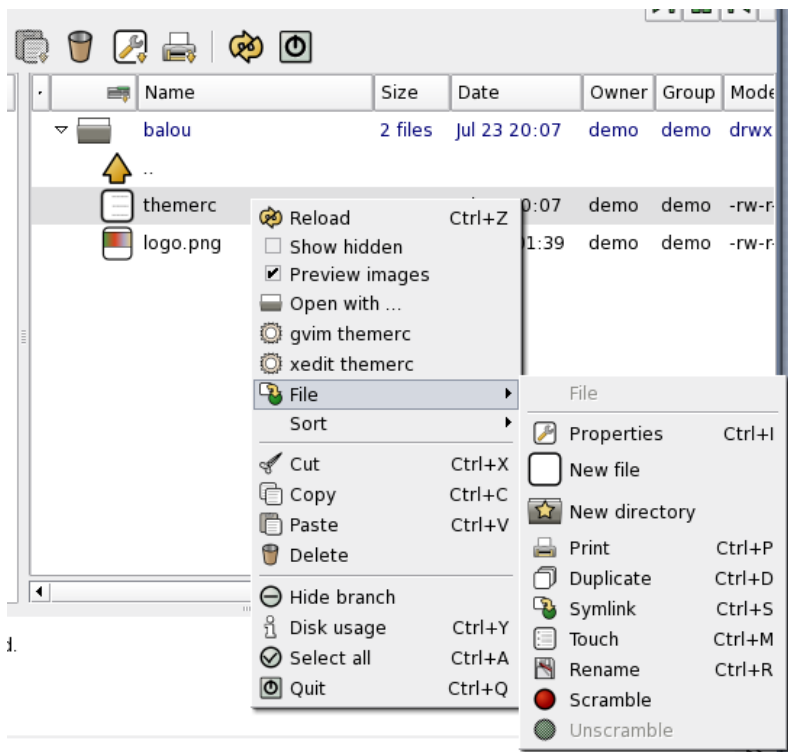
Further file operations are included in the file submenu, described below and which may be quickly accessed with F8.



The file popup submenu

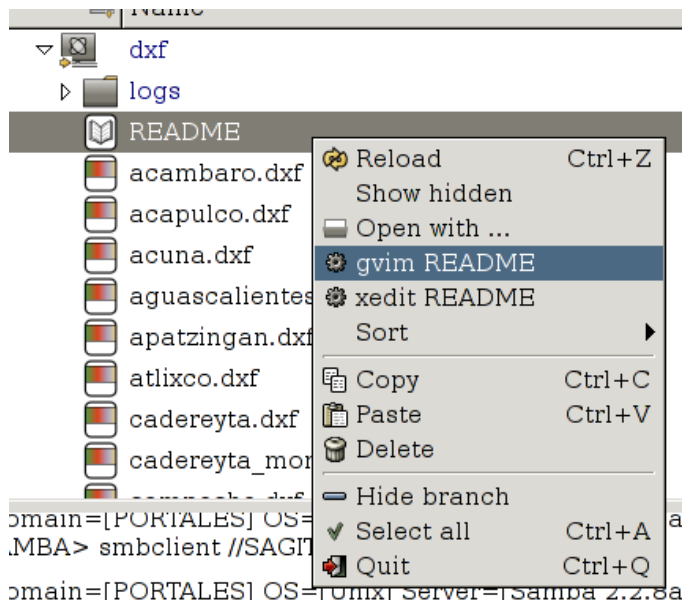
The file submenu which may be quickly accessed with F8, contains the basic operations normally done to the filesystem:

- *Properties*: Modify the file's user/group or mode information (also may be done by inline editing of the fields).
- *New file*: Creates a new file in the selected directory.
- *New directory*: Creates a new directory within the selected directory.
- *Print*: Prints the selected file using **xfpri4**
- *Duplicate*: Creates a duplicate of the selected file or directory.
- *Symlink*: Creates a symlink of the selected directory or file (also available by cti-shift dragging or paste-linking the pasteboard).
- *Touch*: Touch the file or directory.
- *Rename*: Rename the file or directory (also available via inline editing.
- *Scramble*: Password scrambles the file. If the filemanager is compiled with the --enable-scrambledir option, then this item will not be greyed out for directories and the whole directory can be recursively scrambled with the same password.
- *Unscramble*: Unscrambles the file. The mimetype extension for scrambled files is .cyt, so that this option is grayed out if the selected file is not of the scrambled type. If recursive scrambling of directories is enabled at compile time, this option will also be active for directories.



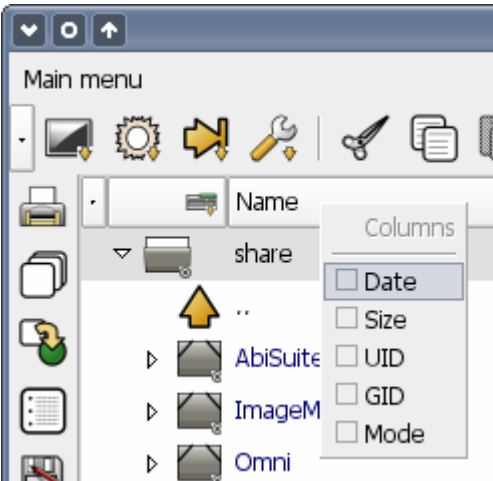
The netfile popup menu

The popup for SMB network files is similar to the one for local files, but does not contain the file submenu.



The columns popup menu

If you right click over the titles of the columns, you get the *columns popup*. With this popup you can toggle which columns you wish to be visible or not. By default configuration, the right pane has all optional columns visible, and the left column has none of the optional columns visible.



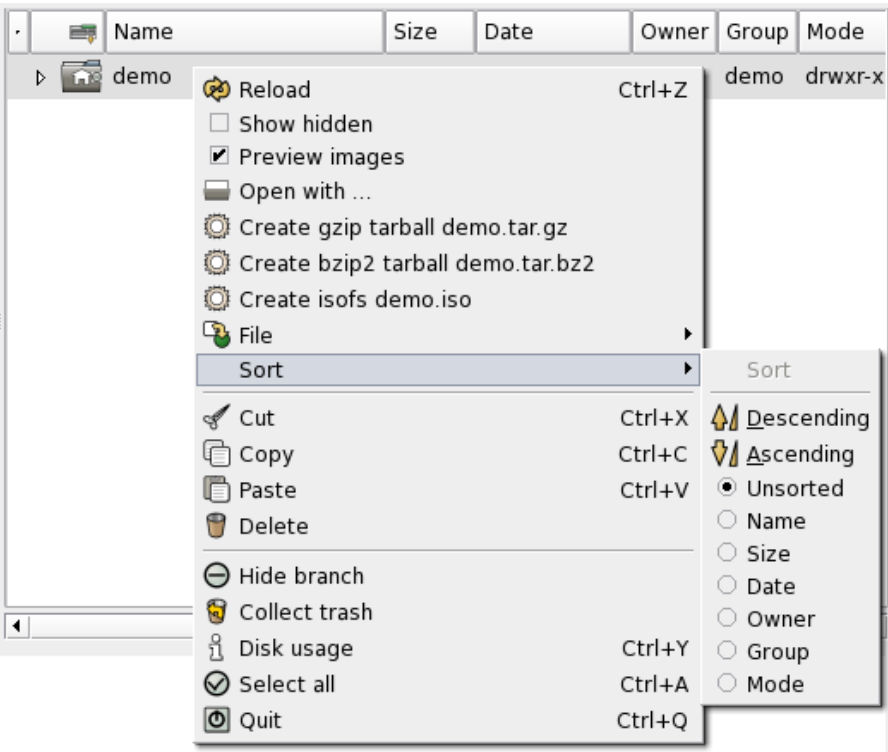
The icon title smart button



This is the popup for the icon column title. By keeping pressed you make a popup of buttons appear. Release on any button determines which one gets the click. These buttons are used to toggle main branches on and off. If you want to see the fstab branch, click on the fstab symbol. If you want to hide the local branch, click on the local branch symbol.

The sort popup submenu

The sort submenu allows you to toggle the set sorting method for the treeview. The *unsorted* method implies a sorting by name and subsorted by filetype. You may also toggle the sorting method by clicking on the column titles. The purpose of this menu is to make a sort method toggle available from the keyboard.



The toolbars

The standard toolbar



The standard toolbar is a shortcut to many menu functions. By right clicking on any button with a down arrow, you can appear or disappear the corresponding side bar. You can also bring up a popup with the sidebar elements by pressing with the button and not releasing. You then release on the popup element you desire to click. The last clicked element of the group becomes the top button visible in the toolbar.

The menu toolbar

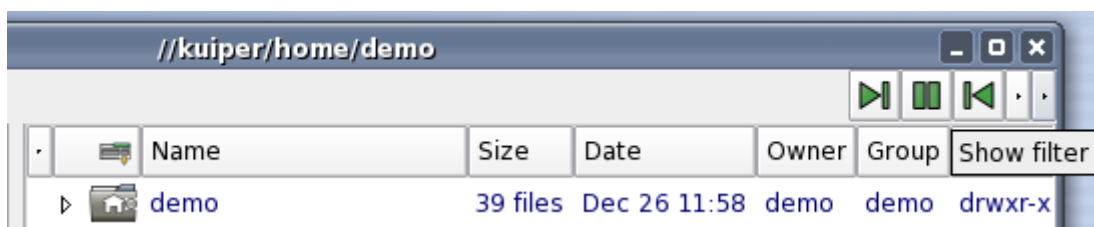
The menu toolbar consists of the following elements:

The filter box



This allows you to filter the contents of a directory before being inserted into the treeview. Regular expressions such as that shown in the figure are also acceptable (besides classic filters like *.c). After changing the filter string, refresh the view. If the treeview is hidden, nothing is filtered.

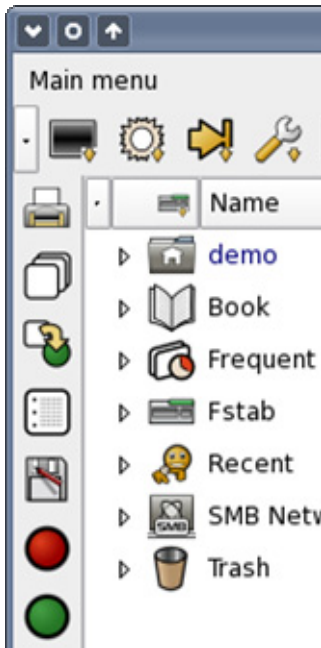
The hide/show buttons



In the menu toolbar, you have buttons for showing only the right treeview (also with F12), the left treeview (also with F11), viewing both treeviews (either F11 or F12 twice), and hiding and showing the filter box and the standard toolbar. (If you compiled with --enable-panel, you will have a second toolbar reflecting your xfce4-panel configuration, complete with hide and unhide buttons, replacing the applications root branch from xffm-4.0).

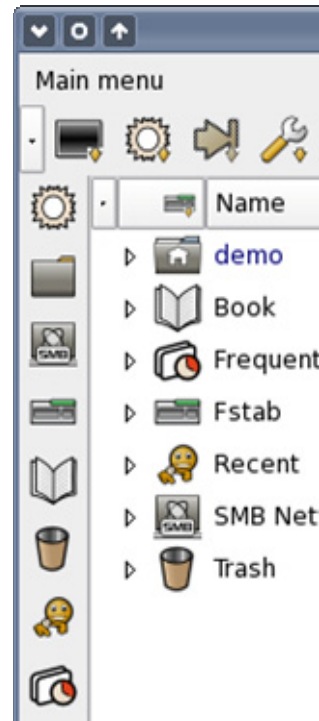
The sidebars

There are several sidebars available in the default configuration.



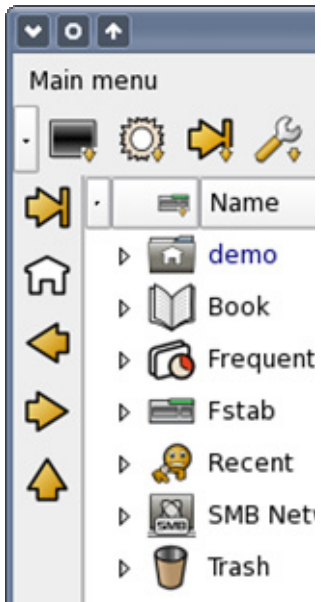
The file submenu sidebar

See "File submenu" for an explanation of the available options. Exactly one item must be selected from the treeview for this sidebar to be active.



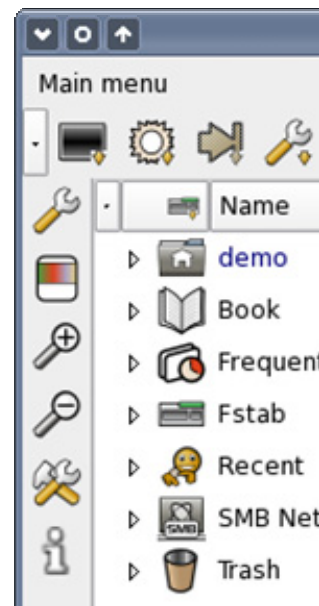
The open sidebar

See "open menu" for an explanation of the available options.



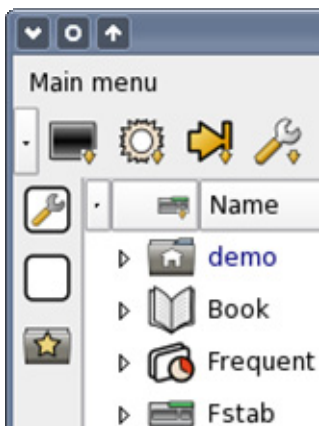
The go sidebar

See "go menu" for an explanation of the available options.



The options sidebar

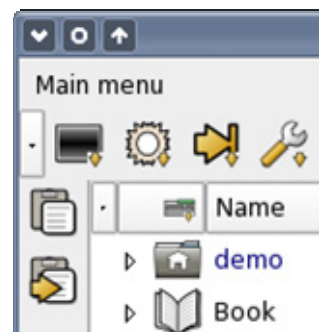
See "options menu" for an explanation of the available options.



The multiple_select sidebar

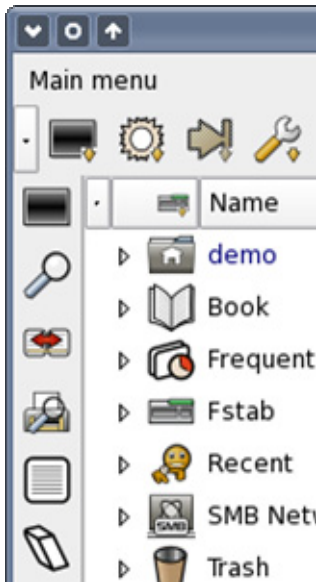
You can create a new file or directory or open the properties dialog from here. At least one item must be selected from the treeview for this to be active. See "File

submenu" for an explanation of the available options.



The paste sidebar

You can paste the contents of the pasteboard, or paste-link the contents of the pasteboard. The paste-link function creates symlinks of the files referenced in the pasteboard.



The tools sidebar

See "tools menu" for an explanation of the available options.

The xfce-mcs-manager

Certain functions perform better if configured with the mcs manager plugin.

On deleting a file, the confirmation dialog will default to one of three buttons: cancel, wastebasket or unlink. Choose whatever you prefer here.

If you don't want any output at all to the diagnostics window, check the *Disable diagnostics* option

If want to hold the output of xterms or xfce4-terminals opened by the filemanager, check the *Hold xterms* option

If want to take full advantage of the the mount/unmount functions provided by the filemanager, it is best you install **sudo** and have it properly configured to allow mount/unmount. If **sudo** requires a password, the filemanager will prompt you accordingly. Check the *Mount with sudo* option for this.



The last part of the mcs plugin allows you to set environment variables on the fly:

- **TERMCMD**: The command used for opening terminals.

- **LANG:** The environment variable LANG passed on by the filemanager to the applications it opens. You only need to change this if you want this variable to be different from that which is used for the filemanager.
- **XFFM_HOME:** The path that the filemanager goes to when the *Go home* function is selected.
- **SMB_USER:** The default username%password used for SMB network queries.
- **SMB_CODESET:** Code set used to interpret non-ascii characters on remote SMB servers.
- **XFFM_STATUS_LINE_LENGTH:** Defines the maximum length of the strings which appear in the status line. This option is provided to avoid the width of the filemanager window to grow beyond the user's choice.
- **XFFM_MAX_PREVIEW_SIZE:** This environment variable defines the maximum size for image preview (by default set at 256 KB). Note that some previews may not be generated if they are too thin or too wide. To view these, install image-magick and use double click to view these these files.

The diagnostics window

This is the window where output from commands performed by the filemanager is displayed. If you wish to increase the verbosity, use the *verbose* preference, and if you want to disable the output altogether, use the *mcs* plugin.

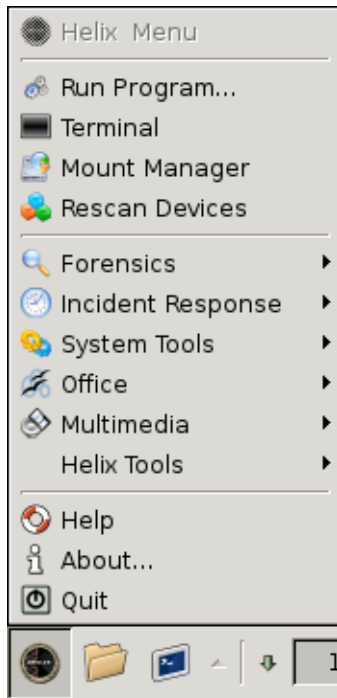




Helix Tools

Clicking on the Helix Start Button reveals a number of commands and submenus. This menu is also available by right-clicking anywhere on the desktop.

The Main Menu



Run Program... opens a simple command line tool to allow users to quickly launch programs.

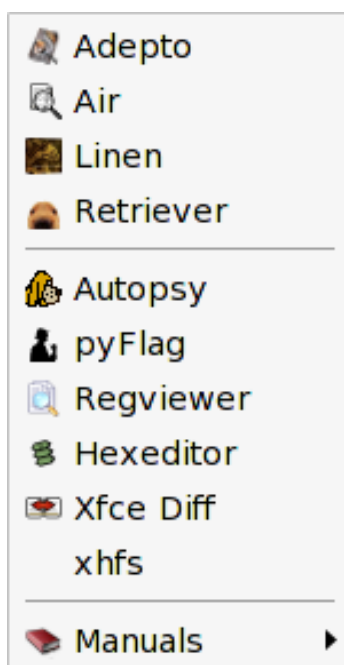
Terminal opens a terminal window, however, the commands in this window are not logged. For logging and replay options, use the terminal icon in the task bar.

Mount Manager will open up a xfstab window and allow the user to manage the devices that have been mounted.

Rescan Devices can be used to access devices that have not been automatically detected.

Help, About, and **Quit** perform the normally expected actions.

The Forensic Menu



On the Forensics Submenu, the following commands are available:

Adepto was developed to perform image acquisition and generate a chain of custody.

Air is a GUI front-end for dd.

Linen is an Image Acquisition Tool from Guidance Software, which can be used to capture suspect media and create images that can be processed by the EnCase Forensic toolkit.

Retriever is an image (picture/video) capturing utility for “knock & talks”, “quick peeks”, and general searches. It can scan a mounted device and locate all of the images and movie files.

Autopsy is a Forensic Browser is a graphical interface to the command line tools in The Sleuth Kit. They can be used to analyze Windows or Linux systems.

pyFlag is designed to simplify the process of log file analysis and forensic investigations.

Regviewer is a Windows registry file navigator.

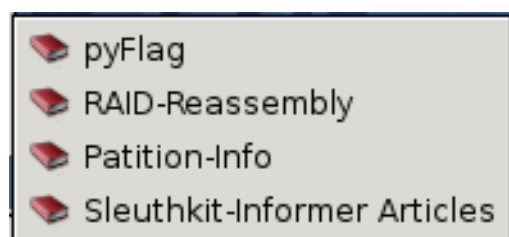
Hexeditor is a simple binary editor. It lets users view and edit a binary file in both hex and ASCII with a multiple level undo/redo mechanism.

Xfce Diff is GUI to the GNU diff and patch commands. With this utility, you can view differences between for files or directories.

xhfs is a Macintosh file system browser

The Manuals Menu

The Manuals menu contains four online reference manuals:



pyFlag covers the basic operations of the pyFlag utility.

RAID-Reassembly details some of the issues involved with reconstructing RAID drives for forensic analysis.

Partition-Info lists the partition IDs for many different partition types.

Sleuthkit-Informer Articles contains 21 issues of the bi-monthly newsletter.

The Incident Response Menu

The Incident Response Menu contains 3 utilities:



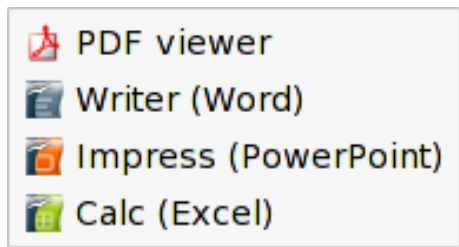
Ethereal is the classic network protocol analyzer.

ClamAV is the GPL Clam AntiVirus toolkit for UNIX.

F-Prot AntiVirus was developed to identify and remove viruses threatening workstations running Linux.

The Office Menu

The Office Menu contains 4 utilities:



PDF viewer to view Acrobat PDF files.

Writer – a MS Word compatible word processor.

Impress – a MS PowerPoint compatible presentation program.

Calc – a MS Excel compatible spread sheet program.

In addition, there are numerous other Linux based tools available in the other menus.



Adepto

Adepto was created by Drew Fahey of e-fense.com

Adepto is a GUI front-end to dd/dcfldd/sdd and was designed to simplify the creating of forensic bit images, and to automatically create a chain of custody.

Adepto Features

Adepto has several features and abilities, they include the following:

- auto-detection of IDE and SCSI drives, CD-ROMs, and tape drives
- choice of using either dd, dcfldd, or sdd
- image verification between source and copy via MD5 or SHA1
- image compression/decompression via gzip/bzip2
- image over a TCP/IP network via Netcat/Cryptcat, or SAMBA (NetBIOS)
- supports SCSI tape drives
- wiping (zeroing) drives or partitions
- splitting images into multiple segments
- Detailed logging with date/times and complete command-line used.

Starting Adepto

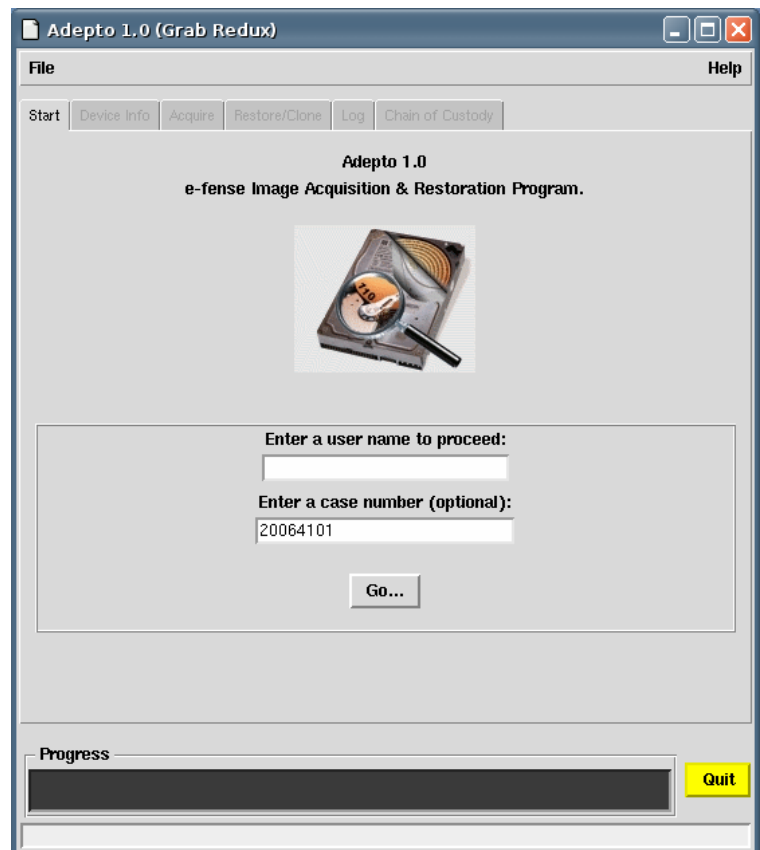
When you start Adepto, it will prompt you for a user and a case number. This is handy for keeping track of multiple cases, as well as maintaining a chain-of-custody.

The case number is based on the current date, but can be modified to fit the format of your case numbering system.

Once the user clicks “Go”, the program allows access to several tabs: Device Info, Acquire, Restore/Clone, Log, Chain of Custody.

Device Info

The Device Info tab will display information about the various devices on the system. Select the name of the device using the drop down box.



The download button next to the display box will list all of the devices connected to the system. If a device is not listed, click on the “[Rescan Devices]” hyperlink.

Once the device is selected, the information, such as the Make, Model, etc., for the device will be displayed.



Acquire

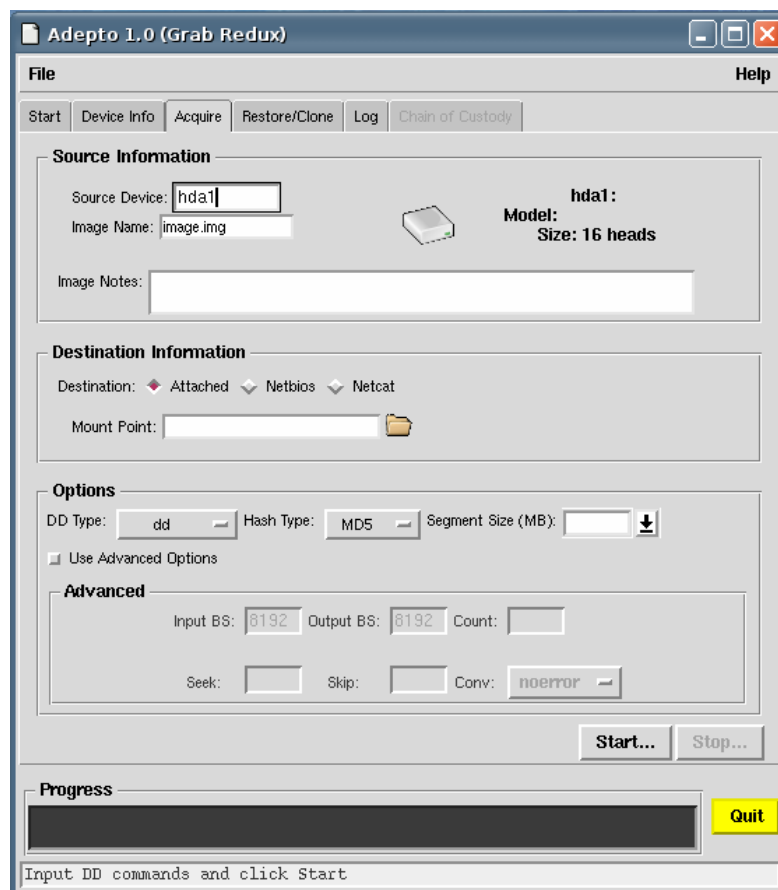
Once a device has been selected, the Acquire tab will become available, where the user can select various options for the actual copy. The user can enter optional image notes.

Under destination information, the user can select devices that are physically attached to the system, or connected to the network via NetBIOS or Netcat.

The user can also specify options for the dd command, including the hash type, the segment size, and advanced options including block sizes and more.

Once the options are selected, the user can select “Start...” to start the acquisition process.

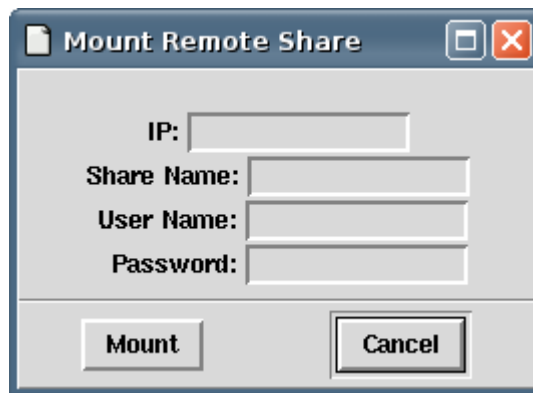
Since Adepto is just a front end to dd, you are really using dd to acquire your images. GRAB just makes the long and sometimes ugly command line easier to



manage. You should be familiar with the dd syntax before you embark on using Adepto as your acquisition tool, but it is not necessary. Just keep in mind **YOU CAN CAUSE PERMANENT DATA LOSS ON YOUR HARD DRIVES** if you reverse the source and destination devices.

If you want to send the acquired image to a network server through Netcat/Cryptcat or Samba using NetBIOS. Upon selecting the destination type you will be prompted for the additional information required to establish the connection.

If you want to use Netcat/Cryptcat, then type in the server's IP address and port number that the Netcat /Cryptcat server is listening on. If you select NetBIOS, you will need to click on the [Get share] hyperlink which will pop up a "Mount Remote Share" dialog box.



Restore/Clone

The Restore/Clone tab allows the user to restore an image to a device, or recombine the split images into a single file.

To restore a split image, the user specifies the first file in the split set (normally ending with a .000), then selects a destination drive, or a destination file. Clicking the Restore button will complete the task.

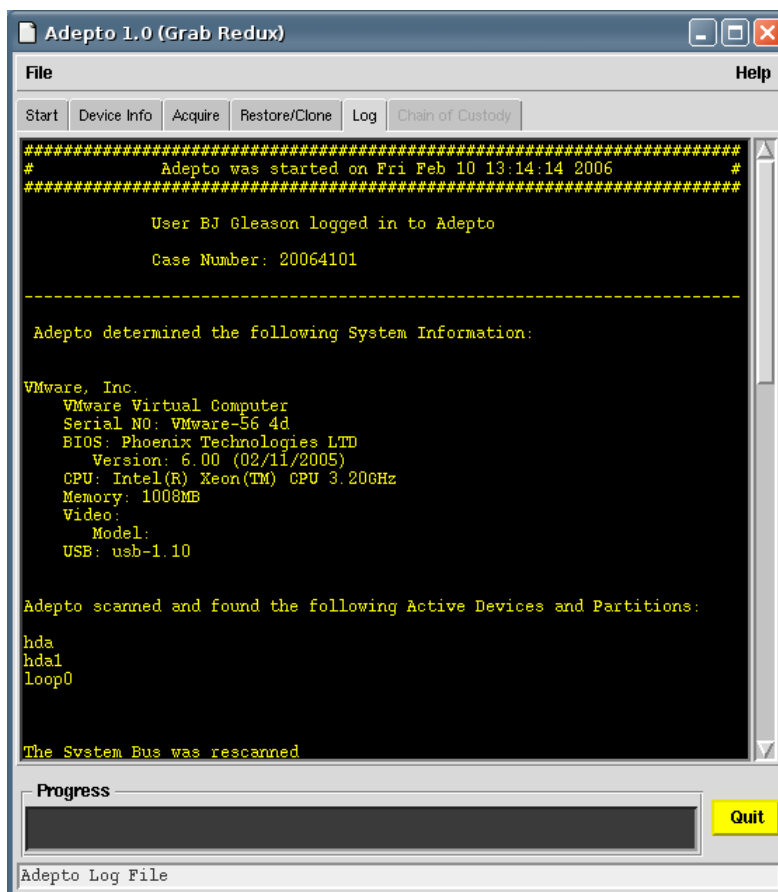
This tab also allows the user to clone one device to another. Similar to utilities like Drive Image and Ghost, except it will make a forensic copy of the source device.

In both cases, the destination device must be mounted as read/write.



Log

The log tab displays a details log of all the actions the user is making.



Chain of Custody

Adepto will automatically create a chain of custody form based on the device that was imaged. The user only needs to fill out only the evidence number and click the create button. A chain of custody form will be saved on your destination drive.



AIR: Automated Image and Restore

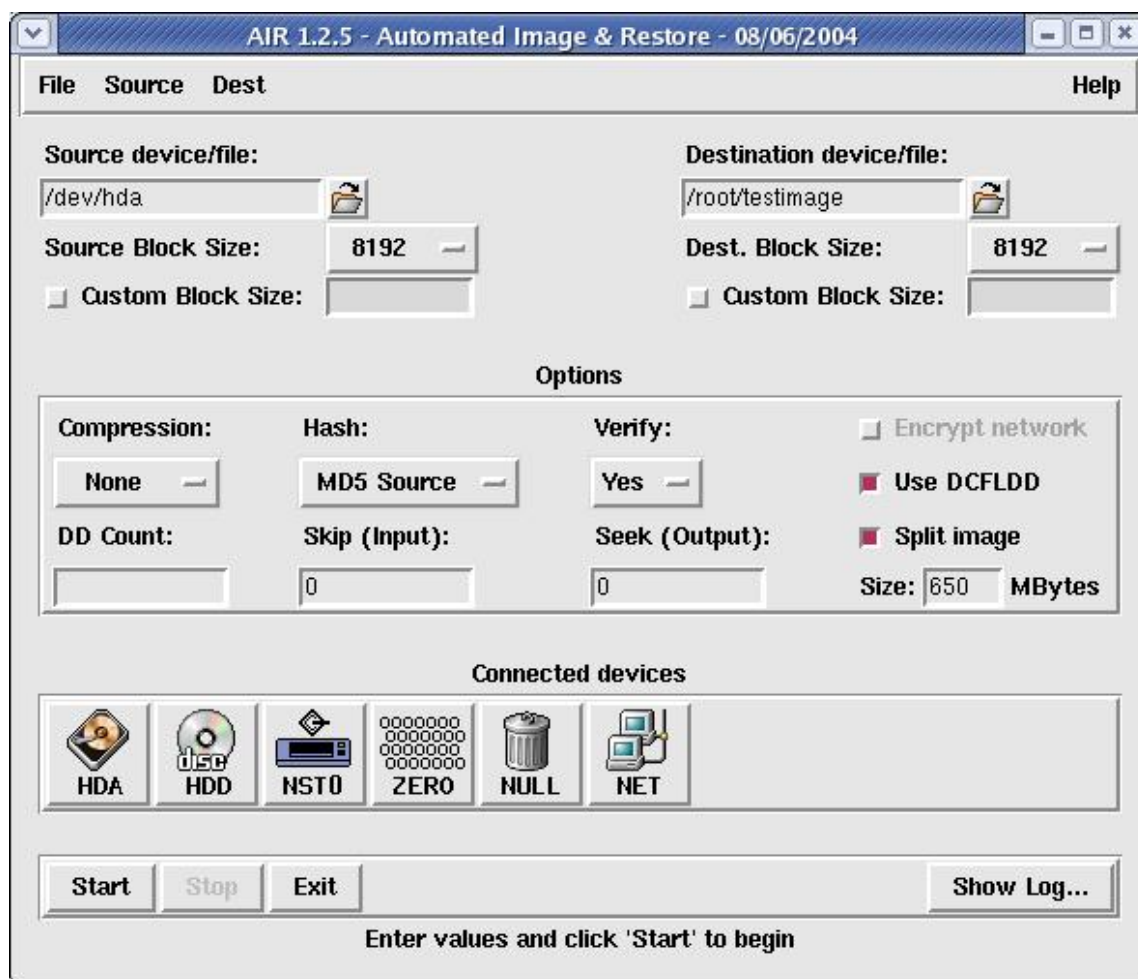
Developed by Steve Gibson. Available from <https://sourceforge.net/projects/air-imager/>

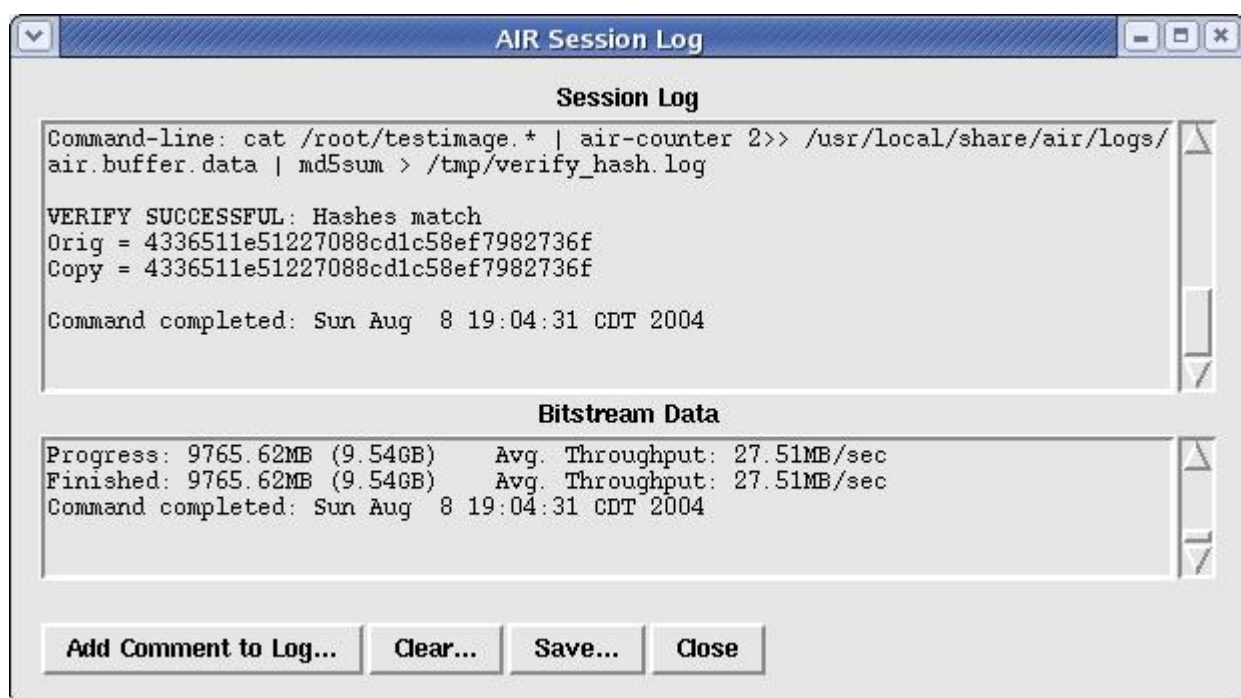
The following is taken from <http://air-imager.sourceforge.net/>

AIR (Automated Image and Restore) is a GUI front-end to dd/dcfldd designed for easily creating forensic bit images.

Features:

- auto-detection of IDE and SCSI drives, CD-ROMs, and tape drives
- choice of using either dd or dcfldd
- image verification between source and copy via MD5 or SHA1/256/384/512
- image compression/decompression via gzip/bzip2
- image over a TCP/IP network via netcat/cryptcat
- supports SCSI tape drives
- wiping (zeroing) drives or partitions
- splitting images into multiple segments
- detailed logging with date/times and complete command-line used







linen: EnCase Image Acquisition Tool.

Developed by EnCase. Available from <http://www.guidancesoftware.com>

The following is taken from http://www.guidancesoftware.com/products/v5_manualexcerpts.asp

The EnCase Linen utility allows you to acquire any device from a Linux-based forensic computer. The Linen utility provides an alternate method of acquiring a device via FastBloc in Windows, or EN.EXE in DOS. This method also allows users to hash any device present on the Linux operating system it is running on. With the introduction of Linen, users are now able to acquire Linux machines via a crossover cable from the Windows EnCase client by putting it into Server Mode. Linen is dependent on the distribution of Linux it is installed on. See the chapter in this document titled EnCase Linen Acquisition Utility for more detailed information.

The following is based on the directions from http://www.guidancesoftware.com/support/articles/acquire_safely.asp, and have been modified for Helix.

Local: "Linux Drive to Drive"

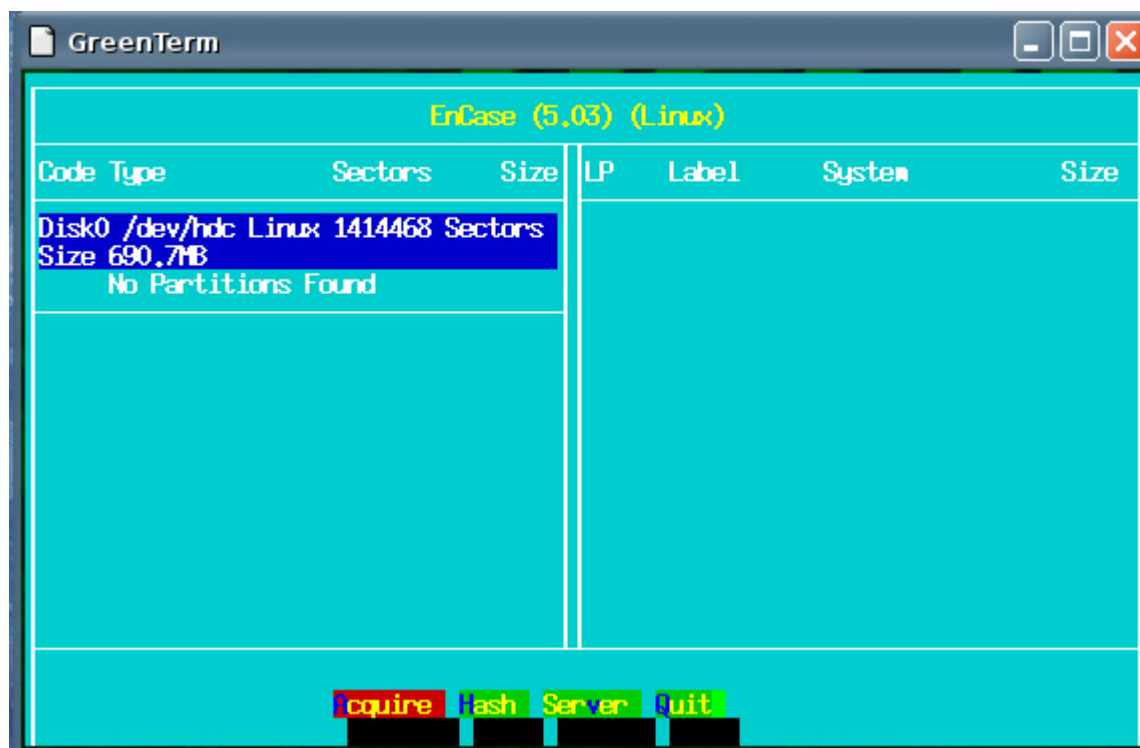
The Linux Local Method means that you are booting into a Non-Auto Mount distribution of Linux and:

- Acquiring the suspect's hard drive from within your own computer, or
- Acquiring the suspect's hard drive in his/her computer with your storage hard drive in his/her computer.

CAUTION: To perform this, you need to make certain that the computer containing the suspect hard drive will boot from your Non-Auto Mount Linux system **ONLY**. This is exceptionally important because if you accidentally boot up into the suspects operating system, or if your distribution of Linux "Auto-Mounts" the suspects hard drive, you will write to the subject's hard drive. Be careful and double check every step.

1. Attach the suspect Hard drive and a FAT32 formatted target drive to the computer
2. Be sure Helix CD is loaded, and the system is set to boot from the CD. Turn on the computer
3. Open a command shell. Change the FAT32 target drive to read/write access.
4. Create a Mount Point for your FAT32 storage hard drive by typing "**mkdir /media/FAT32**"
5. Determine the hard drive device name by examining the output of the command "**fdisk -l**"
 - a. As a general reference, Linux follows the below naming conventions:
 - i. hda - Primary Master
 - ii. hdb - Primary Slave
 - iii. hdc - Secondary Master
 - iv. hdb - Secondary Slave
 - v. SCSI, USB and FireWire devices are labeled as sda, sdb, sdc, etc...

6. Mount the storage partition to the mount point by typing "**mount /dev/hdx# /media/FAT32**"
Where 'hdx#' is the drive and partition you found above in step 5 (Example: hda3)
7. Execute the Linen program by typing "**linen**"



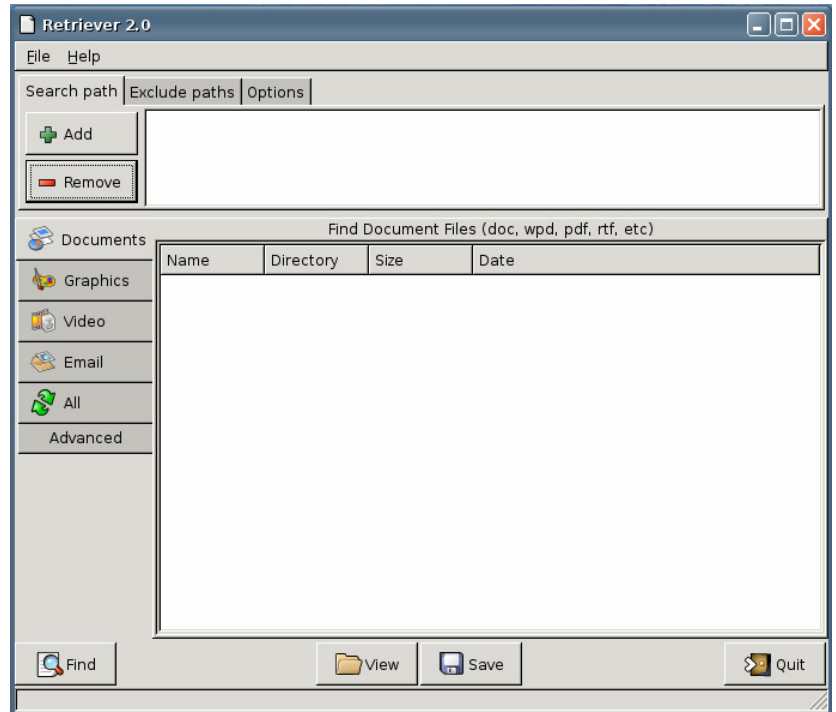
8. Select Acquire
9. Specify the target location, which should be **"/media/FAT32"**
10. Fill in the remaining required fields and the acquisition will begin
11. Once the acquisition is finished, exit EnCase for Linux
12. Shut down the system
13. Now remove the power and data cables from the suspect hard drive



Retriever

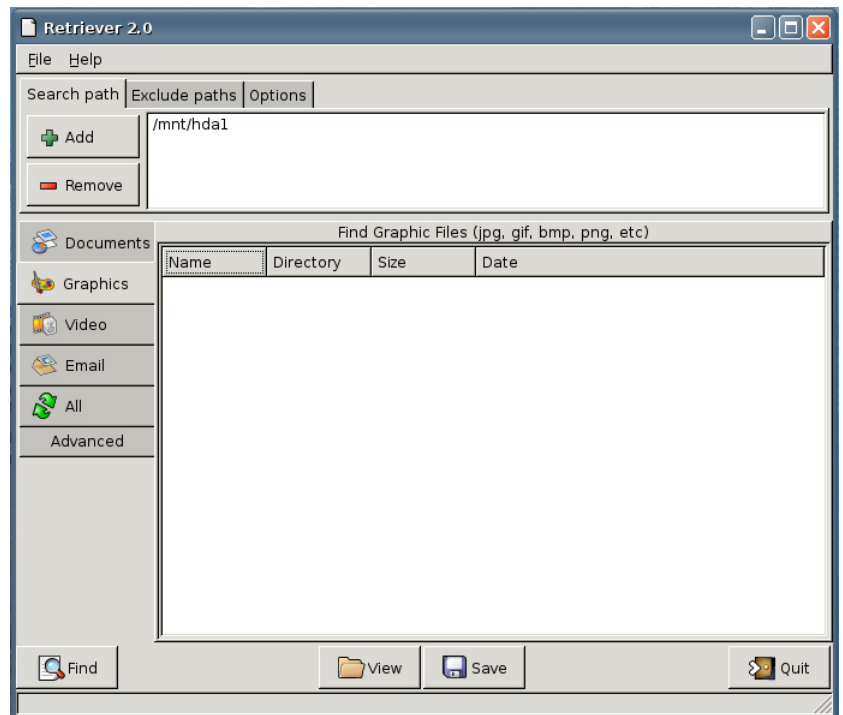
Retriever is a new tool created by me exclusively for the Helix CD. Retriever is an image (picture/video) capturing utility for “knock & talks”, “quick peeks”, and general searches. Retriever will scan a mounted device and locate all of the images and movie files and can place them onto a USB key (or local drive) as well as open an image viewer to view them.

When the program starts, the user can add the paths to examine by clicking on the “+ Add” button, navigating to the directory, and clicking OK.



Once the paths have been added, the user selects the type of files they are looking for.

In this case, the user is searching the target drive mounted at /mnt/hda1, and is looking for graphic files.

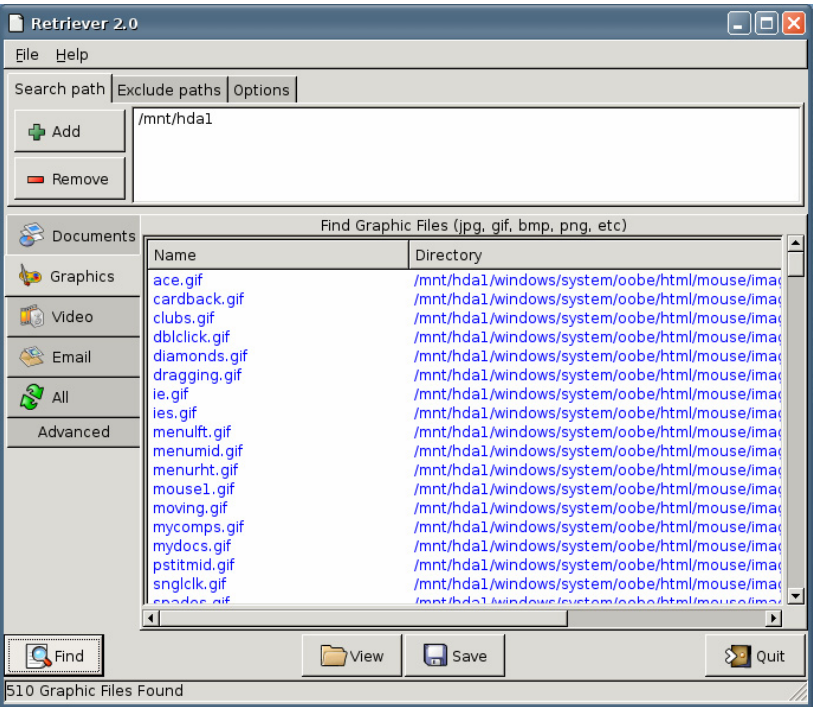


After a few minutes, the list of files matching the user's criteria is displayed.

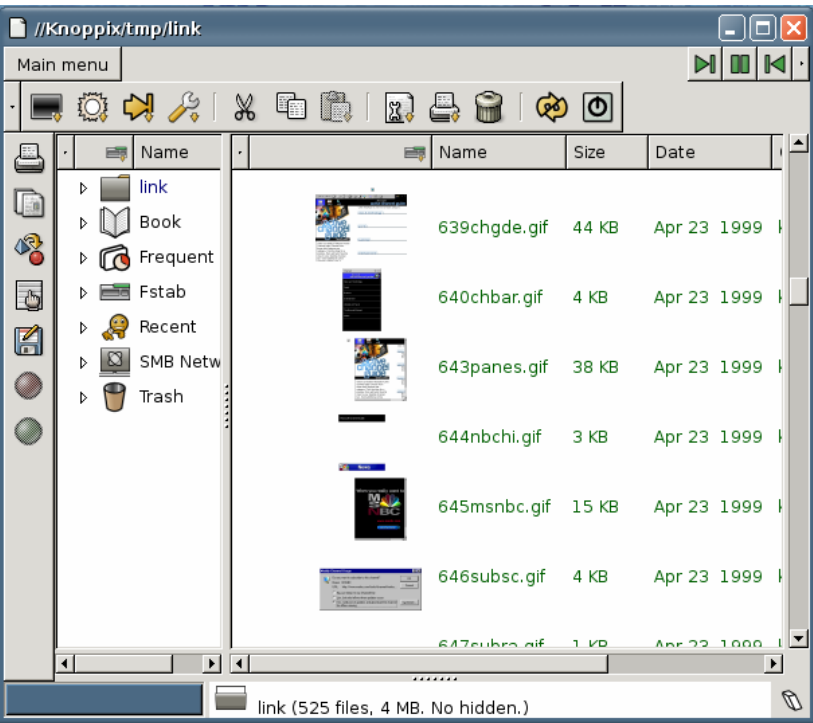
In this case, we see that 510 graphic files have been found.

Retriever has created a series of symbolic links to these files in the /tmp/links directory (this directory can be changed by accessing the "Options" tab).

The Save button will save a list of all the filenames and paths. It will not save the images.



Clicking the "View" button will bring up the file manager, which will display thumbnails of the images. Double-clicking on any of the images will bring up the image full size.



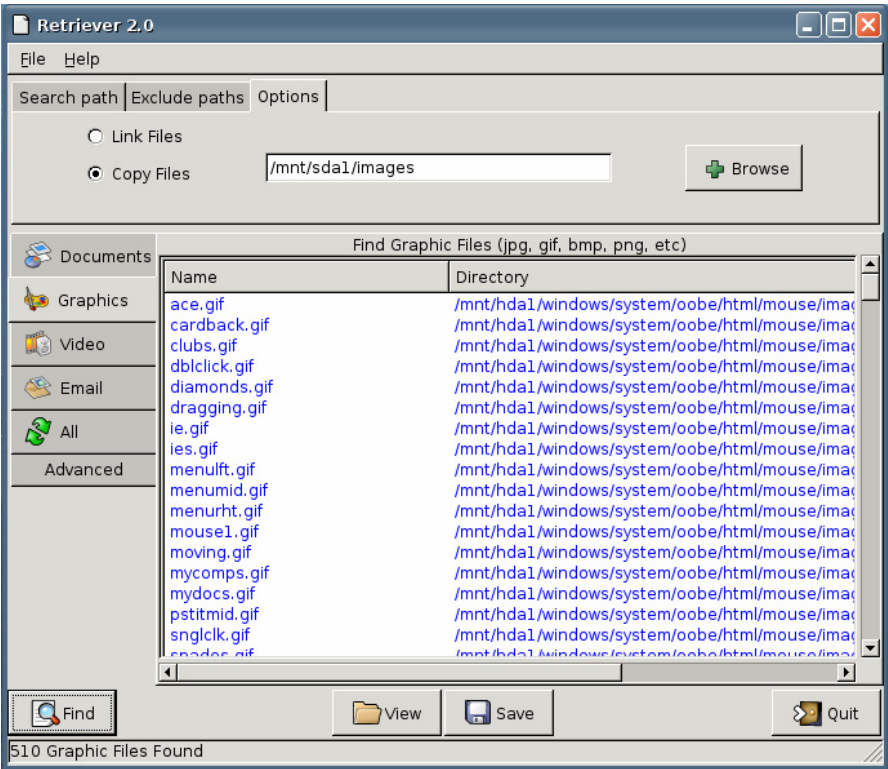
To save the files to a removable device, such as a USB drive, make sure that the drive is mounted as read/write, and then change the settings on the "Options" tab.

In this case, the user is going to copy the files to the /mnt/sda1/images directory, with is located on a removable USB drive.

You must click the “Find” button again, it will search for graphics, and as they are found, they will be copied to the specified directory.

In addition to the images, there will also be a text file called GRAPHIC-logfile containing the paths of the all the images.

Each search type will produce its own logfile.





Autopsy

Developed by Brian Carrier. Available from <http://www.sleuthkit.org/autopsy/>

The following is taken from <http://www.sleuthkit.org/autopsy/desc.php>

The Autopsy Forensic Browser is a graphical interface to the command line digital investigation analysis tools in [The Sleuth Kit](#). Together, they can analyze Windows and UNIX disks and file systems (NTFS, FAT, UFS1/2, Ext2/3).

The Sleuth Kit and Autopsy are both Open Source and run on UNIX platforms. As Autopsy is HTML-based, you can connect to the Autopsy server from any platform using an HTML browser. Autopsy provides a "File Manager"-like interface and shows details about deleted data and file system structures.

Analysis Modes

- A **dead analysis** occurs when a dedicated analysis system is used to examine the data from a suspect system. Autopsy and The Sleuth Kit are run in a trusted environment, typically in a lab.
- A **live analysis** occurs when the suspect system is being analyzed while it is running. In this case, Autopsy and The Sleuth Kit are run from a CD in an untrusted environment. This is frequently used during incident response while the incident is being confirmed. After it is confirmed, the system can be acquired and a dead analysis performed.

Evidence Search Techniques

- **File Listing:** Analyze the files and directories, including the names of deleted files and files with Unicode-based names.

The screenshot displays the Autopsy File Analysis window. The top menu bar includes: FILE ANALYSIS, KEYWORD SEARCH, FILE TYPE, IMAGE DETAILS, META DATA, DATA UNIT, HELP, and CLOSE. The main window is divided into two panes. The left pane shows a 'View Directory:' section with a text input field containing 'E:\', an 'OK' button, and two buttons labeled 'ALL DELETED FILES' and 'EXPAND DIRECTORIES'. The right pane contains a table listing files and directories.

File Name	Creation Time	Modification Time	Access Time	Size	Permissions	Owner	Group	Link
r/r label.exe	1996.10.14 01:38:00 (EDT)	2002.06.13 17:08:40 (EDT)	2002.06.13 17:08:45 (EDT)	32016	48	0		182-128-4
r/r legacy.inf	1996.10.14 01:38:00 (EDT)	2002.06.13 17:08:40 (EDT)	2002.06.13 17:08:40 (EDT)	4654	48	0		183-128-4
r/r lights.exe	1996.10.14 01:38:00 (EDT)	2002.06.13 17:08:40 (EDT)	2002.06.13 17:08:40 (EDT)	35600	48	0		184-128-4
r/- LMREPL.EXE	0000.00.00 00:00:00 (GMT)	0000.00.00 00:00:00 (GMT)	0000.00.00 00:00:00 (GMT)	0	0	0	0	185-128-4
r/r LMREPL.EXE	1996.10.14 01:38:00 (EDT)	2002.06.13 17:08:40 (EDT)	2002.06.13 17:08:45 (EDT)	86800	48	0		186-128-4
r/r loadfix.com	1996.10.14 01:38:00 (EDT)	2002.06.13 17:08:40 (EDT)	2002.06.13 17:08:40 (EDT)	1131	48	0		186-128-4 (realloc)
r/r loadfix.com	1996.10.14 01:38:00 (EDT)	2002.06.13 17:08:40 (EDT)	2002.06.13 17:08:40 (EDT)	1131	48	0		186-128-4

Below the table, there are links: ASCII (display - report) * Strings (display - report) * Export * Add Note. The file type is listed as: File Type: MS Windows PE 32-bit Intel 80386 GUI executable.

The bottom pane shows the 'String Contents Of File: E:\system32\inetins.exe'. The content includes:

```
!This program cannot be run in DOS mode.
.text
.rdata
.data
.rsrc
@.reloc
MSVCRT.dll
KERNEL32.dll
USER32.dll
OSVM
```

- **File Content:** The contents of files can be viewed in raw, hex, or the ASCII strings can be extracted. When data is interpreted, Autopsy sanitizes it to prevent damage to the local analysis system. Autopsy does not use any client-side scripting languages.



- **Hash Databases:** Lookup unknown files in a hash database to quickly identify it as good or bad. Autopsy uses the NIST National Software Reference Library (NSRL) and user created databases of known good and known bad files.
- **File Type Sorting:** Sort the files based on their internal signatures to identify files of a known type. Autopsy can also extract only graphic images (including thumbnails). The extension of the file will also be compared to the file type to identify files that may have had their extension changed to hide them.
- **Timeline of File Activity:** In some cases, having a timeline of file activity can help identify areas of a file system that may contain evidence. Autopsy can create timelines that contain entries for the Modified, Access, and Change (MAC) times of both allocated and unallocated files.

CREATE DATA FILE CREATE TIMELINE VIEW TIMELINE VIEW NOTES HELP CLOSE									
<- May 2002 Jul 2002 ->									
Jun 2002 OK									
Mon Jun 10 2002 19:33:10	3888	m..	-/-rwxrwxrwx	48	0	112-128-4	C:/system32/drivers/NTHANDLE.SYS		
Thu Jun 13 2002 21:01:34	22299	.ac	-/-rwxrwxrwx	48	0	263-128-4	C:/system32/oemnadem.inf		
Thu Jun 13 2002 21:01:35	20263	.ac	-/-rwxrwxrwx	48	0	270-128-4	C:/system32/oemnadlm.inf		
	39386	.c	-/-rwxrwxrwx	48	0	193-128-4	C:/system32/mem.exe		
	56	mac	d/drwxrwxrwx	48	0	49-144-7	C:/system32		
	9488	.c	-/-rwxrwxrwx	48	0	191-128-4	C:/system32/lsass.exe		
	9488	.c	-/-rwxrwxrwx	48	0	191-128-4	C:/system32/lsass.exe (deleted-realloc)		
	33662	.ac	-/-rwxrwxrwx	48	0	268-128-4	C:/system32/oemnadin.inf		
	86800	.c	-/-rwxrwxrwx	48	0	185-128-4	C:/system32/LMREPL.EXE		
	25491	.ac	-/-rwxrwxrwx	48	0	269-128-4	C:/system32/oemnadlb.inf		
	24391	.ac	-/-rwxrwxrwx	48	0	264-128-4	C:/system32/oemnaden.inf		
	22297	.ac	-/-rwxrwxrwx	48	0	266-128-4	C:/system32/oemnadfd.inf		
	85632	.c	-/-rwxrwxrwx	48	0	179-128-4	C:/system32/kml386.exe		
	22296	.ac	-/-rwxrwxrwx	48	0	267-128-4	C:/system32/oemnadim.inf		
	32016	.c	-/-rwxrwxrwx	48	0	182-128-4	C:/system32/label.exe		
	35225	.ac	-/-rwxrwxrwx	48	0	265-128-4	C:/system32/oemnadep.inf		

- **Keyword Search:** Keyword searches of the file system image can be performed using ASCII strings and grep regular expressions. Searches can be performed on either the full file system image or just the unallocated space. An index file can be created for faster searches. Strings that are frequently searched for can be easily configured into Autopsy for automated searching.
- **Meta Data Analysis:** Meta Data structures contain the details about files and directories. Autopsy allows you to view the details of any meta data structure in the file system. This is useful for recovering deleted content. Autopsy will search the directories to identify the full path of the file that has allocated the structure.
- **Data Unit Analysis:** Data Units are where the file content is stored. Autopsy allows you to view the contents of any data unit in a variety of formats including ASCII, hexdump, and strings. The file type is also given and Autopsy will search the meta data structures to identify which has allocated the data unit.
- **Image Details:** File system details can be viewed, including on-disk layout and times of activity. This mode provides information that is useful during data recovery.

Case Management

- **Case Management:** Investigations are organized by *cases*, which can contain one or more *hosts*. Each host is configured to have its own time zone setting and clock skew so that the times shown are the same as the original user would have seen. Each host can contain one or more file system images to analyze.
- **Event Sequencer:** Time-based events can be added from file activity or IDS and firewall logs. Autopsy sorts the events so that the sequence of incident events can be more easily determined.
- **Notes:** Notes can be saved on a per-host and per-investigator basis. These allow you to make quick notes about files and structures. The original location can be easily recalled with the click of a button when the notes are later reviewed. All notes are stored in an ASCII file.
- **Image Integrity:** It is crucial to ensure that files are not modified during analysis. Autopsy, by default, will generate an MD5 value for all files that are imported or created. The integrity of any file that Autopsy uses can be validated at any time.
- **Reports:** Autopsy can create ASCII reports for files and other file system structures. This enables you to quickly make consistent data sheets during the investigation.
- **Logging:** Audit logs are created on a case, host, and investigator level so that actions can be easily recalled. The exact Sleuth Kit commands that are executed are also logged.
- **Open Design:** The code of Autopsy is open source and all files that it uses are in a raw format. All configuration files are in ASCII text and cases are organized by directories. This makes it easy to export the data and archive it. It also does not restrict you from using other tools that may solve the specific problem more appropriately.
- **Client Server Model:** Autopsy is HTML-based and therefore you do not have to be on the same system as the file system images. This allows multiple investigators to use the same server and connect from their personal systems.

Autopsy is written in Perl and runs on the same UNIX platforms as The Sleuth Kit:

- Linux
- Mac OS X
- Open & FreeBSD
- Solaris



Developed by David Collett & Michael Cohen. Available from <http://pyflag.sourceforge.net/>



The following is taken from <http://pyflag.sourceforge.net/>

FLAG (Forensic and Log Analysis GUI) was designed to simplify the process of log file analysis and forensic investigations. Often, when investigating a large case, a great deal of data needs to be analysed and correlated. PyFlag uses a database as a backend to assist in managing the large volumes of data. This allows PyFlag to remain responsive and expedite data manipulation operations.

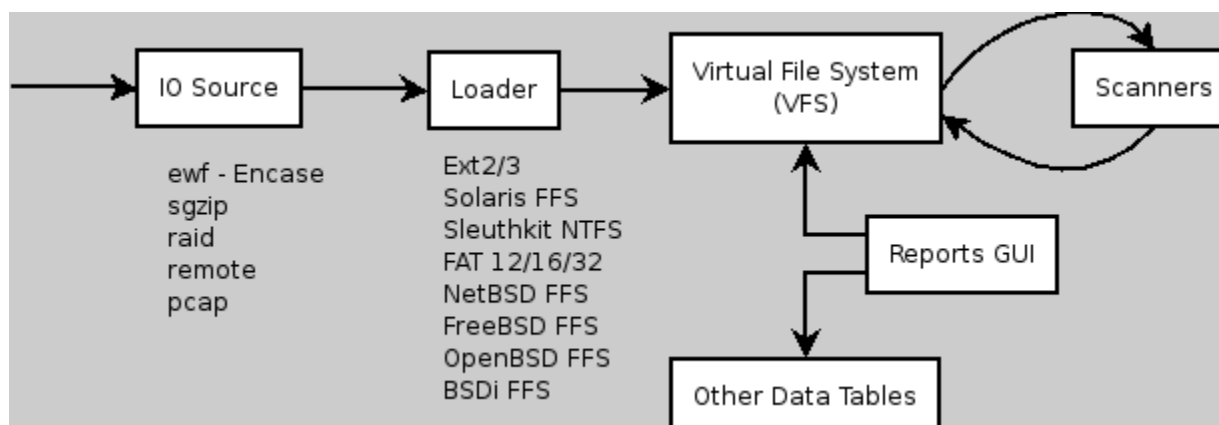
Since PyFLAG is web based, it is able to be deployed on a central server and shared with a number of users at the same time. Data is loaded into cases which keeps information separated.

PyFlag started off as a project in the Australian Department of Defence. It is now hosted on sourceforge.

The following is taken from <http://pyflag.sourceforge.net/Documentation/manual/index.html>

Overview

The general PyFlag architecture is shown below.



The following are the main components of PyFlag:

- **IO Sources:** Forensic data is often available in a variety of different formats. The IO Source is an abstraction allowing PyFlag to handle arbitrary input file types by using different drivers to present a consistent and uniform logical view of the data.
- **The FileSystem Loader:** Forensic images contain a variety of different filesystems. The FileSystem driver allows PyFlag to support different filesystem formats. The FileSystem Driver is responsible to initially populating the VFS with a listing of files found in the filesystem under investigation.

- The Virtual File System: PyFlag uses the original Unix idea that "everything is a file". The VFS is the main arena for presenting information to users. Files in the VFS do not necessarily exist in the image, but represent information which has been deduced about the filesystem.
- Scanners: Scanning is a process that passes all files in a certain directory through one or more scanners. A Scanner is a component which studies the files being scanned and collects information about these files. This might include adding new files to the VFS (which could be scanned again).
- The GUI and table widget: The GUI provides for a mechanism for examining the results of the scanners, and navigating the VFS. A *Report* is a limited set of functionality which provides access to specialised data collected by scanners.
- Scripting and automation: Its great being able to use the GUI for examining the data, but often we want to automate certain tasks so they may be done more efficiently. This section covers *Flash* (The Flag Shell).
- Network Forensics: This section describes the network forensics module of PyFlag.

There are some excellent tutorials available at:

- <http://wiki.lca2006.linux.org.au/PyFlag%2520Tutorial>
- <http://pyflag.sourceforge.net/Documentation/tutorials/index.html>



Regviewer

Developed by Chris Eagle. Available from <http://sourceforge.net/projects/regviewer/>

RegViewer is a windows registry file navigator. It is platform independent allowing for examination of Windows registry files from any platform. It is particularly useful when conducting forensics of Windows files from *nix systems.

Although the registry appears to be in one file, it is actually placed on your computer in several files. Depending on your system configuration, registry files can be found in any of the following locations:

For Windows 95, 98, and Me systems:

C:\Windows\System.dat

C:\Windows\User.dat

C:\Windows\Profiles\Policy.pol

For Windows 2000, and XP systems

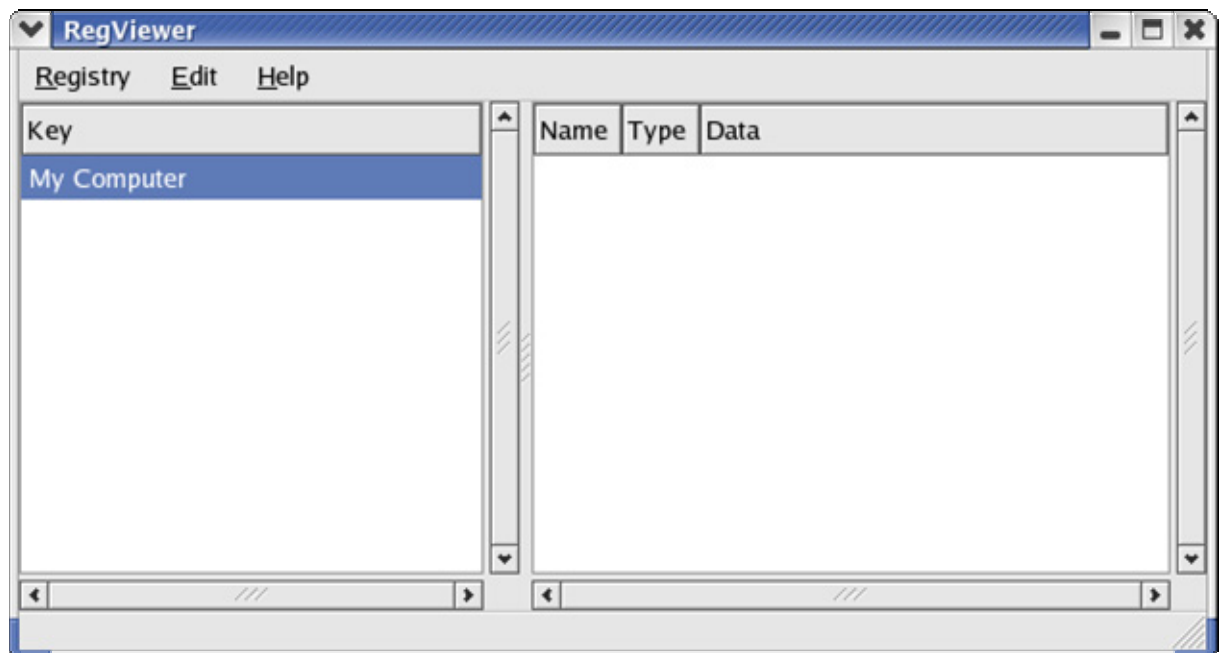
C:\Documents and Settings\User Name\Ntuser.dat

C:\Windows\System32\Config\Security, System.alt, Default, Sam, Software, System

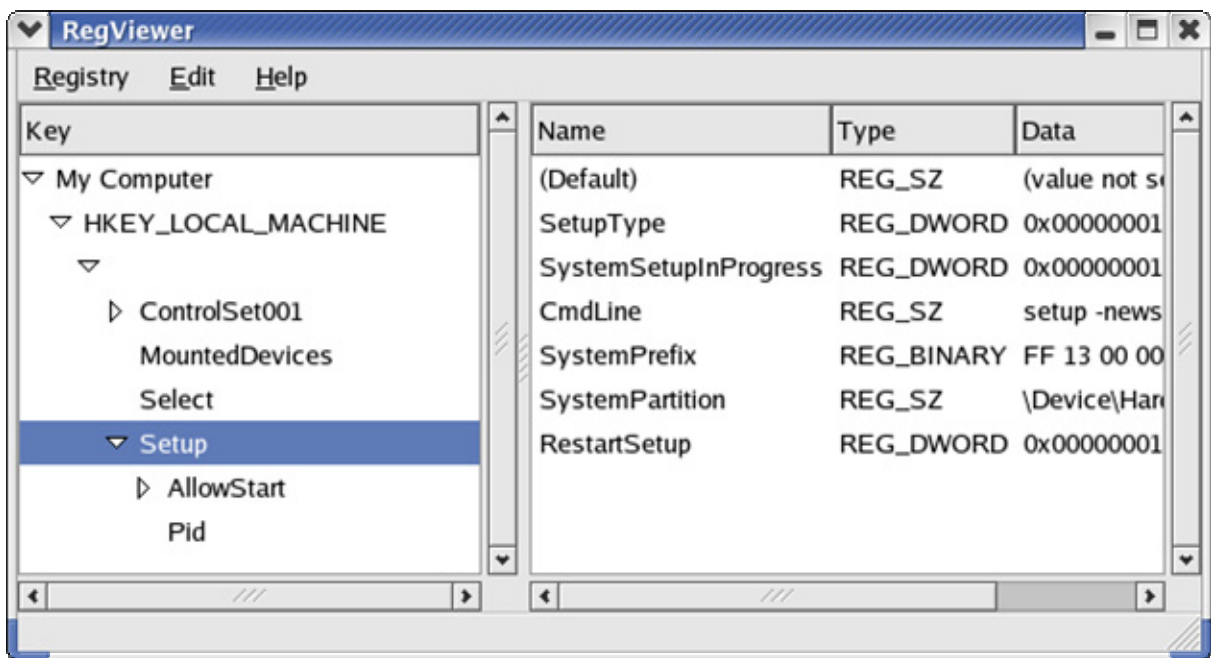
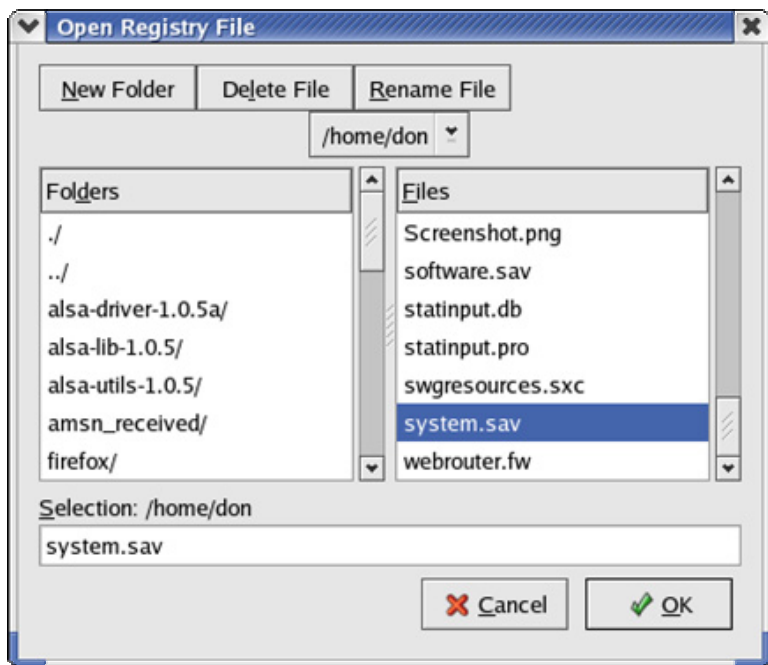
The following mages are taken from

<http://www.cs.usask.ca/undergrads/das322/Cmpt352/DFT/index.html>

When the regviewer application is started, you will see the following screen.



By selecting Registry / Import Registry File..., a directory window will open. Navigate to the location where the registry hive you wish to examine located.



The directory will be displayed, and can be navigated in a way very similar to the Windows Regedit program.

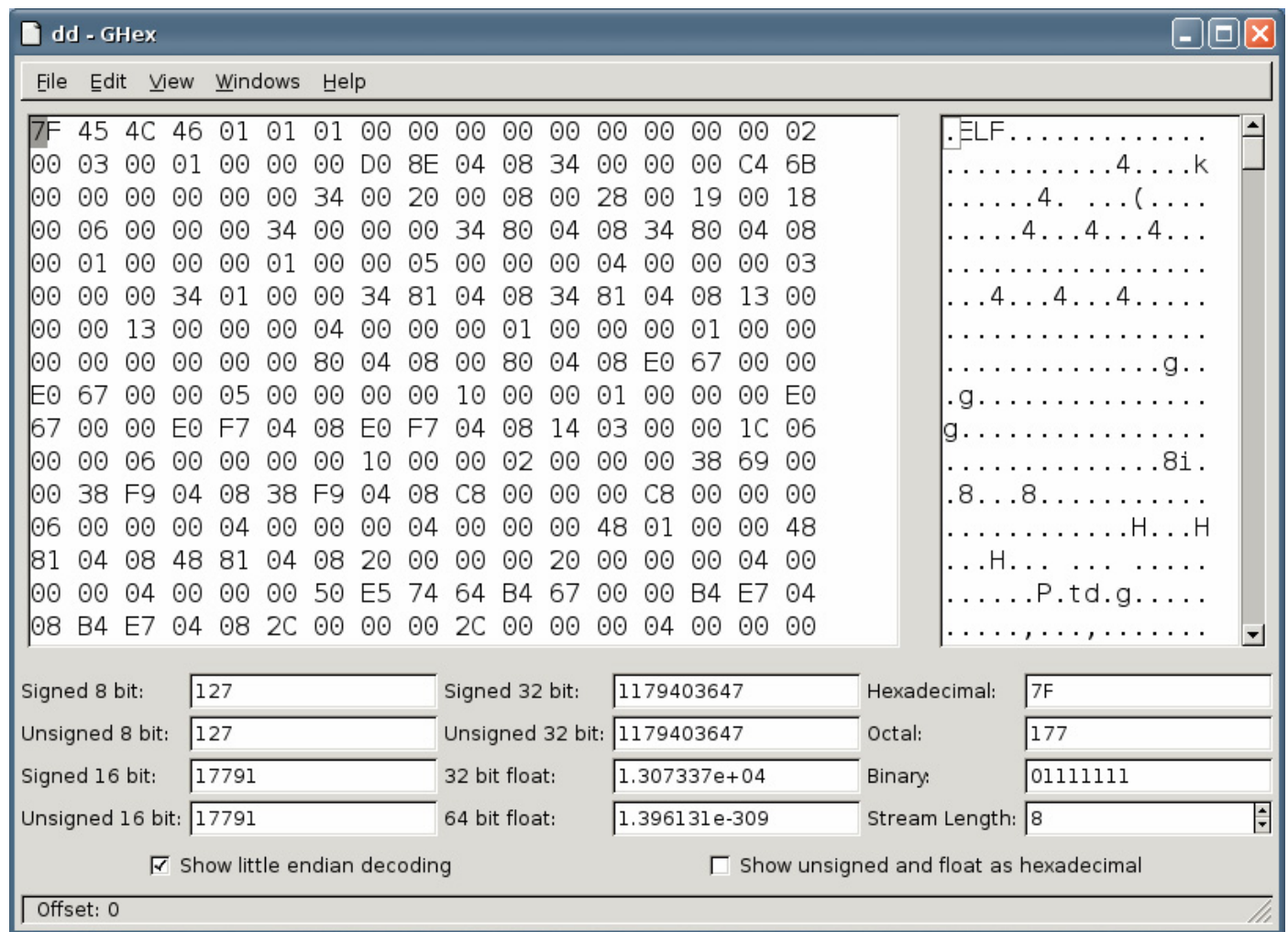


Hexeditor (GHex)

Developed by Jaka Mocnik. Available from <http://directory.fsf.org/text/editors/ghex.html>

GHex is a simple binary editor. It lets users view and edit a binary file in both hex and ascii with a multiple level undo/redo mechanism. Features include find and replace functions, conversion between binary, octal, decimal and hexadecimal values, and use of an alternative, user-configurable MDI concept that lets users edit multiple documents with multiple views of each.

To examine any file, simply select File / Open, and select the file to examine.

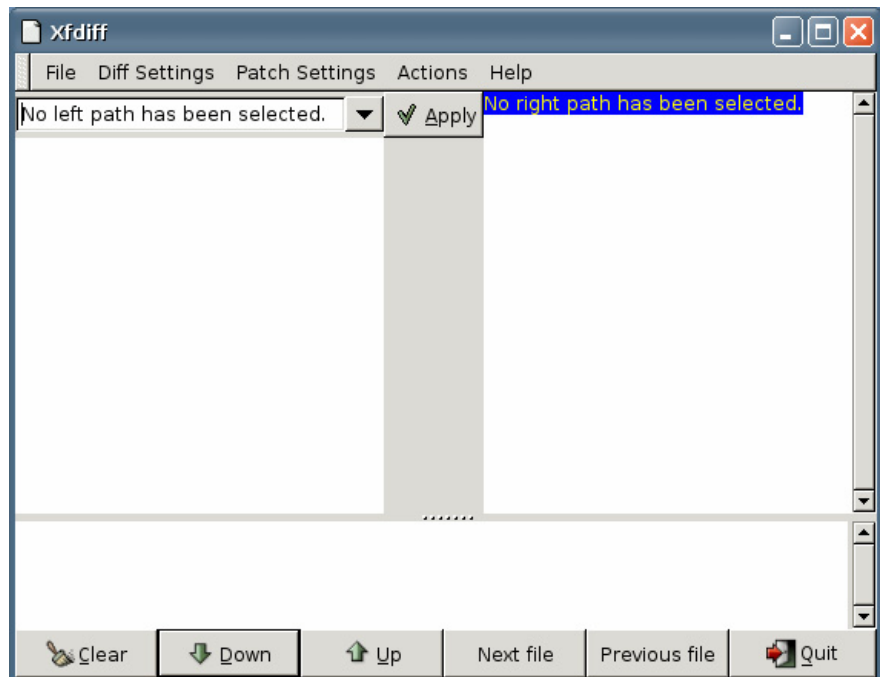




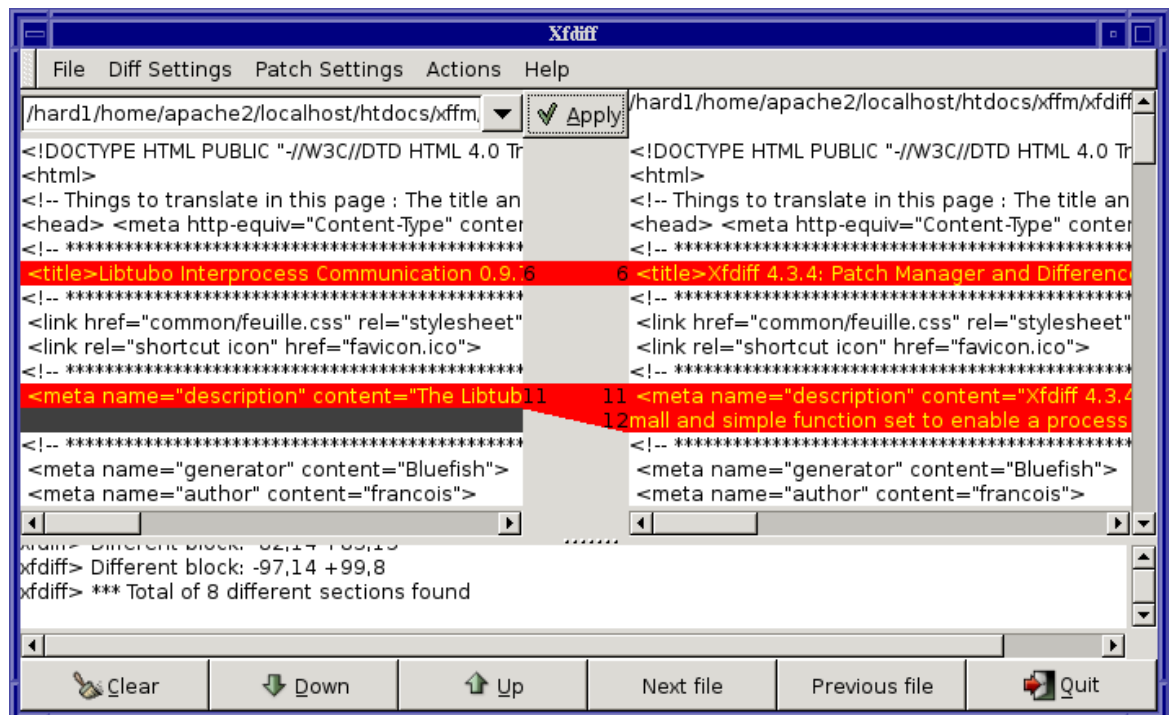
Xfce Diff

Developed by Edscott Garcia. Available from <http://www.xfce.org>

Xfdiff 4.3.4 is graphic interface to the GNU diff and patch commands. With this utility, you can view differences side by side for files or directories. You can also view differences that applying a patch file would imply, without applying the patch. You can also apply patches to the hard disc or create patch files for differences between files or directories.



Here is Xfdiff comparing two different files (image taken from <http://xffm.sourceforge.net/screenshots/xfdiff/>)





Developed by Robert Leslie. Available from <http://www.mars.org/home/rob/proj/hfs/>

The following was taken from http://linuxcommand.org/man_pages/xhfs1.html

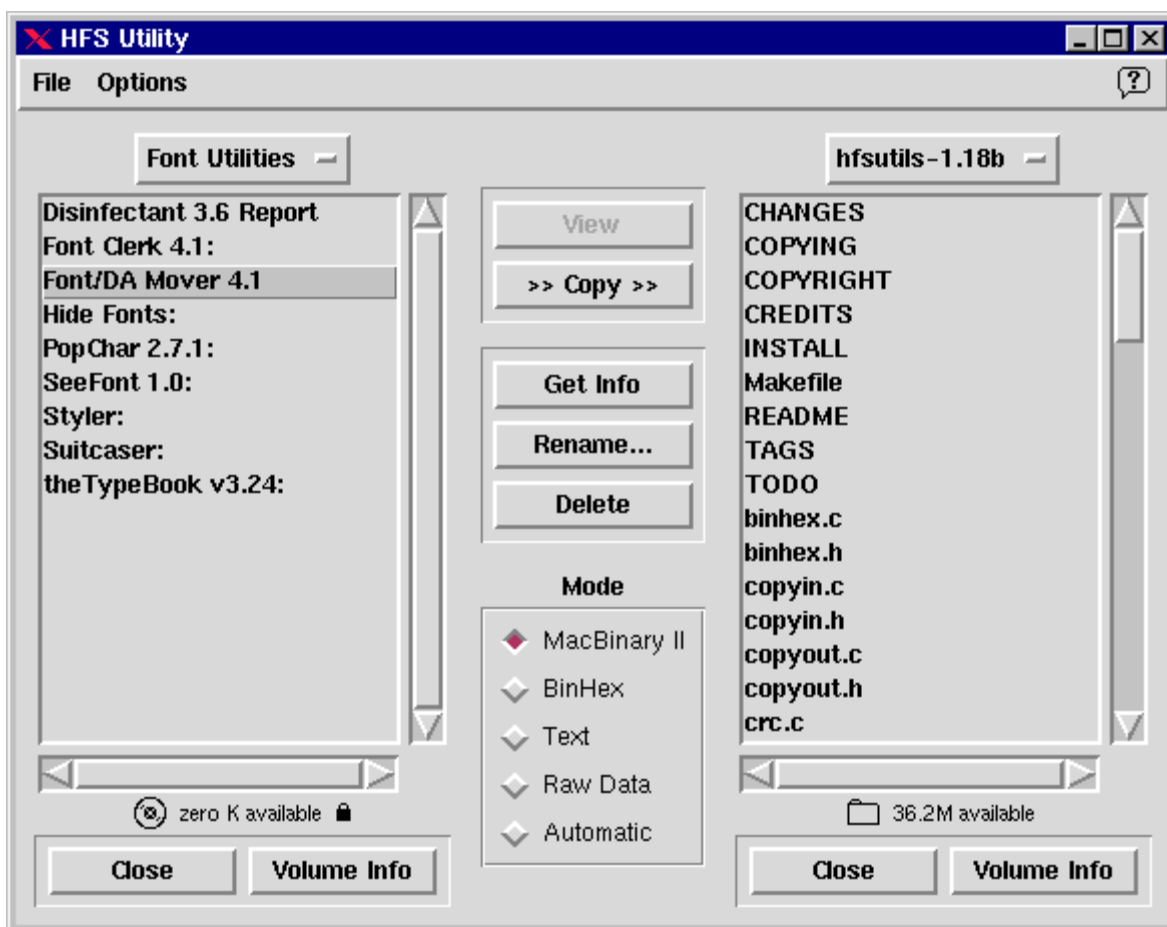
xhfs is a graphical interface for manipulating HFS (Macintosh) volumes

SYNOPSIS

xhfs [left-path [right-path]]

DESCRIPTION

xhfs presents a graphical front-end for browsing and copying files on HFS-formatted volumes.



The display is divided into two parts, left and right, which can each independently view a directory on either an HFS volume or the host (UNIX) filesystem. Double-clicking the name of a directory in either view will open that directory. A pop-up menu at the top of each directory view can be used to navigate to any directory between the current and the beginning of the hierarchy.

Text files can be viewed by double-clicking them. Any file or set of files can be copied to the directory shown in the other view by selecting them and clicking the "Copy" button. Copying is performed according to the selected copy mode:

MacBinary II

The file(s) will be copied using the MacBinary II format. This is the recommended mode for transferring arbitrary Macintosh files.

BinHex

The file(s) will be copied using the BinHex format. This mode should be used to encode Macintosh files into strict ASCII format.

Text

In this mode, only the data fork(s) of the selected file(s) are copied. Furthermore, translation is performed on the data's end-of-line characters to conform to the standard for text files on the destination.

Raw Data

In this mode, only the data fork(s) of the selected file(s) are copied. However, no translation is performed whatsoever on the data.

Automatic

A copy mode will be selected automatically according to a set of heuristics.



Ethereal

Developed by Gerald Combs. Available from <http://www.ethereal.com/>

Ethereal is a sophisticated, complex tool what will allow you to interactively browse network traffic. The next few pages simply have a reprint of the man page – enough to give you a quick look around. To really learn how to use this tool, head over to <http://www.ethereal.com/>, where you can download user manuals and other documentation. Be sure to check out the Ethereal Wiki - <http://wiki.ethereal.com/> - a treasure trove of tutorials, sample files, and how-to guides.

The following is taken from <http://www.ethereal.com/>

Ethereal® is used by network professionals around the world for troubleshooting, analysis, software and protocol development, and education. It has all of the standard features you would expect in a protocol analyzer, and several features not seen in any other product. Its open source license allows talented experts in the networking community to add enhancements. It runs on all popular computing platforms, including Unix, Linux, and Windows.

Ethereal is still technically beta software, but it has a comprehensive feature set and is suitable for production use. Here is the list of features, current as of version 0.9.14, in no particular order:

- Data can be captured "off the wire" from a live network connection, or read from a capture file.
- Ethereal can read capture files from tcpdump (libpcap), NAI's Sniffer™ (compressed and uncompressed), Sniffer™ Pro, NetXray™, Sun snoop and atmsnoop, and many other programs. Any of these files can be compressed with gzip and Ethereal will decompress them on the fly.
- Live data can be read from Ethernet, FDDI, PPP, Token-Ring, IEEE 802.11, Classical IP over ATM, and loopback interfaces (at least on some platforms; not all of those types are supported on all platforms).
- Captured network data can be browsed via a GUI, or via the TTY-mode "tethereal" program.
- Capture files can be programmatically edited or converted via command-line switches to the "editcap" program.
- 750 protocols can currently be dissected:
- Output can be saved or printed as plain text or PostScript®.
- Data display can be refined using a display filter.
- Display filters can also be used to selectively highlight and color packet summary information.
- All or part of each captured network trace can be saved to disk.

The following is from <http://www.ethereal.com/docs/man-pages/ethereal.1.html>

SYNOPSIS

ethereal [**-a** capture autostop condition] ... [**-b** capture ring buffer option] ... [**-B** capture buffer size (Win32 only)] [**-c** capture packet count] [**-f** capture filter] [**-g** packet number] [**-h**] [**-i** capture interface] [**-k**] [**-l**] [**-L**] [**-m** font] [**-n**] [**-N** name resolving flags] [**-o** preference/recent setting] ... [**-p**] [**-Q**] [**-r** infile] [**-R** read (display) filter] [**-S**] [**-s** capture snaplen] [**-t** time stamp format] [**-v**] [**-w** savefile] [**-y** capture link type] [**-z** statistics] [infile]

DESCRIPTION

Ethereal is a GUI network protocol analyzer. It lets you interactively browse packet data from a live network or from a previously saved capture file. **Ethereal**'s native capture file format is **libpcap** format, which is also the format used by **tcpdump** and various other tools.

Ethereal can read / import the following file formats:

- **libpcap, tcpdump and various other tools using tcpdump's capture format**
- **snoop and atmsnoop**
- **Shomiti/Finisar Surveyor captures**
- **Novell LANalyzer captures**
- **Microsoft Network Monitor captures**
- **AIX's iptrace captures**
- **Cinco Networks NetXRay captures**
- **Network Associates Windows-based Sniffer captures**
- **Network General/Network Associates DOS-based Sniffer (compressed or uncompressed) captures**
- **AG Group/WildPackets EtherPeek/TokenPeek/AiroPeek/EtherHelp/PackageGrabber captures**
- **RADCOM's WAN/LAN analyzer captures**
- **Network Instruments Observer version 9 captures**
- **Lucent/Ascend router debug output**
- **files from HP-UX's nettl**
- **Toshiba's ISDN routers dump output**
- **the output from i4btrace from the ISDN4BSD project**
- **traces from the EyeSDN USB S0.**
- **the output in IPLog format from the Cisco Secure Intrusion Detection System**
- **pppd logs (pppdump format)**
- **the output from VMS's TCPIPtrace/TCPtrace/UCX\$TRACE utilities**
- **the text output from the DBS Etherwatch VMS utility**
- **Visual Networks' Visual UpTime traffic capture**
- **the output from CoSine L2 debug**
- **the output from Accellent's 5Views LAN agents**
- **Endace Measurement Systems' ERF format captures**
- **Linux Bluez Bluetooth stack hcidump -w traces**

There is no need to tell **Ethereal** what type of file you are reading; it will determine the file type by itself. **Ethereal** is also capable of reading any of these file formats if they are compressed using gzip. **Ethereal** recognizes this directly from the file; the '.gz' extension is not required for this purpose.

Like other protocol analyzers, **Ethereal**'s main window shows 3 views of a packet. It shows a summary line, briefly describing what the packet is. A packet details display is shown, allowing you to drill down to exact protocol or field that you interested in. Finally, a hex dump shows you exactly what the packet looks like when it goes over the wire.

In addition, **Ethereal** has some features that make it unique. It can assemble all the packets in a TCP conversation and show you the ASCII (or EBCDIC, or hex) data in that conversation. Display filters in **Ethereal** are very powerful; more fields are filterable in **Ethereal** than in other protocol analyzers, and the syntax you can use to create your filters is richer. As **Ethereal** progresses, expect more and more protocol fields to be allowed in display filters.

Packet capturing is performed with the pcap library. The capture filter syntax follows the rules of the pcap library. This syntax is different from the display filter syntax.

Compressed file support uses (and therefore requires) the zlib library. If the zlib library is not present, **Ethereal** will compile, but will be unable to read compressed files.

The pathname of a capture file to be read can be specified with the **-r** option or can be specified as a command-line argument.

OPTIONS

Most users will want to start **Ethereal** without options and configure it from the menus instead. Those users may just skip this section.

-a
Specify a criterion that specifies when **Ethereal** is to stop writing to a capture file. The criterion is of the form *test:value*, where *test* is one of:

duration:value Stop writing to a capture file after *value* seconds have elapsed.

filesize:value Stop writing to a capture file after it reaches a size of *value* kilobytes (where a kilobyte is 1000 bytes, not 1024 bytes). If this option is used together with the **-b** option, **Ethereal** will stop writing to the current capture file and switch to the next one if filesize is reached.

files:value Stop writing to capture files after *value* number of files were written.

-b
Cause **Ethereal** to run in ``multiple files" mode. In ``multiple files" mode, **Ethereal** will write to several capture files. When the first capture file fills up, **Ethereal** will switch writing to the next file and so on.

The created filenames are based on the filename given with the **-w** flag, the number of the file and on the creation date and time, e.g. `savefile_00001_20050604120117.pcap`, `savefile_00001_20050604120523.pcap`, ...

With the *files* option it's also possible to form a ``ring buffer". This will fill up new files until the number of files specified, at which point **Ethereal** will discard the data in the first file and start

writing to that file and so on. If the *files* option is not set, new files filled up until one of the capture stop conditions match (or until the disk is full).

The criterion is of the form *key:value*, where *key* is one of:

duration:*value* switch to the next file after *value* seconds have elapsed, even if the current file is not completely filled up.

filesize:*value* switch to the next file after it reaches a size of *value* kilobytes (where a kilobyte is 1000 bytes, not 1024 bytes).

files:*value* begin again with the first file after *value* number of files were written (form a ring buffer).

-B Win32 only: set capture buffer size (in MB, default is 1MB). This is used by the the capture driver to buffer packet data until that data can be written to disk. If you encounter packet drops while capturing, try to increase this size.

-c Set the maximum number of packets to read when capturing live data.

-f Set the capture filter expression.

-g After reading in a capture file using the **-r** flag, go to the given *packet number*.

-h Print the version and options and exit.

-i Set the name of the network interface or pipe to use for live packet capture.

Network interface names should match one of the names listed in **``tethereal -D''**. If you're using Unix, **``netstat -i''** or **``ifconfig -a''** might also work to list interface names, although not all versions of Unix support the **-a** flag to **ifconfig**.

Pipe names should be either the name of a FIFO (named pipe) or **``-''** to read data from the standard input. Data read from pipes must be in standard libpcap format.

-k Start the capture session immediately. If the **-i** flag was specified, the capture uses the specified interface. Otherwise, **Ethereal** searches the list of interfaces, choosing the first non-loopback interface if there are any non-loopback interfaces, and choosing the first loopback interface if there are no non-loopback interfaces; if there are no interfaces, **Ethereal** reports an error and doesn't start the capture.

-l Turn on automatic scrolling if the packet display is being updated automatically as packets arrive during a capture (as specified by the **-S** flag).

-L List the data link types supported by the interface and exit.

-m Set the name of the font used by **Ethereal** for most text. **Ethereal** will construct the name of the bold font used for the data in the byte view pane that corresponds to the field selected in the packet details pane from the name of the main text font.

- n**
Disable network object name resolution (such as hostname, TCP and UDP port names), the **-N** flag might override this one.
- N**
Turn on name resolving only for particular types of addresses and port numbers, with name resolving for other types of addresses and port numbers turned off. This flag overrides **-n** if both **-N** and **-n** are present. If both **-N** and **-n** flags are not present, all name resolutions are turned on.

The argument is a string that may contain the letters:

m to enable MAC address resolution

n to enable network address resolution

t to enable transport-layer port number resolution

C to enable concurrent (asynchronous) DNS lookups
- o**
Set a preference or recent value, overriding the default value and any value read from a preference/recent file. The argument to the flag is a string of the form *prefname:value*, where *prefname* is the name of the preference/recent value (which is the same name that would appear in the preference/recent file), and *value* is the value to which it should be set. Since **Ethereal** 0.10.12, the recent settings replaces the formerly used **-B**, **-P** and **-T** flags to manipulate the GUI dimensions.
- p**
Don't put the interface into promiscuous mode. Note that the interface might be in promiscuous mode for some other reason; hence, **-p** cannot be used to ensure that the only traffic that is captured is traffic sent to or from the machine on which **Ethereal** is running, broadcast traffic, and multicast traffic to addresses received by that machine.
- Q**
Cause **Ethereal** to exit after the end of capture session (useful in batch mode with **-c** option for instance); this option requires the **-i** and **-w** parameters.
- r**
Read packet data from *infile*.
- R**
When reading a capture file specified with the **-r** flag, causes the specified filter (which uses the syntax of display filters, rather than that of capture filters) to be applied to all packets read from the capture file; packets not matching the filter are discarded.
- S**
Automatically update the packet display as packets are coming in.
- s**
Set the default snapshot length to use when capturing live data. No more than *snaplen* bytes of each network packet will be read into memory, or saved to disk.
- t**
Set the format of the packet timestamp displayed in the packet list window, the default is relative. The format can be one of:

r relative: The relative time is the time elapsed between the first packet and the current packet

a absolute: The absolute time is the actual time the packet was captured, with no date displayed

ad absolute with date: The absolute date and time is the actual time and date the packet was captured

d delta: The delta time is the time since the previous packet was captured

-v

Print the version and exit.

-w

Set the default capture file name.

-y

If a capture is started from the command line with **-k**, set the data link type to use while capturing packets. The values reported by **-L** are the values that can be used.

-z

Get **Ethereal** to collect various types of statistics and display the result in a window that updates in semi-real time. Currently implemented statistics are:

-z dcerpc,srt,uuid,major.minor[,<filter>]

Collect call/reply SRT (Service Response Time) data for DCERPC interface *uuid*, version *major.minor*. Data collected is number of calls for each procedure, MinSRT, MaxSRT and AvgSRT. Example: use **-z dcerpc,srt,12345778-1234-abcd-ef00-0123456789ac,1.0** to collect data for CIFS SAMR Interface. This option can be used multiple times on the command line.

If the optional filterstring is provided, the stats will only be calculated on those calls that match that filter. Example: use **-z dcerpc,srt,12345778-1234-abcd-ef00-0123456789ac,1.0,ip.addr==1.2.3.4** to collect SAMR SRT statistics for a specific host.

-z io,stat

Collect packet/bytes statistics for the capture in intervals of 1 seconds. This option will open a window with up to 5 color-coded graphs where number-of-packets-per-second or number-of-bytes-per-second statistics can be calculated and displayed.

This option can be used multiple times on the command line.

This graph window can also be opened from the Analyze:Statistics:Traffic:IO-Stat menu item.

-z rpc,srt,program,version[,<filter>]

Collect call/reply SRT (Service Response Time) data for *program/version*. Data collected is number of calls for each procedure, MinSRT, MaxSRT and AvgSRT. Example: use **-z rpc,srt,100003,3** to collect data for NFS v3. This option can be used multiple times on the command line.

If the optional filter string is provided, the stats will only be calculated on those calls that match that filter. Example: use **-z rpc,srt,100003,3,nfs.fh.hash==0x12345678** to collect NFS v3 SRT statistics for a specific file.

-z rpc,programs

Collect call/reply RTT data for all known ONC-RPC programs/versions. Data collected is number of calls for each protocol/version, MinRTT, MaxRTT and AvgRTT.

-z smb,srt[,filter]

Collect call/reply SRT (Service Response Time) data for SMB. Data collected is number of calls for each SMB command, MinSRT, MaxSRT and AvgSRT. Example: use **-z smb,srt**.

The data will be presented as separate tables for all normal SMB commands, all Transaction2 commands and all NT Transaction commands. Only those commands that are seen in the capture will have its stats displayed. Only the first command in a xAndX command chain will be used in the calculation. So for common SessionSetupAndX + TreeConnectAndX chains, only the SessionSetupAndX call will be used in the statistics. This is a flaw that might be fixed in the future.

This option can be used multiple times on the command line.

If the optional filterstring is provided, the stats will only be calculated on those calls that match that filter. Example: use **-z ``smb,srt,ip.addr==1.2.3.4''** to only collect stats for SMB packets exchanged by the host at IP address 1.2.3.4 .

-z fc,srt[,filter]

Collect call/reply SRT (Service Response Time) data for FC. Data collected is number of calls for each Fibre Channel command, MinSRT, MaxSRT and AvgSRT. Example: use **-z fc,srt**. The Service Response Time is calculated as the time delta between the First packet of the exchange and the Last packet of the exchange.

The data will be presented as separate tables for all normal FC commands, Only those commands that are seen in the capture will have its stats displayed.

This option can be used multiple times on the command line.

If the optional filterstring is provided, the stats will only be calculated on those calls that match that filter. Example: use **-z ``fc,srt,fc.id==01.02.03''** to only collect stats for FC packets exchanged by the host at FC address 01.02.03 .

-z ldap,srt[,filter]

Collect call/reply SRT (Service Response Time) data for LDAP. Data collected is number of calls for each implemented LDAP command, MinSRT, MaxSRT and AvgSRT. Example: use **-z ldap,srt**. The Service Response Time is calculated as the time delta between the Request and the Response.

The data will be presented as separate tables for all implemented LDAP commands, Only those commands that are seen in the capture will have its stats displayed.

This option can be used multiple times on the command line.

If the optional filterstring is provided, the stats will only be calculated on those calls that match that filter. Example: use **-z ``ldap,srt,ip.addr==10.1.1.1''** to only collect stats for LDAP packets exchanged by the host at IP address 10.1.1.1 .

The only LDAP command that are currently implemented and the stats will be available for are: BIND SEARCH MODIFY ADD DELETE MODRDN COMPARE EXTENDED

-z mgcp,srt[,filter]

Collect requests/response SRT (Service Response Time) data for MGCP. This is similar to **-z smb,srt**. Data collected is number of calls for each known MGCP Type, Minimum SRT, Maximum SRT and Average SRT. Example: use **-z mgcp,srt**.

This option can be used multiple times on the command line.

If the optional filterstring is provided, the stats will only be calculated on those calls that match that filter. Example: use **-z ``mgcp,srt,ip.addr==1.2.3.4``** to only collect stats for MGCP packets exchanged by the host at IP address 1.2.3.4 .

-z conv,type[,filter]

Create a table that lists all conversations that could be seen in the capture. *type* specifies for which type of conversation we want to generate the statistics; currently the supported ones are

"eth"	Ethernet	
"fc"	Fibre Channel addresses	
"fddi"	FDDI addresses	
"ip"	IP addresses	
"ipx"	IPX addresses	
"tcp"	TCP/IP socket pairs	Both IPv4 and IPv6 are supported
"tr"	TokenRing	
"udp"	UDP/IP socket pairs	Both IPv4 and IPv6 are supported

If the optional filter string is specified, only those packets that match the filter will be used in the calculations.

The table is presented with one line for each conversation and displays number of packets/bytes in each direction as well as total number of packets/bytes. By default, the table is sorted according to total number of packets.

These tables can also be generated at runtime by selecting the appropriate conversation type from the menu ``Tools/Statistics/Conversation List``.

-z h225,counter[,filter]

Count ITU-T H.225 messages and their reasons. In the first column you get a list of H.225 messages and H.225 message reasons, which occur in the current capture file. The number of occurrences of each message or reason is displayed in the second column.

Example: use **-z h225,counter**.

This option can be used multiple times on the command line.

If the optional filterstring is provided, the stats will only be calculated on those calls that match that filter. Example: use **-z ``h225,counter,ip.addr==1.2.3.4``** to only collect stats for H.225 packets exchanged by the host at IP address 1.2.3.4 .

-z h225,srt[,filter]

Collect requests/response SRT (Service Response Time) data for ITU-T H.225 RAS. Data collected is number of calls of each ITU-T H.225 RAS Message Type, Minimum SRT, Maximum SRT, Average SRT, Minimum in Packet, and Maximum in Packet. You will also get the number of Open Requests (Unresponded Requests), Discarded Responses (Responses without matching request) and Duplicate Messages. Example: use **-z h225,srt**.

This option can be used multiple times on the command line.

If the optional filterstring is provided, the stats will only be calculated on those calls that match that filter. Example: use **-z ``h225,srt,ip.addr==1.2.3.4''** to only collect stats for ITU-T H.225 RAS packets exchanged by the host at IP address 1.2.3.4 .

-z sip,stat[*filter*]

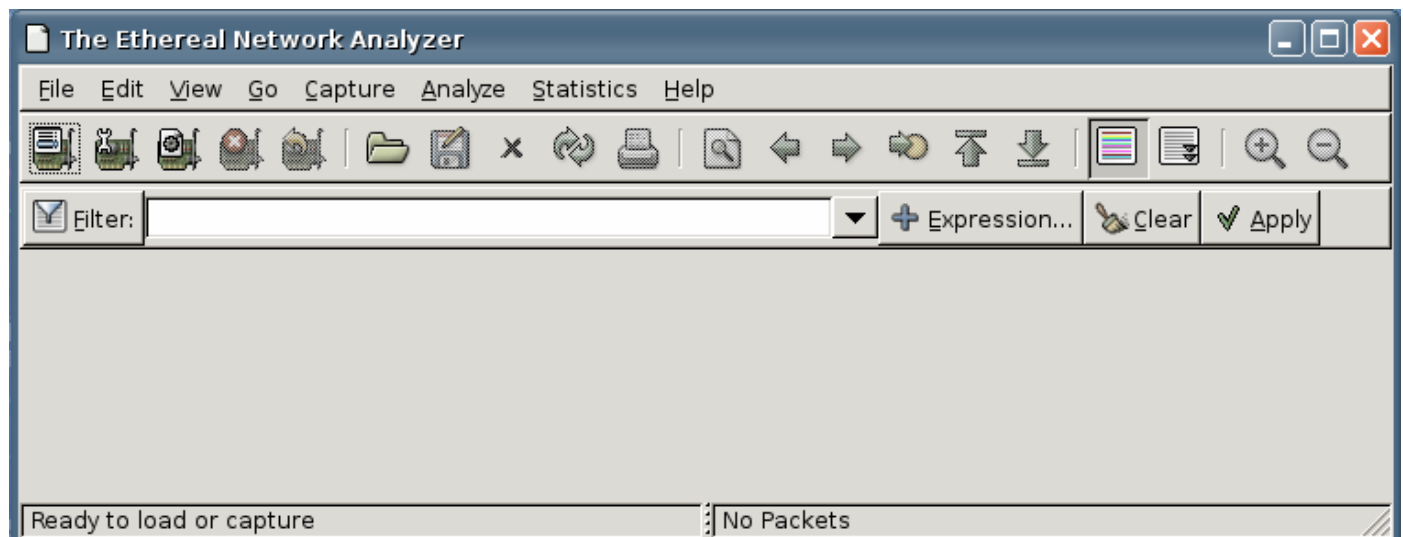
This option will activate a counter for SIP messages. You will get the number of occurrences of each SIP Method and of each SIP Status-Code. Additionally you also get the number of resent SIP Messages (only for SIP over UDP).

Example: use **-z sip,stat**.

This option can be used multiple times on the command line.

If the optional filter string is provided, the stats will only be calculated on those calls that match that filter. Example: use **-z ``sip,stat,ip.addr==1.2.3.4''** to only collect stats for SIP packets exchanged by the host at IP address 1.2.3.4 .

INTERFACE



MENU ITEMS

File:Open

File:Open Recent

File:Close

Open or close a capture file. The *File:Open* dialog box allows a filter to be specified; when the capture file is read, the filter is applied to all packets read from the file, and packets not matching the filter are discarded. The *File:Open Recent* is a submenu and will show a list of previously opened files.

File:Merge

Merge another capture file to the currently loaded one. The *File:Merge* dialog box allows the merge ``Prepended'', ``Chronologically'' or ``Appended'', relative to the already loaded one.

File:Save**File:Save As**

Save the current capture, or the packets currently displayed from that capture, to a file. Check boxes let you select whether to save all packets, or just those that have passed the current display filter and/or those that are currently marked, and an option menu lets you select (from a list of file formats in which at particular capture, or the packets currently displayed from that capture, can be saved), a file format in which to save it.

File:File Set:List Files

Show a dialog box that list all files of the file set matching the currently loaded file. A file set is a compound of files resulting from a capture using the ``multiple files'' / ``ringbuffer'' mode, recognizable by the filename pattern, e.g.: Filename_00001_20050604101530.pcap.

File:File Set:Next File**File:File Set:Previous File**

If the currently loaded file is part of a file set (see above), open the next / previous file in that set.

File:Export

Export captured data into an external format. Note: the data cannot be imported back into Ethereal, so be sure to keep the capture file.

File:Print

Print packet data from the current capture. You can select the range of packets to be printed (which packets are printed), and the output format of each packet (how each packet is printed). The output format will be similar to the displayed values, so a summary line, the packet details view, and/or the hex dump of the packet can be printed.

Printing options can be set with the *Edit:Preferences* menu item, or in the dialog box popped up by this menu item.

File:Quit

Exit the application.

Edit:Find Packet

Search forward or backward, starting with the currently selected packet (or the most recently selected packet, if no packet is selected). Search criteria can be a display filter expression, a string of hexadecimal digits, or a text string.

When searching for a text string, you can search the packet data, or you can search the text in the Info column in the packet list pane or in the packet details pane.

Hexadecimal digits can be separated by colons, periods, or dashes. Text string searches can be ASCII or Unicode (or both), and may be case insensitive.

Edit:Find Next**Edit:Find Previous**

Search forward / backward for a packet matching the filter from the previous search, starting with the currently selected packet (or the most recently selected packet, if no packet is selected).

Edit:Time Reference:Set Time Reference (toggle)

Set (or unset if currently set) the selected packet as a Time Reference packet. When a packet is set as a Time Reference packet, the timestamps in the packet list pane will be replaced with the string ``*REF*`. The relative time timestamp in later packets will then be calculated relative to the timestamp of this Time Reference packet and not the first packet in the capture.

Packets that have been selected as Time Reference packets will always be displayed in the packet list pane. Display filters will not affect or hide these packets.

If there is a column displayed for ``Culmulative Bytes" this counter will be reset at every Time Reference packet.

Edit:Time Reference:Find Next

Edit:Time Reference:Find Previous

Search forward / backward for a time referenced packet.

Edit:Mark Packet (toggle)

Mark (or unmark if currently marked) the selected packet. The field ``frame.marked" is set for packets that are marked, so that, for example, a display filters can be used to display only marked packets, and so that the [Edit:Find Packet](#) dialog can be used to find the next or previous marked packet.

Edit:Mark All Packets

Edit:Unmark All Packets

Mark / Unmark all packets that are currently displayed.

Edit:Preferences

Set the GUI, capture, printing and protocol options (see [Preferences](#) dialog below).

View:Main Toolbar

View:Filter Toolbar

View:Statusbar

Show or hide the main window controls.

View:Packet List

View:Packet Details

View:Packet Bytes

Show or hide the main window panes.

View:Time Display Format

Set the format of the packet timestamp displayed in the packet list window.

View:Name Resolution:Resolve Name

Try to resolve a name for the currently selected item.

View:Name Resolution:Enable for ... Layer

Enable or disable translation of addresses to names in the display.

View:Colorize Packet List

Enable or disable the coloring rules. Disabling will improve performance.

View:Auto Scroll in Live Capture

Enable or disable the automatic scrolling of the packet list while a live capture is in progress.

View:Zoom In

View:Zoom Out

Zoom into / out of the main window data (by changing the font size).

View:Normal Size

Reset the zoom factor of zoom in / zoom out back to normal font size.

View:Resize All Columns

Resize all columns to best fit the current packet display.

View:Expand Subtrees

Expands the currently selected item and it's subtrees in the packet details.

View:Expand All**View:Collapse All**

Expand / Collapse all branches of the packet details.

View:Coloring Rules

Change the foreground and background colors of the packet information in the list of packets, based upon display filters. The list of display filters is applied to each packet sequentially. After the first display filter matches a packet, any additional display filters in the list are ignored. Therefore, if you are filtering on the existence of protocols, you should list the higher-level protocols first, and the lower-level protocols last.

How Colorization Works

Packets are colored according to a list of color filters. Each filter consists of a name, a filter expression and a coloration. A packet is colored according to the first filter that it matches, Color filter expressions use exactly the same syntax as display filter expressions.

When Ethereal starts, the color filters are loaded from:

1. The user's personal color filters file or, if that does not exist,
2. The global color filters file.

If neither of these exist then the packets will not be colored.

View:Show Packet In New Window

Create a new window containing a packet details view and a hex dump window of the currently selected packet; this window will continue to display that packet's details and data even if another packet is selected.

View:Reload

Reload a capture file. Same as *File:Close* and *File:Open* the same file again.

Go:Back

Go back in previously visited packets history.

Go:Forward

Go forward in previously visited packets history.

Go:Go To Packet

Go to a particular numbered packet.

Go:Go To Corresponding Packet

If a field in the packet details pane containing a packet number is selected, go to the packet number specified by that field. (This works only if the dissector that put that entry into the packet details put it into the details as a filterable field rather than just as text.) This can be used, for example, to go to the packet for the request corresponding to a reply, or the reply corresponding to a request, if that packet number has been put into the packet details.

Go:First Packet**Go>Last Packet**

Go to the first / last packet in the capture.

Capture:Interfaces

Shows a dialog box with all currently known interfaces and displaying the current network traffic amount. Capture sessions can be started from here. Beware: keeping this box open results in high system load!

Capture:Options

Initiate a live packet capture (see [Capture Options](#) dialog below). If no filename is specified, a temporary file will be created to hold the capture. The location of the file can be chosen by setting your TMPDIR environment variable before starting **Ethereal**. Otherwise, the default TMPDIR location is system-dependent, but is likely either `/var/tmp` or `/tmp`.

Capture:Start

Start a live packet capture with the previously selected options. This won't open the options dialog box, and can be convenient for repeatedly capturing with the same options.

Capture:Stop

Stop a running live capture.

Capture:Restart

While a live capture is running, stop it and restart with the same options again. This can be convenient to remove irrelevant packets, if no valuable packets were captured so far.

Capture:Capture Filters

Edit the saved list of capture filters, allowing filters to be added, changed, or deleted.

Analyze:Display Filters

Edit the saved list of display filters, allowing filters to be added, changed, or deleted.

Analyze:Apply as Filter

Create a display filter, or add to the display filter strip at the bottom, a display filter based on the data currently highlighted in the packet details, and apply the filter.

If that data is a field that can be tested in a display filter expression, the display filter will test that field; otherwise, the display filter will be based on absolute offset within the packet, and so could be unreliable if the packet contains protocols with variable-length headers, such as a source-routed token-ring packet.

The **Selected** option creates a display filter that tests for a match of the data; the **Not Selected** option creates a display filter that tests for a non-match of the data. The **And Selected**, **Or Selected**, **And Not Selected**, and **Or Not Selected** options add to the end of the display filter in the strip at the bottom an AND or OR operator followed by the new display filter expression.

Analyze:Prepare a Filter

Create a display filter, or add to the display filter strip at the bottom, a display filter based on the data currently highlighted in the packet details, but don't apply the filter.

Analyze:Enabled Protocols

Allow protocol dissection to be enabled or disabled for a specific protocol. Individual protocols can be enabled or disabled by clicking on them in the list or by highlighting them and pressing the space bar. The entire list can be enabled, disabled, or inverted using the buttons below the list.

When a protocol is disabled, dissection in a particular packet stops when that protocol is reached, and Ethereal moves on to the next packet. Any higher-layer protocols that would otherwise have been processed will not be displayed. For example, disabling TCP will prevent the dissection and display of TCP, HTTP, SMTP, Telnet, and any other protocol exclusively dependent on TCP.

The list of protocols can be saved, so that Ethereal will start up with the protocols in that list disabled.

Analyze:Decode As

If you have a packet selected, present a dialog allowing you to change which dissectors are used to decode this packet. The dialog has one panel each for the link layer, network layer and transport layer protocol/port numbers, and will allow each of these to be changed independently. For example, if the selected packet is a TCP packet to port 12345, using this dialog you can instruct Ethereal to decode all packets to or from that TCP port as HTTP packets.

Analyze:User Specified Decodes

Create a new window showing whether any protocol ID to dissector mappings have been changed by the user. This window also allows the user to reset all decodes to their default values.

Analyze:Follow TCP Stream

If you have a TCP packet selected, display the contents of the data stream for the TCP connection to which that packet belongs, as text, in a separate window, and leave the list of packets in a filtered state, with only those packets that are part of that TCP connection being displayed. You can revert to your old view by pressing ENTER in the display filter text box, thereby invoking your old display filter (or resetting it back to no display filter).

The window in which the data stream is displayed lets you select:

- whether to display the entire conversation, or one or the other side of it;
- whether the data being displayed is to be treated as ASCII or EBCDIC text or as raw hex data;

and lets you print what's currently being displayed, using the same print options that are used for the *File:Print Packet* menu item, or save it as text to a file.

Statistics:Summary

Show summary information about the capture, including elapsed time, packet counts, byte counts, and the like. If a display filter is in effect, summary information will be shown about the capture and about the packets currently being displayed.

Statistics:Protocol Hierarchy

Show the number of packets, and the number of bytes in those packets, for each protocol in the trace. It organizes the protocols in the same hierarchy in which they were found in the trace. Besides counting the packets in which the protocol exists, a count is also made for packets in which the protocol is the last protocol in the stack. These last-protocol counts show you how many packets (and the byte count associated with those packets) **ended** in a particular protocol. In the table, they are listed under ``End Packets" and ``End Bytes".

Statistics:IO Graphs

Open a window where up to 5 graphs in different colors can be displayed to indicate number of packets or number of bytes per second for all packets matching the specified filter. By default only one graph will be displayed showing number of packets per second.

The top part of the window contains the graphs and scales for the X and Y axis. If the graph is too long to fit inside the window there is a horizontal scrollbar below the drawing area that can scroll the graphs to the left or the right. The horizontal axis displays the time into the capture and the vertical axis will display the measured quantity at that time.

Below the drawing area and the scrollbar are the controls. On the bottom left there will be five similar sets of controls to control each individual graph such as ``Display:<button>" which button will toggle that individual graph on/off. If <button> is ticked, the graph will be displayed. ``Color:<color>" which is just a button to show which color will be used to draw that graph (color is

only available in Gtk2 version) and finally ``Filter:<filter-text>" which can be used to specify a display filter for that particular graph.

If filter-text is empty then all packets will be used to calculate the quantity for that graph. If filter-text is specified only those packets that match that display filter will be considered in the calculation of quantity.

To the right of the 5 graph controls there are four menus to control global aspects of the draw area and graphs. The ``Unit:" menu is used to control what to measure; ``packets/tick", ``bytes/tick" or ``advanced..."

packets/tick will measure the number of packets matching the (if specified) display filter for the graph in each measurement interval.

bytes/tick will measure the total number of bytes in all packets matching the (if specified) display filter for the graph in each measurement interval.

advanced... see below

``Tick interval:" specifies what measurement intervals to use. The default is 1 second and means that the data will be counted over 1 second intervals.

``Pixels per tick:" specifies how many pixels wide each measurement interval will be in the drawing area. The default is 5 pixels per tick.

``Y-scale:" controls the max value for the y-axis. Default value is ``auto" which means that **Ethereal** will try to adjust the maxvalue automatically.

``advanced..." If Unit:advanced... is selected the window will display two more controls for each of the five graphs. One control will be a menu where the type of calculation can be selected from SUM,COUNT,MAX,MIN,AVG and LOAD, and one control, textbox, where the name of a single display filter field can be specified.

The following restrictions apply to type and field combinations:

SUM: available for all types of integers and will calculate the SUM of all occurrences of this field in the measurement interval. Note that some field can occur multiple times in the same packet and then all instances will be summed up. Example: 'tcp.len' which will count the amount of payload data transferred across TCP in each interval.

COUNT: available for all field types. This will COUNT the number of times certain field occurs in each interval. Note that some fields may occur multiple times in each packet and if that is the case then each instance will be counted independently and COUNT will be greater than the number of packets.

MAX: available for all integer and relative time fields. This will calculate the max seen integer/time value seen for the field during the interval. Example: 'smb.time' which will plot the maximum SMB response time.

MIN: available for all integer and relative time fields. This will calculate the min seen integer/time value seen for the field during the interval. Example: 'smb.time' which will plot the minimum SMB response time.

AVG: available for all integer and relative time fields. This will calculate the average seen integer/time value seen for the field during the interval. Example: 'smb.time' which will plot the average SMB response time.

LOAD: available only for relative time fields (response times).

Example of advanced: Display how NFS response time MAX/MIN/AVG changes over time:

Set first graph to:

```
filter:nfs&&rpc.time
Calc:MAX rpc.time
```

Set second graph to

```
filter:nfs&&rpc.time
Calc:AVG rpc.time
```

Set third graph to

```
filter:nfs&&rpc.time
Calc:MIN rpc.time
```

Example of advanced: Display how the average packet size from host a.b.c.d changes over time.

Set first graph to

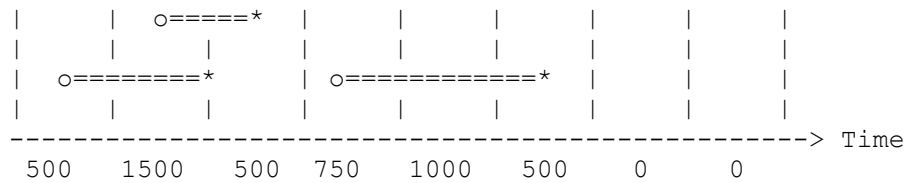
```
filter:ip.addr==a.b.c.d&&frame.pkt_len
Calc:AVG frame.pkt_len
```

LOAD: The LOAD io-stat type is very different from anything you have ever seen before! While the response times themselves as plotted by MIN,MAX,AVG are indications on the Server load (which affects the Server response time), the LOAD measurement measures the Client LOAD. What this measures is how much workload the client generates, i.e. how fast will the client issue new commands when the previous ones completed. i.e. the level of concurrency the client can maintain. The higher the number, the more and faster is the client issuing new commands. When the LOAD goes down, it may be due to client load making the client slower in issuing new commands (there may be other reasons as well, maybe the client just doesn't have any commands it wants to issue right then).

Load is measured in concurrency/number of overlapping i/o and the value 1000 means there is a constant load of one i/o.

In each tick interval the amount of overlap is measured. See the graph below containing three commands: Below the graph are the LOAD values for each interval that would be calculated.

```
|      |      |      |      |      |      |      |      |
|      |      |      |      |      |      |      |      |
```



Statistics:Conversation List

This option will open a new window that displays a list of all conversations between two endpoints. The list has one row for each unique conversation and displays total number of packets/bytes seen as well as number of packets/bytes in each direction.

By default the list is sorted according to the number of packets but by clicking on the column header; it is possible to re-sort the list in ascending or descending order by any column.

By first selecting a conversation by clicking on it and then using the right mouse button (on those platforms that have a right mouse button) ethereal will display a popup menu offering several different filter operations to apply to the capture.

These statistics windows can also be invoked from the Ethereal command line using the **-z conv** argument.

Statistics:Service Response Time:DCE-RPC

Open a window to display Service Response Time statistics for an arbitrary DCE-RPC program interface and display **Procedure**, **Number of Calls**, **Minimum SRT**, **Maximum SRT** and **Average SRT** for all procedures for that program/version. These windows opened will update in semi-real time to reflect changes when doing live captures or when reading new capture files into **Ethereal**.

This dialog will also allow an optional filter string to be used. If an optional filter string is used only such DCE-RPC request/response pairs that match that filter will be used to calculate the statistics. If no filter string is specified all request/response pairs will be used.

Statistics:Service Response Time:Fibre Channel

Open a window to display Service Response Time statistics for Fibre Channel and display **FC Type**, **Number of Calls**, **Minimum SRT**, **Maximum SRT** and **Average SRT** for all FC types. These windows opened will update in semi-real time to reflect changes when doing live captures or when reading new capture files into **Ethereal**. The Service Response Time is calculated as the time delta between the First packet of the exchange and the Last packet of the exchange.

This dialog will also allow an optional filter string to be used. If an optional filter string is used only such FC first/last exchange pairs that match that filter will be used to calculate the statistics. If no filter string is specified all request/response pairs will be used.

Statistics:Service Response Time:ONC-RPC

Open a window to display statistics for an arbitrary ONC-RPC program interface and display **Procedure**, **Number of Calls**, **Minimum SRT**, **Maximum SRT** and **Average SRT** for all procedures for that program/version. These windows opened will update in semi-real time to reflect changes when doing live captures or when reading new capture files into **Ethereal**.

This dialog will also allow an optional filter string to be used. If an optional filter string is used only such ONC-RPC request/response pairs that match that filter will be used to calculate the statistics. If no filter string is specified all request/response pairs will be used.

By first selecting a conversation by clicking on it and then using the right mouse button (on those platforms that have a right mouse button) ethereal will display a popup menu offering several different filter operations to apply to the capture.

Statistics:Service Response Time:SMB

Collect call/reply SRT (Service Response Time) data for SMB. Data collected is number of calls for each SMB command, MinSRT, MaxSRT and AvgSRT.

The data will be presented as separate tables for all normal SMB commands, all Transaction2 commands and all NT Transaction commands. Only those commands that are seen in the capture will have its stats displayed. Only the first command in a xAndX command chain will be used in the calculation. So for common SessionSetupAndX + TreeConnectAndX chains, only the SessionSetupAndX call will be used in the statistics. This is a flaw that might be fixed in the future.

You can apply an optional filter string in a dialog box, before starting the calculation. The stats will only be calculated on those calls matching that filter.

By first selecting a conversation by clicking on it and then using the right mouse button (on those platforms that have a right mouse button) ethereal will display a popup menu offering several different filter operations to apply to the capture.

Statistics:Service Response Time:MGCP

Collect requests/response SRT (Service Response Time) data for MGCP. Data collected is **number of calls** for each known MGCP Type, **Minimum SRT**, **Maximum SRT**, **Average SRT**, **Minimum in Packet**, and **Maximum in Packet**. These windows opened will update in semi-real time to reflect changes when doing live captures or when reading new capture files into **Ethereal**.

You can apply an optional filter string in a dialog box, before starting the calculation. The statistics will only be calculated on those calls matching that filter.

Statistics:Service Response Time:ITU-T H.225 RAS

Collect requests/response SRT (Service Response Time) data for ITU-T H.225 RAS. Data collected is **number of calls** for each known ITU-T H.225 RAS Message Type, **Minimum SRT**, **Maximum SRT**, **Average SRT**, **Minimum in Packet**, and **Maximum in Packet**. You will also get the number of **Open Requests** (Unresponded Requests), **Discarded Responses** (Responses without matching request) and Duplicate Messages. These windows opened will update in semi-real time to reflect changes when doing live captures or when reading new capture files into **Ethereal**.

You can apply an optional filter string in a dialog box, before starting the calculation. The statistics will only be calculated on those calls matching that filter.

Statistics:ITU-T H.225

Count ITU-T H.225 messages and their reasons. In the first column you get a list of H.225 messages and H.225 message reasons, which occur in the current capture file. The number of occurrences of each message or reason will be displayed in the second column. This window opened will update in semi-real time to reflect changes when doing live captures or when reading new capture files into **Ethereal**.

You can apply an optional filter string in a dialog box, before starting the counter. The statistics will only be calculated on those calls matching that filter.

Statistics:SIP

Activate a counter for SIP messages. You will get the number of occurrences of each SIP Method and of each SIP Status-Code. Additionally you also get the number of resent SIP Messages (only for SIP over UDP).

This window opened will update in semi-real time to reflect changes when doing live captures or when reading new capture files into **Ethereal**.

You can apply an optional filter string in a dialog box, before starting the counter. The statistics will only be calculated on those calls matching that filter.

Statistics:ONC-RPC Programs

This dialog will open a window showing aggregated RTT statistics for all ONC-RPC Programs/versions that exist in the capture file.

Help:Contents

Some help texts.

Help:Supported Protocols

List of supported protocols and display filter protocol fields.

Help:Manual Pages

Display locally installed HTML versions of these manual pages in a web browser.

Help:Ethereal Online

Various links to online resources to be open in a web browser, like <http://www.ethereal.com>.

Help>About Ethereal

See various information about Ethereal (see [About](#) dialog below), like the version, the folders used, the available plugins, ...

WINDOWS

Main Window

The main window contains the usual things like the menu, some toolbars, the main area and a statusbar. The main area is split into three panes, you can resize each pane using a "thumb" at the right end of each divider line.

The main window is much more flexible than before. The layout of the main window can be customized by the *Layout* page in the dialog box popped up by *Edit:Preferences*, the following will describe the layout with the default settings.

Main Toolbar

Some menu items are available for quick access here. There is no way to customize the items in the toolbar, however the toolbar can be hidden by *View:Main Toolbar*.

Filter Toolbar

A display filter can be entered into the filter toolbar. A filter for HTTP, HTTPS, and DNS traffic might look like this:

```
tcp.port == 80 || tcp.port == 443 || tcp.port == 53
```

Selecting the *Filter:* button lets you choose from a list of named filters that you can optionally save. Pressing the Return or Enter keys, or selecting the *Apply* button, will cause the filter to be applied to

the current list of packets. Selecting the *Reset* button clears the display filter so that all packets are displayed (again).

There is no way to customize the items in the toolbar, however the toolbar can be hidden by *View:Filter Toolbar*.

Packet List Pane

The top pane contains the list of network packets that you can scroll through and select. By default, the packet number, packet timestamp, source and destination addresses, protocol, and description are displayed for each packet; the *Columns* page in the dialog box popped up by *Edit:Preferences* lets you change this (although, unfortunately, you currently have to save the preferences, and exit and restart Ethereal, for those changes to take effect).

If you click on the heading for a column, the display will be sorted by that column; clicking on the heading again will reverse the sort order for that column.

An effort is made to display information as high up the protocol stack as possible, e.g. IP addresses are displayed for IP packets, but the MAC layer address is displayed for unknown packet types.

The right mouse button can be used to pop up a menu of operations.

The middle mouse button can be used to mark a packet.

Packet Details Pane

The middle pane contains a display of the details of the currently-selected packet. The display shows each field and its value in each protocol header in the stack. The right mouse button can be used to pop up a menu of operations.

Packet Bytes Pane

The lowest pane contains a hex and ASCII dump of the actual packet data. Selecting a field in the packet details highlights the corresponding bytes in this section.

The right mouse button can be used to pop up a menu of operations.

Statusbar

The statusbar is divided into two parts, on the left some context dependant things are shown, like information about the loaded file, on the right the number of packets are displayed: P = Packets captured/loaded, D = Displayed in packet list (after filtering), M = Marked by user.

The statusbar can be hidden by *View:Statusbar*.

Preferences

The *Preferences* dialog lets you control various personal preferences for the behavior of **Ethereal**.

User Interface Preferences

The *User Interface* page is used to modify small aspects of the GUI to your own personal taste:

Scrollbars

The vertical scrollbars in the three panes can be set to be either on the left or the right.

Selection Bars

The selection bar in the packet list and packet details can have either a ``browse" or ``select" behavior. If the selection bar has a ``browse" behavior, the arrow keys will move an outline of the selection bar, allowing you to browse the rest of the list or details without changing the selection

until you press the space bar. If the selection bar has a "select" behavior, the arrow keys will move the selection bar and change the selection to the new item in the packet list or packet details.

Tree Line Style

Trees can be drawn with no lines, solid lines, or dotted lines between items, or can be drawn with "tab" headings.

Tree Expander Style

The expander item that can be clicked to show or hide items under a tree item can be omitted (note that this will prevent you from changing whether those items are shown or hidden!), or can be drawn as squares, triangles, or circles.

Hex Display

The highlight method in the hex dump display for the selected protocol item can be set to use either inverse video, or bold characters.

Save Window Position

If this item is selected, the position of the main Ethereal window will be saved when Ethereal exits, and used when Ethereal is started again.

Save Window Size

If this item is selected, the size of the main Ethereal window will be saved when Ethereal exits, and used when Ethereal is started again.

File Open Dialog Behavior

This item allows the user to select how Ethereal handles the listing of the "File Open" Dialog when opening trace files. "Remember Last Directory" causes Ethereal to automatically position the dialog in the directory of the most recently opened file, even between launches of Ethereal. "Always Open in Directory" allows the user to define a persistent directory that the dialog will always default to.

Directory

Allows the user to specify a persistent File Open directory. Trailing slashes or backslashes will automatically be added.

Layout Preferences

The *Layout* page lets you specify the general layout of the main window. You can choose from six different layouts and fill the three panes with the contents you like.

Column Preferences

The *Columns* page lets you specify the number, title, and format of each column in the packet list.

The *Column title* entry is used to specify the title of the column displayed at the top of the packet list. The type of data that the column displays can be specified using the *Column format* option menu. The row of buttons on the left perform the following actions:

New

Adds a new column to the list.

Delete

Deletes the currently selected list item.

Up / Down

Moves the selected list item up or down one position.

Font Preferences

The *Font* page lets you select the font to be used for most text.

Color Preferences

The *Colors* page can be used to change the color of the text displayed in the TCP stream window and for marked packets. To change a color, simply select an attribute from the "Set:" menu and use the color selector to get the desired color. The new text colors are displayed as a sample text.

Capture Preferences

The *Capture* page lets you specify various parameters for capturing live packet data; these are used the first time a capture is started.

The *Interface*: combo box lets you specify the interface from which to capture packet data, or the name of a FIFO from which to get the packet data.

The *Data link type*: option menu lets you, for some interfaces, select the data link header you want to see on the packets you capture. For example, in some OSes and with some versions of libpcap, you can choose, on an 802.11 interface, whether the packets should appear as Ethernet packets (with a fake Ethernet header) or as 802.11 packets.

The *Limit each packet to ... bytes* check box lets you set the snapshot length to use when capturing live data; turn on the check box, and then set the number of bytes to use as the snapshot length.

The *Filter*: text entry lets you set a capture filter expression to be used when capturing.

If any of the environment variables SSH_CONNECTION, SSH_CLIENT, REMOTEHOST, DISPLAY, or CLIENTNAME are set, Ethereal will create a default capture filter that excludes traffic from the hosts and ports defined in those variables.

The *Capture packets in promiscuous mode* check box lets you specify whether to put the interface in promiscuous mode when capturing.

The *Update list of packets in real time* check box lets you specify that the display should be updated as packets are seen.

The *Automatic scrolling in live capture* check box lets you specify whether, in an "Update list of packets in real time" capture, the packet list pane should automatically scroll to show the most recently captured packets.

Printing Preferences

The radio buttons at the top of the *Printing* page allow you choose between printing packets with the *File:Print Packet* menu item as text or PostScript, and sending the output directly to a command or saving it to a file. The *Command*: text entry box, on UNIX-compatible systems, is the command to send files to (usually **lpr**), and the *File*: entry box lets you enter the name of the file you wish to save to. Additionally, you can select the *File*: button to browse the file system for a particular save file.

Protocol Preferences

There are also pages for various protocols that Ethereal dissects, controlling the way Ethereal handles those protocols.

Edit Capture Filter List

Edit Display Filter List

Capture Filter

Display Filter

Read Filter

Search Filter

The *Edit Capture Filter List* dialog lets you create, modify, and delete capture filters, and the *Edit Display Filter List* dialog lets you create, modify, and delete display filters.

The *Capture Filter* dialog lets you do all of the editing operations listed, and also lets you choose or construct a filter to be used when capturing packets.

The *Display Filter* dialog lets you do all of the editing operations listed, and also lets you choose or construct a filter to be used to filter the current capture being viewed.

The *Read Filter* dialog lets you do all of the editing operations listed, and also lets you choose or construct a filter to be used to as a read filter for a capture file you open.

The *Search Filter* dialog lets you do all of the editing operations listed, and also lets you choose or construct a filter expression to be used in a find operation.

In all of those dialogs, the *Filter name* entry specifies a descriptive name for a filter, e.g. **Web and DNS traffic**. The *Filter string* entry is the text that actually describes the filtering action to take, as described above. The dialog buttons perform the following actions:

New

If there is text in the two entry boxes, creates a new associated list item.

Edit

Modifies the currently selected list item to match what's in the entry boxes.

Delete

Deletes the currently selected list item.

Add Expression...

For display filter expressions, pops up a dialog box to allow you to construct a filter expression to test a particular field; it offers lists of field names, and, when appropriate, lists from which to select tests to perform on the field and values with which to compare it. In that dialog box, the OK button will cause the filter expression you constructed to be entered into the *Filter string* entry at the current cursor position.

OK

In the *Capture Filter* dialog, closes the dialog box and makes the filter in the *Filter string* entry the filter in the *Capture Preferences* dialog. In the *Display Filter* dialog, closes the dialog box and makes the filter in the *Filter string* entry the current display filter, and applies it to the current capture. In the *Read Filter* dialog, closes the dialog box and makes the filter in the *Filter string* entry the filter in the *Open Capture File* dialog. In the *Search Filter* dialog, closes the dialog box and makes the filter in the *Filter string* entry the filter in the *Find Packet* dialog.

Apply

Makes the filter in the *Filter string* entry the current display filter, and applies it to the current capture.

Save

If the list of filters being edited is the list of capture filters, saves the current filter list to the personal capture filters file, and if the list of filters being edited is the list of display filters, saves the current filter list to the personal display filters file.

Close

Closes the dialog without doing anything with the filter in the *Filter string* entry.

The Color Filters Dialog

This dialog displays a list of color filters and allows it to be modified.

THE FILTER LIST

Single rows may be selected by clicking. Multiple rows may be selected by using the ctrl and shift keys in combination with the mouse button.

NEW

Adds a new filter at the bottom of the list and opens the Edit Color Filter dialog box. You will have to alter the filter expression at least before the filter will be accepted. The format of color filter expressions is identical to that of display filters. The new filter is selected, so it may immediately be

moved up and down, deleted or edited. To avoid confusion all filters are unselected before the new filter is created.

EDIT

Opens the Edit Color Filter dialog box for the selected filter. (If this button is disabled you may have more than one filter selected, making it ambiguous which is to be edited.)

DELETE

Deletes the selected color filter(s).

EXPORT

Allows you to choose a file in which to save the current list of color filters. You may also choose to save only the selected filters. A button is provided to save the filters in the global color filters file (you must have sufficient permissions to write this file, of course).

IMPORT

Allows you to choose a file containing color filters which are then added to the bottom of the current list. All the added filters are selected, so they may be moved to the correct position in the list as a group. To avoid confusion, all filters are unselected before the new filters are imported. A button is provided to load the filters from the global color filters file.

CLEAR

Deletes your personal color filters file, reloads the global color filters file, if any, and closes the dialog.

UP

Moves the selected `filter(s)` up the list, making it more likely that they will be used to color packets.

DOWN

Moves the selected `filter(s)` down the list, making it less likely that they will be used to color packets.

OK

Closes the dialog and uses the color filters as they stand.

APPLY

Colors the packets according to the current list of color filters, but does not close the dialog.

SAVE

Saves the current list of color filters in your personal color filters file. Unless you do this they will not be used the next time you start Ethereal.

CLOSE

Closes the dialog without changing the coloration of the packets. Note that changes you have made to the current list of color filters are not undone.

Capture Options

The *Capture Options* dialog lets you specify various parameters for capturing live packet data.

The *Interface:* field lets you specify the interface from which to capture packet data or a command from which to get the packet data via a pipe.

The *Link layer header type:* field lets you specify the interfaces link layer header type. This field is usually disabled, as most interface have only one header type.

The *Capture packets in promiscuous mode* check box lets you specify whether the interface should be put into promiscuous mode when capturing.

The *Limit each packet to ... bytes* check box and field lets you specify a maximum number of bytes per packet to capture and save; if the check box is not checked, the limit will be 65535 bytes.

The *Capture Filter:* entry lets you specify the capture filter using a tcpdump-style filter string as described above.

The *File:* entry lets you specify the file into which captured packets should be saved, as in the *Printer Options* dialog above. If not specified, the captured packets will be saved in a temporary file; you can save those packets to a file with the *File:Save As* menu item.

The *Use multiple files* check box lets you specify that the capture should be done in ``multiple files" mode. This option is disabled, if the *Update list of packets in real time* option is checked.

The *Next file every ... megabyte(s)* check box and fields lets you specify that a switch to a next file should be done if the specified filesize is reached. You can also select the appropriate unit, but beware that the filesize has a maximum of 2 GB. The check box is forced to be checked, as ``multiple files" mode requires a file size to be specified.

The *Next file every ... minute(s)* check box and fields lets you specify that the switch to a next file should be done after the specified time has elapsed, even if the specified capture size is not reached.

The *Ring buffer with ... files* field lets you specify the number of files of a ring buffer. This feature will capture into to the first file again, after the specified amount of files were used.

The *Stop capture after ... files* field lets you specify the number of capture files used, until the capture is stopped.

The *Stop capture after ... packet(s)* check box and field let you specify that Ethereal should stop capturing after having captured some number of packets; if the check box is not checked, Ethereal will not stop capturing at some fixed number of captured packets.

The *Stop capture after ... megabyte(s)* check box and field lets you specify that Ethereal should stop capturing after the file to which captured packets are being saved grows as large as or larger than some specified number of megabytes. If the check box is not checked, Ethereal will not stop capturing at some capture file size (although the operating system on which Ethereal is running, or the available disk space, may still limit the maximum size of a capture file). This option is disabled, if ``multiple files" mode is used,

The *Stop capture after ... second(s)* check box and field let you specify that Ethereal should stop capturing after it has been capturing for some number of seconds; if the check box is not checked, Ethereal will not stop capturing after some fixed time has elapsed.

The *Update list of packets in real time* check box lets you specify whether the display should be updated as packets are captured and, if you specify that, the *Automatic scrolling in live capture* check box lets you specify the packet list pane should automatically scroll to show the most recently captured packets as new packets arrive.

The *Enable MAC name resolution*, *Enable network name resolution* and *Enable transport name resolution* check boxes let you specify whether MAC addresses, network addresses, and transport-layer port numbers should be translated to names.

About

The *About* dialog lets you view various information about Ethereal.

About:Ethereal

The *Ethereal* page lets you view general information about Ethereal, like the installed version, licensing information and such.

About:Authors

The *Authors* page shows the author and all contributors.

About:Folders

The *Folders* page lets you view the directory names where Ethereal is searching it's various configuration and other files.

About:Plugins

The *Plugins* page lets you view the dissector plugin modules available on your system.

The *Plugins List* shows the name and version of each dissector plugin module found on your system. The plugins are searched in the following directories: the *lib/ethereal/plugins/\$VERSION* directory under the main installation directory (for example, */usr/local/lib/ethereal/plugins/\$VERSION*), */usr/lib/ethereal/plugins/\$VERSION*, */usr/local/lib/ethereal/plugins/\$VERSION*, and *\$HOME/.ethereal/plugins* on UNIX-compatible systems, and in the *plugins\VERSION* directory under the main installation directory (for example, *C:\Program Files\Ethereal\plugins\VERSION*) and *%APPDATA%\Ethereal\plugins\VERSION* (or, if *%APPDATA%* isn't defined, *%USERPROFILE%\Application Data\Ethereal\plugins\VERSION*) on Windows systems; *\$VERSION* is the version number of the plugin interface, which is typically the version number of Ethereal. Note that a dissector plugin module may support more than one protocol; there is not necessarily a one-to-one correspondence between dissector plugin modules and protocols. Protocols supported by a dissector plugin module are enabled and disabled using the *Edit:Protocols* dialog box, just as protocols built into Ethereal are.

FILES

These files contains various **Ethereal** configuration values.

Preferences

The *preferences* files contain global (system-wide) and personal preference settings. If the system-wide preference file exists, it is read first, overriding the default values. If the personal preferences file exists, it is read then, overriding these values (again). Note: If the command line flag **-o** is used, it will override these values even once more.

The preferences settings are in the form *prefname:value*, one per line, where *prefname* is the name of the preference (which is the same name that would appear in the preference file), and *value* is the value to which it should be set; white space is allowed between **:** and *value*. A preference setting can be continued on subsequent lines by indenting the continuation lines with white space. A **#** character starts a comment that runs to the end of the line.

The global preferences file is searched in the *ethereal* directory under the *share* subdirectory of the main installation directory (for example, */usr/local/share/ethereal/preferences*) on UNIX-compatible systems, and in the main installation directory (for example, *C:\Program Files\Ethereal\preferences*) on Windows systems.

The personal preferences file, is searched in *\$HOME/.ethereal/preferences* on UNIX-compatible systems and *%APPDATA%\Ethereal\preferences* (or, if *%APPDATA%* isn't defined, *%USERPROFILE%\Application Data\Ethereal\preferences*) on Windows systems.

Note: Whenever the preferences are saved by using the *Save* button in the *Edit:Preferences* dialog box, your personal preferences file will be overwritten with the new settings, destroying any comments that were in the file.

Recent

The *recent* file will store personal settings (mostly GUI related) like the current **Ethereal** window size. The file is saved at program exit and read in at program start automatically (comments in this file are therefore automatically destroyed). Note: If the command line flag **-o** is used, it will override these values.

The settings in this file have the same format as in the *Preferences* files, and the same directory as for the personal preferences file is used.

Disabled (Enabled) Protocols

The *disabled_protos* file contains a list of protocols that have been disabled, so that their dissectors are never called. The file contains protocol names, one per line, where the protocol name is the same name that would be used in a display filter for the protocol. A **#** character starts a comment that runs to the end of the line. The same directory as for the personal preferences file is used.

Note: Whenever the disabled protocols list is saved by using the *Save* button in the *Analyze:Enabled Protocols* dialog box, your disabled protocols file will be overwritten with the new settings, destroying any comments that were in the file.

Name Resolution (hosts)

If the personal *hosts* file exists, the entries in that file are used to resolve IPv4 and IPv6 addresses before any other attempts are made to resolve them. That file has the standard *hosts* file syntax; each line contains one IP address and name, separated by whitespace. The same directory as for the personal preferences file is used.

Name Resolution (ethers)

The *ethers* files, are consulted to correlate 6-byte hardware addresses to names. First the global *ethers* file is tried and if that address is not found there the personal one is tried next.

Each line contains one hardware address and name, separated by whitespace. The digits of the hardware address are separated by either a colon (:), a dash (-), or a period (.). The following three lines are valid lines of an *ethers* file:

ff:ff:ff:ff:ff:ff	Broadcast
c0-00-ff-ff-ff-ff	TR_broadcast
00.00.00.00.00.00	Zero_broadcast

The global *ethers* file is searched in the */etc* directory on UNIX-compatible systems, and in the main installation directory (for example, *C:\Program Files\Ethereal*) on Windows systems.

The personal *ethers* file is searched in the same directory as the personal preferences file.

Name Resolution (manuf)

The *manuf* file is used to match the 3-byte vendor portion of a 6-byte hardware address with the manufacturer's name; it can also contain well-known MAC addresses and address ranges specified with a netmask. The format of the file is the same as the *ethers* file, except that entries of the form:

00:00:0C	Cisco
----------	-------

can be provided, with the 3-byte OUI and the name for a vendor, and entries of the form:

```
00-00-0C-07-AC/40      All-HSRP-routers
```

can be specified, with a MAC address and a mask indicating how many bits of the address must match. Trailing zero bytes can be omitted from address ranges. That entry, for example, will match addresses from 00-00-0C-07-AC-00 through 00-00-0C-07-AC-FF. The mask need not be a multiple of 8.

The *manuf* file is installed in the *etc* directory under the main installation directory (for example, */usr/local/etc/manuf*) on UNIX-compatible systems, and in the main installation directory (for example, *C:\Program Files\Ethereal\manuf*) on Windows systems.

Name Resolution (ipxnets)

The *ipxnets* files are used to correlate 4-byte IPX network numbers to names. First the global *ipxnets* file is tried and if that address is not found there the personal one is tried next.

The format is the same as the *ethers* file, except that each address is four bytes instead of six. Additionally, the address can be represented as a single hexadecimal number, as is more common in the IPX world, rather than four hex octets. For example, these four lines are valid lines of an *ipxnets* file:

```
C0.A8.2C.00      HR
c0-a8-1c-00      CEO
00:00:BE:EF      IT_Server1
110f             FileServer3
```

The global *ipxnets* file is found in the */etc* directory on UNIX-compatible systems, and in the main installation directory (for example, *C:\Program Files\Ethereal*) on Windows systems.

The personal *ipxnets* file is searched in the same directory as the personal preferences file.

Capture Filters

The *cfilters* file, contains personal capture filters.

The personal *cfilters* file uses the same directory as the personal preferences file.

Display Filters

The *dfilters* file, contains personal display filters.

The personal *dfilters* file uses the same directory as the personal preferences file.

Color Filters (Coloring Rules)

The *colorfilters* files contain system-wide and personal color filters.

The global *colorfilters* file is installed in the *ethereal* directory under the *share* subdirectory of the main installation directory (for example, */usr/local/share/ethereal*) on UNIX-compatible systems, and in the main installation directory (for example, *C:\Program Files\Ethereal*) on Windows systems,

The personal *colorfilters* file uses the same directory as the personal preferences file.

The following is taken from <http://wiki.ethereal.com/CaptureFilters>

An overview of the capture filter syntax can be found in the Etherreal's User's Guide.

Ethereal uses the same syntax for capture filters as tcpdump, WinDump, Analyzer, and any other program that uses the libpcap/WinPcap library.

Examples

Capture only traffic to or from IP address 172.18.5.4:

- host 172.18.5.4

Capture only DNS (port 53) traffic:

- port 53

Capture non-HTTP and non-SMTP traffic on your server (both are equivalent):

- host www.example.com and not (port 80 or port 25)
- host www.example.com and not port 80 and not port 25

Capture except all ARP and DNS traffic:

- port not 53 and not arp

Capture only Ethernet type EAPOL:

- ether proto 0x888e

Capture only IP traffic - the shortest filter, but sometimes very useful to get rid of lower layer protocols like ARP and STP:

- ip

Capture only unicast traffic - useful to get rid of noise on the network if you only want to see traffic to and from your machine, not, for example, broadcast and multicast announcements:

- not broadcast and not multicast



ClamAV: ClamAV Anti Virus Scanner.

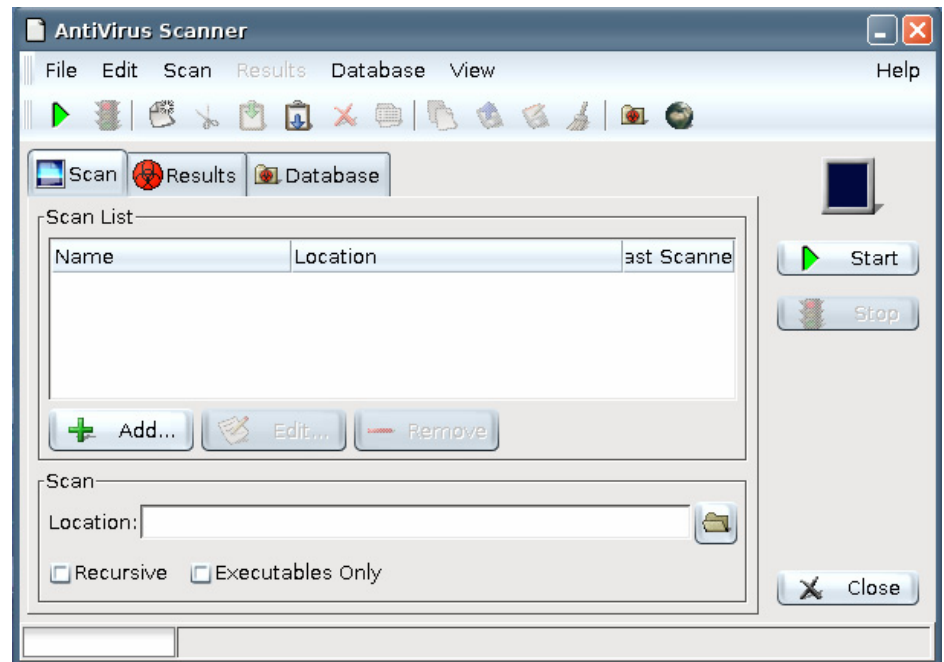


Developed by Tomasz Kojm. Available from <http://clamav.net/>

Clam AntiVirus is an anti-virus toolkit for UNIX, designed for e-mail scanning on mail gateways. It provides a flexible and scalable multi-threaded daemon, a command line scanner, and an advanced tool for automatic database updating via Internet. The package also includes a virus scanner shared library.

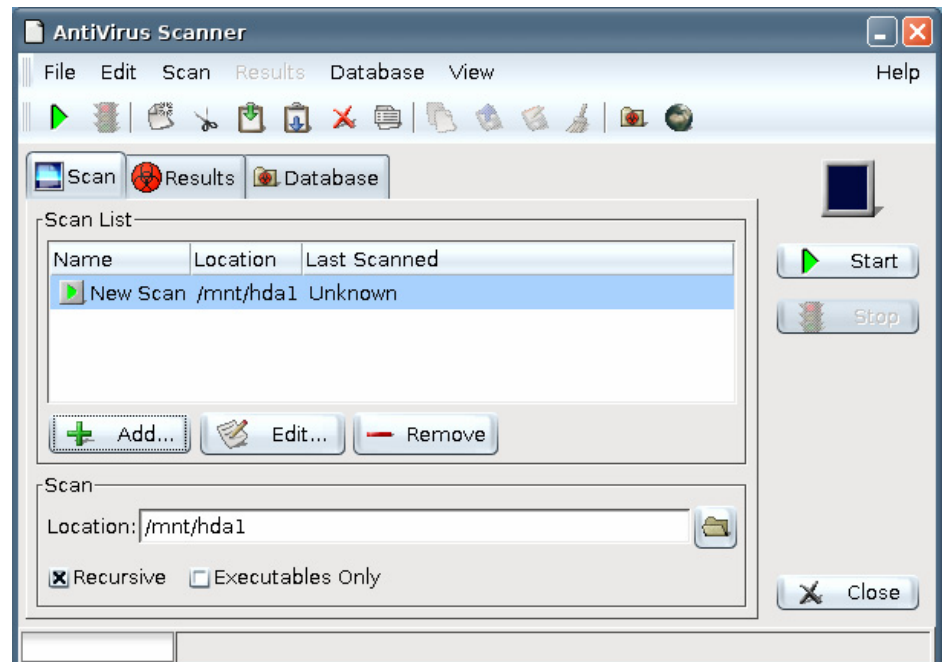
When the program starts, it displays a graphical user interface. The user needs to select where they want the program to scan.

Clicking on the “Add...” button will allow the user to select the directory to scan. Multiple paths can be added by repeatedly selecting the “Add...” button.

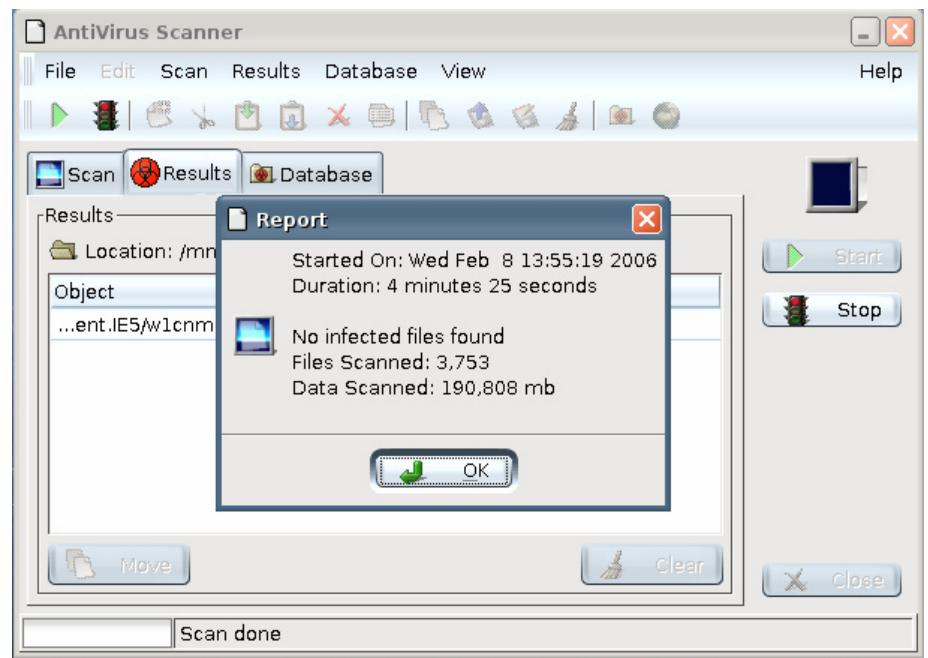


Here the user has added the target system mounted as /mnt/hda1 to be scanned.

Before beginning the scan, it would be a good idea, if the system is connected to the Internet, to update the anti-virus signature. Select Database / Update from Internet.



Once the database is updated, you can select “Start” to start the scan. Depending on the number of files and the speed of the system, this can take quite some time. When the program is finished, it will display the statistics of what it has found.





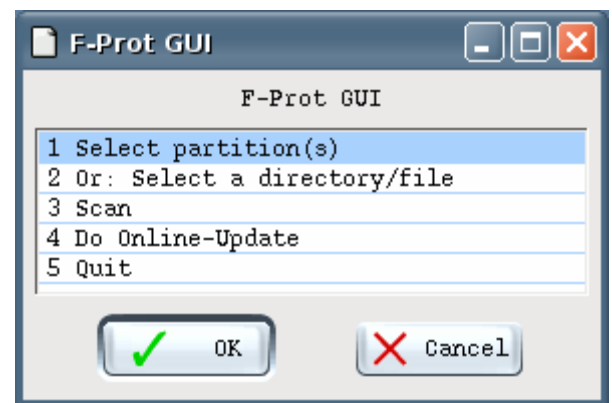
Developed by FRISK Software International. Available from http://www.f-prot.com/products/home_use/linux/

For home users using the Linux open-source operating system, we offer F-Prot Antivirus for Linux Workstations. F-Prot Antivirus for Linux Workstations utilizes the renowned F-Prot Antivirus scanning engine for primary scan but has in addition to that a system of internal heuristics devised to search for unknown viruses (Frisk Software International, 2006).

F-Prot Antivirus for Linux was especially developed to effectively eradicate viruses threatening workstations running Linux. It provides full protection against macro viruses and other forms of malicious software - including Trojans. F-Prot Antivirus can detect a total of 232593 worms, viruses and other malicious programs (Frisk Software International, 2006).

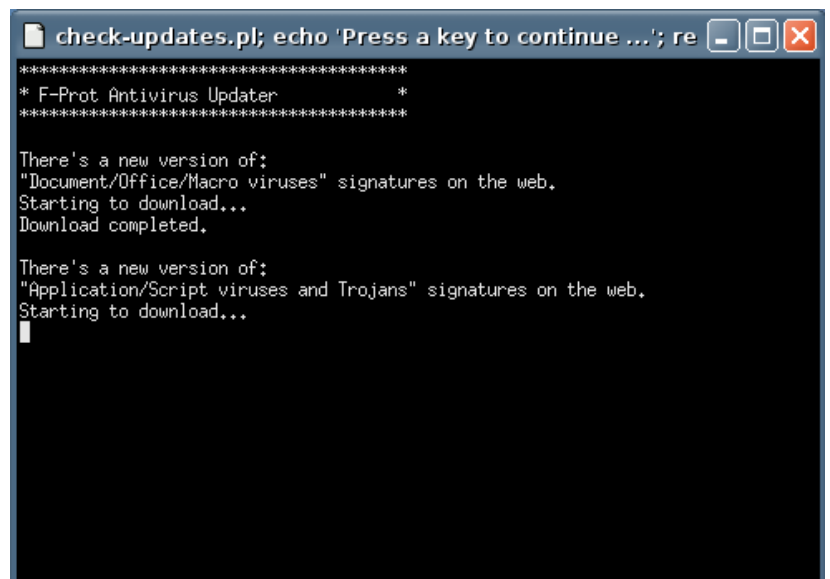
A command line version of this program is also available.

When the program is started, the user is presented with a menu. The user may choose to scan an entire partition, or a file or directory. However, if the system Helix is running on is connected to the Internet, it is recommended that the user first performs choice 4 – Do Online-Update.

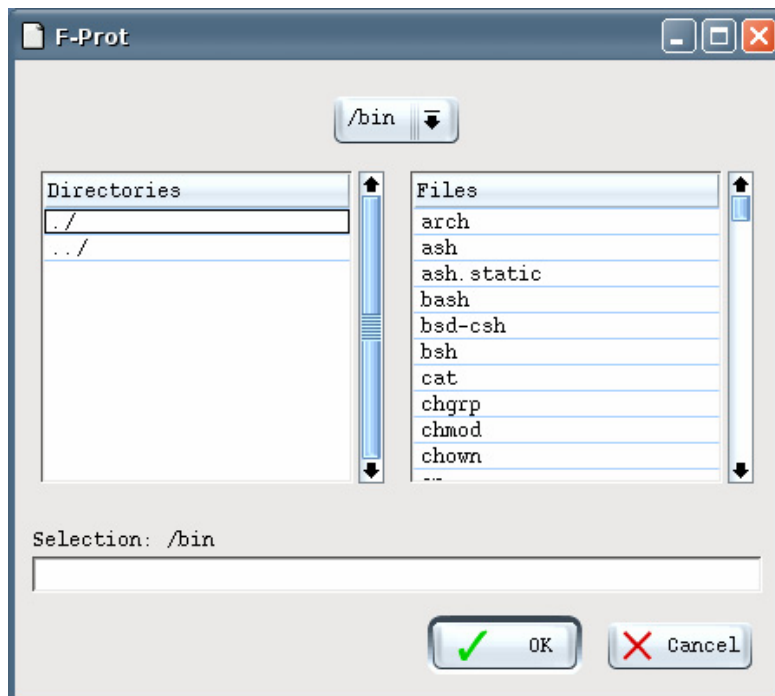


During an online update, the F-Prot program will access the F-Prot server and see if there is a more current version of the anti-virus signatures available. If there is, they will be downloaded.

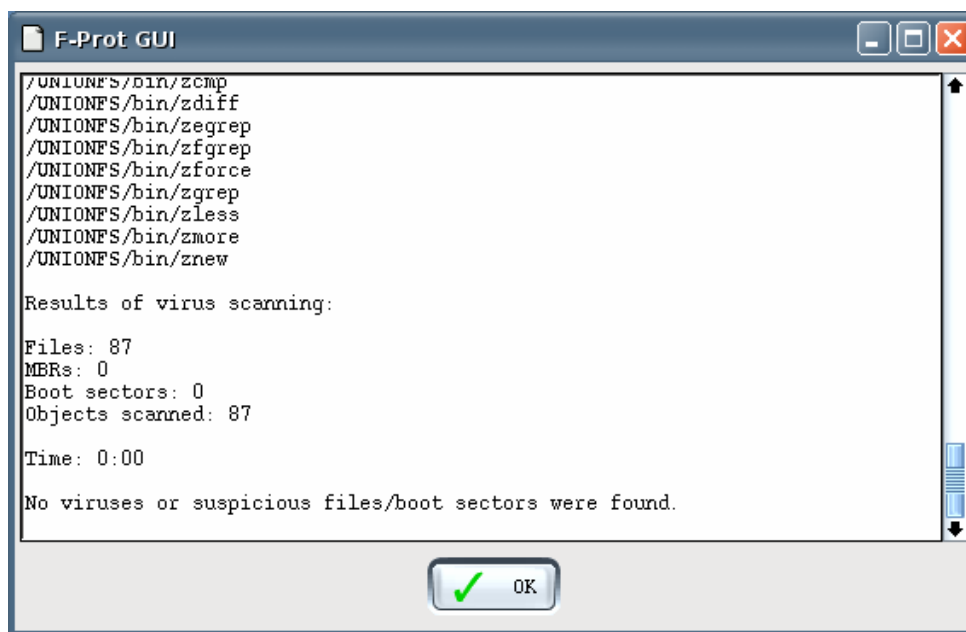
Taking this step will ensure that the program will be able to detect the latest malware hiding on the system.



After the signatures have been updated, you can choose either a partition, directory, or file. In this case, we are going to check the /bin directory on the Helix CD.



F-Prot will examine all the files in the selected directory and identify any suspicious files.





pyFlag



This document covers the basic operations of the pyFlag utility. This document can be found on the menu, or in the /usr/local/RTFM directory.

For more information on pyFlag, go to <http://pyflag.sourceforge.net>

pyFLAG

Basic Concepts

The following section describes a number of basic concepts in PyFlag. We cover the concept of Cases, IO Sources, and the Table Widget. All these are central to the understanding of how PyFlag works.

Flag Cases

A central concept to flag is the case. A case is simply an area to collect related information regarding a particular incident. Internally a case is kept in its own database, and tables are added to the case as different forms of evidence are added.

To create a new case, click the Case Management tab and add a new case.

Resetting the case deletes all data from the case, which is essentially equivalent to dropping the case database and recreating it.

IO Sources



RAID-Reassembly

This document, written by Michael Cohen, deals with the issues involved with forensic analysis of RAID drives. This document can be found on the menu, or in the /usr/local/RTFM directory. This document is also available online from

<http://pyflag.sourceforge.net/Documentation/articles/raid/reconstruction.html>

Abstract

When Forensic examiners and Incident response teams approach a computer system, quite often RAID Drives are involved. RAID arrays are now commonly found on many computers - from the expensive enterprise server machines, all the way to desktop machines. Imaging these machines is often challenging since reconstituting RAID logical images in the lab may be difficult without the identical controller used to create the array, and an identical configuration. Sometimes the controller may not accept the disks as members of the array if the array header is damaged or overwritten, making it impossible to reconstruct the logical image using conventional means.

This paper covers the manual reconstruction of RAID sets. A method is presented to manually recover the RAID reconstruction map, and tools are presented to use this map for reassembling the original RAID set or simply use the entire set as evidence without reassembly.

Introduction

RAID disks have become popular in recent years even in low end systems. When confronted with a RAID system, the investigator is often confronted with a difficult choice.

One reliable way for imaging this system is to attempt booting the system from a CDROM into a forensic platform such as KNOPPIX or Helix for example. These platforms have drivers for many RAID controllers which would often allow the array to be seen as a single logical disk. Then the investigator would image the device over the network, USB or FireWire to another machine.

The above method is very reliable and certainly should be used as the first port of call. However, often this method fails:

- Often the drivers present in the forensic operating system do not support the RAID controller. If the controller does not have native Linux support this might be a problem.
- The RAID is done in software using a proprietary product. There are no Linux drivers that are capable of reading such an array.
- The headers on the disks are damaged or the disks are marked as bad, leading the array controller to refuse to use these disks, despite the fact that the disks themselves might be readable.
- It is impossible to obtain the original controller and BIOS configuration. This might happen if the controller has been destroyed or is simply unavailable.

In these cases the RAID will have to be reconstructed by hand. This paper will detail a method to allow this process to be done reliably.



Partition-Info

This document lists the partition IDs for many different partition types. This can be very useful when trying to mount unknown drives. This document can be found on the menu, or in the /usr/local/RTFM directory.

The following is from http://www.win.tue.nl/%7Eaeb/partitions/partition_types-1.html

List of partition identifiers for PCs

Below a list of the known partition IDs (system indicators) of the various operating systems, file systems, boot managers, etc. For the various systems, short descriptions are given, in the cases where I have some info. There seem to be two other major such lists: Ralf Brown's (see interrupt list under Int 19) and Hale Landis' but the present one is more correct and more complete. (However, these two URLs are a valuable source for other information.) See also the Powerquest table and the specification for DOS-type partition tables.

Copyright (C) Andries E. Brouwer 1995-2004. Link to this list - do not copy it. It is being updated regularly. Additions, corrections, explanations are welcome. (Mail to aeb@cw.nl.)

ID Name

00 Empty

To be precise: this is not used to designate unused area on the disk, but marks an unused partition table entry. (All other fields should be zero as well.) Unused area is not designated. Plan9 assumes that it can use everything not claimed for other systems in the partition table.

01 DOS 12-bit FAT

DOS is a family of single-user operating systems for PCs. 86-DOS ('QDOS' - Quick and Dirty OS) was a CP/M-like operating system written by Tim Paterson of Seattle Computer Products (1979). Microsoft bought it, renamed it to MS-DOS 1.0 and sold it to IBM (1980) to be delivered together with the first IBM PCs (1981). MS-DOS 2.0 (1983) was rather different, and designed to be somewhat Unix-like. It supported a hard disk (up to 16MB; up to 32MB for version 2.1). Version 3.3+ added the concept of partitions, where each partition is at most 32MB. (Compaq DOS 3.31 relaxed this restriction.) Since version 4.0 partitions can be 512 MB. Version 5.0 supports partitions up to 2 GB. Several clones exist: DR-DOS (from Digital Research, later part of Novell and called NovellDOS or NDOS, then owned by Caldera and called OpenDOS, then by its subsidiary Lineo who named it back to DR-DOS. See <http://www.drDOS.com/>), PC-DOS (from IBM), FreeDOS, ... See Types of DOS. See `comp.os.msDOS.*` and MSDOS partitioning summary. The type 01 is for partitions up to 15 MB.

This document is over 27 pages long, and covers partition types from 00 to FF. The full document is available online or on the Helix CD.



Sleuthkit-Informer Articles

The following is taken from <http://www.sleuthkit.org/informer/>

The Sleuth Kit Informer is a bi-monthly newsletter for The Sleuth Kit, Autopsy, and related tools. The goal of the newsletter is to increase awareness, knowledge, and documentation for these tools. The planned topics range from tool design details to techniques on breaking a disk image into partition images.

To subscribe to the e-mail newsletter, go to <http://lists.sourceforge.net/lists/listinfo/sleuthkit-informer>. Starting in 2004, new issues are released on the 15th of odd months (January, March, May etc.).



Table of Contents

Issue #1 - February 15, 2003

- A High-Level Design Overview of Autopsy and TASK
- Placing HTML in Jail

Issue #2 - March 15, 2003

- Autopsy 1.70 Case Management
- Splitting The Disk - Part 1

Issue #3 - April 15, 2003

- Did You Know? - Autopsy Date Stamps
- Sorting Out The Sorter (Part 1 in a series of 3)

Issue #4 - May 15, 2003

- Did You Know? - Group-based File Recovery
- Creating Custom sorter Rule Sets (Part 2 in a series of 3)

Issue #5 - June 15, 2003

- Did You Know? - Importing timelines into spread sheets
- Sorter Internals (Part 3 in a series of 3)

Issue #6 - July 15, 2003

- Hunting for Hashes (Part 1 in a series of 2)

Issue #7 - August 15, 2003

- Did You Know? - Reducing the data in timelines
- NSRL Correction
- Finding Hashes with 'hfind' (Part 2 in a series of 2)

Issue #8 - September 15, 2003

- Did You Know? - New Command Logging
- Locking In On Keywords

Issue #9 - October 15, 2003

- No major article (On vacation because of the Honeynet Challenge grading)

Issue #10 - November 16, 2003

- UNIX Incident Verification with The Sleuth Kit

Issue #11 - December 15, 2003

- 'dd' Acquisitions

Issue #12 - January 15, 2004

- sdd: A 'dd' Variant
- Splitting The Disk With mmls

Issue #13 - March 15, 2004

- Call For Papers
- UNIX Incident Verification with Autopsy

Issue #14 - May 15, 2004

- Call For Papers
- TSK FAT File Recovery

Issue #15 - July 15, 2004

- Partition Recovery With TestDisk (Christophe Grenier)
- File Name Searching In Autopsy (Brian Carrier)

Issue #16 - September 15, 2004

- Searchtools, Indexed Searching in Forensic Images (Paul Bakker)
- sstrings and Unicode Searching (Brian Carrier)
- NTFS Orphan Files (Brian Carrier)

Issue #17 - November 15, 2004

- Detecting Host Protected Areas (HPA) in Linux (Brian Carrier)
- Finding Binary Signatures (Brian Carrier)

Issue #18 - January 15, 2005

- Description of the FAT fsstat Output (Brian Carrier)

Issue #19 - March 15, 2005

- New Image File Support (Brian Carrier)
- Hooking IO Calls for Multi-Format Image Support (Michael Cohen)

Issue #20 - May 15, 2005

- Removing Host Protected Areas (HPA) in Linux (Brian Carrier)
- Automatic Type Detection (Brian Carrier)

Issue #21 - November 15, 2005

- New Sleuth Kit Licenses (Brian Carrier)
- FAT and ils Changes (Brian Carrier)



Command Line Tools

While it would be nice for all tools to have graphical user interfaces, the truth is that there many very powerful tools that are only available from a command line. In fact, some investigators believe that these tools are more powerful than their GUI counterparts.

The following tools are available on the Linux side of the Helix CD, and will only operate from the command shell. When using these tools, it would be best to use a logged command shell (available from the taskbar) so that all your actions are logged.

Some of these tools are very powerful, and can be very destructive, so be very careful when using them.

2hash	MD5 & SHA1 parallel hashing.
bmap	Detect & Recover data in used slackspace.
chaosreader	Trace tcpdump files and extract data.
chkrootkit	Look for rootkits.
chntpw	Change Windows passwords.
dcfldd	dd replacement from the DCFL.
e2recover	Recover deleted files in ext2 file systems.
fatback	Analyze and recover deleted FAT files.
faust.pl	Analyze elf binaries and bash scripts.
fenris	debugging, tracing, decompiling.
foremost	Carve files based on header and footer.
f-prot	F-Prot Anti Virus Scanner.
ftimes	A toolset for forensic data acquisition.
galleta	Cookie analyzer for Internet Explorer.
glimpse	Indexing and query system.
grepmail	Grep through mailboxes.
logfinder.py	EFF logfinder utility.
logsh	Log your terminal session (Borrowed from FIRE).
lshw	Hardware Lister.
mac_grab.pl	e-fense MAC time utility.
mac-robber	TCT's graverobber written in C.
md5deep	Recursive md5sum with db lookups.
outguess	Stego detection suite.
pasco	Forensic tool for Internet Explorer Analysis.
rifiuti	"Recycle BIN" analyzer.
rkhunter	Rootkit hunter.
scalpel	fast file carver
sdd	Specialized dd with better performance.
sha1deep	Recursive sha1sum with db lookups.
sha256deep	Recursive sha1sum with db lookups.
stegdetect	Stego detection suite.
wipe	Secure file deletion.

2hash: MD5 & SHA1 parallel hashing.

Developed by Thomas Akin. Available from <http://crossrealm.com/2hash/>

From the website: 2hash simultaneously performs a md5 and a sha1 checksum on file(s). If you want two checksums for additionally integrity checking, you previously had to run md5sum and sha1sum serially causing the integrity checks to take 100% longer than running a single check alone. 2hash runs both hashes in parallel, only having to read the file once. It allows you to get both hash values with only about an 8% time increase over md5 alone, and only about a 2% time increase over sha1 alone. It runs about 90% quicker than using both md5 and sha1 one after the other...

Sample Ouput:

```
# 2hash recovered.txt
(md5)  547e3d9033620b83d6fb93a9106af672      recovered.txt
(sha1) f949d01a59b889aa1f448d2bb8a4c493ae56a84b  recovered.txt
#
```

The program will accept standard wildcards (?, *), and regular expressions.

bmap: Detect & Recover data in used slackspace.

Developed by Daniel Ridge. Available from

<http://www.packetstormsecurity.org/linux/security/bmap-1.0.17.tar.gz>

bmap can be used to store, recover, and delete information stored in the slack space of a file. Note: Users should be very careful when using this tool, according to the developer, "WARNING: This may spank your hard drive." There are concerns if this tool will operate on the ext3 filesystem. According to Henry Owen (2004), "bmap has not been updated since 2000 and was never tested with ext3. This foray into ext3 is not mentioned by the bmap author or anywhere on the internet and may damage your filesystem. "

The following is the output of the command: `bmap --help`

```
bmap:1.0.20 (05/17/04) newt@scyld.com
```

```
Usage: bmap [OPTION]... [<target-filename>]
```

```
use block-list knowledge to perform special operations on files
```

```
--doc VALUE
```

```
where VALUE is one of:
```

version	display version and exit
help	display options and exit
man	generate man page and exit
sgml	generate SGML invocation info

```
--mode VALUE
```

```
where VALUE is one of:
```

map	list sector numbers
carve	extract a copy from the raw device
slack	display data in slack space
putslack	place data into slack
wipeslack	wipe slack
checkslack	test for slack (returns 0 if file has slack)
slackbytes	print number of slack bytes available
wipe	wipe the file from the raw device
frag	display fragmentation information for the file
checkfrag	test for fragmentation (returns 0 if file is fragmented)

```
--outfile <filename> write output to ...
```

```
--label                useless bogus option
```

```
--name                 useless bogus option
```

```
--verbose              be verbose
```

```
--log-thresh <none | fatal | error | info | branch | progress | entryexit> logging threshold ...
```

```
--target <filename>  operate on ...
```

The following is from <http://www.linuxsecurity.com/content/view/117638/> (Chuvakin, 2002).

A more detailed look at ext2 internals reveals the existence of slack space. Filesystem uses addressable parts of disk called blocks, that have the same size. Ext2 filesystems typically use 1,2

or 4 KB blocks. If a file is smaller than the block size, the remaining space is wasted. It is called slack space. This problem long plagued early Windows 9x users with FAT16 filesystems, which had to use block sizes of up to 32K, thus wasting a huge amount of space if storing small files.

On a 4GB Linux partition, the block size is typically 4K (chosen automatically when the `mke2fs` utility is run to create a filesystem). Thus one can reliably hide up to 4KB of data per file if using a small file. The data will be invulnerable to disk usage, invisible from the filesystem, and, which is more exciting for some people, undetectable by file integrity checkers using file checksumming algorithms and MAC times. Ext2 floppy (with a block size of 1KB) allows hiding data as well, albeit in smaller chunks.

The obscure tool `bmap` exists to jam data in slack space, take it out and also wipe the slack space, if needed. Some of the examples follow:

The following command puts the data in slack space produced by `/etc/passwd` file

```
# echo "evil data is here" | bmap --mode putslack /etc/passwd
```

This command will show the data stored in a file's slack space

```
# bmap --mode slack /etc/passwd
getting from block 887048
file size was: 9428
slack size: 2860
block size: 4096
evil data is here
```

This command will delete any data stored in a file's slack space

```
# bmap --mode wipeslack /etc/passwd
```

Hiding data in slack space can be used to store secrets, plant evidence (forensics software will find it, but the suspect probably will not) and maybe hide tools from integrity checkers (if automated splitting of the larger file into slack-sized chunks is implemented).

ChaosReader: Trace tcpdump files and extract data.

Developed by Brendan Gregg. Available from
<http://users.tpg.com.au/bdgcvb/chaosreader.html>



ChaosReader is a freeware tool to trace TCP/UDP/... sessions and fetch application data from snoop or tcpdump logs. This is a type of "any-snarf" program, as it will fetch telnet sessions, FTP files, HTTP transfers (HTML, GIF, JPEG,...), SMTP emails, and such, from the captured data inside network traffic logs. A html index file is created that links to all the session details, including realtime replay programs for telnet, rlogin, IRC, X11 or VNC sessions; and reports such as image reports and HTTP GET/POST content reports. Chaosreader can also run in standalone mode - where it invokes tcpdump or snoop (if they are available) to create the log files and then processes them (Gregg, 2004).

The following information was taken from <http://users.tpg.com.au/adsl4yb/Chaos/readme.txt>

QUICK USAGE:

```
tcpdump -s9000 -w out1; chaosreader out1; netscape index.html
or,
snoop -o out1; chaosreader out1; netscape index.html
or,
ethereal (save as "out1"); chaosreader out1; netscape index.html
or,
chaosreader -s 5; netscape index.html
```

```
USAGE: chaosreader [-aehikqrvxAHIRTUXY] [-D dir]
               [-b port[,...]] [-B port[,...]]
               [-j IPaddr[,...]] [-J IPaddr[,...]]
               [-l port[,...]] [-L port[,...]] [-m bytes[k]]
               [-M bytes[k]] [-o "time"|"size"|"type"|"ip"]
               [-p port[,...]] [-P port[,...]]
               infile [infile2 ...]
```

```
chaosreader -s [mins] | -S [mins[,count]]
               [-z] [-f 'filter']
```

```
chaosreader      # Create application session files, indexes

-a, --application # Create application session files (default)
-e, --everything  # Create HTML 2-way & hex files for everything
-h               # Print a brief help
--help          # Print verbose help (this) and version
--help2         # Print massive help
-i, --info       # Create info file
-q, --quiet      # Quiet, no output to screen
-r, --raw        # Create raw files
-v, --verbose    # Verbose - Create ALL files .. (except -e)
-x, --index      # Create index files (default)
-A, --noapplication # Exclude application session files
-H, --hex        # Include hex dumps (slow)
-I, --noinfo     # Exclude info files
-R, --noraw      # Exclude raw files
-T, --notcp      # Exclude TCP traffic
-U, --noudp      # Exclude UDP traffic
-Y, --noicmp     # Exclude ICMP traffic
-X, --noindex    # Exclude index files
-k, --keydata    # Create extra files for keystroke analysis
```

```

-D dir      --dir dir          # Output all files to this directory
-b 25,79    --playtcp 25,79    # replay these TCP ports as well (playback)
-B 36,42    --playudp 36,42    # replay these UDP ports as well (playback)
-l 7,79     --htmltcp 7,79     # Create HTML for these TCP ports as well
-L 7,123    --htmludp 7,123    # Create HTML for these UDP ports as well
-m 1k       --min 1k           # Min size of connection to save ("k" for Kb)
-M 1024k    --max 1k           # Max size of connection to save ("k" for Kb)
-o size     --sort size        # sort Order: time/size/type/ip (Default time)
-p 21,23    --port 21,23       # Only examine these ports (TCP & UDP)
-P 80,81    --noport 80,81     # Exclude these ports (TCP & UDP)
-s 5        --runonce 5        # Standalone. Run tcpdump/snoop for 5 mins.
-S 5,10     --runmany 5,10     # Standalone, many. 10 samples of 5 mins each.
-S 5        --runmany 5        # Standalone, endless. 5 min samples forever.
-z          --runredo          # Standalone, redo. Rereads last run's logs.
-j 10.1.2.1 --ipaddr 10.1.2.1   # Only examine these IPs
-J 10.1.2.1 --noipaddr 10.1.2.1 # Exclude these IPs
-f 'port 7' --filter 'port 7'   # With standalone, use this dump filter.

```

eg1,

```

tcpdump -s9000 -w output1      # create tcpdump capture file
chaosreader output1            # extract recognised sessions, or,
chaosreader -ve output1        # gimme everything, or,
chaosreader -p 20,21,23 output1 # only ftp and telnet...

```

eg2,

```

snoop -o output1              # create snoop capture file instead
chaosreader output1            # extract recognised sessions...

```

eg3,

```

chaosreader -S 2,5            # Standalone, sniff network 5 times for 2 mins
                              # each. View index.html for progress (or .text)

```

Output Files: Many will be created, run this in a clean directory.

Short example,

index.html	Html index (full details)
index.text	Text index
index.file	File index for standalone redo mode
image.html	HTML report of images
getpost.html	HTML report of HTTP GET/POST requests
session_0001.info	Info file describing TCP session #1
session_0001.telnet.html	HTML coloured 2-way capture (time sorted)
session_0001.telnet.raw	Raw data 2-way capture (time sorted)
session_0001.telnet.raw1	Raw 1-way capture (assembled) server->client
session_0001.telnet.raw2	Raw 1-way capture (assembled) client->server
session_0002.web.html	HTML coloured 2-way
session_0002.part_01.html	HTTP portion of the above, a HTML file
session_0003.web.html	HTML coloured 2-way
session_0003.part_01.jpeg	HTTP portion of the above, a JPEG file
session_0004.web.html	HTML coloured 2-way
session_0004.part_01.gif	HTTP portion of the above, a GIF file
session_0005.part_01.ftp-data.gz	An FTP transfer, a gz file.

...

The convention is,

session_*	TCP Sessions
stream_*	UDP Streams
icmp_*	ICMP packets
index.html	HTML Index
index.text	Text Index
index.file	File Index for standalone redo mode only
image.html	HTML report of images
getpost.html	HTML report of HTTP GET/POST requests
*.info	Info file describing the Session/Stream
*.raw	Raw data 2-way capture (time sorted)

*.raw1	Raw 1-way capture (assembled) server->client
*.raw2	Raw 1-way capture (assembled) client->server
*.replay	Session replay program (perl)
.partial.	Partial capture (tcpdump/snoop were aware of drops)
*.hex.html	2-way Hex dump, rendered in coloured HTML
*.hex.text	2-way Hex dump in plain text
*.X11.replay	X11 replay script (talks X11)
*.textX11.replay	X11 communicated text replay script (text only)
*.textX11.html	2-way text report, rendered in red/blue HTML
*.keydata	Keystroke delay data file. Used for SSH analysis.

Modes:

- * Normal - eg "chaosreader infile", this is where a tcpdump/snoop file was created previously and chaosreader reads and processes it.
- * Standalone, once - eg "chaosreader -s 10", this is where chaosreader runs tcpdump/snoop and generates the log file, in this case for 10 minutes, and then processes the result. Some OS's may not have tcpdump or snoop available so this will not work (instead you may be able to get Ethereal, run it, save to a file, then use normal mode). There is a master index.html and the report index.html in a sub dir, which is of the format out_YYYYMMDD-hhmm, eg "out_20031003-2221".
- * Standalone, many - eg "chaosreader -S 5,12", this is where chaosreader runs tcpdump/snoop and generates many log files, in this case it samples 12 times for 5 minutes each. While this is running, the master index.html can be viewed to watch progress, which links to minor index.html reports in each sub directory.
- * Standalone, redo - eg "chaosreader -ve -z", (the -z), this is where a standalone capture was previously performed - and now you would like to reprocess the logs - perhaps with different options (in this case, "-ve"). It reads index.file to determine which capture logs to read.
- * Standalone, endless - eg "chaosreader -S 5", like standalone many - but runs forever (if you ever had the need?). Watch your disk space!

Note: this is a work in progress, some of the code is a little unpolished.

Advice:

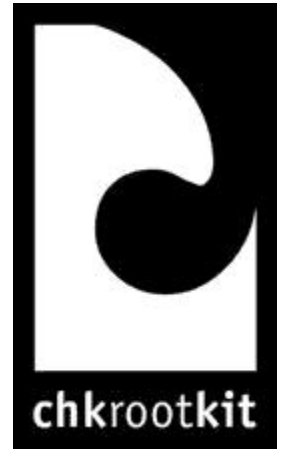
- * Run chaosreader in an empty directory.
- * Create small packet dumps. Chaosreader uses around 5x the dump size in memory. A 100Mb file could need 500Mb of RAM to process.
- * Your tcpdump may allow "-s0" (entire packet) instead of "-s9000".
- * Beware of using too much disk space, especially standalone mode.
- * If you capture too many small connections giving a huge index.html, try using the -m option to ignore small connections. eg "-m 1k".
- * snoop logs may actually work better. Snoop logs are based on RFC1761, however there are many variants of tcpdump/libpcap and this program cannot read them all. If you have Ethereal you can create snoop logs during the "save as" option. On Solaris use "snoop -o logfile".
- * tcpdump logs may not be portable between OSs that use different sized timestamps or endian.
- * Logs are best created in a memory filesystem for speed, usually /tmp.
- * For X11 or VNC playbacks, first practise by replaying a recent captured session of your own. The biggest problem is colour depth, your screen must match the capture. For X11 check authentication (xhost +), for VNC check the viewers options (-8bit, "Hextile", ...)
- * SSH analysis can be performed with the "sshkeydata" program as demonstrated on <http://www.brendangregg.com/sshanalysis.html> . chaosreader provides the input files (*.keydata) that sshkeydata analyses.

chkrootkit: Look for rootkits.

Developed by Nelson Murilo and Klaus Steding-Jessen. It is available from <http://www.chkrootkit.org>

chkrootkit is a tool to locally check for signs of a rootkit. It will check the major utilities for infection, and can currently detect 60 rootkits and their variations.

The following information was taken from <http://www.chkrootkit.org/README>



Usage

chkrootkit must run as root. The simplest way is:

```
# ./chkrootkit
```

This will perform all tests. You can also specify only the tests you want, as shown below:

Usage: `./chkrootkit [options] [testname ...]`

Options:

-h	show this help and exit
-V	show version information and exit
-l	show available tests
-d	debug
-q	quiet mode
-x	expert mode
-r dir	use dir as the root directory
-p dir1:dir2:dirN	path for the external commands used by chkrootkit
-n	skip NFS mounted dirs

Where *testname* stands for one or more from the following list:

aliens	chkutmp	egrep	init	pop2	tcpd
asp	amd	env	killall	pop3	tcpdump
bindshel	basenam	find	ldsopreloa	ps	top
l	e	fingerd	d login	pstree	telnetd
lkm	biff	gpm	ls	rpcinfo	timed
rexedcs	chfn	grep	lsf	rlogind	traceroute
sniffer	chsh	hdparm	mail	rshd	vdir
w55808	cron	su	mingetty	slogin	w
wted	date	ifconfig	netstat	sendmail	write
scalper	du	inetd	named	shd	
slapper	dirname	inetdconf	passwd	syslogd	
z2	echo	identd	pidof	tar	

For example, the following command checks for trojaned ps and ls binaries and also checks if the network interface is in promiscuous mode.

```
# ./chkrootkit ps ls sniffer
```

The ``-q'` option can be used to put chkrootkit in quiet mode – in this mode only output messages with ``infected'` status are shown.

With the ``-x'` option the user can examine suspicious strings in the binary programs that may indicate a trojan -- all the analysis is left to the user.

Lots of data can be seen with:

```
# ./chkrootkit -x | more
```

Pathnames inside system commands:

```
# ./chkrootkit -x | egrep '^/'
```

chkrootkit uses the following commands to make its tests: `awk`, `cut`, `egrep`, `find`, `head`, `id`, `ls`, `netstat`, `ps`, `strings`, `sed`, `uname`. It is possible, with the ``-p'` option, to supply an alternate path to chkrootkit so it won't use the system's (possibly) compromised binaries to make its tests.

To use, for example, binaries in `/cdrom/bin`:

```
# ./chkrootkit -p /cdrom/bin
```

It is possible to add more paths with a ``:'`

```
# ./chkrootkit -p /cdrom/bin:/floppy/mybin
```

Sometimes is a good idea to mount the disk from a compromised machine on a machine you trust. Just mount the disk and specify a new rootdir with the ``-r'` option.

For example, suppose the disk you want to check is mounted under `/mnt`, then:

```
# ./chkrootkit -r /mnt
```

Output Messages

The following messages are printed by chkrootkit (except with the `-x` and `-q` command options) during its tests:

"INFECTED": the test has identified a command probably modified by a known rootkit;

"not infected": the test didn't find any known rootkit signature.

"not tested": the test was not performed -- this could happen in the following situations:

- a) the test is OS specific;
- b) the test depends on an external program that is not available;
- c) some specific command line options are given. (e.g. `-r`).

"not found": the command to be tested is not available;

"Vulnerable but disabled": the command is infected but not in use. (not running or commented in inetd.conf)

A trojaned command has been found. What should I do now?

Your biggest problem is that your machine has been compromised and this bad guy has root privileges.

Maybe you can solve the problem by just replacing the trojaned command -- the best way is to reinstall the machine from a safe media and to follow your vendor's security recommendations.

More Information

"Adding Chkrootkit to Your Unix Auditing Arsenal", by Bill Hutchison, available from http://www.giac.org/practical/gsec/Bill_Hutchison_GSEC.pdf

"Understanding Rootkits", by Oktay Altunergil, available from <http://www.linuxdevcenter.com/pub/a/linux/2001/12/14/rootkit.html>

"Scanning for Rootkits", by Oktay Altunergil, available from <http://www.linuxdevcenter.com/pub/a/linux/2002/02/07/rootkits.html>

chntpw: Change Windows passwords.

Offline NT Password & Registry Editor

Developed by Petter Nordahl-Hagen.

Available from <http://home.eunet.no/~pnordahl/ntpasswd/>

This is an incredibly useful tool to reset the password on any Windows NT, 2000, and XP account. In order for this to work, the drive containing the OS must be mounted as read/write to allow the program to modify the registry of the target system.

The following is taken from: <http://home.eunet.no/~pnordahl/ntpasswd/README.txt>

The Offline NT Password Editor

What does it do?

This little program will enable you to view some information and change user passwords in a Windows NT SAM userdatabase file. You do not need to know the old passwords. However, you need to get at the file some way or another yourself. In addition it contains a simple registry editor with full write support, and hex-editor which enables you to fiddle around with bits&bytes in the file as you wish yourself.

Why?

I often forget passwords. Especially on test installations (that I just must have some stuff out of half a year later..) On most unix-based boxes you just boot the thingy off some kind of rescue bootmedia (cd/floppy etc), and simply edit the password file. On Windows NT however, as far as I know, there is no way except reinstalling the userdatabase, losing all users except admin. (ok, some companies let you pay lotsa \$\$\$\$ for some rescue service..)

How?

Currently, this thing only runs under linux, but it may just happen to compile on other platforms, too. (there are dos-versions available, look for links on my webpage) So, to set a new adminpassword on your NT installation you either:

- 1) Take the harddrive and mount it on a linux-box
- 2) Use a linux-bootdisk or CD

one is available at: <http://home.eunet.no/~pnordahl/ntpasswd/>

ie. you do it offline, with the NT system down.

Usage:

```
chntpw version 0.99.2 040105, (c) Petter N Hagen
chntpw: change password of a user in a NT SAM file, or invoke registry editor.
chntpw [OPTIONS] <samfile> [systemfile] [securityfile] [otherrehive] [...]
-h          This message
-u <user>   Username to change, Administrator is default
-l          list all users in SAM file
-i          Interactive. List users (as -l) then ask for username to change
-e          Registry editor. Now with full write support!
-d          Enter buffer debugger instead (hex editor),
-t          Trace. Show hexdump of structs/segments. (deprecated debug function)
-v          Be a little more verbose (for debugging)
-L          Write names of changed files to /tmp/changed
-N          No allocation mode. Only (old style) same length overwrites possible
```

Normal usage is:

```
> chntpw sam system security
- open registry hives 'sam' and 'system' and change administrator account.
Versions dated later from Feb 1999 and later also supports
and will find the admin account, even if the name has been changed,
or the name has been localized (different languageversion of NT
use different admin-names)
```

The -u option:

Specifies user to change:

```
> chntpw -u jabbathehutt mysam
- Prompt for password for 'jabbathehutt', if found (otherwise do nothing)
```

Or you may give RID number in hex:

```
> chntpw -u 0x1f4 mysam
- Will edit administrator.
```

Names does not support multibyte (unicode) characters like some russian and asian locales. Give RID in hex to edit users with such names. Must start with 0x. Ex: 0x2fa

The -l option:

Will list all users in the sam-file.

The -i option:

Go into the interactive menu system.

The -d option:

This will load the file, and then immediately enter the buffer debugger.
This is a simple hex-editor with only a few commands, enter ? at the . prompt to see
a short command overview.
'q' exits without saving, 's' exit and saves.

The -e option:

Will enter the registry editor.
You can navigate the registry like a filesystem at the command-line prompt:
See regedit.txt file for more info.

The -t option:

This is a debug function (extended -l) to show how it traces the chain of structs in
the file. This also includes a raw interpretation of the different registry
structures + a hex dump.

The -L option:

Drops the filenames of the changed hives in /tmp/changed
Used by the floppy scripts.

The -N option:

Will fall back to old edit mode, disable the block allocations and only support
overwrite-same-size. Used to ensure safety in testing period.

dcfldd: dd replacement from the DCFL.

Developed by Nicholas Harbour. Available from <http://dcfldd.sourceforge.net/>

This tool was originally developed at the Department of Defense Computer Forensics Lab (DCFL), hence the name. While Nick Harbour is no longer affiliated with the DCFL, he still maintains the package. The DCFL does not maintain, support, or have any other affiliation with dcfldd (Harbour, 2006). Helix contains dcfldd 1.3.4 which is now undergoing forensic accreditation.

dcfldd is an enhanced version of GNU dd with features useful for forensics and security. Based on the dd program found in the GNU Coreutils package, dcfldd has the following additional features (Harbour, 2006):

- Hashing on-the-fly - dcfldd can hash the input data as it is being transferred, helping to ensure data integrity.
- Status output - dcfldd can update the user of its progress in terms of the amount of data transferred and how much longer operation will take.
- Flexible disk wipes - dcfldd can be used to wipe disks quickly and with a known pattern if desired.
- Image/wipe Verify - dcfldd can verify that a target drive is a bit-for-bit match of the specified input file or pattern.
- Multiple outputs - dcfldd can output to multiple files or disks at the same time.
- Split output - dcfldd can split output to multiple files with more configurability than the split command.
- Piped output and logs - dcfldd can send all its log data and output to commands as well as files natively.

The following is the man page for dcfldd.

NAME

dcfldd - manual page for dcfldd (dcfldd) 1.2.4

SYNOPSIS

dcfldd [OPTION]...

DESCRIPTION

Copy a file, converting and formatting according to the options.

bs=BYTES

force ibs=BYTES and obs=BYTES

cbs=BYTES

convert BYTES bytes at a time

conv=KEYWORDS

convert the file as per the comma separated keyword list

count=BLOCKS

copy only BLOCKS input blocks

ibs=BYTES

read BYTES bytes at a time

if=FILE
 read from FILE instead of stdin

obs=BYTES
 write BYTES bytes at a time

of=FILE
 write to FILE instead of stdout

NOTE: of=FILE may be used several times to write
 output to multiple files simultaneously

seek=BLOCKS
 skip BLOCKS obs-sized blocks at start of output

skip=BLOCKS
 skip BLOCKS ibs-sized blocks at start of input

pattern=HEX
 use the specified binary pattern as input

textpattern=TEXT
 use repeating TEXT as input

hashwindow=BYTES
 perform a hash on every BYTES amount of data

hash=NAME
 either md5, sha1, sha256, sha384 or sha512

default algorithm is md5. To select multiple algorithms to run
 simultaneously enter the names in a comma separated list

hashlog=FILE
 send MD5 hash output to FILE instead of stderr

if you are using multiple hash algorithms you can send each to a
 separate file using the convention ALGORITHMlog=FILE, for
 example md5log=FILE1, sha1log=FILE2, etc.

hashformat=FORMAT
 display each hashwindow according to FORMAT

the hash format mini-language is described below

totalhashformat=FORMAT display the total hash value according to
 FORMAT status=[on|off] display a continual status message
 on stderr

default state is "on"

statusinterval=N
 update the status message every N blocks

default value is 256

sizeprobe=[if|of]
 determine the size of the input or output file

for use with status messages. (this option gives you a percent-

age indicator) WARNING: do not use this option against a tape device.

split=BYTES
write every BYTES amount of data to a new file

This operation applies to any of=FILE that follows

splitformat=TEXT
the file extension format for split operation.

you may use any number of 'a' or 'n' in any combo the default format is "nnn" NOTE: The split and splitformat options take effect

only for output files specified AFTER these options appear in the command line. Likewise, you may specify these several times for for different output files within the same command line. you may use as many digits in any combination you would like. (e.g. "anaannnaana" would be valid, but quite insane)

vf=FILE
verify that FILE matches the specified input

verifylog=FILE
send verify results to FILE instead of stderr

--help display this help and exit

--version
output version information and exit

The structure of of FORMAT may contain any valid text and special variables. The built-in variables are used the following format: #variable_name# To pass FORMAT strings to the program from a command line, it may be necessary to surround your FORMAT strings with "quotes." The built-in variables are listed below:

window_start
The beginning byte offset of the hashwindow

window_end
The ending byte offset of the hashwindow

block_start
The beginning block (by input blocksize) of the window

block_end
The ending block (by input blocksize) of the hash window

hash The hash value

algorithm
The name of the hash algorithm

For example, the default FORMAT for hashformat and totalhashformat are:
hashformat="#window_start# - #window_end#: #hash#" totalhashformat="Total (#algorithm#): #hash#"

The FORMAT structure accepts the following escape codes:

\n Newline
 \t Tab
 \r Carriage return
 \\ Insert the '\' character
 ## Insert the '#' character as text, not a variable

BLOCKS and BYTES may be followed by the following multiplicative suffixes: xM M, c 1, w 2, b 512, kD 1000, k 1024, MD 1,000,000, M 1,048,576, GD 1,000,000,000, G 1,073,741,824, and so on for T, P, E, Z, Y. Each KEYWORD may be:

ascii from EBCDIC to ASCII
 ebcdic from ASCII to EBCDIC
 ibm from ASCII to alternated EBCDIC
 block pad newline-terminated records with spaces to cbs-size
 unblock replace trailing spaces in cbs-size records with newline
 lcase change upper case to lower case
 notrunc do not truncate the output file
 ucase change lower case to upper case
 swab swap every pair of input bytes
 noerror continue after read errors
 sync pad every input block with NULs to ibs-size; when used with block or unblock, pad with spaces rather than NULs

e2recover: Recover deleted files in ext2 file systems.

Developed by Aaron Crane. Available from <http://www.ibiblio.org/linsearch/lsmis/e2recover-1.0.html>

The following is taken from the command: `e2recover --help`

Usage: `e2recover [OPTION]... [LSDEL-FILES]...`

Attempt to recover deleted files from an ext2 file system, using the output (in LSDEL-FILES) from the "lsdel" command in debugfs. Standard input is read if no file names are given or on a file name of '-'. Recovered files are written to the appropriate temporary directory, with names like 'e2rec.PID.DEV.INUM'. Uses environment variables \$FSGRAB and \$DEBUGFS to find those programs, or looks in \$PATH if they are unset.

<code>-b, --block-size=BLOCKSIZE</code>	the file system has blocks of BLOCKSIZE bytes (default: 1024)
<code>-d, --device=DEVICE</code>	the file system is on DEVICE (default: /dev/hda1)
<code>-g, --guess-indirects</code>	try to recover files with zeroed indirect blocks by assuming that there was no fragmentation in that file
<code>-t, --tmpdir=TMPDIR,</code> <code>--tempdir=TMPDIR</code>	write recovered files to TMPDIR (default: \${TMPDIR:-/tmp})
<code>--help</code>	display this help and exit
<code>--version</code>	display version information and exit

BLOCKSIZE may have an optional multiplier suffix: w for 2, b for 512, k for 1024, m for 1Meg. BLOCKSIZE must be an exact multiple of 512 bytes.

For more information see the Linux Ext2fs Undeletion mini-HOWTO by Aaron Crane, located at <http://www.faqs.org/docs/Linux-mini/Ext2fs-Undeletion.html>

f-prot: F-Prot Anti Virus Scanner.

Developed by FRISK Software International. Available from http://www.f-prot.com/products/home_use/linux/

For home users using the Linux open-source operating system, we offer F-Prot Antivirus for Linux Workstations. F-Prot Antivirus for Linux Workstations utilizes the renowned F-Prot Antivirus scanning engine for primary scan but has in addition to that a system of internal heuristics devised to search for unknown viruses (Frisk Software International, 2006).



F-Prot Antivirus for Linux was especially developed to effectively eradicate viruses threatening workstations running Linux. It provides full protection against macro viruses and other forms of malicious software - including Trojans. F-Prot Antivirus can detect a total of 232593 worms, viruses and other malicious programs (Frisk Software International, 2006).

A GUI version of this program is also available.

The following is the man page for f-prot.

NAME

f-prot - F-Prot Antivirus for UNIX, Command-Line Scanner

SYNTAX

f-prot [options] [file or directory]

DESCRIPTION

f-prot f-prot is a tool for scanning individual files or directory trees for viruses. The options selected determine which methods are used for scanning. By default f-prot scans all files, including inside archives, and reports to STDOUT. F-prot only lists files which are found to be infected.

REPORTING OPTIONS

By default f-prot reports to STDOUT, and only lists files which have been found to be infected.

-append

Append to existing report file.

-help Displays short summary of available options for F-Prot Antivirus.

-list Show a list of all files which have been checked.

-nobreak

Do not abort scan if ESC is pressed.

-old Do not give a warning message when using outdated DEF files.

-page Only show one screen output at a time.

- report=<report_name>
Save output to the <report_name> file.
- silent
Do not generate any screen output. This can be useful in the case of running f-prot in a cron job and using the -report option.
- wrap Wrap text output so it fits in 78 columns. This also applies to the file used with the -report option.

SCANNING OPTIONS

By default f-prot scans all files, including inside archives.

- ai Enable neural-network virus detection. The -ai option should not be used with the -noheur option.
- archive=n [default is 5]
Scan inside supported archives n levels deep, the supported range is between 1 and 99. The older form '-archive' is supported for compatibility reasons, in which case n is set to 5. Supported archives are .zip, .cab, .tar, .gz, .lzh and .arj files. Currently F-Prot Antivirus does not support disinfection or removal of infected files within archives. Unix mailboxes are considered to be archives and therefore F-Prot Antivirus is not able to remove infected attachments from mailboxes.
- server [default]
Attempts to identify infections within password protected archives. "-server" implies "-archive=5".
- noserver
Does not attempt to identify infections within password protected archives.
- auto Automatically remove detected viruses. As noted above, this will not work on archived files.
- collect
Scan a virus collection. This option is intended for advanced users. When this option is used it will, e.g. scan for bootsector viruses within files, even though the virus resides within a file instead of a bootsector.
- delete
Delete infected files. User confirmation is required. However, the -auto option can be used to automatically confirm the action. F-Prot Antivirus does not support removal of infected objects located in archives. Also, the -delete option has no effect on office documents, since that could cause the loss of work.
- disinf
Disinfect whenever possible. User confirmation is required. However, the -auto option can be used to automatically confirm the action. F-Prot Antivirus does not support disinfection of infected objects located in archives.
- dumb [default]

Scans all files

`-type` Scan files by content. By default `f-prot` scans all files. By using the `-type` option, you are instructing the scanner to limit the search to scanning by content.

`-ext` Scan only files with default extensions. By default `f-prot` scans all files. By using the `-ext` option, you are instructing the scanner to limit the search to files with default extensions.

`-follow`
Follow symbolic links. This should be used with care, as the program does not detect "circular" directories, and may get stuck in an endless loop.

`-noheur`
Disable heuristic scanning. The `-noheur` option should not be used with the `-ai` option.

`-nosub` Do not scan subdirectories.

`-onlyheur`
Only use heuristics, do not scan for known virus signatures. By using this option `F-Prot Antivirus` will only detect a fraction of infected files.

`-packed` [default]
Unpack compressed executables. There is no corresponding `-nopacked` option. This option is provided for legacy reasons.

`-rename`
Rename extensions of infected files to prevent them from being executed, e.g. renaming `file.com` to `file.vom` and `file.exe` to `file.vxe`. This will not prevent files from being executed on Unix since on the one hand `.exe` files and `.com` files from Windows are not executable on a Unix platform by default, and on the other hand file extensions are not used on Unix systems with regards to executability.

MACRO SCANNING OPTIONS

By default `f-prot` scans for macro's within known file-types.

`-nomacro`
Do not scan for macro viruses.

`-onlymacro`
Only scan for macro viruses.

`-removeall`
Remove all macros from all documents. When this option is used with `-disinf` or `-delete` all identified macros will be removed.

`-removenew`
Remove new variants of macro viruses by removing all macros from infected documents.

`-saferemove`
Remove all macros from documents, if a known virus is found.

INFORMATION OPTIONS

These information options are stand-alone, you can not combine them with other options. (The version information displayed by -verno are included in the beginning of every scan report by default).

-verno Show version information.

-virlist

List viruses known to F-Prot Antivirus with the current virus signature files.

-virno Give statistical information about viruses known to F-Prot Antivirus with the current virus signature files.

PROGRAM EXIT CODES

- 0 Normal exit. Nothing found, nothing done.
- 1 Unrecoverable error (e.g., missing virus signature files).
- 2 Selftest failed (program has been modified).
- 3 At least one virus-infected object was found.
- 4 Reserved, not currently in use.
- 5 Abnormal termination (scanning did not finish).
- 6 At least one virus was removed.
- 7 Error, out of memory.
- 8 At least one suspicious object was found.
- 9 At least one object was not scanned (encrypted file, unsupported/unknown compression method, unsupported/unknown file format, corrupted or invalid file).
- 10 At least one archive object was not scanned (contains more than N levels of nested archives, as specified with -archive switch).

fatback: Analyze and recover deleted FAT files.

Developed by Nicholas Harbour of the DoD Computer Forensics Lab. Available from <http://prdownloads.sourceforge.net/biatchux/>

Fatback is a forensic tool for undeleting files from Microsoft FAT file systems. Fatback is different from other undelete tools in that it does the following :

- Runs under UNIX environments (only Linux and FreeBSD tested so far)
- Can undelete files automatically
- Supports Long File Names
- Supports FAT12, FAT16, and FAT32
- Powerful interactive mode
- Recursively undeletes deleted directories
- Recovers lost cluster chains
- Works with single partitions or whole disks

The following is taken from the command: fatback

```
Usage: fatback [FILE] -l [LOG] [OPTION]...
Undelete files from FAT filesystems.
Fatback v1.3
(c) 2000-2001 DoD Computer Forensics Lab
-o, --output=DIR          specifies a directory to place output files
-a, --auto                auto undelete mode. non-interactively
                           recovers all deleted files
-l, --log=LOGFILE         specifies a file to audit log to.
-v, --verbose             display extra information to the screen.
-p, --partition=PNUM      go directly to PNUM partition
-d, --delprefix=PREFIX    use PREFIX to signify deleted files instead
                           of the default "?"
-s, --single              force into single partition mode
-z, --sectsize=SIZE       adjust the sector size. default is 512
-m, --mmap                use mmap() file I/O for improved performance
-h, --help                display this help screen
Report bugs to <harbourn@dcfl.gov>
```

The following it taken from the fatback-manual.info from the <http://prdownloads.sourceforge.net/biatchux/fatback-1.3.tar.gz>

Using Fatback

In order to cater to users with a variety of experience levels, Fatback provides two ways of interacting. The first method is called "automated" mode and input is solely given on the command line. This method is for users who simply want to recover all files (or just deleted files) from a partition and not be bothered by the details. The second method is called "interactive" mode. In interactive mode, a user interacts with Fatback through a command interpreter which mimics the look and feel of a traditional UNIX shell. Interactive mode is recommended for users that want to do more advanced undeleting.

There is no difference in the undelete technique of the two different modes. When a user runs Fatback in automated mode, it is actually running predefined or "canned" commands through the fatback interpreter.

The only limitation of the automated mode (as of version 1.3) is that it will only process a single partition.

To run Fatback, type the program name ('fatback'), then type any options you wish to pass to Fatback. The last argument on the command line should be the name of the input file. Here is the command syntax:

```
fatback OPTIONS INPUT-FILE
```

The options can either be a letter or a word and may or may not require any arguments. For example, to specify a file to place the audit log into, you may use the '-l' flag or the '--log' flag. These options require an argument. To specify the required argument with the '-l' option, use '-l FILE'. To specify the argument with the '--log' option, use '--log=FILE'.

The input file can be either a device (a file in the '/dev' directory) or an image of a drive or partition.

Audit Logs

=====

Fatback uses audit logs to keep a record of operations performed in a session. The data it logs includes the commands the user types, the command line used to execute the program, the users environment, information about the partition being analyzed, and information about each file that was recovered.

By default, the audit log will be written to a file called 'fatback.log' in the current directory. To store the audit log to a different location, use the '-l FILE' or '--log=FILE' switch.

Command Line Options

=====

Fatback version 1.3 provides the following command line options:

'-a'

'--auto'

Run Fatback in automatic undelete mode. This mode will attempt to recover all deleted files in a given partition, and only that partition. If the input data is a partitioned drive, use the '-p NUMBER' or '--partition=NUMBER' option to specify which partition to use.

'-o DIRECTORY'

'--output=DIRECTORY'

Place recovered files into the directory specified. If Fatback is run in automatic undelete mode, or if a recursive copy is performed, sub directories will be created underneath the output directory that correspond to directories in the partition that Fatback is working with.

'-l LOG-FILE'

'--log=LOG-FILE'

Place the audit log into the specified file.


```

'-v'
'--verbose'
    Display extra information to the screen.

'-p PARTITION-NUMBER'
'--partition=PARTITION-NUMBER'
    Process a specific partition of a partitioned drive. This is
    necessary to use auto mode with a partitioned drive. In
    interactive mode, the partition menu will be bypassed.

'-d'

'--delprefix=PREFIX'
    Use PREFIX as the beginning of the name of deleted files. The
    default value is '?'.

'-s'
'--single'
    Treat input as a single partition without checking for partitions.

'-z SECTOR-SIZE'
'--sectsize=SECTOR-SIZE'
    Use SECTOR-SIZE as the sector size of the input data instead of the
    default value of 512.

'-h'
'--help'
    Display a help screen and terminate

'-V'
'--version'
    Display the Fatback version number and terminate.

```

The Fatback Interpreter

=====

If Fatback is run without the `'-a'` or `'--auto'` option, it enters what is called "interactive" mode. In interactive mode, Fatback gives you a prompt to which you can enter commands and direct Fatback to perform more specific tasks than the automatic undelete mode.

If the input is a partitioned drive, Fatback will first display a menu of possible partitions and prompt you for which you would like to work with. Fatback will then enter the partition and you may begin exploring and recovering files!

The command interpreter is loosely modeled after the classic UNIX shell environment. The interpreter provides a prompt (`'fatback>'` by default), and mimics several UNIX shell commands such as `'ls'`, `'cd'`, `'pwd'`, `'cp'`, and many others.

Fatback version 1.3 has the following commands:

```

'cd'
    Change to a specified directory

'copy'
'cp'
    Copy files out to an external file system

'help'
    Display a list of commands and a brief description of each

```

```

`dir'
`ls'
    List entries in a directory

`pwd'
    Print the name of the current directory

`stat'
    Display detailed information about a directory entry

`chain'
    Display the cluster chain for a directory entry

`cpchain'
    Copy a cluster chain out to a file

`lostchains'
    Display a list of lost cluster chains in the current partition

`sh'
    Execute a command in the outside environment

`set'
    Set run-time variables within Fatback

`done'
    Stop working with the current partition, or exit fatback if in
    single partition mode.

`quit'
    Exit Fatback

```

The ``copy'` command is synonymous with ``cp'`, and the ``dir'` command is synonymous with ``ls'`. The ``copy'` and ``dir'` aliases were created to give users who primarily use DOS a familiar interface. However, the Fatback interpreter was designed to mimic a UNIX shell, so the ``cp'` and ``ls'` forms are preferred and used by all the documentation.

It is important to note that Fatback is very case sensitive. All directory entries are in upper case, and some may have a long file name (*note Long File Names::) associated with it that can be mixed case. When specifying directory entries you must use either the exact uppercase name, or the long file name. To specify a long file name that contains white space, put the whole name in double quotes. For example, the ``Program Files'` directory in a windows system can be specified by either ``PROGRA~1'` or ``"Program Files"'`.

The ``cd'` Command

The ``cd'` command has the following syntax:

```
cd DIRECTORY
```

This will set the current directory to DIRECTORY. DIRECTORY may be any number of layers deeper than the current directory. For example, to change to the ``system'` directory underneath the ``windows'` directory from the root directory, you would run the following command:

```
cd /windows/system
```

The directory names ``.`` and ``..`` are reserved for relative path specification purposes. The ``.`` is a directory entry that represents its parent directory. For example, specifying ``MYDIR/`.`` is the same as specifying ``MYDIR`` because the ``.`` specifies its parent, which is ``MYDIR``. Similarly, the ``..`` entry specifies the parent directory of the parent directory of itself. An example of this would be ``MYDIR/SUBDIR/..``, which would of course be the same as ``MYDIR``.

The ``cp`` Command

The ``cp`` command is used to copy files from the fatback environment out to the host file system. It has the following syntax:

```
cp OPTIONS FILES TO-DIRECTORY
```

FILES can be specified as any number of file names, or patterns. Patterns are used to specify many files at once by using special sequences of characters. The most commonly used patterns are ``*``, ``?``, and ``[]``. The ``*`` character is used to specify zero or more characters of any kind, ``?`` specifies one character of any kind, and ``[]`` specifies a single character of a specific set.

Patterns

.....

When used by its self, the ``*`` character will match all files in a directory. For example the following command would copy all the files in the current directory to the ``/mnt/data`` directory in the hosts file system:

```
cp * /mnt/data
```

The ``*`` character can also be used in conjunction with other. For example, the following command will copy all files that end in ``.exe`` to the ``/mnt/data`` directory:

```
cp *.exe /mnt/data
```

Here is an example of using the ``?`` character to copy all the files in the ``SETUP`` directory that have a single character for an extension to the ``/mnt/data`` directory:

```
cp SETUP/*.? /mnt/data
```

The ``[]`` pattern is a bit more complex than the previous examples. Between the left and right bracket is where a specific set of matching characters is specified. For example, the pattern ``[abc]`` would match the letter ``a``, ``b``, or ``c``. Ranges or characters can also be specified using the ``-`` character in between two other characters. Using this syntax, all the letters in the alphabet can be specified using the pattern ``[a-z]``.

Patterns can be combined for even greater power. If you copy all the files in the current directory that begin with a number and end with the extension ``.dat`` to the ``/mnt/data`` directory, the following command could be used:

```
cp [0-9]*.dat /mnt/data
```

For more information on the syntax of the patterns, consult your systems man pages under `globs(7)`.

``cp`` command options

.....

The ``cp'` command accepts two options, ``-d'` and ``-R'`. The ``-d'` option tells ``cp'` to only copy files that are deleted, and skip over active file entries. The ``-R'` option makes the command recurse down any sub directories it finds. To undelete all the files in a partition to the ``/mnt/data'` directory, use the following command:

```
cp -d -R /* /mnt/data
```

The ``ls'` Command

The ``ls'` is used to display entries in a directory. The syntax for ``ls'` is as follows:

```
ls DIRECTORY
```

The entries in the specified DIRECTORY are displayed. If no DIRECTORY is specified, entries in the current directory are displayed. Multiple directories can also be displayed at the same time by specifying more than one directory, or by using a pattern.

The ``stat'` Command

The ``stat'` command displays detailed information about a directory entry. This information includes all information displayed with ``ls'`, plus additional information such as the cluster chain, and creation date. The ``stat'` command has the following syntax:

```
stat FILES
```

The ``chain'` Command

The ``chain'` command displays the cluster chain of a given directory entry or entries. The syntax for the ``chain'` is:

```
chain FILES
```

The output of running the ``chain'` command will be a series of numbers. Each number represents a cluster in the FAT table that the entry occupies.

The ``cpchain'` Command

The ``cpchain'` command writes the data in a cluster chain out to a file. It's syntax is as follows:

```
cpchain CHAIN TO-FILE
```

CHAIN is a number value of the starting cluster of the cluster chain to be written out. TO-FILE is where fatback will store the data in the host file system.

The ``sh'` Command

The ``sh'` command executes a command in the outside environment. It's syntax is simply the command ``sh'` followed by any commands that you would normally execute at a shell prompt. This can be convenient if, for example, you accidentally ran Fatback before you mounted the file system where you intended to place the files you are going to undelete to. In this case, you could execute the mount command within a ``sh'` command like this:

```
sh mount /dev/ad0s1 /mnt/extra-hd
```

On a more advanced note, the ``sh'` is implemented with improved signal handling which is not present in the standard UNIX ``system()'` function. This makes it possible to run even dangerous processes without the risk of crashing the parent process (fatback in this case). In other words, fear not the ``sh'` command, for it will only bring good fortune to thee.

The ``set'` Command

The ``set'` command is used to set run-time variables as well as modify the current FAT table. To set run-time variables, use the following syntax:

```
set VARNAME=VALUE
```

The FAT table can be modified by using the following syntax:

```
set CLUSTER-NUMBER=VALUE
```

CLUSTER-NUMBER represents an entry in the FAT table and VALUE is the cluster that that entry points to. When a FAT table entry is modified with ``set'`, the changes are not purely temporary and memory resident only.

If the command ``set'` is run with no arguments, then it will display a list of the run-time variables and their associated values.

The ``done'` Command

If the input to Fatback is a partitioned drive, then executing the ``done'` command will cause Fatback to finish editing the current partition and return to the partition menu. Otherwise, if the input is a single partition, executing the ``done'` command will cause Fatback to terminate.

The ``quit'` Command

Unlike the ``done'` command, executing the ``quit'` will cause Fatback to terminate regardless of whether the input is only a partition or multiple partitions.

Run-time Variables

=====

Fatback provides run-time variables as a way of dynamically configuring the behavior of its execution during run-time. Variables are set and viewed with the ``set'` command.

Here is a list of the run-time variables in Fatback version 1.3:

``verbose'`
The variable that determines whether or not to display extra information to the screen.

``sectsize'`
The sector size for fatback to use when making calculations. This defaults to 512, but if an input drive uses a different size and Fatback does not detect it properly, then set this by hand. This variable can also be set via the command line using the ``-z'` or ``--sectsize'` option.

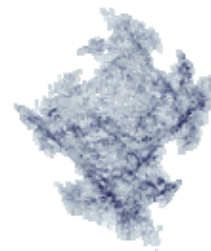
``prompt'`
The string that Fatback uses to prompt the user. This is set by default to ``fatback>'`. This probably will be of little interest to most end users, however it is important to note for someone who, for example, plans to write custom automation scripts using Expect(1).

``showall'`
The variable that determines whether or not to display non-deleted files when the ``ls'` command is executed. This variable can be set to either ``on'` or ``off'`. If it set to ``on'` then all files will be displayed with the ``ls'` command. Otherwise, if it is set to ``off'` then only deleted files will be displayed.

``deleted_prefix'`
The string that Fatback uses as the first part of the name of deleted files. The default value is `'?'`.

faust.pl: Analyze elf binaries and bash scripts.

Developed by Frederic Raynal. Available from <http://www.security-labs.org/index.php3?page=faust>



The following is taken from <http://www.security-labs.org/index.php3?page=faust>

faust is a perl script that helps to analyze files found after an intrusion or the compromising of a honeypot. Its goal is not to make the analysis, but to extract the pieces of information that you will use afterward in your analysis.

Elf analysis

- General information: MD5, type, stat, header, dynamic libraries.
- Elf sections: select the Elf sections you want to look in, and how you want to display them (asm code or strings for instance).
- Symbols: if the binary is not stripped, symbols are extracted and sorted by categories.
- strings: all strings you can extract using the string (take care that you get more strings by looking directly in some sections).
- live analysis (risky): select the mode you want (cmd or trace) to run the analyzed program and get the associated information.
-

Bash Scripts

- General information: MD5, type.
- Texts: comments in the script, and echoed messages.
- Commands: by default cp, mv, ftp, wget and mail are displayed.
- Directories: access to /etc, /dev and /home are reported.
- cross references: for each line matching one of the above categories, faust keeps track of where it belongs to.

The analysis of a binary is composed of 2 parts: dead and live analysis

A "dead" analysis focuses on information contained in the binary itself. It can come from several places, depending on the binary format, and the programmer. For instance, I retrieve in the text displayed by a slightly modified exploit the url of a web site of someone probably related to the intruder. Then I got pictures of him, and some of his (girl)friends among other tools! The names of the functions of some global variables are also very instructive if the code is related to something known. Unfortunately, if this binary is new, then you can't afford to perform a real reverse-engineering work.

A "live" analysis looks at what does the program by running it. You immediately understand how dangerous it can be: imagine there is a malware embedded in the binary, or a hidden instruction ("rm -rf /" even as a non root user is rather destructive). So, by default, this analysis is not done by faust ... but it can do it in a very simplistic way.

Usage:

```
faust.pl [-c configuration file] [-q quoted line] <file1 file2 ...>
```

Example (check the local ls executable):

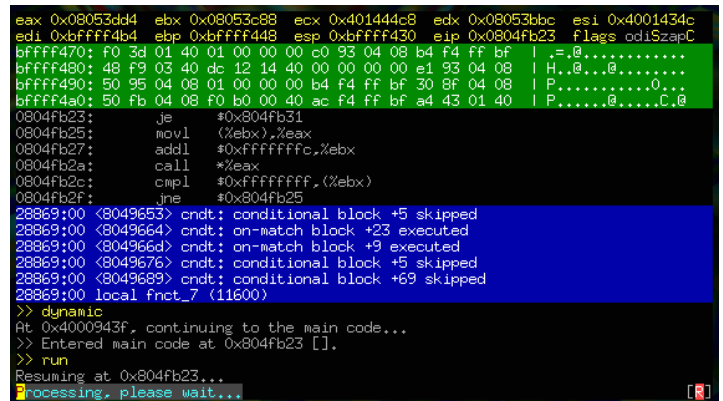
```
faust.pl -c /usr/local/bin/faust.conf ls
```

fenris: debugging, tracing, decompiling.

Developed by Michal Zalewski. Available from

http://www.bindview.com/Services/RAZOR/Utilities/Unix_Linux/fenris_index.cfm

Fenris is a multipurpose tracer, GUI debugger, stateful analyzer and partial decompiler intended to simplify bug tracking, security audits, code, algorithm, protocol analysis and computer forensics - providing a structural program trace, interactive debugging capabilities, general information about internal constructions, execution path, memory operations, I/O, conditional expressions and much more. Because it does not require sources or any particular compilation method, this multi-component project can be very helpful for black-box tests and evaluations - but it will also be a great tool for open-source project audits, as an unmatched real-time reconnaissance tool - especially when sources are too complex or too badly written to be analyzed by hand in a reliable way and reasonable time. Fenris does not rely on GNU libbfd for any critical tasks, and because of that, it is possible and feasible to trace and analyze binaries modified to fool debuggers, crypted, or otherwise tweaked. Fenris components also support other, independent debuggers or disassemblers, thanks to its capabilities to reconstruct symbol tables for stripped, static binaries with no debugging or symbol information whatsoever. (Zalewski, 2002)

The screenshot shows the Fenris debugger's main window. The top pane displays a disassembly of code, with registers (eax, ebx, ecx, edx, esi) and memory addresses (e.g., 0x08053dd4, 0xbffff4b4) visible. The bottom pane shows a control flow graph (CFG) with nodes representing code blocks and edges representing control flow. The graph includes labels like 'conditional block *5 skipped' and 'on-match block *23 executed'. The interface is dark-themed with green and blue highlights.

This project is not intended to find problems, bugs or security vulnerabilities automatically. It is supposed to be a reliable, useful tool that works in real world and can deliver valuable information which can be used to detect known problems, but also to spot unique or not so obvious dynamic conditions. (Zalewski, 2002)

Among many other features, Fenris is able to perform traditional, instruction by instruction or breakpoint to breakpoint interactive debugging enhanced by additional structural data about the code delivered to the user; it is able to fingerprint functions in static binaries, reconstruct symbol tables in ELF files based on that information, automatically detect common library code; able to deliver text-based and graphical, browsable output that documents different aspects of program activity on different abstraction layers; able to perform partial analysis of single structural blocks. It is designed to make things easier, filling the gap between existing code analysis and debugging tools - but not to replace all of them. (Zalewski, 2002)

Fenris is an amazing power, and complex tool.

The following was taken from <http://lcamtuf.coredump.cx/fenris/README>

Starting Fenris

```
fenris [ -E PAR=VAL ] [ -u user ] [ -o file ] [ -L dbase ] [ -R a:b ]  
      [ -t nnn ] [ -P ip:off:val ] [ -sdyiCSfFmGxpAeq ] program  
      [ params... ]
```

Mandatory parameter is program name, eventually followed by program parameters. If, for some reason, program name has to start with '-',

it should be preceded with '--' parameter. Before program name, you can place one or more optional parameters, such as:

`-o filename`

This options writes results to file instead of stderr. It is faster and recommended in all cases.

`-E PAR=VAL`

Puts PAR in the environment. This is especially useful if you want to trace a program with unusual LD_PRELOAD or other settings that would affect the functionality of 'fenris' itself if modified earlier. Multiple -E options are allowed.

`-u user`

Run as user. This option is available for root (see section 0x04, security issues), and will cause program to effectively run with uids, gids and supplementary groups of given user.

`-R a:b`

This option traces code from the moment when eip reaches point a to the moment when it reaches b. Incomplete range can be provided - for example, `-R :0x12345678` will trace code from the beginning to eip 0x12345678, and `-R 0x12345678:` will start tracing at 0x12345678 and continue as long as possible. NOTE: think of it as trigger points, not a continous range. For example, if you use `-R 0x12345678:`, but eip 0x12345678 is never reached, even if 0x23456789 is being executed, trace will never start.

This option is wonderful for starting trace at certain nest level and continuing it until this execution level is exited.

`-L dbase`

Load additional (supplementary) fingerprints database. Multiple -L options allowed. If filename does not contain slashes, Fenris will look for it in directories described later in this section.

`-t nnn`

Main function is nnn rets from `_do_global_ctors_aux`. By default, this is set to 2, and does not have to be changed unless something is really wrong. You should use this option if you see that trace ends with '...return from main' almost immediately at the beginning (try increasing -t parameter) or somewhere in the middle or does not reach main at all (try decreasing). However, this should not happen, in general. The only case I'm aware of are HMM 3 binaries (patchlevel 1.3.1a, does not affect 1.3), they require -t 3 instead.

`-X seg`

Use this segment prefix instead of the default (determined for a typical binary on your system) as a code segment. Code segment is the segment Fenris actively traces. Some ELF binaries can be altered to start in a different segment - a good example is a burneye ELF crypting tool. Code segment address is used by Fenris for some operations, such as describing parameters, handling signal handlers, function fingerprinting. While not absolutely necessary, it is wise to pass this parameter when

suitable. Pass the most significant byte of code segment starting address as this parameter (for example, if your code segment starts at 0x050a0000, use 0x05).

-P ip:off:val

This directive means: change a byte at address 'off' to 'val' when eip reaches 'ip'. If 'ip' is omitted or zero, this rule will be applied immediately to the freshly mapped binary (keep in mind that some memory regions mapped later may be not available at this moment). Read-only flag is generally overridden, and for files mapped into memory in read-only mode, a local copy of the modified page is spawned. All values passed to this parameter can be in decimal or in hex if preceeded with 0x, and multiple options are possible. Non-IP entries will be applied only once, at the beginning. All others will be applied every time a given IP is reached.

There are some additional considerations to be aware of when used in conjunction with tracing across `execve()`s - see -e option description for details.

-s

This option disables automatic prolog detection. It is not recommended, as it makes `./fenris trace` whole linking process and libc initialization. However, in rare cases when binary is compiled on odd, not supported system, this might be a solution. For long-term operations, however, it is recommended to contact the author providing his with this binary (or parts of it), so he'll be able to add support for this specific construction.

-y

Reports memory writes and reads immediately (without -y, memory access is reported per function on return).

-C

Inhibits tracing conditional expressions. This option is useful if output will be read by human, as it might decrease amount of reported information.

-S

Inhibits resolving library functions. This might effect in some speed improvement, but is generally not recommended without a good reason.

-f

Trace child processes after `fork()` or `vfork()`. Might be useful for tracing daemons and such (however it might cause some problems due to signal delivery semantics changes, see 0x07, known bugs)

-d

Do not describe function parameters. Reduces amount of generated output.

-F

Do not fingerprint functions. This option is effective for static binaries only, and will disable loading and displaying fingerprints. This is not really recommended - for stripped binaries, it makes your life more difficult, for binaries with symbols has almost no effect. However it might reduce memory usage and improve speed.

-m

Do not trace memory writes. This option reduces amount of generated output.

-i

This option disables indenting, reporting of pid and nesting level. It makes output non-structural, non-standard, but shorter. This will also break compatibility with ragnarok.

-x

This option causes Fenris to ignore 'return from main' and to continue tracing, returning to nest level 0. Generally speaking, this is not recommended at any time. If you have problems with 'return from main' appearing too early in the trace, try re-adjusting -t parameter instead. If this do not help, apparently one or more of calling or return conventions used by traced application are not supported, and you shouldn't rely on results anyway.

-p

Prefix every message with eip. Some commands report eip, some not, this might be useful for debugging, and is a must if you want to modify the code later with -P option. This option is compatible with ragnarok. Note that information is not displayed in some uniform way. For example, syscalls are displayed after return, local functions are displayed before call - so it takes some time to get the idea.

-A

Assume that all functions return some value, regardless of all other conditions. This will trigger some meaningless return values reported, but is useful if the binary is very optimized.

-q

Do not report last line of output to the debugger. This is meaningful only with -W, and makes sense when you use a multi-window debugger shell that already reports Fenris output (we're working on such a shell right now).

-G

"Go away" option. Can be used only in conjunction with -W, and it basically turns off all analysis capabilities of Fenris - from tracing nesting level, detecting function / library / system calls, thru many other capabilities. It is useful for troublesome non-C code. Fenris output will be practically completely disabled, and only some debugging messages will be supported (such as single-step, getmem, address breakpoint, etc).

-e

Trace new code loaded by `execve()`. This option might be convenient in some cases, but should be used with caution. Also, be warned that `-P` option will be global and apply to both old and new image in memory, except for no-IP entries that would be applied only once.

For more information on computer forensics applications, you may want to visit <http://lcamtuf.coredump.cx/fenris/reverse.txt>, where I tried to give few hints on approaching May 2002 reverse engineering challenge from Project Honeynet.

Managing fingerprints database

Managing fingerprints database is relatively simple. First of all, Fenris looks for a database in the following places:

```
./fnprints.dat
$HOME/.fenris/fnprints.dat
/etc/fnprints.dat
$HOME/fnprints.dat
```

Additionally, custom fingerprints database can be specified by `-L` option (multiple databases allowed). Same search logic applies to `-L` parameters, unless they contain path components ('/'). This is reasonable to maintain separate fingerprint databases, as it allows you to be selective. For example, if you are about to trace 'sash', you can be pretty sure it won't use `libX*` libraries, so first, you can make lookups faster, and then, you minimize eventual false positives or confusion caused by identifying some functions incorrectly. As an example, I provide fingerprints for pretty old, but still used `glibc 2.0.7` in `support/fn-2.0.7.dat`, and fingerprints for `libc5` (`support/fn-libc5.dat`). Note that, as for today, Fenris will probably not work on `libc5` systems (I have to port it), but this can be used against statically linked binaries taken from such systems.

The main database shipped with Fenris right now is a composite database for all major libraries for `x86 libc 2.1.x` and `2.2.x` generated by `gcc 2.9x` to `3.1`. It is pretty huge, but also versatile. If you believe it makes sense to maintain smaller libraries, feel free to do it and send me your selection!

Fingerprints database is a plain text file in the following format:

```
[debug info] function_name MD5_SIGN
```

Where 'debug info' is used by 'fprints' utility to indicate the source (filename+offset) of given symbol, `function_name` is self-explanatory, and `MD5_SIGN` is 8-digit hexadecimal MD5 shortcut for given function (see section 0x05, tracing mechanism for more details on hashing algorithm).

'fprints' utility accepts any ELF file (executable, shared library or relocatable `.o` file / `.a` archive) as a parameter and generates signatures for all functions. It does not really make any sense to grab signatures from shared libraries, as they are not used to build static binaries, so you should target `.o` files instead. However, it is possible and sometimes reasonable to gather signatures from ELF executables. It allows you to fingerprint some frequently used functions (e.g. `__non_dynamic_init` or some custom common code used by others; let's say Loki uses some common engine for all their games, you can easily index functions in this engine once and benefit from automated recognition later). Typical output looks like that:

```
[printf.o+52] printf CC6E587C
[printf.o+52] _IO_printf CC6E587C
--> printf.o: done (2 functions)
```

As you see, one of entries is just an alias.

Selected 'fprints' results can be appended to fnprints.dat file of your choice. It is important to mention that many libraries have multiple entries for the same function, so 'fprints' shouldn't be really used to gather fingerprints for large .a archives, like libc. This task can be accomplished by invoking 'getfprints' utility, which is a shell script wrapper around fprints. It can process whole .a archive or even multiple archives at once, eliminate dupes, and such. Please note that it is perfectly possible to copy .a files from a system that is not directly supported by Fenris, for example, libc5 box, and extract signatures on a different system.

When invoked with no parameters, 'getfprints' will extract default set of symbols from:

```
/usr/lib/libc.a
/usr/lib/libm.a
/usr/lib/libdl.a
/usr/lib/libresolv.a
/usr/lib/libreadline.a
/usr/lib/libtermcap.a
/usr/lib/libssl.a
/usr/lib/libBrokenLocale.a
/usr/lib/libcrypt.a
(one static binary)
```

This is the way it is invoked by ./build script, and can be used at any time to restore defaults or to update signatures (for new libc version, for example). If invoked with one parameter, 'getfprints' will go thru this .a file or set of .a files. An example would be:

```
./getfprints "/usr/lib/libcrypto.a /usr/lib/libmd5.a"
```

It is important to quote the list so it effectively makes one parameter. Otherwise, only first file will be processed. Call it laziness on my end ;-)

Default output file for 'getfprints' is NEW-fnprints.dat in current directory. When integrating it with existing fnprints.dat, please make sure you eliminate dupes by issuing the following command:

```
cat NEW-fnprints.dat fnprints.dat | sort | uniq >clean-new.dat
```

This utility requires ./fprints to be in current directory or in your path.

Another tool provided with the project is called 'dress', roughly an opposite to 'strip'. It will accept a stripped static ELF binary as a parameter, and will try to detect library functions. Detected names will be placed in the symbol table and a new ELF file will be generated. Usage is rather simple:

```
./dress input_elf - this will dump symbols to stdout
./dress input_elf output_elf - this will create a new ELF with symbols
```

Additional options:

```
-F nnn          - use this file for fingerprint database
-S xxx          - use this name as a code section (override .text)
```

Note that symbols generated are not GDB debugging info. In other words, you can view them with `nm`, `objdump`, they will be shown in `gdb` disassembly, but you might have problems setting an explicit breakpoint such as `"break printf"`. Blame GDB. As a workaround, you can run `dress` without a second parameter once again, and grab interesting addresses from the output. Enjoy.

Note that `'dress'` has nothing to do with `'unstrip'`, which is used to, quote, "replace the symbol table in dynamically linked executables".

Aegir, the interactive debugger

The last component discussed here is Aegir, the interactive debugger. For information for programmers, please refer to `doc/debug-api.txt`. This brief write-up should help you with developing modules for Aegir or even replacing it with your own debugging shell in easy way.

Whole interactive debugging functionality in Fenris is designed to provide instruction-by-instruction, breakpoint-to-breakpoint and watchpoint-to-watchpoint capabilities within the local code. This means that while it is possible to set up a breakpoint in library code, it is not really possible to walk thru library functions instruction by instruction. This is done for your own good, I doubt you really want to debug `libc` with Fenris. Fenris does not trace nesting level, function calls and so on within `libc`, so your possibilities are very limited anyway. Keep in mind that Fenris is an executable tracker, not a library debugger, and will treat library space pretty much like kernel space - a black hole. We are not trying to understand library functions, they are documented and predictable (note: this, obviously, won't be true for suspected code loaded as a shared library; Fenris will support such code in the future).

Right now, I'm going to focus on front-end functionality. Running Aegir is very straightforward, as all parameters are controlled by whatever you passed to Fenris, and the only parameter you have to pass is the path you gave to Fenris using `-W` option earlier. Fenris **MUST** be already running with `-W` option to launch Aegir. Aegir will shut down as soon as Fenris exits. Aegir provides some basic `gdb`-alike functionality, but also several more interesting features. In its current version, it also lacks several features, such as support for symbolic names in many functions, which can be a minor annoyance and should be fixed in 0.07.

The GUI version of Aegir, `nc-aegir`, works basically the same way, but provides an organized debugging screen with register, memory and code views, integrated Fenris output view, and automatic control over Fenris parameters. `nc-aegir` integrates Fenris session with the debugger, and it uses either `'screen'` utility (when running on a text terminal) or `xterm` session (when running under X Window system) to provide comfortable multi-view debugging environment. The GUI is documented by a self-explanatory help available after pressing `Alt-H`, so I will not cover it extensively here.

Please note that there is a fundamental difference between how Aegir / `nc-aegir` and `gdb` handle interruptions. If you hit `Ctrl+C` in Aegir or `nc-aegir`, or issue a `"stop"` command, it will not stop immediately if the

process is in the middle of a blocking call. It will schedule stop for as soon as the control returns to userspace. This is to avoid problems with interrupted syscalls, gdb style. To terminate the program immediately, hit Ctrl+C again, or use "halt" command. You will be then instructed whether the syscall will resume upon continuation or not, and what to do to avoid problems.

Once Aegir is running, you should have access to its internal help, and all messages are rather user-friendly. The following list of commands is provided for more detailed reference:

- dynamic

This is probably the first command to issue for a standard dynamically linked binary. Fenris stops at first instruction, which, for dynamic executables, would be the linker. To skip whole linking process and libc prolog, simply type "dyn" and wait a while. Of course, nothing stops you from walking thru the linking process and libc entry, but in most standard applications, it is pointless. On the other hand, it can happen that ELF is tweaked to hide some code in this phase, before "main" is reached, so this feature is not automatic in interactive debugging mode.

On some system, execution will stop at the very end of libc intro, and additional "ret" might be necessary.

- disass [x [len]]

Called without parameters, will provide a disassembly of the instruction at current eip. Called with one parameter, will disassemble one instruction at any given address. With two parameters, will disassemble "len" bytes starting at address x. Disassembly uses something that should match AT&T assembler notation, and all direct addresses are associated with their symbolic names, if any found. Note that "disass", like most other commands, does not understand symbolic names passed instead of 'x'. In other words, you can't just type "disass function_foo", this is a limitation of current Aegir implementation. On the other hand, you can use "info" directive to look up an address for a given name.

Note that this command called with a single parameter in nc-aegir will change the view in code window instead of disassembling to the console.

- regs

Shows general purpose registers. Note that Fenris does not really support floating point commands in any way (thanks to its internal disassembler), and I decided not to include fp registers in Aegir for now. Most of registers are displayed in hex and decimal; eflags are displayed as hex and octal.

- back

Displays stack backtrace - calls history. Note that what happens in libc is not covered here. If you set up a breakpoint on syscall "nanosleep", and this syscall is called from a library function called from another library function, all you'll see in stack backtrace will lead to the point where first library function was called. Backtrace includes stack address range that belongs to this function and other information, such as from where was it called. This capability is not affected by -fomit-frame-pointer,

or any other options that can confuse gdb.

- cur

Displays last output from Fenris. "Last output" stands for last "output entity", which typically means last sequence of output caused by some code construction. One instruction, such as RET, can result in multiple lines being written by Fenris. All of them are considered a single entity. But if next instruction generates another line, this line is considered a new entity.

Fenris generally reports back to Aegir the last entity produced before a running process reached a breakpoint. "Cur" will return this entity until the process is continued and stopped once again, and any message from Fenris was generated in between. This mechanism is a bit complex, but works pretty well. You probably don't want to get all lines from Fenris on your debugging console, but perhaps would appreciate knowing where you stopped.

- info x

Displays the information associated with name or address x. First, if x is non-numeric, the address associated with 'x' is resolved, then, additional information about this address is obtained. This additional information is what Fenris knows about the address - associated name, first sight, last modification, size.

- fdinfo x

Displays what Fenris knows about file descriptor x. This typically includes associated file / socket, and first sight record.

- break x

Sets a breakpoint at address x.

Note: breakpoints are ignored inside libc, except for ones being set at the beginning of a libcall. This way, you can breakpoint at 'printf', but it is generally pointless to set a breakpoint at, say, printf+10. In this situation, "step" will continue until library code is left, all other commands will affect the code that called this library function, not the function itself (so "down" would continue until underlying local function returns, and so on). If you can avoid it, don't set breakpoints inside libc :-). If you do, try to do little more than "step" to get back to where it was called.

- sbreak x

Sets a breakpoint on syscall x (x can be either numeric value or symbolic name). Breakpoint trap will be generated when this particular syscall is called. If this breakpoint trap is generated within libc, special rules mentioned above (for "break") apply.

The best use for this type of breakpoint is to hook clone, vfork, and fork, so you can always react on them before they are being executed. Aegir, in its current form, is capable of tracing only one process at once, so you probably want to overwrite fork()s and move a desired value to %eax to choose one branch or another.

- ibreak x

Sets a breakpoint on signal x (x can be either numeric value or symbolic name). Breakpoint trap will be generated when this particular signal is delivered. Note that default action for Fenris is not to stop on any signals unless you want to, which is different from gdb. Signals can be delivered anywhere, and special rules for libc code apply.

- rwatch start end

Sets a watchpoint on read access to the memory area start-end. Breakpoint trap will be generated if any known syscall, known library function or any local code is trying to access the memory. Fenris does not trace inside libcalls, so unknown libcalls accessing memory will be not reported (if the pointer is passed as a parameter and is auto-detected, it will be considered "read" anyway).

This, generally speaking, can be a problem. There's only one reasonable way to solve it, that is, implementing more libcalls in Fenris.

Another important issue is that when a parameter is passed to some library function, this is reported as a read of first four bytes of the parameter. Please consider library read and write reporting only a hint - Fenris does not physically trace this code, and makes certain assumptions. For many functions, it is impossible to determine how much data will be actually read or written (think "scanf", for example), and Fenris does not try to make up numbers.

- wwatch start end

Sets a watchpoint on write access to the memory area start-end. Breakpoint trap will be generated if any known syscall, known library function or any local code is trying to write the memory. Fenris does not trace inside libcalls, so unknown libcalls writing memory will be not reported (unless, for example, it is modified directly by a syscall called from this libcall or such).

This, generally speaking, can be a problem. There's only one reasonable way to solve it, that is, implementing more libcalls in Fenris.

Another important issue is that when a parameter is passed to some library function, this is reported as a write of first four bytes of the parameter. Please consider library read and write reporting only a hint - Fenris does not physically trace this code, and makes certain assumptions. For many functions, it is impossible to determine how much data will be actually read or written (think "scanf", for example), and Fenris does not try to make up numbers.

- step [x]

Make one or x single steps over the code; note that libcall functions are considered a single step. See notes for "break".

- ret [x]

Continue to next or x-th RET in the code. Note that this command will ignore library code.

- libc

Continue to next libcall. Note that libcalls called from libcalls are ignored by Fenris.

- sys

Continue to next syscall. You can end up in library code, see notes for "break".

- call

Continue to next local function call.

- down

Continue until the code leaves the current function. This is different from "ret", as ret can occur in a function called from current function before current function itself reaches RET. This command uses Fenris nest level tracing capabilities to stop the program.

- next

Continue to next output entity from Fenris. This is useful for line-by-line debugging, and is different from "step".

- run

Continue execution until next breakpoint (or until the program exits). Once again, keep in mind that signals do not interrupt the program unless you used "ibreak".

- stop

Stop program as soon as possible. This is typically done as soon as control returns to userspace ("stop" will not abort blocking syscalls).

- halt

Stop program NOW. This will return from blocking syscalls aborting them (and can cause problems, just like Ctrl-C in gdb).

- fprint x

Fingerprint code at address x. Fenris returns a signature and matching names for the function at this address. This is done on Fenris side so it is matched against currently loaded fingerprint database, ensuring that results are coherent with automatic fingerprinting.

- x y [z]

Displays memory at address 'y' as a hexdump. If no third parameter is given, first 16 bytes are displayed, otherwise, z is used to specify length. The format is pretty simple: address, 16 bytes as hex, and 16 printable characters per line.

Note that this command called with a single parameter in nc-aegir will change the view in data window instead of displaying on the console.

- y x

Displays a string (more precisely, its first 128 bytes), from address x.

- setreg nnn y

Sets general purpose register 'nnn' to value y. Not all registers can be set with ptrace().

- setmem x y

Sets memory byte at address x to value y.

- list

Lists all watchpoints and breakpoints, and their ID numbers.

- del x

Deletes a breakpoint or watchpoint with ID x.

- memmap [not implemented in 0.04b]

Displays memory map - all objects that are known, along with the information about them.

- fdmap [not implemented in 0.04b]

Displays all known file descriptors with short descriptions.

- fnmap [not implemented in 0.04b]

Displays all known local functions.

- signals

Displays handlers for all signals.

- load xxx

Loads a module "xxx". Modules can be used to implement custom functionality in Aegir, see doc/debug-api.txt for more information.

- exec xxx

Execute a shell command 'xxx'.

- log [x]

A command available only in nc-aegir. Because Fenris output is tunneled directly to one of nc-aegir windows, if you want to create a copy of this data, you have to use this command. To start logging to a new file, type "log /path/to/log". To stop logging, type "log" with no parameters.

- help

Get help.

- quit yes

Terminate the session (can be abbreviated as 'q y' for convenience).

All commands can be abbreviated as long as they are not ambiguous. There is no step by step introduction to using Aegir, because it is assumed that its users will have some background with gdb, assembly language, and debugging in general, and I believe that the above command reference and a sample "demo session" discussed earlier are more than enough to get started.

foremost: Carve files based on header and footer.

Developed by Jesse Kornblum and Kris Kendall from the United States Air Force Office of Special Investigations and The Center for Information Systems Security Studies and Research. Available from <http://foremost.sourceforge.net/>

Foremost is a console program to recover files based on their headers, footers, and internal data structures. This process is commonly referred to as data carving. Foremost can work on image files, such as those generated by dd, Safeback, Encase, etc, or directly on a drive. The headers and footers can be specified by a configuration file or you can use command line switches to specify built-in file types. These built-in types look at the data structures of a given file format allowing for a more reliable and faster recovery (Kornblum, 2006).

The man page presented below is from <http://foremost.sourceforge.net/foremost.html>

NAME

foremost - Recover files using their headers, footers, and data structures

SYNOPSIS

foremost[-h][-V][-d][-vqwQT][-b<blocksize>][-o<dir>] [-t<type>][-s<num>][-i<file>]

BUILTIN FORMATS

Recover files from a disk image based on file types specified by the user using the -t switch.

jpg	Support for the JFIF and Exif formats including implementations used in modern digital cameras.
gif	
png	
bmp	Support for windows bmp format.
avi	
mpg	Support for most MPEG's (must begin with 0x000001BA)
exe	Windows PE executables (also extracts compile time to audit file)
rar	
wav	
riff	This will extract AVI and RIFF since they use the same file format (RIFF). note faster than running each separately.
wmv	Note may also extract -wma files as they have similar format.
mov	
pdf	
ole	This will grab any file using the OLE file structure. This includes PowerPoint, Word, Excel, Access, and StarWriter
doc	Note it is more efficient to run OLE as you get more bang for your buck. If you wish to ignore all other ole files then use this.
Zip	Note is will extract .jar files as well because they use a similar format. Open Office docs are just zipped XML files so they are extracted as well. These include SXW, SXC, SXI, and SX? for undetermined OpenOffice files.
htm	
cpp	C source code detection, note this is primitive and may generate documents other than C code.

all Run all pre-defined extraction methods. [Default if no **-t** is specified]

DESCRIPTION

Recover files from a disk image based on headers and footers specified by the user.

- h** Show a help screen and exit.
- V** Show copyright information and exit.
- d** Turn on indirect block detection, this works well for Unix file systems.
- T** Time stamp the output directory so you don't have to delete the output dir when running multiple times.
- v** Enables verbose mode. This causes more information regarding the current state of the program to be displayed on the screen, and is highly recommended.
- q** Enables quick mode. In quick mode, only the start of each sector is searched for matching headers. That is, the header is searched only up to the length of the longest header. The rest of the sector, usually about 500 bytes, is ignored. This mode makes foremost run considerably faster, but it may cause you to miss files that are embedded in other files. For example, using quick mode you will not be able to find JPEG images embedded in Microsoft Word documents. Quick mode should not be used when examining NTFS file systems. Because NTFS will store small files inside the Master File Table, these files will be missed during quick mode.
- Q** Enables Quiet mode. Most error messages will be suppressed.
- w** Enables write audit only mode. No files will be extracted.
- a** Enables write all headers, perform no error detection in terms of corrupted files.
- b number** Allows you to specify the block size used in foremost. This is relevant for file naming and quick searches. The default is 512. ie. `foremost -b 1024 image.dd`
- k number** Allows you to specify the chunk size used in foremost. This can improve speed if you have enough RAM to fit the image in. It reduces the checking that occurs between chunks of the buffer. For example if you had > 500MB of RAM. ie. `foremost -k 500 image.dd`
- i file** The *file* is used as the input file. If no input file is specified or the input file cannot be read then stdin is used.
- o directory** Recovered files are written to the directory *directory*.
- c file** Sets the configuration file to use. If none is specified, the file "foremost.conf" from the current directory is used, if that doesn't exist then "/etc/foremost.conf" is used. The format for the configuration file is described in the default configuration file included with this program. See the *CONFIGURATION FILE* section below for more information.
- s number** Skips *number* blocks in the input file before beginning the search for headers. ie. `foremost -s 512 -t jpeg -i /dev/hda1`

CONFIGURATION FILE

The configuration file is used to control what types of files foremost searches for. A sample configuration file, `foremost.conf`, is included with this distribution. For each file type, the configuration file describes the file's extension, whether the header and footer are case sensitive, the maximum file size, and the header and footer for the file. The footer field is optional, but header, size, case sensitivity, and extension are not! Any line that begins with a pound sign is considered a comment and ignored. Thus, to skip a file type just put a pound sign at the beginning of that line.

Headers and footers are decoded before use. To specify a value in hexadecimal use \x[0-f][0-f], and for octal use \[1-9][1-9][1-9]. Spaces can be represented by \s. Example: "\x4F\123\\sCCI" decodes to "OSI CCI".

To match any single character (aka a wildcard) use a ?. If you need to search for the ? character, you will need to change the wildcard line *and* every occurrence of the old wildcard character in the configuration file. Do not forget those hex and octal values! ? is equal to \x3f and \063.

EXAMPLES

Search for jpeg format skipping the first 100 blocks

```
foremost -s 100 -t jpg -i image.dd
```

Only generate an audit file, and print to the screen (verbose mode)

```
foremost -av image.dd
```

Search all defined types

```
foremost -t all -i image.dd
```

Search for gif and pdf

```
foremost -t gif,pdf -i image.dd
```

Search for office documents and jpeg files in a Unix file system in verbose mode.

```
foremost -v -t ole,jpeg -i image.dd
```

Run the default case

```
foremost image.dd
```

ftimes: A toolset for forensic data acquisition.

Developed by Klayton Monroe. Available from
<http://ftimes.sourceforge.net/FTimes/>



The following is from <http://ftimes.sourceforge.net/FTimes/>

FTimes is a system baselining and evidence collection tool. The primary purpose of FTimes is to gather and/or develop information about specified directories and files in a manner conducive to intrusion analysis.

FTimes is a lightweight tool in the sense that it doesn't need to be "installed" on a given system to work on that system, it is small enough to fit on a single floppy, and it provides only a command line interface.

Preserving records of all activity that occurs during a snapshot is important for intrusion analysis and evidence admissibility. For this reason, FTimes was designed to log four types of information: configuration settings, progress indicators, metrics, and errors. Output produced by FTimes is delimited text, and therefore, is easily assimilated by a wide variety of existing tools.

FTimes basically implements two general capabilities: file topography and string search. File topography is the process of mapping key attributes of directories and files on a given file system. String search is the process of digging through directories and files on a given file system while looking for a specific sequence of bytes. Respectively, these capabilities are referred to as map mode and dig mode.

FTimes supports two operating environments: workbench and client-server. In the workbench environment, the operator uses FTimes to do things such as examine evidence (e.g., a disk image or files from a compromised system), analyze snapshots for change, search for files that have specific attributes, verify file integrity, and so on. In the client-server environment, the focus shifts from what the operator can do locally to how the operator can efficiently monitor, manage, and aggregate snapshot data for many hosts. In the client-server environment, the primary goal is to move collected data from the host to a centralized system, known as an Integrity Server, in a secure and authenticated fashion. An Integrity Server is a hardened system that has been configured to handle FTimes GET, PING, and PUT HTTP/S requests.

The following is from <http://ftimes.sourceforge.net/FTimes/ManPage.shtml>

NAME

ftimes - A system baselining and evidence collection tool.

SYNOPSIS

ftimes --cfgtest file mode [-s]

ftimes --compare mask baseline snapshot [-l level]

ftimes --decoder snapshot [-l level]

ftimes --digauto file [-l level] [list]

ftimes --digfull file [-l level] [list]

ftimes --diglean file [-l level] [list]

ftimes --getmode file [-l level]

ftimes --mapauto mask [-l level] [list]

ftimes --mapfull file [-l level] [list]

ftimes --maplean file [-l level] [list]

ftimes --putmode file [-l level]

ftimes --version

DESCRIPTION

FTimes is a system baselining and evidence collection tool. The primary purpose of **FTimes** is to gather and/or develop information about specified directories and files in a manner conducive to intrusion analysis.

FTimes is a lightweight tool in the sense that it doesn't need to be "installed" on a given system to work on that system, it is small enough to fit on a single floppy, and it provides only a command line interface.

Preserving records of all activity that occurs during a snapshot is important for intrusion analysis and evidence admissibility. For this reason, **FTimes** was designed to log four types of information: configuration settings, progress indicators, metrics, and errors. Output produced by **FTimes** is delimited text, and therefore, is easily assimilated by a wide variety of existing tools.

FTimes basically implements two general capabilities: file topography and string search. File topography is the process of mapping key attributes of directories and files on a given file system. String search is the process of digging through directories and files on a given file system while looking for a specific sequence of bytes. Respectively, these capabilities are referred to as map mode and dig mode.

FTimes supports two operating environments: workbench and client-server. In the workbench environment, the operator uses **FTimes** to do things such as examine evidence (e.g., a disk image or files from a compromised system), analyze snapshots for change, search for files that have specific attributes, verify file integrity, and so on. In the client-server environment, the focus shifts from what the operator can do locally to how the operator can efficiently monitor, manage, and aggregate snapshot data for many hosts. In the client-server environment, the primary goal is to move collected data from the host to a centralized system, known as an Integrity Server, in a secure and authenticated fashion. An Integrity Server is a hardened system that has been configured to handle **FTimes** GET, PING, and PUT HTTP/S requests.

The **FTimes** distribution contains a script called `nph-ftimes.cgi` that may be used in conjunction with a Web server to implement a public Integrity Server interface. Deeper topics such as the construction and internal mechanics of an Integrity Server are not addressed in this document.

FTimes provides several modes of operation that either implement its basic capabilities or support them in some way. These modes are described in the MODES OF OPERATION section of this document and are outlined here:

- **cfgtest** - check config file syntax for a given file and mode
- **compare** - compare two map snapshots to detect change
- **decoder** - decode a compressed map snapshot
- **digauto** - search for strings in files using a default configuration
- **digfull** - search for strings in files using a specified configuration
- **diglean** - same as digfull except that the range of controls is limited and output can be written directly to std{err,out}
- **getmode** - download a config file from an Integrity Server
- **mapauto** - collect directory and file attributes using a default configuration
- **mapfull** - collect directory and file attributes using a specified configuration
- **maplean** - same as mapfull except that the range of controls is limited and output can be written directly to std{err,out}
- **putmode** - upload a dig or map snapshot to an Integrity Server
- **version** - display version information and exit

FTimes also has many controls which dictate how it will execute. Some modes support very few controls while others support quite a few. The following table summarizes what controls apply to each mode of operation. An 'X' indicates that the given control applies to the selected mode.

	MODES										
	c	c	d	d	d	g	m	m	m	p	v
	f	o	i	i	i	e	a	a	a	u	e
	g	m	g	g	g	t	p	p	p	t	r
	t	p	a	f	l	m	a	f	l	m	s
	e	a	u	u	e	o	u	u	e	o	i
	s	r	t	l	a	d	t	l	a	d	o
	t	e	o	l	n	e	o	l	n	e	n
===== CONTROL =====											
AnalyzeBlockSize	.	.	.	X	X	.	.	X	X	.	.
AnalyzeCarrySize	.	.	.	X	X
AnalyzeDeviceFiles	.	.	.	X	X	.	.	X	X	.	.
AnalyzeRemoteFiles	.	.	.	X	X	.	.	X	X	.	.
BaseName	.	.	.	X	X	X	.	X	X	X	.
BaseNameSuffix	.	.	.	X	X	.	.	X	X	.	.
Compress	X	X	.	.
DataType	X	.
DateTime	X	.
DigString	.	.	X	X	X
DigStringNoCase	.	.	X	X	X
DigStringNormal	.	.	X	X	X
DigStringRegExp	.	.	X	X	X
EnableRecursion	.	.	.	X	X	.	.	X	X	.	.
Exclude	.	.	.	X	X	.	.	X	X	.	.
ExcludesMustExist	.	.	.	X	X	.	.	X	X	.	.
FieldMask	.	X	X	X	X	X	.
FileSizeLimit	.	.	.	X	X	.	.	X	X	.	.
GetAndExec	X
GetFileName	X
HashDirectories	X	X	.	.
HashSymbolicLinks	X	X	.	.
Import	.	.	.	X	X	X	.	X	X	X	.
Include	.	.	.	X	X	.	.	X	X	.	.

IncludesMustExist	.	.	.	X	X	.	.	X	X	.	.
LogDir	.	.	.	X	X	.	.	X	X	.	.
LogFileName	X	.
MagicFile	X	X	.	.
MapRemoteFiles	.	.	.	X	X	.	.	X	X	.	.
MatchLimit	.	.	.	X	X
NewLine	.	.	.	X	X	.	.	X	X	.	.
OutDir	.	.	.	X	X	.	.	X	X	.	.
OutFileHash	X	.
OutFileName	X	.
RequirePrivilege	.	.	.	X	X	.	.	X	X	.	.
RunType	.	.	.	X	.	.	.	X	.	X	.
SSLBundledCAsFile	.	.	.	X	.	X	.	X	.	X	.
SSLExpectedPeerCN	.	.	.	X	.	X	.	X	.	X	.
SSLMaxChainLength	.	.	.	X	.	X	.	X	.	X	.
SSLPassPhrase	.	.	.	X	.	X	.	X	.	X	.
SSLPrivateKeyFile	.	.	.	X	.	X	.	X	.	X	.
SSLPublicCertFile	.	.	.	X	.	X	.	X	.	X	.
SSLUseCertificate	.	.	.	X	.	X	.	X	.	X	.
SSLVerifyPeerCert	.	.	.	X	.	X	.	X	.	X	.
URLAuthType	.	.	.	X	.	X	.	X	.	X	.
URLCreateConfig	.	.	.	X	.	.	.	X	.	.	.
URLGetRequest	X
URLGetURL	X
URLPassword	.	.	.	X	.	X	.	X	.	X	.
URLPutSnapshot	.	.	.	X	.	.	.	X	.	.	.
URLPutURL	.	.	.	X	.	.	.	X	.	X	.
URLUnlinkOutput	.	.	.	X	.	.	.	X	.	.	.
URLUsername	.	.	.	X	.	X	.	X	.	X	.

MODES OF OPERATION

The modes of operation described in this section are mutually exclusive. In other words, only one mode may be specified per invocation. Unless otherwise stated, the value for the **baseline**, **snapshot**, and **file** arguments may be the name of a regular file or '-'. If the latter form is given, **FTimes** expects to read the equivalent input from stdin. Note, however, that the **baseline** and **snapshot** arguments can not be '-' simultaneously. The elements and syntax rules of for all configuration files are described in the CONFIGURATION CONTROLS section of this document. The **level** option is described in the OPTIONS section of this document. The **list** option specifies one or more directories, files, or symbolic links that are to be scanned. Collectively, these items represent an **Include** list. See the **Include** control for more information.

--cfgtest {file|-} mode [-s]

Verify the syntax of a given configuration **file** in the context of a specified **mode** where **mode** can be one of: **digauto**, **digfull**, **diglean**, **getmode**, **mapfull**, **maplean**, or **putmode**. The given configuration **file** is parsed with the same methods that would be used if **FTimes** had been invoked in that particular run **mode**. By default, directories and files are not checked for existence. This allows config files to be tested in a separate environment from where they will be used. Strict testing (i.e., directories and files must exist) may be enabled with the **-s** option.

The value 'Syntax Passed' is written to stdout, if all syntax checks are satisfied. Otherwise, the value 'Syntax Failed' and a description of the failure will be written to stdout.

Note: The fact that a given file passes all syntax checks does not guarantee that its use will lead to a successful outcome. It merely ensures that specified controls are valid for a given mode, and the values for those controls meet basic syntax requirements.

--compare mask {baseline|-} {snapshot|-} [-l level]

Compare **baseline** and **snapshot** data according to the specified **mask** where mask identifies the attributes to be analyzed. Output is written to stdout and has the following format:

```
category|name|changed|unknown
```

The category field indicates what type of change has occurred. It can have one of the following values:

```
C - Changed
M - Missing
N - New
U - Unknown (i.e., one or both fields were NULL)
X - Cross (i.e., Changed and Unknown)
```

The changed field contains a comma separated list of fields that have changed. This field will be NULL if the category is New or Missing.

The unknown field contains a comma separated list of fields that could not be compared due to lack of information. This field will be NULL if the category is New or Missing.

The specified **mask** must comply with the syntax rules set forth for the **FieldMask** control.

Note: The **baseline** and **snapshot** arguments can not be '-' simultaneously.

--decoder {snapshot|-} [-l level]

Decode a compressed **snapshot**. A compressed snapshot can be created by running **FTimes** in map mode (i.e., **mapfull** or **maplean**) with **Compress** enabled. Output is written to stdout.

--digauto {file|-} [-l level] [list]

Use default configuration settings to search an **Include** list for a set of user defined strings. These strings are defined in **file** according to the syntax for the **DigStringNormal**, **DigStringNoCase**, and **DigStringRegExp** controls. If **list** is not specified, **FTimes** will search the entire system including remote shares or mount points. Any device files specifically included in the **list** will be searched (i.e., **AnalyzeDeviceFiles** is always enabled in this mode of operation). Output is written to stdout and has the following format.

```
name|type|offset|string
```

The offset field, represented as a decimal value, contains the location in the file identified by name where the specified string was found.

--digfull {file|-} [-l level] [list]

Use the configuration settings in **file** to search an **Include** list for a set of user defined strings. The **Include** list may be specified by a combination of **Include** controls and **list** arguments. If an **Include** list is not specified, **FTimes** will search the entire system. Remote shares or mount points will only be searched if **AnalyzeRemoteFiles** is enabled. Any device files specifically included in the **list** will only be searched if **AnalyzeDeviceFiles** is enabled.

--diglean {file|-} [-l level] [list]

Use the configuration settings in **file** to search an **Include** list for a set of user defined strings. The **Include** list may be specified by a combination of **Include** controls and **list** arguments. If an **Include** list is not specified, **FTimes** will search the entire system. Remote shares or mount points will only be searched if **AnalyzeRemoteFiles** is enabled. Any device files specifically included in the **list** will only be searched if **AnalyzeDeviceFiles** is enabled. The difference between this mode and **--digfull** is that fewer controls are defined/available and output can be written directly to `std{err,out}`.

--getmode {file|-}

Use the configuration settings in **file** to download **digfull**, **diglean**, **mapfull**, or **maplean** configuration information. One of three possible actions, depending on how **getmode** is configured, will take place once the download is complete:

- **FTimes** writes the downloaded information to `stdout`,
- **FTimes** writes the downloaded information to the file specified by **GetFileName**, or
- **FTimes** restarts in **digfull**, **diglean**, **mapfull**, or **maplean** using the downloaded information as its new configuration file

The first action is effected when **GetAndExec** is disabled and **GetFileName** is not specified. The second action is effected when **GetAndExec** is disabled and **GetFileName** is specified. The third action is effected when **GetAndExec** is enabled and **GetFileName** is specified.

--mapauto mask [-l level] [list]

Use default configuration settings to map an **Include** list according to the specified **mask** where **mask** identifies the attributes to be collected. If **list** is not specified, **FTimes** will map the entire system including remote shares or mount points. Any device files specifically included in the **list** will be mapped (i.e., **AnalyzeDeviceFiles** is always enabled in this mode of operation). Output is written to `stdout`, and its format depends on the value of **mask**.

The specified **mask** must comply with the syntax rules set forth for the **FieldMask** control.

--mapfull {file|-} [-l level] [list]

Use configuration settings in **file** to map an **Include** list. The **Include** list may be specified by a combination of **Include** controls and **list** arguments. If an **Include** list is not specified, **FTimes** will map the entire system. Remote shares or mount points will only be mapped if **AnalyzeRemoteFiles** is enabled. Any device files specifically included in the **list** will only be mapped if **AnalyzeDeviceFiles** is enabled.

--maplean {file|-} [-l level] [list]

Use configuration settings in **file** to map an **Include** list. The **Include** list may be specified by a combination of **Include** controls and **list** arguments. If an **Include** list is not specified, **FTimes** will map the entire system. Remote shares or mount points will only be mapped if **AnalyzeRemoteFiles** is enabled. Any device files specifically included in the **list** will only be mapped if **AnalyzeDeviceFiles** is enabled. The difference between this mode and **--mapfull** is that fewer controls are defined/available and output can be written directly to `std{err,out}`.

--putmode {file|-} [-l level]

Use the configuration settings in **file** to upload an existing snapshot to an Integrity Server configured to receive such.

--version

Display version information and exit.

OPTIONS

-l level

The **LogLevel** option controls the amount of log output. As **level** decreases, the amount of output increases. The range of values that may be assigned to **level** is stated below. In cases where evidence collection is of primary concern, **LogLevel** should be no higher than Landmark. The default **LogLevel** is Landmark.

```
6 - Critical
5 - Failure
4 - Warning
3 - Information
2 - Landmark
1 - Waypoint
0 - Debug
```

-s

Enforce strict testing. This requires that specified directories and files exist on the system running the test. Controls affected by this option include: **LogDir**, **OutDir**, **SSLPublicCertFile**, **SSLPrivateKeyFile**, and **SSLBundledCAsFile**.

CONFIGURATION CONTROLS

This section describes the various controls that **FTimes** recognizes. In general, controls either shape runtime behavior or provide information needed by the application to perform a specific function. Controls and their values, one pair/line, are written to a file having the following format.

```
<control> = <value>
```

All controls are case insensitive, but, in general, their values are not. Comments may occur anywhere on a given line, and must begin with a pound character (i.e., '#'). In any given line, all text to the right of the first comment will be ignored. White space surrounding controls and values is ignored.

CONTROL DESCRIPTIONS

This section describes each control that may be specified, defines what values it may have, and states which modes of operation recognize the control.

AnalyzeBlockSize: [1-1048576]

Applies to **digfull**, **diglean**, **mapfull**, and **maplean**.

AnalyzeBlockSize is optional. It instructs the analysis engine to use the specified block size (in bytes) when reading and processing file data. The default value for this control is 16384 (16 KB).

AnalyzeCarrySize: [1-1048576]

Applies to **digfull**, **diglean**.

AnalyzeCarrySize is optional. It instructs the analysis engine to use the specified block size (in bytes) when saving (or carrying) data from one dig operation to the next. The default value for this control is 1024 (1 KB).

Note: The value for this control must not exceed **AnalyzeBlockSize**, and it must be equal to or larger than the maximum string length for normal and case insensitive strings. If either condition is not met, the program will abort.

AnalyzeDeviceFiles: [Y|N]

Applies to **digfull**, **diglean**, **mapfull**, and **maplean**.

AnalyzeDeviceFiles is optional. When enabled ('Y' or 'y'), it instructs **FTimes** to analyze block/character device files that have been specifically included by name on the command line or through an **Include** (e.g., **Include=/dev/ad0**). Device files that reside in an included directory (e.g., **Include=/dev**) are not analyzed simply because their parent was included -- you must specifically call them out. Also, any device files that were specifically included will be pruned if their parent or any higher level directory was included too. The default value is 'N'.

Note: Analyzing block/character device files can take a very long time or forever (e.g., **/dev/zero**).

AnalyzeRemoteFiles: [Y|N]

Applies to **digfull**, **diglean**, **mapfull**, and **maplean**.

AnalyzeRemoteFiles is optional. When enabled ('Y' or 'y'), it instructs **FTimes** to ignore the fact that a given **Include** does not reside on the local system. The result is that **FTimes** will attempt to analyze remote files. The default value is 'N'.

Note: Analyzing remote file systems can create large amounts of network traffic. Just remember that you may be mapping an entire disk.

BaseName: <name|->

Applies to **digfull**, **diglean**, **getmode**, **mapfull**, **maplean**, and **putmode**.

BaseName is required. It specifies the name prefix that will be attached to the various output files. It also serves as the CLIENTID parameter in GET/PING/PUT requests. The recommended name format is one that matches the following regular expression:

```
^[0-9A-Za-z_-]{1,64}$
```

This is because `nph-ftimes.cgi` uses that expression to validate the CLIENTID parameter in GET/PING/PUT requests. Typically, **BaseName** and **URLUsername** will be the same when basic authentication is enabled, but this is not a requirement.

If you're using FTimes in a lean mode, a good naming convention would be to use the hostname of the system being baselined. Also, both lean modes allow you to specify a **BaseName** value of '-'. This causes FTimes to write its output to stdout/stderr.

BaseNameSuffix: [datetime|none|pid]

Applies to **digfull**, **diglean**, **mapfull**, and **maplean**.

BaseNameSuffix is optional. It specifies the type of suffix that will be attached to the **BaseName**. If **BaseNameSuffix** is set to `datetime`, a suffix of the following format will be appended to the **BaseName**: `YYYYMMDDHHMMSS`. If it is set to `none`, no suffix will be appended, and if it is set to `pid`, the value of the current process ID will be appended. The default value is `none`.

Compress: [Y|N]

Applies to **mapfull** and **maplean**.

Compress is optional. When enabled ('Y' or 'y'), it activates a form of lossless ASCII compression. This yields a compression ratio that can be as good as three to one. The default value is 'N'.

As a side note, compressing compressed snapshots with a program like `gzip(1)` yields better compression than if `gzip(1)` was used alone on the same data in its uncompressed form.

DataType: [dig|map]

Applies to **putmode**.

DataType is required. It represents the DATATYPE parameter in PUT requests, and specifies the type of data being posted.

DateTime: <YYYYMMDDHHMMSS>

Applies to **putmode**.

DateTime is required. It represents the DATETIME parameter in PUT requests, and specifies the date/time of the snapshot being posted.

DigString: <string>

Applies to **digauto**, **digfull**, and **diglean**.

DigString is an alias for **DigStringNormal**, and it is being phased out. Please use **DigStringNormal** instead.

DigStringNoCase: <string>

Applies to **digauto**, **digfull**, and **diglean**.

DigStringNoCase is conditionally required. It specifies a case insensitive search string. This string must be URL encoded in the same manner as a normal **DigString** -- refer to that control description for the details. Internally, all alpha characters (i.e., [A-Za-z]) are converted to lower case.

DigStringNormal: <string>

Applies to **digauto**, **digfull**, and **diglean**.

DigStringNormal is conditionally required. It specifies a search string. This string must be URL encoded if it contains '%', '+', ' ', or any non-printable characters. When in doubt about whether or not a value should be encoded, encode it. To encode a character, convert its hex value according to the format %HH where H is a hex digit. Spaces may alternatively be encoded as '+'.

DigStringRegExp: <regexp>

Applies to **digauto**, **digfull**, and **diglean**.

DigStringRegExp is conditionally required. It specifies a Perl compatible regular expression. Unlike the strings specified in the **DigString** and **DigStringNoCase** controls, this string must not be URL encoded. With **DigStringRegExp** patterns, you must specify no more than one capturing '()' subpattern. You can use '(?:)' if you require additional parentheses for grouping purposes. If you do not specify a capturing subpattern, the entire match will be captured.

Note: This control is only available if PCRE support was compiled into the binary. As of version 3.5.0, PCRE support is enabled by default.

EnableRecursion: [Y|N]

Applies to **digfull**, **diglean**, **mapfull**, and **maplean**.

EnableRecursion is optional. When enabled ('Y' or 'y'), it instructs **FTimes** to recursively process directories. The default value is 'Y'.

Exclude: [directory|file|link]

Applies to **digfull**, **diglean**, **mapfull**, and **maplean**.

Exclude controls are optional, and there is no predefined limit on the number that may be specified. However, there can be only one **Exclude** control/value pair per line. It is not necessary to explicitly exclude special file systems such as PROCFS as **FTimes** will detect their presence and automatically exclude them. **Exclude** values must be specified as a fully qualified path (see **Include** control). If **ExcludesMustExist** is enabled, then each **Exclude** must reference an existing file, directory, or symbolic link. Otherwise, **FTimes** will abort.

Note: Symbolic links are not supported in WIN32-based file systems.

Note: The exclude mechanism works on an exact match basis, but it can be used to produce a recursive effect. For example, if you include '/' and exclude '/etc', then '/etc' and anything below it will not be processed. However, if you include '/etc/hosts' and exclude '/etc', then '/etc/hosts' will not be processed because the recursive effect would not be in play, and there is no **Exclude** that exactly matches it.

ExcludesMustExist: [Y|N]

Applies to **digfull**, **diglean**, **mapfull**, and **maplean**.

ExcludesMustExist is optional. When enabled ('Y' or 'y'), it instructs **FTimes** to check that every **Exclude** exists prior to mapping or digging. If this control is enabled and any **Exclude** does not exist, **FTimes** will abort. The default value is 'N'.

FieldMask: <mask>

Applies to **compare**, **mapauto**, **mapfull**, **maplean**, and **putmode**.

FieldMask is required in **compare**. Its value indicates what fields are to be compared for change.

FieldMask is required in **mapauto**, **mapfull**, and **maplean**. Its value dictates what attributes get collected or derived during a scan.

FieldMask is required in **putmode**. Its value indicates what fields are contained in the data being posted.

There can be no embedded white space in a given mask specification, and it must comply with the following case insensitive syntax:

```
ALL[<+|-><field>...]
```

or

```
NONE<+|-><field>[<+|-><field>...]
```

The following fields may be specified on Windows platforms with two caveats: (1) ctime is only available on Windows NT/2K systems and (2) altstreams is only available if the target file system is NTFS.

```
volume      - Volume serial number
findex      - File serial number
```

attributes	- File attributes
atime	- Time of last file access
mtime	- Time of last file modification
ctime	- Creation time
chtime	- Change time (undocumented)
size	- File size in bytes
altstreams	- Number of alternate or named streams
sha1	- SHA1 digest of the file's data stream
md5	- MD5 digest of the file's data stream
magic	- File type

The following fields may be specified on UNIX platforms:

dev	- Device identification number
inode	- File identification number
mode	- File attributes and permissions
nlink	- Number of hard links
uid	- User identification number
gid	- Group identification number
rdev	- Device type (contains major/minor numbers)
atime	- Time of last file access
mtime	- Time of last file modification
ctime	- Time of last file status change
size	- File size in bytes
sha1	- SHA1 digest of the file's data stream
md5	- MD5 digest of the file's data stream
magic	- File type

FileSizeLimit: <integer>

Applies to **digfull**, **diglean**, **mapfull**, and **maplean**.

FileSizeLimit is optional. It instructs the analysis engine to skip files that are larger than the specified size limit. The default value is zero, which means do not impose a limit.

GetAndExec: [Y|N]

Applies to **getmode**.

GetAndExec is optional. When enabled ('Y' or 'y'), it causes **FTimes** to start a new snapshot once the download is complete. This is accomplished through an `exec()` call. **GetAndExec** depends on **GetFileName**. The default value is 'N'.

Note: Take care when specifying **GetFileName**. If you choose a location that is writeable by other processes, **FTimes** may not read the config file you intended it to. That is to say, some other process may have modified or replaced the original file.

GetFileName: <file>

Applies to **getmode**.

GetFileName is required if **GetAndExec** is enabled. Its value is the name of the file in which the downloaded configuration information is to be stored. **GetFileName** may be specified as a relative path.

HashDirectories: [Y|N]

Applies to **mapfull** and **maplean**.

HashDirectories is optional. When enabled ('Y' or 'y'), it instructs **FTimes** to compute digests for directories. This is done by hashing the digests of all files and directories contained in a given directory in the order they are encountered. Thus, if a directory has the following structure where $D\{1|2\}$ and $F\{1|2\}$ represent directories and files respectively,

```
D1
|
- F1
+ D2
  |
  - F2
```

then, assuming that F1 is mapped before D2, D1 and D2 have the following hashes:

```
Hash(D2) = H(H(F2))
Hash(D1) = H(H(F1), Hash(D2))
```

where H represents the hash algorithm (e.g., MD5, SHA1, etc.).

If an entry within the directory is a special file (e.g., a device) or cannot be opened/hashed, a 16 byte string consisting of all zeros is used for the computation. The default value is 'N'.

HashSymbolicLinks: [Y|N]

Applies to **mapfull** and **maplean**.

HashSymbolicLinks is optional. When enabled ('Y' or 'y'), it instructs **FTimes** to compute digests for symbolic links. This is done by hashing the data returned by `readlink()`. The default value is 'Y'.

Import: <file>

Applies to **digfull**, **diglean**, **mapfull**, **maplean**, **putmode**, and **getmode**.

Import is optional. When specified, the directives contained in the file referenced by this control are included in the current configuration. Multiple instances of this control are allowed per file, and recursion is permitted up to three levels deep. Imports may be specified using a relative path.

Include: [directory|file|link]

Applies to **digfull**, **diglean**, **mapfull**, and **maplean**.

Include controls are optional, and there is no predefined limit on the number that may be specified. However, there can be only one **Include** control/value pair per line. If no **Include** controls are specified, **FTimes** will attempt to map the entire system. If **IncludesMustExist** is enabled, then each **Include** must reference an existing file, directory, or symbolic link. Otherwise, **FTimes** will abort.

Include values must be a regular file, directory, or symbolic link specified as a fully qualified path. For WIN32 file systems, this means that each **Include** must begin with a drive designator ([A-Za-z]:) and be followed by the target path (e.g., 'c:\temp'). For UNIX file systems, each **Include** must begin with a forward slash (e.g., '/tmp').

Note: Symbolic links are not supported in WIN32-based file systems.

Note: Take care when including file systems that reside on remote shares because **FTimes** may attempt to map them. To prevent this from happening, you can either exclude the remote file system or disable **AnalyzeRemoteFiles**.

Note: Directory hashing only applies to directories. This means that each file **Include** is treated as an isolated object apart from any specific tree. The exception to this is any **Include** that gets automatically pruned because it's part of a larger branch.

IncludesMustExist: [Y|N]

Applies to **digfull**, **diglean**, **mapfull**, and **maplean**.

IncludesMustExist is optional. When enabled ('Y' or 'y'), it instructs **FTimes** to check that every **Include** exists prior to mapping or digging. If this control is enabled and any **Include** does not exist, **FTimes** will abort. The default value is 'N'.

LogDir: <directory>

Applies to **digfull**, **diglean**, **mapfull**, and **maplean**.

LogDir is optional. It tells **FTimes** where to write log data. If not specified, the value of **OutDir** will be used as the value for **LogDir**. **LogDir** may be specified as a relative path.

LogFileName: <file>

Applies to **putmode**.

LogFileName is required. Its value is the name of the log file to be posted. **LogFileName** must be specified as a full path.

MagicFile: <file>

Applies to **mapfull** and **maplean**.

MagicFile is optional. If the magic field in **FieldMask** is set and **MagicFile** references a valid XMagic file, **FTimes** attempts to determine the type of each file it maps. If the magic field is not set, this control is ignored. **MagicFile** may be specified as a relative path.

Note: XMagic is not built into **FTimes** by default. If your version of **FTimes** does not have this support and the magic field is requested, **FTimes** will produce null fields.

Note: XMagic is automatically disabled if compression is enabled.

MapRemoteFiles: [Y|N]

Applies to **digfull**, **diglean**, **mapfull**, and **maplean**.

MapRemoteFiles is an alias for **AnalyzeRemoteFiles**, and it is being phased out. Please use **AnalyzeRemoteFiles** instead.

MatchLimit: <integer>

Applies to **digfull** and **diglean**.

MatchLimit is optional. It instructs the search engine to stop searching for a particular pattern within a file once the specified match limit has been reached. The default value is zero, which means do not impose a limit.

Note: Searching for a string such as 'A' with a MatchLimit of zero may produce a tremendous amount of output.

NewLine: [LF|CRLF]

Applies to **digfull**, **diglean**, **mapfull**, and **maplean**.

NewLine is optional. When set to the value CRLF, it generates Windows style line feeds (i.e., carriage return and linefeed). When **NewLine** is set to the value LF, it generates UNIX style line feeds (i.e., linefeed). This control is useful if you review/analyze data sets from different platforms on one particular computer. The default value is the native value for the operating system running the program.

OutDir: <directory>

Applies to **digfull**, **diglean**, **mapfull**, and **maplean**.

OutDir is required. It tells **FTimes** where to write output data. If not specified, the program will abort. **OutDir** may be specified as a relative path.

OutFileHash: <MD5>

Applies to **putmode**.

OutFileHash is required. Its value is the MD5 digest of the output file. This value must match the actual file hash before the upload will be allowed. The value is represented as 32 hex characters.

OutFileName: <file>

Applies to **putmode**.

OutFileName is required. Its value is the name of the output file being posted. **OutFileName** must be specified as a full path.

RequirePrivilege: [Y|N]

Applies to **digfull**, **diglean**, **mapfull**, and **maplean**.

RequirePrivilege is optional. When enabled ('Y' or 'y'), it indicates that the operator wants to ensure that the snapshot is run from a privileged account. On UNIX systems this means that **FTimes** must be run from an account that has a real user id of zero (i.e., root). On NT/2K systems this means that it must be run from an account that has the backup and restore user rights. The default value is 'N'.

Note: **FTimes** will work without privilege, but it's likely to generate more errors due to permission problems.

RunType: [baseline|linktest|snapshot]

Applies to **digfull**, **mapfull**, and **putmode**.

RunType is optional. This control sets a corresponding flag in the log file that classifies output data as baseline, snapshot, or linktest. The value of this control does not affect the format or content of the output. It simply classifies the data so that automated analysis applications can process it accordingly. The default value is baseline.

SSLBundledCAsFile: <file>

Applies to **digfull**, **mapfull**, **putmode**, and **getmode**.

SSLBundledCAsFile is required when **SSLVerifyPeerCert** is enabled. This control specifies the name of a PEM (Privacy Enhanced Mail) encoded file that contains a bundled set of Certificate Authority (CA) certificates. Any validated peer certificate that is signed by one of these certificate authorities will be accepted provided that the **SSLMaxChainLength** and **SSLExpectedPeerCN** checks are also satisfied. **SSLBundledCAsFile** may be specified as a relative path.

SSLExpectedPeerCN: <name>

Applies to **digfull**, **mapfull**, **putmode**, and **getmode**.

SSLExpectedPeerCN is required when **SSLVerifyPeerCert** is enabled. The value of this control represents the peer's expected Common Name (CN). Conventionally, CNs are specified as fully qualified domain names. This control eliminates the need to perform a DNS lookup at the time of certificate validation. This, in turn, may help to prevent attacks involving DNS spoofing.

SSLMaxChainLength: [1-10]

Applies to **digfull**, **mapfull**, **putmode**, and **getmode**.

SSLMaxChainLength is optional when **SSLVerifyPeerCert** is enabled. The value of this control determines how deep a certificate chain may be before it is considered invalid. The default value is one.

SSLPassPhrase: <passphrase>

Applies to **digfull**, **mapfull**, **putmode**, and **getmode**.

SSLPassPhrase is optional when **SSLUseCertificate** is enabled. Its value, if specified, is used to decrypt the contents of the client's private key file (see **SSLPrivateKeyFile**).

SSLPrivateKeyFile: <file>

Applies to **digfull**, **mapfull**, **putmode**, and **getmode**.

SSLPrivateKeyFile is required when **SSLUseCertificate** is enabled. This control specifies the name of a PEM (Privacy Enhanced Mail) encoded key file that can be used to sign SSL certificates. **SSLPrivateKeyFile** may be specified as a relative path.

SSLPublicCertFile: <file>

Applies to **digfull**, **mapfull**, **putmode**, and **getmode**.

SSLPublicCertFile is required when **SSLUseCertificate** is enabled. This control specifies the name of a PEM (Privacy Enhanced Mail) encoded certificate that will be provided during SSL handshakes. **SSLPublicCertFile** may be specified as a relative path.

SSLUseCertificate: [Y|N]

Applies to **digfull**, **mapfull**, **putmode**, and **getmode**.

SSLUseCertificate is optional. When enabled ('Y' or 'y'), it instructs the application to provide client side certificate authentication, if necessary. The default value is 'N'.

SSLVerifyPeerCert: [Y|N]

Applies to **digfull**, **mapfull**, **putmode**, and **getmode**.

SSLVerifyPeerCert is optional. When enabled ('Y' or 'y'), it instructs the application to verify the credentials of the peer server. The default value is 'N'.

URLAuthType: [basic|none]

Applies to **digfull**, **mapfull**, **putmode**, and **getmode**.

URLAuthType is optional. It identifies what authentication scheme to use when issuing HTTP/HTTPS requests. The value specified by this control applies to any requests involving **URLGetURL** or **URLPutURL**. When **URLAuthType** is set to basic, user credentials are base 64 encoded and incorporated into the request header. User credentials specified in the URL take precedence over credentials specified in the **URLUsername** and **URLPassword** controls. The default value is none.

URLCreateConfig: [Y|N]

Applies to **digfull** and **mapfull**.

URLCreateConfig is optional when **URLPutSnapshot** is enabled. When enabled ('Y' or 'y'), it instructs the application to create a configuration file suitable for reposting the snapshot at a later time. The default value is 'N'.

URLGetRequest: [Dig{Full,Lean}Config|Map{Full,Lean}Config]

Applies to **getmode**.

URLGetRequest is required. This control specifies what kind of config file the client is requesting when it issues a GET request. It also determines the next runmode when **GetAndExec** is enabled. Thus, values of Dig{Full,Lean}Config will cause **FTimes** to restart in **digfull** or **diglean** mode, and values of Map{Full,Lean}Config will cause **FTimes** to restart in **mapfull** or **maplean** mode.

URLGetURL: <url>

Applies to **getmode**.

URLGetURL is required. It defines the scheme, user credentials, host address, port, and CGI application to be used when making requests. If a username/password pair is specified in the URL, that pair takes precedence over the values specified by **URLUsername/URLPassword**, if any. URLs must use a scheme of http or https and satisfy the following regular expression:

scheme://(user(:pass)?@)?host(:port)?/(path(\?query)?)?

URLPassword: <password>

Applies to **digfull**, **mapfull**, **putmode**, and **getmode**.

URLPassword is optional. It identifies the password to use when accessing an Integrity Server. The value specified by this control is used in conjunction with **URLGetURL** and **URLPutURL** unless those controls supply their own username/password pair.

URLPutSnapshot: [Y|N]

Applies to **digfull** and **mapfull**.

URLPutSnapshot is optional. When enabled ('Y' or 'y'), **FTimes** attempts to post the snapshot to an Integrity Server. Prior to starting its scan, **FTimes** will transmit an application layer PING to the server to verify that it is accessible and functional. If the remote CGI application is functioning properly, the extended HTTP response code 250 (Ping Received) is returned. **URLPutSnapshot** depends on **URLPutURL**. If basic authentication is required, the controls **URLUsername**, **URLPassword**, and **URLAuthType** may need to be specified as well. The default value is 'N'.

URLPutURL: <url>

Applies to **digfull**, **mapfull**, and **putmode**.

URLPutURL is optional. It defines the scheme, user credentials, host address, port, and CGI application to be used when making PUT requests. If a username/password pair is specified in the URL, that pair takes precedence over the values specified by **URLUsername/URLPassword**, if any. In any event, user credentials are only sent when basic authentication has been requested (See **URLAuthType**). **URLPutURL** uses the same syntax as **URLGetURL**.

URLUnlinkOutput: [Y|N]

Applies to **digfull** and **mapfull**.

URLUnlinkOutput is optional when **URLPutSnapshot** is enabled. When enabled ('Y' or 'y'), any output files are overwritten and unlinked before the program exits. The default value is 'N'.

URLUsername: <username>

Applies to **digfull**, **mapfull**, **putmode**, and **getmode**.

URLUsername is optional. It identifies the username to use when accessing an Integrity Server. The value specified by this control is used in conjunction with **URLGetURL** and **URLPutURL** unless those controls supply their own username/password pair.

RETURN VALUES

Upon successful completion, a value of **XER_OK** (0) is returned. If the program encountered a critical error and had to abort immediately, the value **XER_Abort** (1) is returned. If the command line does not conform to the required syntax, a value of **XER_Usage** (2) is returned. Otherwise, one of the following error codes is returned. These codes indicate which subsystem encountered the fatal error.

- **XER_BootStrap** (3)
- **XER_ProcessArguments** (4)
- **XER_Initialize** (5)
- **XER_CheckDependencies** (6)
- **XER_Finalize** (7)
- **XER_WorkHorse** (8)
- **XER_FinishUp** (9)
- **XER_FinalStage** (10)

FILES

Several different files may be required as input or produced as output in the course of running **FTimes**. These files are generically described below.

BaseName(_BaseNameSuffix)?.cfg

The upload config file. This file is created automatically when **URLCreateConfig** is enabled, and may be used in conjunction with put mode to upload a previously created snapshot.

BaseName(_BaseNameSuffix)?.log

The main log file. This file contains a record of the activities that took place during a dig or map run provided that **LogLevel** was set at an appropriate level.

BaseName(_BaseNameSuffix)?.{dig,map}

The main output file. This file contains possibly compressed map or dig data collected during a scan. The first line in this file contains a header that specifies the name of each field collected.

Bundled Certificate Authorities

When **SSLUseCertificate** has been enabled, **FTimes** expects to find a bundled certificate authorities file in the location specified by **SSLBundledCAs** control. If this file does not exist or have the proper format, **FTimes** will abort.

Config

This file contains directives used to configure **FTimes**. In general a particular config file applies to a single mode of operation. This is because different modes support different controls. Refer to the CONFIGURATION CONTROLS section of this document to determine what controls apply to each mode.

Public Certificate and Private Key

When **SSLUseCertificate** has been enabled, **FTimes** expects to find certificate and key files in the locations specified by **SSLPublicCert** and **SSLPrivateKey** controls, respectively. If these files do not exist or have the proper format, **FTimes** will abort.

Snapshot

This is a previously created and possibly compressed map file that is being supplied as input. In compare mode it represents an uncompressed map file and applies to both the **baseline** and **snapshot** arguments. In decoder mode it represents a compressed map file.

XMagic

This file contains magic tests and descriptions. The magic format used by **FTimes** is XMagic. **FTimes** searches up to three locations for magic: (1) the location specified by the **MagicFile** control, (2) /usr/local/ftimes/etc/xmagic or c:\ftimes\etc\xmagic, and (3) the current working directory. **FTimes** will not abort, if a suitable file is not found, but magic output will be limited.

NOTES

The name attribute may be partially encoded if it contains any special characters. Special characters are defined as [` ' " | % +] or any non-printable character. If a name contains Unicode characters, the high byte is always encoded, and the low byte is encoded as described above. To decode an encoded name, first convert all pluses to spaces (i.e., '+' -> ' '). Then, moving from left to right, convert each %HH sequence, to its byte value. Here 'H' denotes a hex digit.

As a matter of security and good configuration control, you should create a baseline of your entire system on a periodic basis and whenever major events occur such as the addition/removal of software packages or the application of system/security patches.

In critical situations it is best to enable **RequirePrivilege** and run **FTimes** from a privileged account. This will avoid common hang-ups associated with having too little privilege. Backup and restore rights are required to satisfy **RequirePrivilege** on NT systems, and root privilege is required for UNIX systems. If you are simply scanning your own files, you probably do not need to enable **RequirePrivilege**. On NT systems the operator may need the bypass traverse checking privilege in addition to backup and restore rights. Also, make sure that domain policies aren't rendering the operator's privileges ineffective.

In general, use the default LogLevel and review the log data after each invocation as it may reveal errors or issues that need to be addressed.

In general, create distinct config files for each profile you intend to maintain. Here, profile refers to the set of files and directories within a given host that you intend to scan on a regular basis. If you have a large number of profiles that share a common set of controls, you may want to move these controls to a single file. Then, you may simply refer to this file using the **Import** control.

Set **OutDir** to a directory that has the capacity to hold all resultant data. If external media such as jaz, zip, or floppy are available, consider using them. However, using a floppy is not generally recommended unless you know that it has the capacity to hold the data. You may also want to consider writing output to an NFS, Samba, or Windows share. A third option is to use **FTimes'** built-in upload capability, and post your data directly to an Integrity Server.

Setting **LogDir** to a path that represents a floppy device may be useful if data integrity is a concern and there is no removable or protectible media available to store all of the anticipated output. The reason for this is that the log file will contain an MD5 digest of the output file upon the completion of the map. Because the log data was written to floppy, it would be possible to remove and write protect this information for safe keeping. This would provide additional time to offload the results of the map, which can be quite large. Obviously, both files can still be altered by a skilled attacker positioned to take advantage of the situation.

Manage dig and map config files on your Integrity Server, and use **FTimes'** get mode (i.e., **--getmode**) to automatically retrieve them at runtime. This facilitates centralized management and helps to protect potentially sensitive configuration information.

Use the **URLPutSnapshot**, **URLPutURL**, **URLUsername**, **URLPassword**, and **URLAuthType** controls when you want to automatically upload data to an Integrity Server that is configured to handle **FTimes'** PUT requests.

If you have an SSL enabled version of **FTimes**, you'll need to ensure that OpenSSL and/or its libraries are installed and accessible to **FTimes** on the target platform.

When possible, mount target file systems as read only. This will help to ensure minimal time stamp perturbation. In any event, **FTimes** accurately records time stamp information before it is modified by any subsequent access.

EXAMPLES

The following examples are intended to demonstrate different ways of configuring and using **FTimes**. Any text encapsulated between '--- XXX ---' delimiters represents a file's content.

Example 1. Using map mode to baseline and upload

This example demonstrates how to baseline a system and upload the resulting snapshot to an Integrity Server.

The first thing that needs to be done is to obtain the necessary upload information. Assume that an Integrity Server has already been configured to receive snapshots. Also, assume that the following information has been provided:

```
URL = https://192.168.1.50:443/cgi-client/nph-ftimes.cgi
Authentication Type = Basic
Username/ClientID = client_1
Password = password
AllowedFieldMask = ALL-magic
AllowedDataType = map
Server Validates Client Certificates = N
```

Observe that remote server speaks HTTPS. Therefore, an SSL enabled version of **FTimes** is needed. To determine if **FTimes** has SSL support, run the following command:

```
ftimes --version
```

If **FTimes** has SSL support, the output will have the following format:

```
ftimes X.X.X ssl
```

where X.X.X is a version number.

The next item to tackle is the creation of a suitable config file. The following file contains the necessary directives to complete this task.

```
--- example1.cfg ---
BaseName=client_1
OutDir=.
RunType=baseline
FieldMask=ALL-magic
URLPutSnapshot=Y
URLPutURL=https://192.168.1.50:443/cgi-client/nph-ftimes.cgi
URLAuthType=basic
URLUsername=client_1
URLPassword=password
URLCreateConfig=Y
Compress=Y
--- example1.cfg ---
```

Note that there are no **Include** directives. This omission causes **FTimes** to map everything (i.e., the entire system).

Also, note that **OutDir** has been set to the current directory (i.e., '.'). This, while entirely legal, assumes that (1) '.' is writeable and (2) there is enough disk space to hold all generated output.

Compression was enabled to expedite data transfer, and **URLCreateConfig** was enabled to create an upload config file. This would come in handy if, for example, the upload fails. Otherwise, it is not necessary.

The last step is to run **FTimes** and review the corresponding log output to ensure that snapshot was uploaded successfully.

```
ftimes --mapfull example1.cfg
```

Example 2. Using dig mode to search for strings

This example demonstrates how to search files and directories for a set of HEX/ASCII and regular expression strings.

Given that the target file system is /disk1, and the list of strings to be sought are:

```
String1 = /dev/ptyqa
String2 = 0xda 0xbe
String3 = 0x00 A 0x00 A 0x00 A 0x00 A
String4 = A date of the form <YYYY-MM-DD HH:MM:SS>
String5 = Line oriented IP addresses of the form <BOL|tab><192.168.1.*><tab|EOL>
```

where

```
BOL = Beginning Of Line
EOL = End Of Line
```

Strings 1-3 are expressed as (ignore whitespace) ASCII, HEX, and HEX/ASCII, respectively. Strings 4 and 5, on the other hand, are textual descriptions that must be translated into regular expressions. Assume that the required **MatchLimit** is three. That is to say, no more than three matches per file of any single string should be recorded.

The following config file contains those directives required to search all files in /disk1.

```
--- example2.cfg ---
BaseName=digger
OutDir=.
MatchLimit=3
DigStringNormal=/dev/ptyqa
DigStringNormal=%da%be
DigStringNoCase=%00A%00A%00A%00A
DigStringRegExp=\d{4}-\d{2}-\d{2} \d{2}:\d{2}:\d{2}
DigStringRegExp=(?m) (?:^(\t) (192\.168\.1\.\d{1,3})) (?=\t|$)
Include=/disk1
--- example2.cfg ---
```

Observe that strings two and three are URL encoded and that string three will be processed in a case insensitive manner. Read the **DigStringNormal** control description for more details on URL encoding. Also notice that the fourth and fifth strings correspond to Perl compatible regular expressions -- these types of dig strings are available if you compiled **FTimes** with PCRE support. With **DigStringRegExp** patterns, however, you must specify no more than one capturing '()' subpattern. You can use '(?:)' if you require

additional parentheses for grouping purposes. If you do not specify a capturing subpattern, then the entire match will be captured. In this particular case, only the IP address will be captured.

At this point **FTimes** may be run as follows:

```
ftimes --diglean example2.cfg
```

If spontaneity is desired over strict configuration, switch to the **digauto** mode of operation. In this mode elaborate config files aren't necessary, and output is written to stdout. Also, no match limit is imposed. The following example shows how to search an image file (e.g., example2.image) for a set of specific strings. The output, if any, can be piped into other tools that take their input on stdin.

```
--- strings.cfg ---
DigStringNormal=This+box+is+0wn3d
DigStringNormal=3l33t
DigStringNormal=175.20.1.7
DigStringNormal=hacklist@foo.bar.com
--- strings.cfg ---
ftimes --digauto strings.cfg example2.image | some-other-tool
```

Just remember that dig strings must be URL encoded. That's why the spaces in the first string have been replaced with '+'.

Example 3. Change analysis

This example demonstrates how to detect change between two snapshots.

Given that the following files are valid, uncompressed snapshots:

```
Baseline = daily_20001230203000.map
Snapshot = daily_20001231203000.map
```

Compare MD5 digests to determine (C)hanged, (M)issing, and (N)ew files.

The only critical observation needed here is that daily_20001230203000.map is considered to be the baseline. In other words, a baseline is a snapshot taken at an arbitrary point in time to which subsequent snapshots are compared.

This comparison is carried out with the following command:

```
ftimes --compare none+md5 daily_20001230203000.map daily_20001231203000.map
```

To compare multiple attributes at the same time, simply specify the additional fields in the **FieldMask**. For example, the following mask would compare MD5, SHA1, and size attributes:

```
none+md5+sha1+size
```

To compare all attributes at the same time, use a **FieldMask** of 'all'.

Example 4. Using put mode to upload a snapshot

This example demonstrates how to manually upload a snapshot to an Integrity Server. While this is definitely not the easiest way to transfer data, it may be necessary at some point.

Suppose that you have a snapshot that was previously created using the following command line:

```
ftimes --maplean maplean.cfg
```

where maplean.cfg contained the following directives:

```
--- maplean.cfg ---
BaseName=client_1
BaseNameSuffix=datetime
HashDirectories=Y
FieldMask=all-magic
Include=/etc
OutDir=/maps
--- maplean.cfg ---
```

Now, suppose that you wish to upload the associated snapshot files (see below) to an account on an Integrity Server.

```
/maps/client_1_20001230203000.log
/maps/client_1_20001230203000.map
```

The first thing that needs to be done is to obtain the necessary upload information. Assume that an Integrity Server has already been configured to receive snapshots. Also, assume that the following information has been provided:

```
URL = https://192.168.1.50:443/cgi-client/nph-ftimes.cgi
Authentication Type = Basic
Username/ClientID = client_1
Password = password
AllowedFieldMask = ALL-magic
Server Validates Client Certificates = N
```

The next thing to do is to create a config file. Except for **RunType**, **URLPutURL**, **URLAuthType**, **URLUsername**, and **URLPassword**, the controls in the following config file may be obtained or derived by reviewing the log file. Here RunType has been arbitrarily set to snapshot -- it could have also been set to linktest or baseline depending on your needs.

```
--- example4.cfg ---
BaseName=client_1
LogFileName=/maps/client_1_20001230203000.log
OutFileName=/maps/client_1_20001230203000.map
OutFileHash=6958d1337f4f11312a402d0c8f9c050b
DataType=map
FieldMask=all-magic
DateTime=20001230203000
RunType=snapshot
URLPutURL=https://192.168.1.50:443/cgi-client/nph-ftimes.cgi
URLAuthType=basic
URLUsername=client_1
URLPassword=password
```



```
--- example4.cfg ---
```

The last step is to run **FTimes** and review the corresponding log output to ensure that snapshot was uploaded successfully.

```
ftimes --putmode example4.cfg
```

Example 5. Using get mode to download a config file

This example demonstrates how to download a config file from an Integrity Server.

The first thing that needs to be done is to obtain the necessary download information. Assume that an Integrity Server has already been configured to serve config files. Also, assume that the following information has been provided:

```
URL = https://www.integrity.net:443/cgi-client/nph-ftimes.cgi
Authentication Type = Basic
Username/ClientID = client_1
Password = password
Server Validates Client Certificates = Y
Server Common Name = www.integrity.net
Maximum Certificate Chain Length = 2
```

Observe that remote server requires certificates. This means that you'll need three additional PEM encoded files: Public Certificate, Private Key, and Bundled Certificate Authorities. Assume that these files are located on the target system as follows:

```
/usr/local/ftimes/etc/PublicCert.pem
/usr/local/ftimes/etc/PrivateKey.pem
/usr/local/ftimes/etc/BundledCAs.pem
```

Armed with that information, the following config file may be constructed.

```
--- example5.cfg ---
BaseName=client_1
URLGetURL=https://www.integrity.net:443/cgi-client/nph-ftimes.cgi
URLGetRequest=MapFullConfig
GetAndExec=N
URLAuthType=basic
URLUsername=client_1
URLPassword=password
SSLUseCertificate=Y
SSLPublicCertFile=/usr/local/ftimes/etc/PublicCert.pem
SSLPrivateKeyFile=/usr/local/ftimes/etc/PrivateKey.pem
SSLPassPhrase=passphrase
SSLVerifyPeerCert=Y
SSLBundledCAsFile=/usr/local/ftimes/etc/BundledCAs.pem
SSExpectedPeerCN=www.integrity.net
SSLMaxChainLength=2
--- example5.cfg ---
```

The following command will attempt to download a mapfull config file from the specified Integrity Server. If successful, the contents of the config file will be written to stdout.

```
ftimes --getmode example5.cfg
```

If you want to download directly to a file, you can redirect the output to a file or add the **GetFileName** control to your config file. Then, if you wanted to download a config file and take a snapshot in one operation, simply enable **GetAndExec**. This causes **FTimes** to restart in **--mapfull** mode. If **URLGetRequest** was set to **DigFullConfig**, then **FTimes** would request a digfull config file and subsequently restart in **--digfull** mode.

Another way to achieve the same effect is to use the original config file and construct the following pipeline:

```
ftimes --getmode example5.cfg -l 6 | ftimes --mapfull -
```

This has the benefit that potentially sensitive configuration information is not specifically written to a file on disk.

Finally, note that the **LogLevel** for the first command was set to its highest value. This was done simply to reduce log output from that process.

galleta: Cookie analyzer for Internet Explorer.

Developed by Keith J. Jones. Available from
<http://www.foundstone.com/resources/proddesc/galleta.htm>

The following is from <http://www.foundstone.com/resources/proddesc/galleta.htm>

An Internet Explorer Cookie Forensic Analysis Tool.

Author: Keith J. Jones, Principal Computer Forensic Consultant; Foundstone, Inc.

keith.jones@foundstone.com

Copyright 2003 (c) by Foundstone, Inc.

<http://www.foundstone.com>

Many important files within Microsoft Windows have structures that are undocumented. One of the principals of computer forensics is that all analysis methodologies must be well documented and repeatable, and they must have an acceptable margin of error. Currently, there are a lack of open source methods and tools that forensic analysts can rely upon to examine the data found in proprietary Microsoft files.

Many computer crime investigations require the reconstruction of a subject's Internet Explorer Cookie files. Since this analysis technique is executed regularly, we researched the structure of the data found in the cookie files. Galleta, the Spanish word meaning "cookie", was developed to examine the contents of the cookie files. The foundation of Galleta's examination methodology will be documented in an upcoming whitepaper. Galleta will parse the information in a Cookie file and output the results in a field delimited manner so that it may be imported into your favorite spreadsheet program. Galleta is built to work on multiple platforms and will execute on Windows (through Cygwin), Mac OS X, Linux, and *BSD platforms.

Usage: `galleta [options] <filename>`

Options:

`-d` Field Delimiter (TAB by default)

Example Usage:

```
#galleta antihackertoolkit.txt > cookies.txt
```

Open cookies.txt as a TAB delimited file in MS Excel to further sort and filter your results

glimpse: Indexing and query system.

Developed by Golda Velez. Available from <http://webglimpse.net/>

The following is from <http://webglimpse.net/gdocs/glimpsehelp.html>

Glimpse (which stands for GLobal IMPLICIT SEarch) is a popular UNIX indexing and query system that allows you to search through a large set of files very quickly. Glimpse supports most of *agrep*'s options (*agrep* is our powerful version of *grep*) including approximate matching (e.g., finding misspelled words), Boolean queries, and even some limited forms of regular expressions. It is used in the same way, except that you don't have to specify file names. So, if you are looking for a *needle* anywhere in your file system, all you have to do is say *glimpse needle* and all lines containing *needle* will appear preceded by the file name.

To use glimpse you first need to index your files with *glimpseindex*. For example, *glimpseindex -o ~* will index everything at or below your home directory.

Glimpse is also available for web sites, as a set of tools called WebGlimpse. Glimpse includes all of *agrep* and can be used instead of *agrep* by giving a file name(s) at the end of the command. This will cause glimpse to ignore the index and run *agrep* as usual. For example, *glimpse -1 pattern file* is the same as *agrep -1 pattern file*. *Agrep* is distributed as a self-contained package within glimpse, and can be used separately. We added a new option to *agrep*: *-r* searches recursively the directory and everything below it (see *agrep* options below); it is used only when glimpse reverts to *agrep*.

SYNOPSIS

glimpse [**-almost all letters**] *pattern*

INTRODUCTION

We start with simple ways to use glimpse and describe all the options in detail later on. Once an index is built, using *glimpseindex*, searching for *pattern* is as easy as saying

glimpse pattern

The output of glimpse is similar to that of *agrep* (or any other *grep*). The pattern can be any *agrep* legal pattern including a regular expression or a Boolean query (e.g., searching for Tucson AND Arizona is done by *glimpse `Tucson;Arizona`*).

The speed of glimpse depends mainly on the number and sizes of the files that contain a match and only to a second degree on the total size of all indexed files. If the pattern is reasonably uncommon, then all matches will be reported in a few seconds even if the indexed files total 500MB or more. Some information on how glimpse works and a reference to a detailed article are given below.

Most of *agrep* (and other *grep*'s) options are supported, including approximate matching. For example,

glimpse -1 `Tucson;Arezona`

will output all lines containing both patterns allowing one spelling error in any of the patterns (either insertion, deletion, or substitution), which in this case is definitely needed.

glimpse -w -i `parent`

specifies case insensitive (-i) and match on complete words (-w). So `Parent` and `PARENT` will match, `parent/child` will match, but `parenthesis` or `parents` will not match. (Starting at version 3.0, glimpse can be much faster when these two options are specified, especially for very large indexes. You may want to set an alias especially for "glimpse -w -i".)

The -F option provides a pattern that must match the file name. For example,

glimpse -F `\.c\$` needle

will find the pattern *needle* in all files whose name ends with .c. (Glimpse will first check its index to determine which files may contain the pattern and then run agrep on the file names to further limit the search.) The -F option *should not* be put at the end after the main pattern (e.g., "glimpse needle -F hay" is incorrect).

DETAILED DESCRIPTION OF ALL GLIMPSE OPTIONS

-# # is an integer between 1 and 8 specifying the maximum number of errors permitted in finding the approximate matches (the default is zero). Generally, each insertion, deletion, or substitution counts as one error. It is possible to adjust the relative cost of insertions, deletions and substitutions (see -l -D and -S options). Since the index stores only lower case characters, errors of substituting upper case with lower case may be missed. Allowing errors in the match requires more time and can slow down the match by a factor of 2-4. Be very careful when specifying more than one error, as the number of matches tend to grow very quickly.

-a prints attribute names. This option applies only to Harvest SOIF structured data (used with glimpseindex -s).

-A used for glimpse internals.

-b prints the byte offset (from the beginning of the file) of the end of each match. The first character in a file has offset 0.

-B Best match mode. (Warning: -B sometimes misses matches. It is safer to specify the number of errors explicitly.) When -B is specified and no exact matches are found, glimpse will continue to search until the closest matches (i.e., the ones with minimum number of errors) are found, at which point the following message will be shown: "the best match contains x errors, there are y matches, output them? (y/n)" This message refers to the number of matches found in the index. There may be many more matches in the actual text (or there may be none if -F is used to filter files). When the -#, -c, or -l options are specified, the -B option is ignored. In general, -B may be slower than -#, but not by very much. Since the index stores only lower case characters, errors of substituting upper case with lower case may be missed.

-c Display only the count of matching records. Only files with count > 0 are displayed.

-C tells glimpse to send its queries to *glimpseserver*.

-d 'delim' Define *delim* to be the separator between two records. The default value is '\$', namely a record is by default a line. *delim* can be a string of size at most 8 (with possible use of ^ and \$), but not a regular expression. Text between two *delim*'s, before the first *delim*, and after the last *delim* is considered as one record. For example, -d '\$\$' defines paragraphs as records and -d '^From\ ' defines mail messages as records. *glimpse* matches each record separately. This option does not currently work with regular expressions. The -d option is especially useful for Boolean AND queries, because the patterns need not appear in the same line but in the same record. For example,

```
glimpse -F mail -d '^From\ ' 'glimpse;arizona;announcement'
```

will output all mail messages (in their entirety) that have the 3 patterns anywhere in the message (or the header), assuming that files with 'mail' in their name contain mail messages. If you want the scope of the record to be the whole file, use the -W option. *Glimpse warning:* Use this option with care. If the delimiter is set to match mail messages, for example, and glimpse finds the pattern in a regular file, it may not find the delimiter and will therefore output the whole file. (The -t option - see below - can be used to put the *delim* at the end of the record.) *Performance Note:* Agrep (and glimpse) resorts to more complex search when the -d option is used. The search is slower and unfortunately no more than 32 characters can be used in the pattern.

-D k Set the cost of a deletion to *k* (*k* is a positive integer). This option does not currently work with regular expressions.

-e Useful when the pattern begins with a -.

-E prints the lines in the index (as they appear in the index) which match the pattern. Used mostly for debugging and maintenance of the index. This is not an option that a user needs to know about.

-f file_name this option has a different meaning for agrep than for glimpse: In glimpse, only the files whose names are listed in *file_name* are matched. (The file names have to appear as in *.glimpse_filenames*.) In agrep, the *file_name* contains the list of the patterns that are searched.

-F file_pattern limits the search to those files whose name (including the whole path) matches *file_pattern*. This option can be used in a variety of applications to provide limited search even for one large index. If *file_pattern* matches a directory, then all files with this directory on their path will be considered. To limit the search to actual file names, use \$ at the end of the pattern. *file_pattern* can be a regular expression and even a Boolean pattern. This option is implemented by running agrep *file_pattern* on the list of file names obtained from the index. Therefore, searching the index itself takes the same amount of time, but limiting the second phase of the search to only a few files can speed up the search significantly. For example,

```
glimpse -F 'src#\c$' needle
```

will search for needle in all .c files with src somewhere along the path. The -F *file_pattern* must appear before the search pattern (e.g., glimpse needle -F '\c\$' will not work). It is possible to use some of agrep's options when matching file names. In this case all options as well as the *file_pattern* should be in quotes. (-B and -v do not work very well as part of a *file_pattern*.) For example,

`glimpse -F '-1 \.html' pattern`

will allow one spelling error when matching .html to the file names (so ".htm" and ".shtml" will match as well).

`glimpse -F '-v \.c$' counter`

will search for 'counter' in all files *except* for .c files.

-g prints the file number (its position in the .glimpse_filenames file) rather than its name.

-G Output the (whole) files that contain a match.

-h Do not display filenames.

-H *directory_name* searches for the index and the other .glimpse files in *directory_name*. The default is the home directory. This option is useful, for example, if several different indexes are maintained for different archives (e.g., one for mail messages, one for source code, one for articles).

-i Case-insensitive search -- e.g., "A" and "a" are considered equivalent. Glimpse's index stores all patterns in lower case. *Performance Note:* When -i is used together with the -w option, the search may become much faster. It is recommended to have -i and -w as defaults, for example, through an alias. We use the following alias in our .cshrc file .br alias glwi 'glimpse -w -i'

-I k Set the cost of an insertion to *k* (*k* is a positive integer). This option does not currently work with regular expressions.

-j If the index was constructed with the -t option, then -j will output the files last modification dates in addition to everything else. There are no major performance penalties for this option.

-J *host_name* used in conjunction with glimpserver (-C) to connect to one particular server.

-k No symbol in the pattern is treated as a meta character. For example, glimpse -k 'a(b|c)*d' will find the occurrences of a(b|c)*d whereas glimpse 'a(b|c)*d' will find substrings that match the regular expression 'a(b|c)*d'. (The only exception is ^ at the beginning of the pattern and \$ at the end of the pattern, which are still interpreted in the usual way. Use \^ or \\$ if you need them verbatim.)

-K *port_number* used in conjunction with glimpserver (-C) to connect to one particular server at the specified TCP port number.

-l Output only the files names that contain a match. This option differs from the -N option in that the files themselves *are* searched, but the matching lines are not shown.

-L x | x:y | x:y:z if one number is given, it is a limit on the total number of matches. Glimpse outputs only the first *x* matches. If -l is used (i.e., only file names are sought), then the limit is on the number of files; otherwise, the limit is on the number of records. If two numbers are given (x:y), then *y* is an added limit on the total number of files. If three numbers are given (x:y:z), then *z* is an added limit on the number of matches per file. If any of the *x*, *y*, or *z* is set to 0, it means to

ignore it (in other words 0 = infinity in this case); for example, -L 0:10 will output all matches to the first 10 files that contain a match. This option is particularly useful for servers that needs to limit the amount of output provided to clients.

-m used for glimpse internals.

-M used for glimpse internals.

-n Each matching record (line) is prefixed by its record (line) number in the file. *Performance Note:* To compute the record/line number, agrep needs to search for all record delimiters (or line breaks), which can slow down the search.

-N searches only the index (so the search is faster). If -o or -b are used then the result is the number of files that have a potential match plus a prompt to ask if you want to see the file names. (If -y is used, then there is no prompt and the names of the files will be shown.) This could be a way to get the matching file names without even having access to the files themselves. However, because only the index is searched, some potential matches may not be real matches. In other words, with -N you will not miss any file but you may get extra files. For example, since the index stores everything in lower case, a case-sensitive query may match a file that has only a case-insensitive match. Boolean queries may match a file that has all the keywords but not in the same line (indexing with -b allows glimpse to figure out whether the keywords are close, but it cannot figure out from the index whether they are exactly on the same line or in the same record without looking at the file). If the index was not build with -o or -b, then this option outputs the number of *blocks* matching the pattern. This is useful as an indication of how long the search will take. All files are partitioned into usually 200-250 blocks. The file *.glimpse_statistics* contains the total number of blocks (or *glimpse -N a* will give a pretty good estimate; only blocks with no occurrences of 'a' will be missed).

-o the opposite of -t: the delimiter is not output at the tail, but at the beginning of the matched record.

-O the file names are not printed before every matched record; instead, each filename is printed just once, and all the matched records within it are printed after it.

-p (from version 4.0B1 only) Supports reading compressed set of filenames. The -p option allows you to utilize compressed 'neighborhoods' (sets of filenames) to limit your search, without uncompressing them. Added mostly for WebGlimpse. The usage is: "-p filename:X:Y:Z" where "filename" is the file with compressed neighborhoods, X is an offset into that file (usually 0, must be a multiple of sizeof(int)), Y is the length glimpse must access from that file (if 0, then whole file; must be a multiple of sizeof(int)), and Z must be 2 (it indicates that "filename" has the sparse-set representation of compressed neighborhoods: the other values are for internal use only). Note that any colon ":" in filename must be escaped using a backslash \.

-P used for glimpse internals.

-q prints the offsets of the beginning and end of each matched record. The difference between -q and -b is that -b prints the offsets of the actual matched string, while -q prints the offsets of the whole record where the match occurred. The output format is @x{y}, where x is the beginning offset and y is the end offset.

-Q when used together with **-N** glimpse not only displays the filename where the match occurs, but the exact occurrences (offsets) as seen in the index. This option is relevant only if the index was built with **-b**; otherwise, the offsets are not available in the index. This option is ignored when used not with **-N**.

-r This option is an **agrep** option and it will be ignored in **glimpse**, unless **glimpse** is used with a file name at the end which makes it run as **agrep**. If the file name is a directory name, the **-r** option will search (recursively) the whole directory and everything below it. (The **glimpse** index will not be used.)

-R *k* defines the maximum size (in bytes) of a record. The maximum value (which is the default) is 48K. Defining the maximum to be lower than the default may speed up some searches.

-s Work silently, that is, display nothing except error messages. This is useful for checking the error status.

-S *k* Set the cost of a substitution to *k* (*k* is a positive integer). This option does not currently work with regular expressions.

-t Similar to the **-d** option, except that the delimiter is assumed to appear at the *end* of the record. Glimpse will output the record starting from the end of **.I** *delim* to (and including) the next **.I** *delim*. (See warning for the **-d** option.)

-T *directory* Use *directory* as a place where temporary files are built. (Glimpse produces some small temporary files usually in **/tmp**.) This option is useful mainly in the context of structured queries for the Harvest project, where the temporary files may be non-trivial, and the **/tmp** directory may not have enough space for them.

-U (starting at version 4.0B1) Interprets an index created with the **-X** or the **-U** option in **glimpseindex**. Useful mostly for WebGlimpse or similar web applications. When **glimpse** outputs matches, it will display the filename, the URL, and the title automatically.

-v (This option is an **agrep** option and it will be ignored in **glimpse**, unless **glimpse** is used with a file name at the end which makes it run as **agrep**.) Output all records/lines that do *not* contain a match.

-V prints the current version of **glimpse**.

-w Search for the pattern as a word (i.e., surrounded by non-alphanumeric characters. For example, *glimpse -w car* will match *car*, but not *characters* and not *car10*. The non-alphanumeric *must* surround the match; they cannot be counted as errors. This option does not work with regular expressions. *Performance Note:* When **-w** is used together with the **-i** option, the search may become much faster. The **-w** will not work with **\$**, **^**, and **_** (see **BUGS** below). It is recommended to have **-i** and **-w** as defaults, for example, through an alias. We use the following alias in our **.cshrc** file

```
alias glwi 'glimpse -w -i'
```

-W The default for Boolean AND queries is that they cover one record (the default for a record is one line) at a time. For example, *glimpse 'good;bad'* will output all lines containing both 'good' and

'bad'. The -W option changes the scope of Booleans to be the whole file. Within a file glimpse will output all matches to any of the patterns. So, glimpse -W 'good;bad' will output all lines containing 'good' or 'bad', but only in files that contain both patterns. For structured queries, the scope is always the whole attribute or file.

-x The pattern must match the whole line. (This option is translated to -w when the index is searched and it is used only when the actual text is searched. It is of limited use in glimpse.)

-X (from version 4.0B1 only) Output the names of files that contain a match even if these files have been deleted since the index was built. Without this option glimpse will simply ignore these files.

-y Do not prompt. Proceed with the match as if the answer to any prompt is y. Servers (or any other scripts) using glimpse will probably want to use this option.

-Y k If the index was constructed with the -t option, then -Y x will output only matches to files that were created or modified within the last x days. There are no major performance penalties for this option.

-z Allow customizable filtering, using the file .glimpse_filters to perform the programs listed there for each match. The best example is compress/decompress. If .glimpse_filters include the line

*.Z uncompress <

(separated by tabs) then before indexing any file that matches the pattern "*.Z" (same syntax as the one for .glimpse_exclude) the command listed is executed first (assuming input is from stdin, which is why uncompress needs <) and its output (assuming it goes to stdout) is indexed. The file itself is not changed (i.e., it stays compressed). Then if glimpse -z is used, the same program is used on these files on the fly. Any program can be used (we run 'exec'). For example, one can filter out parts of files that should not be indexed. Glimpseindex tries to apply all filters in .glimpse_filters in the order they are given. For example, if you want to uncompress a file and then extract some part of it, put the compression command (the example above) first and then another line that specifies the extraction. Note that this can slow down the search because the filters need to be run before files are searched. (See also glimpseindex.)

-Z No op. (It's useful for glimpse's internals. Trust us.)

PATTERNS

glimpse supports a large variety of patterns, including simple strings, strings with classes of characters, sets of strings, wild cards, and regular expressions.

Strings

Strings are any sequence of characters, including the special symbols '^' for beginning of line and '\$' for end of line. The following special characters ('\$', '^', '*', '[', '^', '|', '(', ')', '!', and '\ ') as well as the following meta characters special to glimpse (and agrep): ';', ',', '#', '<', '>', '-', and '.', should be preceded by '\' if they are to be matched as regular characters. For example, ^abc\ corresponds to the string ^abc\, whereas ^abc corresponds to the string abc at the beginning of a line.

Classes of characters

a list of characters inside [] (in order) corresponds to any character from the list. For example, [a-ho-z] is any character between a and h or between o and z. The symbol '^' inside [] complements the list. For example, [^i-n] denote any character in the character set except character 'i' to 'n'. The symbol '^' thus has two meanings, but this is consistent with egrep. The symbol '.' (don't care) stands for any symbol (except for the newline symbol).

Boolean operations

Glimpse supports an 'AND' operation denoted by the symbol '&'; an 'OR' operation denoted by the symbol '|', a limited version of a 'NOT' operation (starting at version 4.0B1) denoted by the symbol '~', or any combination. For example, *glimpse 'pizza;cheeseburger'* will output all lines containing both patterns. *glimpse -F 'gnu;\.c\$' 'define;DEFAULT'* will output all lines containing both 'define' and 'DEFAULT' (anywhere in the line, not necessarily in order) in files whose name contains 'gnu' and ends with '.c'. *glimpse '{political,computer};science'* will match 'political science' or 'science of computers'. The NOT operation works only together with the -W option and it is generally applies only to the whole file rather to individual records. It currently does not work with approximate matching. Its output may sometimes seem counterintuitive. Use with care. *glimpse -W 'fame;~glory'* will output all lines containing 'fame' in all files that contain 'fame' but do not contain 'glory'; This is the most common use of NOT, and in this case it works as expected. *glimpse -W '~{fame;glory}'* will be limited to files that do not contain both words, and will output all lines containing one of them.

Wild cards

The symbol '#' is used to denote a sequence of any number (including 0) of arbitrary characters. The symbol # is equivalent to .* in egrep. In fact, .* will work too, because it is a valid regular expression (see below), but unless this is part of an actual regular expression, # will work faster. (Currently glimpse is experiencing some problems with #.)

Combination of exact and approximate matching Any pattern inside angle brackets <> must match the text exactly even if the match is with errors. For example, <mathemat>ics matches mathematical with one error (replacing the last s with an a), but mathe<mat>ics does not match mathematical no matter how many errors are allowed. (This option is buggy at the moment.)

Regular expressions

Since the index is word based, a regular expression must match words that appear in the index for glimpse to find it. Glimpse first strips the regular expression from all non-alphabetic characters, and searches the index for all remaining words. It then applies the regular expression matching algorithm to the files found in the index. For example, *glimpse 'abc.*xyz'* will search the index for all files that contain both 'abc' and 'xyz', and then search directly for 'abc.*xyz' in those files. (If you use *glimpse -w 'abc.*xyz'*, then 'abcxyz' will not be found, because glimpse will think that abc and xyz need to be matches to whole words.) The syntax of regular expressions in **glimpse** is in general the same as that for **agrep**. The union operation '|', Kleene closure '*', and parentheses () are all supported. Currently '+' is not supported. Regular expressions are currently limited to approximately 30 characters (generally excluding meta characters). Some options (-d, -w, -t, -x, -D, -l, -S) do not currently work with regular expressions. The maximal number of errors for regular expressions that use '*' or '|' is 4.

structured queries

Glimpse supports some form of structured queries using Harvest's SOIF format. See [STRUCTURED QUERIES](#) for details.

EXAMPLES

`glimpse -F `haystack.h$` needle`

finds all needles in all haystack.h's files.

`glimpse -2 -F html Anesthesiology`

outputs all occurrences of Anesthesiology with two errors in files with html somewhere in their full name.

`glimpse -l -F `.c$` variablename`

lists the names of all .c files that contain variablename (the -l option lists file names rather than output the matched lines).

`glimpse -F `mail;1993` `windsurfing;Arizona``

finds all lines containing *windsurfing* and *Arizona* in all files having `mail` and `1993` somewhere in their full name.

`glimpse -F mail `t.j@#uk``

finds all mail addresses (search only files with mail somewhere in their name) from the uk, where the login name ends with t.j, where the . stands for any one character. (This is very useful to find a login name of someone whose middle name you don't know.)

`glimpse -F mbox -h -G . > MBOX`

concatenates all files whose name matches `mbox` into one big one.

grepmail: Grep through mailboxes.

Developed by David Coppit. Available from <http://grepmail.sourceforge.net/>

The following is taken from <http://grepmail.sourceforge.net/>

Search for emails in a normal or compressed mailbox using a regular expression or date constraint.

Features:

- Gzip, bzip2, tzip support
- Piped input supported (compressed or not)
- Supports complex dates like "between Jan 15, 1999 and 5 weeks ago"
- Can ignore non-text MIME attachments
- Can search only the header or only the body of an email
- Can recurse subdirectories
- Automatically optimizes for speed vs. flexibility when searching based on date constraints.

The following is taken from the command: grepmail

```
grepmail 5.3032
```

usage:

```
grepmail [--help|--version] [-abBDFhHilmrRuvVw] [-C <cache-file>]
  [-j <status>] [-s <sizespec>] [-d <date-specification>]
  [-X <signature-pattern>] [-Y <header-pattern>]
  [-e] <pattern> <files...>
```

```
grepmail [--help|--version] [-abBDFhHilmrRuvVw] [-C <cache-file>]
  [-j <status>] [-s <sizespec>] [-d <date-specification>]
  [-X <signature-pattern>] [-Y <header-pattern>]
  -E <expr> <files...>
```

```
grepmail [--help|--version] [-abBDFhHilmrRuvVw] [-C <cache-file>]
  [-j <status>] [-s <sizespec>] [-d <date-specification>]
  [-X <signature-pattern>] [-Y <header-pattern>]
  -f <pattern-file> <files...>
```

At least one of -s, -d, -u, -e, and -E must be specified, and can appear in any relative order following the other flags. The -e flag is optional if pattern appears immediately before -s or -d. Files can be plain ASCII or ASCII files compressed with gzip, tzip, or bzip2. -E allows for complex pattern matches involving logical operators. If no file is provided, normal or compressed ASCII input is taken from STDIN.

-a Use received date instead of sent date for -d matching
-b Search must match body
-B Print message bodies but with only limited headers
-C Specify the location of the cache file
-d Specify a required date range (see below)
-D Debug mode

- e Explicitly name pattern (when searching for strings beginning with "-")
- E Specify a complex search expression
- f Read patterns from a file
- F Force processing of all data as mailboxes
- h Search must match header
- H Print headers but not bodies of matching emails
- i Ignore case in the search expression
- j Search must match status (A=answered, R=read, D=deleted, O=old, F=flagged)
- l Output the names of files having an email matching the expression
- M Do not search non-text mime attachments
- m Append "X-Mailfolder: <folder>" to all headers to indicate in which folder the match occurred
- n Print the line number info (and filename if necessary) for the emails
- q Quiet mode -- don't output warnings
- r Output the names of the files and the number of emails matching the expression
- R Recurse directories
- s Specify a size range in bytes (see below)
- S Ignore signatures
- u Ensure that no duplicate emails are output
- v Output emails that don't match the expression
- V Display the version number
- w Match word boundaries
- X Specify a regular expression for the signature separator
- Y Specify a header to search (implies -h)
- help Print a help message

Date constraints require Date::Parse. Date specifications must be of the form of:

- a date like "today", "1st thursday in June 1992" (requires Date::Manip), "05/18/93", "12:30 Dec 12th 1880", "8:00pm december tenth",
- "before", "after", or "since", followed by a date as defined above,
- "between <date> and <date>", where <date> is defined as above.

Size constraints must be of the form of:

- 12345: match size of exactly 12345
- <12345, <=12345, >12345, >=12345: match size less than, less than or equal, greater than, or greater than or equal to 12345
- 10000-12345: match size between 10000 and 12345 inclusive

logfinder.py: EFF logfinder utility.

Developed by Ben Laurie and Seth Schoen. Available from <http://www.eff.org/osp/>

The following is taken from <http://www.eff.org/osp/>

logfinder 0.1 - Locate Log Files on Your Systems

Many system administrators don't know exactly what logs they have until they have looked into the question. Often, logging was enabled by defaults -- or by previous system administrators -- and so your systems may be keeping logs you never intended. We have created a program called logfinder as a simple means of locating files that might be logs on an existing system. logfinder uses regular expressions to find local files with "log-like" contents; you can customize those expressions if necessary to meet your needs.

logfinder requires Python 2 or greater and finds logs in text files on a POSIX-like system. (It might also find some log-like data in binary files if the binary files represent that data in textual form.)

The following is taken from http://www.eff.org/news/archives/2005_02.php

Logfinder Helps Eliminate Unwanted Logging of Personal Data

San Francisco, CA - Today the Electronic Frontier Foundation (EFF) released logfinder, a software tool to help people reduce the unnecessary collection of personal information about computer users. Often computer network servers automatically log information about who has visited a website and when, or who has sent and received email. Such data tells a lot about a user's browsing and email habits and could be used in privacy-invasive ways. Moreover, log data must be turned over to government entities with court orders and can be subpoenaed by opposing sides in court cases.

By finding unwanted log files, logfinder informs system administrators when their servers are collecting personal data and gives them the opportunity to turn logging off if it isn't gathering information necessary for administering the system.

Logfinder was conceived by security consultant Ben Laurie and written by EFF Staff Technologist Seth Schoen. It's intended to complement EFF's recent white paper, "Best Practices for Online Service Providers," in which the organization argues that administrators should remove as many logs as possible and delete all personally identifying data from them.

"People who choose to follow our recommendations in the white paper might not know what kinds of logs they have," said Schoen. "Logfinder is an example of one way a system administrator could become aware of the presence of logs, as well as discover sensitive information being collected in known logs."

The following is from the command: logfinder.py -h

Please consider the limitations of this program, which is not able to find every possible kind of log file or identify every potential data retention concern. See README for more information.

For more information about data retention, please consult EFF's resources for on-line service providers at <<http://www.eff.org/osp/>>.

Usage:

```
# logfinder.py [-w] [-l lines] [-c] [-h | --help] [path] [path] [...]
```

With -c or no path specified, look for current logging activity in open files system wide.

With a path or paths specified, look for log-like text in files within the specified path or paths. By default, look at the first 100 lines of such files; if -l is specified, look at the specified number of lines instead; if -w is specified, look at the whole file.

For maximum coverage, specify the path "/".

logsh: Log your terminal session

Developed by Amir Guindehi. Available from <http://open.datacore.ch/download/>

The following is taken from the README file.

The Log Shell is a restricted shell allowing a logged in user to do nothing more than look at a logfile. The Log Shell does in essence a 'tail -f' on a hardcoded log file on the filesystem.

The Log Shell is capable of filtering that log file according to simple regular expressions defined in filter list and grep list config files. These filters allow the system administrator to filter away regular - not important - log messages allowing the user to view only those messages really important. Log filter lists and grep lists can be combined to logical expressions using & (and) and | (or) expressions.

Furthermore the user has the ability to further view only log messages containing specific patterns (eg. grep) and to view only log messages originating from a specific host. You can use & (and) as well as | (or) to give Log Shell logical expressions of patterns which have to match.

The Log Shell allows the user drop marks into the output and to add new lines to the output for log message separation purposes. It's also possible to clear the output screen.

The user is able to reload the configuration and filter lists of running Log Shells by sending a SIGUSR1 to the process or by user command.

At any moment the user is able to scroll back 5kB or 15kB of log messages by single keypress.

Helix Implementation

When logsh is started in Helix, it displays the following information:

```
All activity is logged to ~/ttylog/  
Stop logging by typing exit  
Play back log files with:  
Replay DATE_tty.log.timing DATE_tty.log
```

In the ~/ttylog/ directory, the investigator will find logs with names like:

```
Feb07-171136-tty_0.log  
Feb07-171136-tty_0.log.timing
```

The .log file is a copy of all the console I/O, while the .timing file is keep track of the sequencing of the commands so the session can be played back using the replay command.

Ishw: Hardware Lister.

Created by Lyonel Vincent, available from <http://ezix.sourceforge.net/software/lshw.html>

Ishw (Hardware Lister) is a small tool to provide detailed information on the hardware configuration of the machine. It can report exact memory configuration, firmware version, mainboard configuration, CPU version and speed, cache configuration, bus speed, etc. on DMI-capable x86 or EFI (IA-64) systems and on some PowerPC machines (PowerMac G4 is known to work). Information can be output in plain text, XML or HTML. It currently supports DMI (x86 and EFI only), OpenFirmware device tree (PowerPC only), PCI/AGP, ISA PnP (x86), CUID (x86), IDE/ATA/ATAPI, PCMCIA (only tested on x86), USB and SCSI (Vincent, 2006).

Usage

`lshw [format] [options...]`

where format can be

<code>-X</code>	to launch the GUI (if available)
<code>-html</code>	to activate HTML mode
<code>-xml</code>	to activate XML mode
<code>-short</code>	to print hardware paths
<code>-businfo</code>	to print bus information

and options can be

<code>-enable <i>TEST</i></code>	to enable a test
<code>-disable <i>TEST</i></code>	to disable a test
<code>-class <i>CLASS</i></code>	to limit the output to a given class
<code>-C <i>CLASS</i></code>	alias for <code>-class <i>CLASS</i></code>

NOTE: to use some features (like DMI on x86 platforms), you need to run lshw as root or it will only report partial information.

mac-robber: TCT's graverobber written in C.

Developed by Brian Carrier. Available from <http://www.sleuthkit.org/mac-robber/desc.php>

The following is taken from <http://www.sleuthkit.org/mac-robber/desc.php>

mac-robber is a digital investigation tool that collects data from allocated files in a mounted file system. This is useful during incident response when analyzing a live system or when analyzing a dead system in a lab. The data can be used by the mactime tool in The Sleuth Kit to make a timeline of file activity. The mac-robber tool is based on the grave-robber tool from TCT and is written in C instead of Perl.

mac-robber requires that the file system be mounted by the operating system, unlike the tools in The Sleuth Kit that process the file system themselves. Therefore, mac-robber will not collect data from deleted files or files that have been hidden by rootkits. mac-robber will also modify the Access times on directories that are mounted with write permissions.

"What is mac-robber good for then", you ask? mac-robber is useful when dealing with a file system that is not supported by The Sleuth Kit or other file system analysis tools. mac-robber is very basic C and should compile on any UNIX system. Therefore, you can run mac-robber on an obscure, suspect UNIX file system that has been mounted read-only on a trusted system. I have also used mac-robber during investigations of common UNIX systems such as AIX.

USAGE

mac-robber takes a list of directories to analyze as arguments. For example, to analyze the 'mnt' and 'mnt2' directories and send the output to a file:

```
# mac-robber mnt mnt2 > data/body.mac
```

If you want to analyze the system from the root directory and send the data to a server running netcat, use:

```
# mac-robber / | nc 10.0.0.1 8000
```

The server would be running something like:

```
# nc -l -p 8000 > body.mac
```

To analyze the data, the mactime tool from The Sleuth Kit is required. Use the -b flag to import the body file:

```
# mactime -b body.mac 01/01/2001 > timeline.01-01-2001
```

COMMENTS

- This file uses the readdir function and therefore will update the Access time on directories. Therefore, if you are going to make an image of the disks, do that first. Also, malicious kernel modules could produce incorrect data when run on a compromised host.
- Make sure that you do not write the output of this program to a drive on the compromised system, it may overwrite unallocated data.

mac_grab.pl: e-fense MAC time utility.

Developed by Drew Fahey. Available from www.e-fense.com

The following is taken from the mac_grab.pl command.

Usage: mac_grab.pl [-DRv] {directory}

-D turn on debugging option
-R turn off Recursion
-v verbose

Sample output :

```
# mac_grab.pl bin
```

```
-----  
|                               MAC_GRAB BY E-FENSE, INC.                               |  
----- MAC TIMES Output -----  
  
DATE          TIME          SIZE MAC PERMS          OWNER    GROUP    FILE  
=====
```

Dec 10 2003	22:35:46	61994	mac	-rwxr-xr-x	root	knoppix	bin/root-tail
May 16 2004	01:02:33	109	mac	-rwxr-xr-x	root	knoppix	bin/firewire_start.sh
May 16 2004	01:02:57	84	mac	-rwxr-xr-x	root	knoppix	bin/firewire_stop.sh
May 17 2004	22:56:35	79618	mac	-rwxr-xr-x	root	knoppix	bin/bclump

md5deep: Recursive md5sum with db lookups.

Developed by Special Agent Jesse Kornblum of the United States Air Force Office of Special Investigations. It is available from <http://md5deep.sourceforge.net/>

md5deep is a cross-platform tool that can calculate the MD5 signatures of files. md5deep is similar to the md5sum but it can also process recursive directories, produce an estimated completion time, compare files to known hash sets, and be set to only process certain types of files. There are also companion tools that will calculate SHA-1, SHA-256 Tiger, or Whirlpool message digests for files as well.

The following was taken from <http://md5deep.sourceforge.net/manpage.html>

SYNOPSIS

```
md5deep -v | -V | -h
md5deep [-m|-M|-x|-X <file>] [-a|-A <hash>] [-nwzres0lbkq] [-o <fbcp1sd>] [FILES]
```

DESCRIPTION

Computes the hashes, or message digest, for any number of files while optionally recursively digging through the directory structure. Can also take a list of known hashes and display the filenames of input files whose hashes either do or do not match any of the known hashes. Errors are reported to standard error. If no FILES are specified, reads from standard input.

-r Enables recursive mode. All subdirectories are traversed. Please note that recursive mode cannot be used to examine all files of a given file extension. For example, calling md5deep -r *.txt will examine all files in *directories* that end in .txt.

-e Displays a progress indicator and estimate of time remaining for each file being processed. Time estimates for files larger than 4GB are not available on Windows.

-m <file>

Enables matching mode. The file given should be a list of known hashes. The input files are examined one at a time, and only those files that match the list of known hashes are output. This flag may be used more than once to add multiple sets of known hashes. Acceptable formats for lists of known hashes are plain (such as those generated by md5deep or md5sum), Hashkeeper files, iLook, and the National Software Reference Library (NSRL) as produced by the National Institute for Standards in Technology.

If standard input is used with the -m flag, displays "stdin" if the input matches one of the hashes in the list of known hashes. If the hash does not match, the program displays no output.

This flag may not be used in conjunction with the -x, -X, or -A flags.

-x <file>

Same as the -m flag above, but does negative matching. That is, only those files NOT in the list of known hashes are displayed.

This flag may not be used in conjunction with the `-m`, `-M`, or `-a` flags.

-M and -X <file>

Same as `-m` and `-x` above, but displays the hash for each file that does (or does not) match the list of known hashes.

-a <hash>

Adds a single hash to the list of known hashes used for matching mode, and if not already enabled, enables matching mode. Adding single hashes cannot, by itself, be used to print the hashes of matching files like the `-M` flag does. When used in conjunction with the `-w` flag, the filename displayed is just the hash submitted on the command line.

This flag may not be used in conjunction with the `-x`, `-X`, or `-A` flags.

-A <hash>

Same as `-a` above, but does negative matching. This flag may not be used in conjunction with the `-m`, `-M`, or `-A` flags.

-w During any of the matching modes (`-m`, `-M`, `-x`, or `-X`), displays the filename of the known hash that matched the input file.

-n During any of the matching modes (`-m`, `-M`, `-x`, or `-X`), displays only the filenames of any known hashes that were not matched by any of the input files.

-s Enables silent mode. All error messages are suppressed.

-z Enables file size mode. Prepends the hash with a ten digit representation of the size of each file processed. If the file size is greater than 9999999999 bytes (about 9.3GB) the program displays 9999999999 for the size.

-q Quiet mode. File names are omitted from the output.

-0 Uses a NULL character (`/0`) to terminate each line instead of a newline. Useful for processing filenames with strange characters.

-l Enables relative file paths. Instead of printing the absolute path for each file, displays the relative file path as indicated on the command line. This flag may not be used in conjunction with the `-b` flag.

-b Enables bare mode. Strips any leading directory information from displayed filenames.

This flag may not be used in conjunction with the `-l` flag.

-k Enables asterisk mode. An asterisk is inserted in lieu of a second space between the filename and the hash, just like `md5sum` in its binary (`-b`) mode.

-o <bcplsd>

Enables expert mode. Allows the user specify which (and only which) types of files are processed. Directory processing is still controlled with the `-r` flag. The expert mode options

allowed are:
f - Regular files
b - Block Devices
c - Character Devices
p - Named Pipes
l - Symbolic Links
s - Sockets
d - Solaris Doors

- h** Show a help screen and exit.
- v** Show the version number and exit.
- V** Show copyright information and exit.

RETURN VALUE

Returns a bit-wise value based on the success of the operation and the status of any matching operations.

- 0 Success. Note that the program considers itself successful even when it encounters read errors, permission denied errors, or finds directories when not in recursive mode.
- 1 Unused hashes. Under any of the matching modes, returns this value if one or more of the known hashes was not matched by any of the input files.
- 2 Unmatched inputs. Under any of the matching modes, returns this value if one or more of the input values did not match any of the known hashes.
- 64 User error, such as trying to do both positive and negative matching at the same time.
- 128 Internal error, such as memory corruption or uncaught cycle. All internal errors should be reported to the developer! See the section "Reporting Bugs" below.

outguess : Steganography detection suite.

Developed by Niels Provos. Available from
<http://www.outguess.org/>

The following is taken from <http://www.outguess.org/>



What is OutGuess?

OutGuess is a universal steganographic tool that allows the insertion of hidden information into the redundant bits of data sources. The nature of the data source is irrelevant to the core of OutGuess. The program relies on data specific handlers that will extract redundant bits and write them back after modification. In this version the PNM and JPEG image formats are supported. In the next paragraphs, images will be used as concrete example of data objects, though OutGuess can use any kind of data, as long as a handler is provided.

What is Steganography

Steganography is the art and science of hiding that communication is happening. Classical steganography systems depend on keeping the encoding system secret, but modern steganography is detectable only if secret information is known, e.g. a secret key. Because of their invasive nature, steganography systems leave detectable traces within a medium's characteristics. This allows an eavesdropper to detect media that has been modified, revealing that secret communication is taking place. Although the secrecy of the information is not degraded, its hidden nature is revealed, defeating the main purpose of Steganography.

What does OutGuess do differently

For JPEG images, OutGuess preserves statistics based on frequency counts. As a result, statistical tests based on frequency counts are unable to detect the presence of steganographic content. Before embedding data into an image, OutGuess can determine the maximum message size that can be hidden while still being able to maintain statistics based on frequency counts. This approach has been described in

Niels Provos (2001) Defending Against Statistical Steganalysis, 10th USENIX Security Symposium. Washington, DC, August 2001. Available from
<http://www.citi.umich.edu/u/provos/papers/defending.ps>

OutGuess uses a generic iterator object to select which bits in the data should be modified. A seed can be used to modify the behavior of the iterator. It is embedded in the data along with the rest of the message. By altering the seed, OutGuess tries to find a sequence of bits that minimizes the number of changes in the data that have to be made.

Data Embedding

Below you can see an example run of OutGuess. The table gives an explanation of the different columns in the output.

```

$ outguess -k "my secret key" -d hidden.txt demo.jpg out.jpg
Reading demo.jpg....
JPEG compression quality set to 75
Extracting usable bits: 40059 bits
Correctable message size: 21194 bits, 52.91%
Encoded 'snark.bz2': 14712 bits, 1839 bytes
Finding best embedding...
    0: 7467(50.6%)[50.8%], bias 8137(1.09), saved: -13, total: 18.64%
    1: 7311(49.6%)[49.7%], bias 8079(1.11), saved: 5, total: 18.25%
    4: 7250(49.2%)[49.3%], bias 7906(1.09), saved: 13, total: 18.10%
    59: 7225(49.0%)[49.1%], bias 7889(1.09), saved: 16, total: 18.04%
59, 7225: Embedding data: 14712 in 40059
Bits embedded: 14744, changed: 7225(49.0%)[49.1%], bias: 7889, tot:
40032, skip: 25288
Foiling statistics: corrections: 2590, failed: 1, offset: 122.585494 +-
239.664983
Total bits changed: 15114 (change 7225 + bias 7889)
Storing bitmap into data...
Writing foil/out.jpg....

```

Data Retrieval

You can retrieve data from an image in the following way:

```

$ outguess -k "my secret key" -r out.jpg hidden.txt
Reading out.jpg....
Extracting usable bits: 40059 bits
Steg retrieve: seed: 7225, len: 1839

```

Usage

The following is the manpage for outguess

OutGuess 0.2 Universal Stego (c) 1999-2001 Niels Provos

```

outguess [options] [<input file> [<output file>]]
-[sS] <n>      iteration start, capital letter for 2nd dataset
-[iI] <n>      iteration limit
-[kK] <key>    key
-[dD] <name>   filename of dataset
-[eE]          use error correcting encoding
-p <param>    parameter passed to destination data handler
-r            retrieve message from data
-x <n>        number of key derivations to be tried
-m            mark pixels that have been modified
-t            collect statistic information
-F[+-]        turns statistical steganalysis foiling on/off.
                The default is on.

```

NAME

outguess - universal steganographic tool

SYNOPSIS

```
outguess [ -emt ] [ -r ] [ -k key ] [ -F [+ -] ] [ -d datafile ] [ -s
seed ] [ -i limit ] [ -x maxkeys ] [ -p param ] [ inputfile [ output-
file ] ]
```

DESCRIPTION

Outguess is a universal steganographic tool that allows the insertion of hidden information into the redundant bits of data sources. The nature of the data source is irrelevant to the core of outguess. The program relies on data specific handlers that will extract redundant bits and write them back after modification. Currently only the PPM, PNM, and JPEG image formats are supported, although outguess could use any kind of data, as long as a handler were provided.

Outguess uses a generic iterator object to select which bits in the data should be modified. A seed can be used to modify the behavior of the iterator. It is embedded in the data along with the rest of the message. By altering the seed, outguess tries to find a sequence of bits that minimizes the number of changes in the data that have to be made.

A bias is introduced that favors the modification of bits that were extracted from a high value, and tries to avoid the modification of bits that were extracted from a low value.

Additionally, Outguess allows for the hiding of two distinct messages in the data, thus providing plausible deniability. It keeps track of the bits that have been modified previously and locks them. A (23,12,7) Golay code is used for error correction to tolerate collisions on locked bits. Artificial errors are introduced to avoid modifying bits that have a high bias.

OPTIONS

The following command line options, when specified as capital letters, indicate options for the second message.

-F [+ -]

Specifies that OutGuess should preserve statistics based on frequency counts. As a result, no statistical test that is based on frequency counts will be able to detect steganographic content. This option is on by default.

-kK key

Specify the secret key used to encrypt and hide the message in the provided data.

-dD datafile

Specify the filename containing a message to be hidden in the data.

-sS seed

Specify the initial seed the iterator object uses for selecting bits in the redundant data. If no upper limit is specified, the iterator will use this seed without searching for a more optimal embedding.

-iI limit

Specify the upper limit for finding an optimal iterator seed.
The maximum value for the limit is 65535.

`-eE` Use error correction for data encoding and decoding.

Other options that apply to the general execution of outguess:

`-r` Retrieve a message from a data object. If this option is not specified, outguess will embed messages.

`-x maxkeys`
If the second key does not create an iterator object that is successful in embedding the data, the program will derive up to specified number of new keys.

`-p param`
Passes a string as parameter to the destination data handler. For the JPEG image format, this is the compression quality, it can take values between 75 and 100. The higher the quality the more bits to hide a message in the data are available.

`-m` Mark pixels that have been modified.

`-t` Collect statistics about redundant bit usage. Repeated use increases output level.

For embedding messages, you need to specify a source and a destination filename. Outguess determines the data format by the filename extension. If no filenames are specified outguess operates as a filter and assumes the PPM data format.

EXAMPLES

To embed the message hidden.txt into the monkey.jpg image:

```
outguess -k "my secret pass phrase" -d hidden.txt monkey.jpg  
out.jpg
```

And in the other direction:

```
outguess -k "my secret pass phrase" -r out.jpg message.txt
```

will retrieve the hidden message from the image.

If you want to embed a second message, use:

```
outguess -k "secret1" -d hide1.txt -E -K "secret2" -D hide2.txt  
monkey.jpg out.jpg
```

Outguess will first embed hide1.txt and then hide2.txt on top of it, using error correcting codes. The second message hide2.txt can be retrieved with

```
outguess -k "secret2" -e -r out.jpg message.txt
```

pasco: Forensic tool for Internet Explorer Analysis.

Developed by Keith J. Jones. Available from
<http://www.foundstone.com/resources/proddesc/pasco.htm>

The following is from <http://www.foundstone.com/resources/proddesc/pasco.htm>

An Internet Explorer activity forensic analysis tool.

Author: Keith J. Jones, Principal Computer Forensic Consultant; Foundstone, Inc.

keith.jones@foundstone.com

Copyright 2003 (c) by Foundstone, Inc.

<http://www.foundstone.com>

Many important files within Microsoft Windows have structures that are undocumented. One of the principals of computer forensics is that all analysis methodologies must be well documented and repeatable, and they must have an acceptable margin of error. Currently, there are a lack of open source methods and tools that forensic analysts can rely upon to examine the data found in proprietary Microsoft files.

Many computer crime investigations require the reconstruction of a subject's internet activity. Since this analysis technique is executed regularly, we researched the structure of the data found in Internet Explorer activity files (index.dat files). Pasco, the latin word meaning "browse", was developed to examine the contents of Internet Explorer's cache files. The foundation of Pasco's examination methodology is presented in the white paper located here. Pasco will parse the information in an index.dat file and output the results in a field delimited manner so that it may be imported into your favorite spreadsheet program. Pasco is built to work on multiple platforms and will execute on Windows (through Cygwin), Mac OS X, Linux, and *BSD platforms.

Usage: pasco [options] <filename>

Options:

-d Undelete Activity Records

-t Field Delimiter (TAB by default)

Example Usage:

```
# pasco index.dat > index.txt
```

Open index.txt as a TAB delimited file in MS Excel to further sort and filter your results.

rifiuti: "Recycle BIN" analyzer.

Developed by Keith J. Jones. Available from
<http://www.foundstone.com/resources/proddesc/rifiuti.htm>

The following is taken from <http://www.foundstone.com/resources/proddesc/rifiuti.htm>

A Recycle Bin Forensic Analysis Tool.

Author: Keith J. Jones, Principal Computer Forensic Consultant; Foundstone, Inc.

keith.jones@foundstone.com

Copyright 2003 (c) by Foundstone, Inc.

<http://www.foundstone.com>

Many important files within Microsoft Windows have structures that are undocumented. One of the principals of computer forensics is that all analysis methodologies must be well documented and repeatable, and they must have an acceptable margin of error. Currently, there are a lack of open source methods and tools that forensic analysts can rely upon to examine the data found in proprietary Microsoft files.

Many computer crime investigations require the reconstruction of a subject's Recycle Bin. Since this analysis technique is executed regularly, we researched the structure of the data found in the Recycle Bin repository files (INFO2 files). Rifiuti, the Italian word meaning "trash", was developed to examine the contents of the INFO2 file in the Recycle Bin. The foundation of Rifiuti's examination methodology is presented in the white paper located here. Rifiuti will parse the information in an INFO2 file and output the results in a field delimited manner so that it may be imported into your favorite spreadsheet program. Rifiuti is built to work on multiple platforms and will execute on Windows (through Cygwin), Mac OS X, Linux, and *BSD platforms.

Usage: rifiuti [options] <filename>

Options:

-t Field Delimiter (TAB by default)

Example Usage:

```
# rifiuti INFO2 > INFO2.txt
```

Open INFO2.txt as a TAB delimited file in MS Excel to further sort and filter your results.

rkhunter: Rootkit hunter.

Developed by Michael Boelen. Available from <http://www.rootkit.nl/>

The following is taken from http://www.rootkit.nl/projects/rootkit_hunter.html

Rootkit Hunter is an easy-to-use tool which checks machines running UNIX (clones) for the presence of rootkits and other unwanted tools. Rootkits are selfhiding toolkits used by blackhats/crackers/scriptkiddies to avoid the eye of the sysadmin.



The rootkit hunter is capable of detecting the following threats:

55808 Trojan - Variant A	HjC Rootkit	SHV5 Rootkit
ADM W0rm	ignoKit	Sin Rootkit
AjaKit	ImperalsS-FBRK	Slapper
aPa Kit	Irix Rootkit	Sneakin Rootkit
Apache Worm	Kitko	Suckit
Ambient (ark) Rootkit	Knark	SunOS Rootkit
Balaur Rootkit	Li0n Worm	Superkit
BeastKit	Lockit / LJK2	TBD (Telnet BackDoor)
beX2	mod_rootme (Apache backdoor)	TeLeKiT
BOBKit	MRK	T0rn Rootkit
CiNIK Worm (Slapper.B variant)	Ni0 Rootkit	Trojanit Kit
Danny-Boy's Abuse Kit	NSDAP (RootKit for SunOS)	URK (Universal RootKit)
Devil RootKit	Optic Kit (Tux)	VcKit
Dica	Oz Rootkit	Volc Rootkit
Dreams Rootkit	Portacelo	X-Org SunOS Rootkit
Duarawkz Rootkit	R3dstorm Toolkit	zaRwT.KiT Rootkit
Flea Linux Rootkit	RH-Sharpe's rootkit	Anti Anti-sniffer
FreeBSD Rootkit	RSHA's rootkit	LuCe LKM
Fuck`it Rootkit	Scalper Worm	THC Backdoor
GasKit	Shutdown	
Heroin LKM	SHV4 Rootkit	

The following is taken from the command: `rkhunter --help`

Rootkit Hunter 1.2.7, Copyright 2003-2005, Michael Boelen

Rootkit Hunter comes with ABSOLUTELY NO WARRANTY. This is free software, and you are welcome to redistribute it under the terms of the GNU General Public License. See LICENSE for details.

Valid parameters:

<code>--checkall (-c)</code>	: Check system
<code>--createlogfile*</code>	: Create logfile
<code>--cronjob</code>	: Run as cronjob (removes colored layout)
<code>--display-logfile</code>	: Show logfile at end of the output
<code>--help (-h)</code>	: Show this help
<code>--nocolors*</code>	: Don't use colors for output
<code>--report-mode*</code>	: Don't show uninteresting information for reports
<code>--report-warnings-only*</code>	: Show only warnings (lesser output than --report-mode, more than --quiet)
<code>--skip-application-check*</code>	: Don't run application version checks
<code>--skip-keypress*</code>	: Don't wait after every test (non-interactive)
<code>--quick*</code>	: Perform quick scan (instead of full scan)

```

--quiet*                : Be quiet (only show warnings)
--update                : Run update tool and check for database updates
--version               : Show version and quit
--versioncheck          : Check for latest version

--bindir <bindir>*      : Use <bindir> instead of using default binaries
--configfile <file>*   : Use different configuration file
--dbdir <dir>*          : Use <dbdir> as database directory
--rootdir <rootdir>*    : Use <rootdir> instead of / (slash at end)
--tmpdir <tmpdir>*     : Use <tmpdir> as temporary directory

Explicit scan options:
--allow-ssh-root-user*  : Allow usage of SSH root user login
--disable-md5-check*    : Disable MD5 checks
--disable-passwd-check* : Disable passwd/group checks
--scan-knownbad-files*  : Perform besides 'known good' check a 'known bad' check

```

Multiple parameters are allowed

*) Parameter can only be used with other parameters

The following is taken from http://www.rootkit.nl/articles/rootkit_hunter_faq.html

Rootkit Hunter tells me there is something wrong with my system, what to do?

(1) If your system is infected with a rootkit, it's almost impossible to clean it up (lets say with a full warranty it's clean). Never trust a machine which has been infected with a rootkit, because hiding is his main purpose.

A clean install of the system is recommended after backing up the full system. So follow the next steps:

1. Get the host offline
2. Backup your data (as much as possible, including binaries and logfiles)
3. Verify the integrity of this data
4. Install your host with a fresh install
5. Investigate the old log files and the possible used tools. Also investigate the services which were vulnerable at the time of hack.

(2) If just one check fails, it is possible you have a so called false positive. Sometimes this will happen due custom configurations or changed binaries. If so, please validate:

Files:

- "strings <file>" and check for untrusted file paths (things like /dev/.hiddendir)
- recently updated binaries and their original source. If it is due an update, please sent me an URI to the changed file (like a RPM), so I can add new hashes to the databases.
- "file <file>" and compare them with others (especially trusted binaries). If some binaries are linked static and others are all dynamic, than they could have been trojaned.

Other warnings:

If you have a warning about another part of the checks, please fill in the contact form and tell me something about your system configuration.

scalpel: Fast File Carver

Developed by Golden G. Richard III. Available from <http://www.microforensics.com/>



The following is taken from <http://www.microforensics.com/>

Scalpel is a computer forensic tool designed to identify, isolate and recover data artifacts from computer media during forensics investigations. Scalpel searches hard drives, bit-stream images, unallocated space files, or any computer file for selected characteristics, contents or attributes, and produces reports of the locations and contents of artifacts it finds during the electronic discovery process. Scalpel also 'carves' (produces copies of) the found artifacts as individual files.

The following is from the command: `scalpel -h`

```
scalpel version 1.53
```

```
Written by Golden G. Richard III, based on foremost 0.69
```

```
Carves files from a disk image based on file headers and footers.
```

```
Usage: scalpel [-b] [-h|V] [-v] [-s num] [-i <file>] [-o <outputdir>]
        [-n] [-c <config file>] <imgfile> [<imgfile>] ...
```

- b Carve files even if defined footers aren't discovered within
maximum carve size for file type [foremost 0.69 compat mode]
- c Choose configuration file
- h Print this help message and exit
- i Read names of files to dig from a file
- o Set output directory for carved files
- n Don't add extensions to extracted files
- r Find only first of overlapping headers/footers [foremost 0.69 compat mode]
- s Skip n bytes in each disk image before carving
- V Print copyright information and exit
- v Verbose mode

sdd: Specialized dd w/better performance.

Developed by Jörg Schilling. Available from <http://directory.fsf.org/sysadmin/Backup/sdd.html>

The following is taken from <http://directory.fsf.org/sysadmin/Backup/sdd.html>

sdd is a replacement for a program called dd. sdd is much faster than dd in cases where input block size (ibs) is not equal to the output block size (obs). Statistics are more easily understood than those from dd. Timing available, -time option will print transfer speed Timing & Statistics available at any time with SIGQUIT (^) Can seek on input and output Fast null input Fast null output. Support for the RMT (Remote Tape Server) protocol makes remote I/O fast and easy.

The following is the man page for sdd.

NAME

sdd - disk dump and restore to and from tape or file; copy and/or reblock

SYNOPSIS

sdd [option=value] [-flag]

DESCRIPTION

Sdd copies the specified input file to a specified output file performing the requested conversions. The standard input and output are used by default. The input and output block size may be specified to take advantage of raw physical I/O.

After completion, sdd reports the number of whole records, the sum of bytes from partial input and output blocks and the total amount in kilo bytes on input and output.

If ibs and obs differ, sdd is faster than dd due to the use of an intelligent algorithm.

OPTIONS

-help Print a summary of the available options.

if=name

Input is taken from file name; default is stdin.

If sdd is installed suid root, name may be in remote syntax: user@host:filename as in rcp(1) even if invoked by non root users. See SUID NOTES for more information.

To make a file local although it includes a colon (:), the filename must start with: '/', './' or '../'

of=name

Output is taken from file name; default is stdout. Note that sdd creates and truncates the output file by default; therefore the oseek=# option is useless without the -notrunc option except in special cases such as using magnetic tape or disk special files.

If sdd is installed suid root, name may be in remote syntax: user@host:filename as in rcp(1) even if invoked by non root users.

Note that if sdd talks to an old rmt remote tape server, it does

not open a remote file with the O_CREAT open flag because this would be extremely dangerous. If the rmt server on the other side is the rmt server that comes with star or the GNU rmt server, sdd may use the symbolic mode for the open flags. Only the symbolic open modes allow to send all possible open modes in a portable way to remote tape servers.

It is recommended to use the rmt server that comes with star. It is the only rmt server that gives platform independent compatibility with BSD, Sun and GNU rmt clients and it includes security features that may be set up in /etc/default/rmt.

-inull Do not read input from file. This is similar to if=/dev/zero but much faster. Sdd uses a prepared cleared buffer to satisfy writes.

-onull Do not produce any output. This is similar to of=/dev/null but actually does not write to any file.

ibs=#, obs=#, bs=#

Set input block size, output block size or both to # (default 512 Bytes).

cbs=# Set Conversion buffer size to #.

ivsize=#, ovsize=#

Set input volume size or output volume size to #. You can make copies from devices of different size by using this option. If you want to make a copy to a tape having a size of 60 MBytes you should use the option ovsize=60M. If the capacity of the tape is exceeded, sdd will ask for a second volume. In case ivsize is exceeded, if N<cr> is typed, it is treated as an EOF condition and sdd writes any buffered data to output and exits. In case ovsize is exceeded, if N<cr> is typed, sdd stops and the statistics it prints show that more data were read than written.

count=#

Transfer # of input records or until EOF.

iseek=#, iskip=#

Seek/skip the first # Bytes from input before beginning transfer.

oseek=#, oskip=#

Seek/skip the first # Bytes from output before beginning transfer.

seek=#, skip=#

Seek/skip the first # Bytes from input and output before beginning transfer.

ivseek=#, ovseek=#

Seek # Bytes from input/output at the beginning of each input/output volume before beginning transfer. (You can skip labels of disks and floppies with this option.) Note that the isseek/oseek options still work, but only apply to the first volume. Their values are added to the values of ivseek and ovseek.

-notrunc

Do not truncate an already existing output file before beginning

transfer. This enables it to copy one file into another.

-pg Print a dot to stderr each time a record is written to indicate progress.

-time, -t
Report the total time and the transfer rate.

-noerror
Do not stop transfer on I/O errors. Error messages will appear on the screen.

-noerrwrite
Do not write blocks that are not read corretly. Seek on the output to skip the bad block. The output file must be seekable or **-noerrwrite** will not work correctly.

-noseek
Do not seek after I/O errors. This implies **try=1**.

try=# Set retry count to #. Only if **-noerror** was specified. (default 2)

-debug Turn on debugging messages. You can get knowledge about record sizes on tapes with variable record size with this option.

-fill Pad every output record with zeros up to obs. If **ibs** equals **obs**, or only **bs** was specified, every record will be padded with zeros, otherwise this only applies to the last record.

-swab Swaps bytes (except for the last byte in odd block sizes and odd transfers due to EOF).

-block, -unblock
Convert fixed length records to variable records and vice versa.

-lcase, -ucase
Map alphabetics to lower/upper case.

-ascii, -ebcdic, -ibm
Convert EBCDIC to ASCII resp. ASCII to EBCDIC resp. ASCII to the IBM variant of EBCDIC.

-help Prints a short summary of the **sdd** options and exists.

-version
Prints the **sdd** version number string and exists.

EXAMPLES

```
sdd if=/dev/rsd0a of=/dev/nrst8 bs=2x7x17b
```

Copies the disk /dev/rsd0a to the tape /dev/nrst8 using a record size of 2*7*17 blocks. (this is 2 Cylinders.)

```
sdd if=/dev/rsd0c of=/dev/rsd1c seek=1b bs=63k
```

Copy the whole disk sd0 to sd1 preserving the old label on disk sd1.

FILES

None.

SEE ALSO

dd(1), star(1), rmt(1), tr(1), cp(1), copy(1)

DIAGNOSTICS

sdd: Read f records + p bytes (total of x bytes = d.nnk).
sdd: Wrote f records + p bytes (total of x bytes = d.nnk).

The number of full records, the number of bytes in partial records and the total amount of data in KBytes.

With the QUIT signal (usually ^\) the actual state is displayed.

NOTES

Opposed to dd, sdd is able to handle -iseek -oseek -seek as well as -iskip -oskip -skip regardless to the buffer size. You can make a whole physical copy of a disk without copying the label in one pass of sdd.

When numbers are unspecified they are taken to be bytes.

You can make them 'words' (2 bytes) if they are followed by a 'w' or 'W'.

You can make them blocks (512 bytes) if they are followed by a 'b' or 'B'.

You can make them Kbytes (1024 bytes) if they are followed by a 'k' or 'K'.

You can make them Mbytes (1024 * 1024 bytes) if they are followed by a 'm' or 'M'.

You can make them Gbytes (1024 * 1024 * 1024 bytes) if they are followed by a 'g' or 'G'.

A pair of numbers may be separated by '*' or 'x' to indicate a product.

SUID NOTES

If sdd is installed suid root, sdd is able to make connections to remote files for non root users. This is done by using the rcmd(3) interface to get a connection to a rmt(1) server.

Sdd resets its effective uid back to the real user id immediately after setting up the remote connection to the rmt server and before opening any other file.

BUGS

The option iskip=# and oskip=# and skip=# as well as -block and -unblock are not implemented.

It is confusing to allow the use of all additions together with the record counter -count as they are possible with obs=#.

sha1deep: Recursive sha1sum with db lookups.

Developed by Special Agent Jesse Kornblum of the United States Air Force Office of Special Investigations. It is available from <http://md5deep.sourceforge.net/>

sha1deep is a cross-platform tool that can calculate the SHA1 signatures of files. sha1deep is similar to the md5sum but it can also process recursive directories, produce an estimated completion time, compare files to known hash sets, and be set to only process certain types of files. There are also companion tools that will calculate MD5, SHA-256 Tiger, or Whirlpool message digests for files as well.

The following was taken from <http://md5deep.sourceforge.net/manpage.html>

SYNOPSIS

```
sha1deep -v | -V | -h  
sha1deep [-m|-M|-x|-X <file>] [-a|-A <hash>] [-nwzres0lbkq] [-o <fbcp1sd>] [FILES]
```

DESCRIPTION

Computes the hashes, or message digest, for any number of files while optionally recursively digging through the directory structure. Can also take a list of known hashes and display the filenames of input files whose hashes either do or do not match any of the known hashes. Errors are reported to standard error. If no FILES are specified, reads from standard input.

-r Enables recursive mode. All subdirectories are traversed. Please note that recursive mode cannot be used to examine all files of a given file extension. For example, calling md5deep -r *.txt will examine all files in directories that end in .txt.

-e Displays a progress indicator and estimate of time remaining for each file being processed. Time estimates for files larger than 4GB are not available on Windows.

-m <file>

Enables matching mode. The file given should be a list of known hashes. The input files are examined one at a time, and only those files that match the list of known hashes are output. This flag may be used more than once to add multiple sets of known hashes. Acceptable formats for lists of known hashes are plain (such as those generated by md5deep or md5sum), Hashkeeper files, iLook, and the National Software Reference Library (NSRL) as produced by the National Institute for Standards in Technology.

If standard input is used with the -m flag, displays "stdin" if the input matches one of the hashes in the list of known hashes. If the hash does not match, the program displays no output.

This flag may not be used in conjunction with the -x, -X, or -A flags.

-x <file>

Same as the -m flag above, but does negative matching. That is, only those files NOT in the list of known hashes are displayed.

This flag may not be used in conjunction with the `-m`, `-M`, or `-a` flags.

-M and -X <file>

Same as `-m` and `-x` above, but displays the hash for each file that does (or does not) match the list of known hashes.

-a <hash>

Adds a single hash to the list of known hashes used for matching mode, and if not already enabled, enables matching mode. Adding single hashes cannot, by itself, be used to print the hashes of matching files like the `-M` flag does. When used in conjunction with the `-w` flag, the filename displayed is just the hash submitted on the command line.

This flag may not be used in conjunction with the `-x`, `-X`, or `-A` flags.

-A <hash>

Same as `-a` above, but does negative matching. This flag may not be used in conjunction with the `-m`, `-M`, or `-A` flags.

-w During any of the matching modes (`-m`, `-M`, `-x`, or `-X`), displays the filename of the known hash that matched the input file.

-n During any of the matching modes (`-m`, `-M`, `-x`, or `-X`), displays only the filenames of any known hashes that were not matched by any of the input files.

-s Enables silent mode. All error messages are suppressed.

-z Enables file size mode. Prepends the hash with a ten digit representation of the size of each file processed. If the file size is greater than 9999999999 bytes (about 9.3GB) the program displays 9999999999 for the size.

-q Quiet mode. File names are omitted from the output.

-0 Uses a NULL character (`/0`) to terminate each line instead of a newline. Useful for processing filenames with strange characters.

-l Enables relative file paths. Instead of printing the absolute path for each file, displays the relative file path as indicated on the command line. This flag may not be used in conjunction with the `-b` flag.

-b Enables bare mode. Strips any leading directory information from displayed filenames.

This flag may not be used in conjunction with the `-l` flag.

-k Enables asterisk mode. An asterisk is inserted in lieu of a second space between the filename and the hash, just like `md5sum` in its binary (`-b`) mode.

-o <bcplsd>

Enables expert mode. Allows the user specify which (and only which) types of files are processed. Directory processing is still controlled with the `-r` flag. The expert mode options

allowed are:
f - Regular files
b - Block Devices
c - Character Devices
p - Named Pipes
l - Symbolic Links
s - Sockets
d - Solaris Doors

- h** Show a help screen and exit.
- v** Show the version number and exit.
- V** Show copyright information and exit.

RETURN VALUE

Returns a bit-wise value based on the success of the operation and the status of any matching operations.

- 0 Success. Note that the program considers itself successful even when it encounters read errors, permission denied errors, or finds directories when not in recursive mode.
- 1 Unused hashes. Under any of the matching modes, returns this value if one or more of the known hashes was not matched by any of the input files.
- 2 Unmatched inputs. Under any of the matching modes, returns this value if one or more of the input values did not match any of the known hashes.
- 64 User error, such as trying to do both positive and negative matching at the same time.
- 128 Internal error, such as memory corruption or uncaught cycle. All internal errors should be reported to the developer! See the section "Reporting Bugs" below.

sha256deep: Recursive sha1sum with db lookups.

Developed by Special Agent Jesse Kornblum of the United States Air Force Office of Special Investigations. It is available from <http://md5deep.sourceforge.net/>

sha256deep is a cross-platform tool that can calculate the SHA1 signatures of files. sha1deep is similar to the md5sum but it can also process recursive directories, produce an estimated completion time, compare files to known hash sets, and be set to only process certain types of files. There are also companion tools that will calculate MD5, SHA-1, Tiger, or Whirlpool message digests for files as well.

The following was taken from <http://md5deep.sourceforge.net/manpage.html>

SYNOPSIS

```
sha256deep -v | -V | -h  
sha256deep [-m|-M|-x|-X <file>] [-a|-A <hash>] [-nwzres0lbnkq] [-o <fbcp1sd>] [FILES]
```

DESCRIPTION

Computes the hashes, or message digest, for any number of files while optionally recursively digging through the directory structure. Can also take a list of known hashes and display the filenames of input files whose hashes either do or do not match any of the known hashes. Errors are reported to standard error. If no FILES are specified, reads from standard input.

-r Enables recursive mode. All subdirectories are traversed. Please note that recursive mode cannot be used to examine all files of a given file extension. For example, calling md5deep -r *.txt will examine all files in directories that end in .txt.

-e Displays a progress indicator and estimate of time remaining for each file being processed. Time estimates for files larger than 4GB are not available on Windows.

-m <file>

Enables matching mode. The file given should be a list of known hashes. The input files are examined one at a time, and only those files that match the list of known hashes are output. This flag may be used more than once to add multiple sets of known hashes. Acceptable formats for lists of known hashes are plain (such as those generated by md5deep or md5sum), Hashkeeper files, iLook, and the National Software Reference Library (NSRL) as produced by the National Institute for Standards in Technology.

If standard input is used with the -m flag, displays "stdin" if the input matches one of the hashes in the list of known hashes. If the hash does not match, the program displays no output.

This flag may not be used in conjunction with the -x, -X, or -A flags.

-x <file>

Same as the -m flag above, but does negative matching. That is, only those files NOT in the list of known hashes are displayed.

This flag may not be used in conjunction with the -m, -M, or -a flags.

-M and -X <file>

Same as -m and -x above, but displays the hash for each file that does (or does not) match the list of known hashes.

-a <hash>

Adds a single hash to the list of known hashes used for matching mode, and if not already enabled, enables matching mode. Adding single hashes cannot, by itself, be used to print the hashes of matching files like the -M flag does. When used in conjunction with the -w flag, the filename displayed is just the hash submitted on the command line.

This flag may not be used in conjunction with the -x, -X, or -A flags.

-A <hash>

Same as -a above, but does negative matching. This flag may not be used in conjunction with the -m, -M, or -A flags.

-w During any of the matching modes (-m, -M, -x, or -X), displays the filename of the known hash that matched the input file.

-n During any of the matching modes (-m, -M, -x, or -X), displays only the filenames of any known hashes that were not matched by any of the input files.

-s Enables silent mode. All error messages are suppressed.

-z Enables file size mode. Prepends the hash with a ten digit representation of the size of each file processed. If the file size is greater than 9999999999 bytes (about 9.3GB) the program displays 9999999999 for the size.

-q Quiet mode. File names are omitted from the output.

-0 Uses a NULL character (/0) to terminate each line instead of a newline. Useful for processing filenames with strange characters.

-l Enables relative file paths. Instead of printing the absolute path for each file, displays the relative file path as indicated on the command line. This flag may not be used in conjunction with the -b flag.

-b Enables bare mode. Strips any leading directory information from displayed filenames.

This flag may not be used in conjunction with the -l flag.

-k Enables asterisk mode. An asterisk is inserted in lieu of a second space between the filename and the hash, just like md5sum in its binary (-b) mode.

-o <bcplsd>

Enables expert mode. Allows the user specify which (and only which) types of files are processed. Directory processing is still controlled with the -r flag. The expert mode options

allowed are:
f - Regular files
b - Block Devices
c - Character Devices
p - Named Pipes
l - Symbolic Links
s - Sockets
d - Solaris Doors

- h** Show a help screen and exit.
- v** Show the version number and exit.
- V** Show copyright information and exit.

RETURN VALUE

Returns a bit-wise value based on the success of the operation and the status of any matching operations.

- 0 Success. Note that the program considers itself successful even when it encounters read errors, permission denied errors, or finds directories when not in recursive mode.
- 1 Unused hashes. Under any of the matching modes, returns this value if one or more of the known hashes was not matched by any of the input files.
- 2 Unmatched inputs. Under any of the matching modes, returns this value if one or more of the input values did not match any of the known hashes.
- 64 User error, such as trying to do both positive and negative matching at the same time.
- 128 Internal error, such as memory corruption or uncaught cycle. All internal errors should be reported to the developer! See the section "Reporting Bugs" below.

stegdetect: Steganography detection suite.

Developed by Niels Provos. Available from
<http://www.outguess.org/detection.php>

The following is taken from
<http://www.outguess.org/detection.php>



Stegdetect is an automated tool for detecting steganographic content in images. It is capable of detecting several different steganographic methods to embed hidden information in JPEG images.

Currently, the detectable schemes are

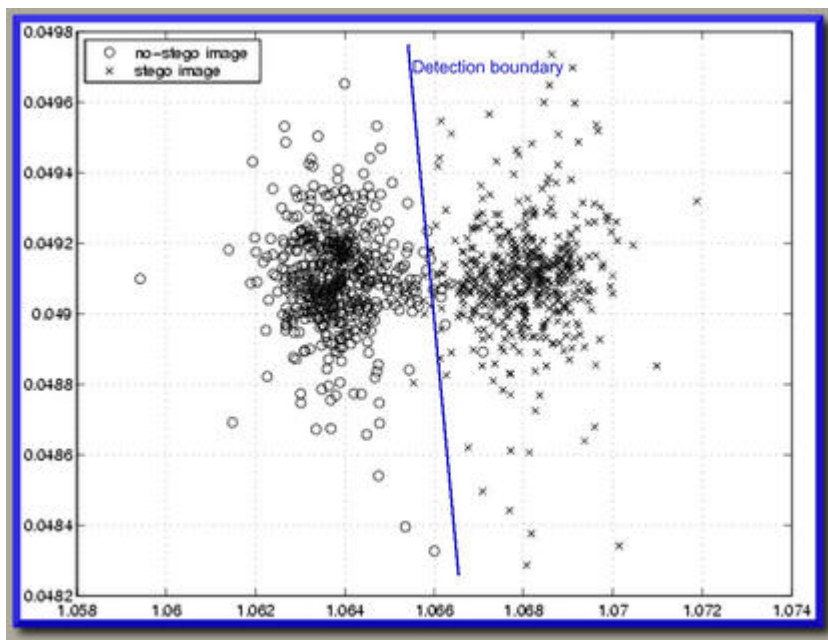
- jsteg,
- jphide (unix and windows),
- invisible secrets,
- outguess 01.3b,
- F5 (header analysis),
- appendX and camouflage.

Stegbreak is used to launch dictionary attacks against JSteg-Shell, JPHide and OutGuess 0.13b.

Automated Detection of New Steganographic Methods

Stegdetect 0.6 supports linear discriminant analysis. Given a set of normal images and a set of images that contain hidden content by a new steganographic application, Stegdetect can automatically determine a linear detection function that can be applied to yet unclassified images.

Linear discriminant analysis computes a dividing hyperplane that separates the no-stego images from the stego images. The hyperplane is characterized as a linear function. The learned function can be saved for later use on new images.



Stegdetect supports several different feature vectors and automatically computes receiver operating characteristic which can be used to evaluate the quality of the automatically learned detection function.

Example

```
# stegdetect *.jpg
cold_dvd.jpg : outguess(old) (***) jphide(*)
dscf0001.jpg : negative
dscf0002.jpg : jsteg(***)
```

```
dscf0003.jpg : jphide(***)
[...]
# stegbreak -tj dscf0002.jpg
Loaded 1 files...
dscf0002.jpg : jsteg(wonderland)
Processed 1 files, found 1 embeddings.
Time: 36 seconds: Cracks: 324123, 8915 c/s
```

For more information on how stegdetect works and on how to use it can be found see:

Niels Provos and Peter Honeyman (2003) Hide and Seek: An Introduction to Steganography, IEEE Security & Privacy Magazine, May/June. Available from <http://niels.xtdnet.nl/papers/practical.pdf>

The following is the man page for stegdetect

NAME

stegdetect - finds image files with steganographic content

SYNOPSIS

```
stegdetect [-qhnV] [-s float] [-C num,tfname] [-c file ... name]
           [-D file] [-d num] [-t tests] [file ...]
```

DESCRIPTION

The stegdetect utility analyses image files for steganographic content. It runs statistical tests to determine if steganographic content is present, and also tries to find the system that has been used to embed the hidden information.

The options are as follows:

- q Only reports images that are likely to have steganographic content.
- h Only calculates the DCT histogram. Use the -d option to display the values.
- n Enables checking of JPEG header information to suppress false positives. If enabled, all JPEG images that contain comment fields will be treated as negatives. OutGuess checking will be disabled if the JFIF marker does not match version 1.1.
- V Displays the version number of the software.
- s float Changes the sensitivity of the detection algorithms. Their results are multiplied by the specified number. The higher the number the more sensitive the test will become. The default is 1.
- C num,tfname Feature vectors are being extracted from the images. The argument num can either be zero or one. A zero indicates that the provided images do not contain steganographic content, a one indicates that they do. The argument tfname is the name of transform used for feature extraction. The features vectors

are printed to stdout.

- c file Reads the data created by the -C options and computes the necessary values to detect steganographic content in yet unknown images. The option can be used multiple times. It expects that the name of the scheme provided as additional argument. The result is a decision object that can be used with the -D option. The decision object contains a the parameters for a linear discriminant function based on the Neyman-Pearson theorem.
- D file Reads a decision object that contains detection information about a new steganographic scheme.
- d num Prints debug information.
- t tests Sets the tests that are being run on the image. The following characters are understood:
 - j Tests if information has been embedded with jsteg.
 - o Tests if information has been embedded with outguess.
 - p Tests if information has been embedded with jphide.
 - i Tests if information has been hidden with invisible secrets.
 - f Tests if information has been hidden with F5.
 - F Tests if information has been hidden with F5 using a more sophisticated but fairly slow detection algorithm.
 - a Tests if information has been added at the end of file, for example by camouflage or appendX.

The default value is jopifa.

The stegdetect utility indicates the accuracy of the detection with a number of stars behind the detected system. If no filenames have been specified, stegdetect will read the filenames from stdin.

EXAMPLES

```
stegdetect -t p auto.jpg
```

Tries to detect the presence of jphide embedded information in auto.jpg.

ERRORS

stegdetect works only for JPEG images.

Currently, there is no support for parameter training. The only exported knob is the sensitivity level. Future versions will export all detection parameters via a configuration file.

wipe: Secure file deletion.

Developed by Berke Durak. Available from <http://abaababa.ouvaton.org/wipe/>

The following is taken from <http://abaababa.ouvaton.org/wipe/wipe.1.html>

SYNOPSIS

```
wipe [options] path1 path2 ... pathn
```

DESCRIPTION

Recovery of supposedly erased data from magnetic media is easier than what many people would like to believe. A technique called Magnetic Force Microscopy (MFM) allows any moderately funded opponent to recover the last two or three layers of data written to disk; **wipe** repeatedly overwrites special patterns to the files to be destroyed, using the fsync() call and/or the O_SYNC bit to force disk access. In normal mode, 34 patterns are used (of which 8 are random). These patterns were recommended in an article from Peter Gutmann (pgut001@cs.auckland.ac.nz) entitled "Secure Deletion of Data from Magnetic and Solid-State Memory". A quick mode allows you to use only 4 passes with random patterns, which is of course much less secure.

IMPORTANT WARNING -- READ CAREFULLY

The author, the maintainers or the contributors of this package can NOT be held responsible in any way if **wipe** destroys something you didn't want it to destroy. Let's make this very clear. I want you to assume that this is a nasty program that will wipe out parts of your files that you didn't want it to wipe. So whatever happens after you launch **wipe** is your entire responsibility. In particular, no one guarantees that **wipe** will conform to the specifications given in this manual page.

Similarly, we cannot guarantee that **wipe** will actually erase data, or that wiped data is not recoverable by advanced means. So if nasties get your secrets because you sold a wiped harddisk to someone you don't know, well, too bad for you.

The best way to sanitize a storage medium is to subject it to temperatures exceeding 1500K. As a cheap alternative, you might use **wipe** at your own risk. Be aware that it is very difficult to assess whether running **wipe** on a given file will actually wipe it -- it depends on an awful lot of factors, such as : the type of file system the file resides on (in particular, whether the file system is a journaling one or not), the type of storage medium used, and the least significant bit of the phase of the moon.

Wiping over NFS or over a journaling filesystem (ReiserFS etc.) will most probably not work.

Therefore I strongly recommend to call **wipe** directly on the corresponding block device with the appropriate options. However *THIS IS AN EXTREMELY DANGEROUS THING TO DO*. Be sure to be sober. Give the right options. In particular : don't wipe a whole harddisk (eg. wipe -kD /dev/hda is bad) since this will destroy your master boot record. Bad idea. Prefer wiping partitions (eg. wipe -kD /dev/hda2) is good, provided, of course, that you have backed up all necessary data.

COMMAND-LINE OPTIONS

-f (force; disable confirmation query)

By default **wipe** will ask for confirmation, indicating the number of regular and special files and directories specified on the command line. You must type "yes" for confirmation, "no" for rejection. You can disable the confirmation query with the **-f** (force) option.

-r (recurse into subdirectories)

Will allow the removal of the entire directory tree. Symbolic links are not followed.

-c (chmod if necessary)

If a file or directory to be wiped has no write permissions set, will do a chmod to set the permission.

-i (informational, verbose mode)

This enables reporting to stdout. By default all data is written to stderr.

-s (silent mode)

All messages, except the confirmation prompt and error messages, are suppressed.

-q (quick wipe)

If this option is used, **wipe** will only make (by default) 4 passes on each file, writing random data. See option **-Q**

-Q <number-of-passes>

Sets the number of passes for quick wiping. Default is 4.

-a (abort on error)

The program will exit with EXIT_FAILURE if a non-fatal error is encountered.

-R (set random device OR random seed command)

With this option which requires an argument you can specify an alternate /dev/random device, or a command whose standard output will be hashed using MD5-hashed. The distinction can be made using the **-S** option.

-S (random seed method)

This option takes a single-character argument, which specifies how the random device/random seed argument is to be used. The default random device is /dev/random. It can be set using the **-R** option.

The possible single-character arguments are:

r

If you want the argument to be treated like a regular file/character device. This will work with /dev/random, and might also work with FIFOs and the like.

c

If you want the argument to be executed as a command. The output from the command will be hashed using MD5 to provide the required seed. See the WIPE_SEEDPIPE environment variable for more info.

p

If you want wipe to get its seed by hashing environment variables, the current date and time, its process id. etc. (the random device argument will not be used). This is of course the least secure setting.

-M (select pseudo-random number generator algorithm)

During the random passes, **wipe** overwrites the target files with a stream of binary data, created by the following choice of algorithms:

l

will use (depending on your system) your libc's random() or rand() pseudorandom generator. Note that on most systems, rand() is a linear congruential generator, which is awfully weak. The choice is made at compile-time with the HAVE_RANDOM define (see the Makefile).

a

will use the Arcfour stream cipher as a PRNG. Arcfour happens to be compatible with the well-known RC4 cipher. This means that under the same key, Arcfour generates exactly the same stream as RC4...

r

will use the fresh RC6 algorithm as a PRNG; RC6 is keyed with the 128-bit seed, and then a null block is repeatedly encrypted to get the pseudo-random stream. I guess this could be quite secure. Of course RC6 with 20 rounds is slower than random(); the compile-time option WEAK_RC6 allows you to use a 4-round version of RC6, which is faster. In order to be able to use RC6, wipe must be compiled with ENABLE_RC6 defined; see the Makefile for warnings about patent issues.

In all cases the PRNG is seeded with the data gathered from the random device (see -R and -S options).

-l <length>

As there can be some problems in determining the actual size of a block device (as some devices do not even have fixed sizes, such as floppy disks or tapes), you might need to specify the size of the device by hand; <length> is the device capacity expressed as a number of bytes. You can use **K** (Kilo) to specify multiplication by 1024, **M** (Mega) to specify multiplication by 1048576, **G** (Giga) to specify multiplication by 1073741824 and **b** (block) to specify multiplication by 512. Thus

$$1024 = 2b = 1K$$

$$20K33 = 20480+33 = 20513$$

$$114M32K = 114*1024*1024+32*1024.$$

-o <offset>

This allows you to specify an offset inside the file or device to be wiped. The syntax of <offset> is the same as for the -l option.

-e

Use exact file size: do not round up file size to wipe possible remaining junk on the last block.

-Z

Don't try to wipe file sizes by repeatedly halving the file size. Note that this is only attempted on regular files so there is no use if you use **wipe** for cleaning a block or special device.

-F

Don't try to wipe file names. Normally, **wipe** tries to cover file names by renaming them; this does NOT guarantee that the physical location holding the old file name gets overwritten. Furthermore, after renaming a file, the only way to make sure that the name change is physically carried out is to call sync (), which flushes out ALL the disk caches of the system, whereas for reading and writing one can use the O_SYNC bit to get synchronous I/O for one

file. As `sync ()` is very slow, calling `sync ()` after every `rename ()` makes filename wiping extremely slow.

- k**
Keep files: do not unlink the files after they have been overwritten. Useful if you want to wipe a device, while keeping the device special file. This implies **-F**.
- D**
Dereference symlinks: by default, **wipe** will never follow symlinks. If you specify **-D** however, **wipe** will consent to, well, wipe the targets of any symlinks you might happen to name on the command line. You can't specify both **-D** and **-r** (recursive) options, first because of possible cycles in the symlink-enhanced directory graph, I'd have to keep track of visited files to guarantee termination, which, you'll easily admit, is a pain in C, and, second, for fear of having a (surprise!!) block device buried somewhere unexpected.
- v**
Show version information and quit.
- h**
Display help.

EXAMPLES

wipe -rcf /home/berke/plaintext/

Wipe every file and every directory (option **-r**) listed under `/home/berke/plaintext/`, including `/home/berke/plaintext/`.

Regular files will be wiped with 34 passes and their sizes will then be halved a random number of times. Special files (character and block devices, FIFOs...) will not. All directory entries (files, special files and directories) will be renamed 10 times and then unlinked. Things with inappropriate permissions will be `chmod()`'ed (option **-c**). All of this will happen without user confirmation (option **-f**).

wipe -kq /dev/hda3

Assuming `/dev/hda3` is the block device corresponding to the third partition of the master drive on the primary IDE interface, it will be wiped in quick mode (option **-q**) i.e. with four random passes. The inode won't be renamed or unlinked (option **-k**). Before starting, it will ask you to type ```yes"`.

wipe -kqD /dev/floppy

Since **wipe** never follows symlinks unless explicitly told to do so, if you want to wipe `/dev/floppy` which happens to be a symlink to `/dev/fd0u1440` you will have to specify the **-D** option. Before starting, it will ask you to type ```yes"`.

wipe -rfi >wipe.log /var/log/*

Here, **wipe** will recursively (option **-r**) destroy everything under `/var/log`, excepting `/var/log`. It will not attempt to `chmod()` things. It will however be verbose (option **-i**). It won't ask you to type ```yes"` because of the **-f** option.

wipe -Kq -l 1440k /dev/fd0

Due to various idiosyncracies of the operating system, it's not always easy to obtain the number of bytes a given device might contain (in fact, that quantity can be variable). This is why you sometimes need to tell **wipe** the amount of bytes to destroy. That's what the **-l**

option is for. Plus, you can use b,K,M and G as multipliers, respectively for 2^9 (512), 2^{10} (1024 or a Kilo), 2^{20} (a Mega) and 2^{30} (a Giga) bytes. You can even combine more than one multiplier !! So that 1M416K = 1474560 bytes.

BUGS/LIMITATIONS

Wipe should work on harddisks and floppy disks; however the internal cache of some harddisks might prevent the necessary writes to be done to the magnetic surface. It would be funny to use it over NFS. Under CFS (Cryptographic File System) the `fsync()` call has no effect; wipe has not much use under it anyway - use wipe directly on the corresponding encrypted files. Also, under Linux, when using a device mounted thru a loopback device, synchronous I/O does not get propagated cleanly.

For wiping floppy disks, at least under Linux, there is no way, besides obscure floppy-driver specific `ioctl`'s to determine the block size of the disk. In particular, the `BLKGETSIZE` `ioctl` is not implemented in the floppy driver. So, for wiping floppies, you must specify the size of the floppy disk using the `-l` option, as in the last example. This option is normally not needed for other fixed block devices, like IDE and SCSI devices.

File name wiping is implemented since version 0.12. I don't know how efficient it is. It first changes the name of the file to a random- generated name of same length, calls `sync ()`, then changes the name to a random-generated name of maximal length.

File size wiping is implemented by repeatedly truncating the file to half of its size, until it becomes empty; `sync ()` is called between such operations.

Note that it is still not possible to file creation date and permission bits portably. A wipe utility working at the block device level could be written using the `ext2fs` library.



Static Binaries

The Need for Static Binaries

When performing an incident response, it is possible that the command commands on the target machine has been compromised, and modified to hide signs that the system has been attacked. Rootkits will often replace command system utilities such as *ls*, *dir*, and *ps*, with modified version to prevent users from detecting rogue processes and files. Even the command dynamic libraries on the target machine can not be trusted.

That is where static binaries come in. They are complete, self-contained executables that do not rely on or use any of the commands on the target system. Since these commands are on a CD-ROM, they can not be compromised or replace by the viruses, worms or rootkits. These are trusted tools.

One of the more common ways to determine if a system has been compromised is to compare the output of the *ps* command from the target system with the static binary *ps* command on the helix CD. If there is a difference, it is possible that the target system has been compromised.

These commands can be run on a live system without rebooting. Once the CD-ROM is mounted, change into the \Static-Binaries and then change into the specific directory for the system being investigated. Helix provides static binaries for Windows, Linux and Solaris.



Windows

On a windows system, these tools are available, as well the Helix graphical user interface. The GUI also provides a number of other incident response tools.

The Win32 binaries are GNU from <http://unxutils.sourceforge.net>, courtesy of Karl M. Syring.

These tools are now all located under the /ir directory in respective directories.

bunzip2.exe	diff.exe	id.exe	pwd.exe	uniq.exe
cat.exe	du.exe	less.exe	rm.exe	unrar.exe
chgrp.exe	echo.exe	libfl.a	rmdir.exe	unzip.exe
chmod.exe	env.exe	libfl.lib	sed.exe	uudecode.exe
chown.exe	expand.exe	ln.exe	sleep.exe	uuencode.exe
cksum.exe	find.exe	ls.exe	sort.exe	wc.exe
compress.exe	fsplit.exe	mkdir.exe	su.exe	which.exe
cp.exe	gawk.exe	mv.exe	sync.exe	whoami.exe
csplit.exe	grep.exe	mvdir.exe	tail.exe	zip.exe
cut.exe	gunzip.exe	pathchk.exe	tar.exe	
date.exe	gzip.exe	pclip.exe	touch.exe	
df.exe	head.exe	printenv.exe	uname.exe	



On a Linux system, in the \Static-Binaries directory, there is a shell script called linux-ir.sh. This is a simple incident response shell script that will collect system information using the static binaries. The output will be displayed on the console, but can be redirected using the standard Linux >> option.

```
linux-ir.sh >> /mnt/sda1/IRoutput.txt
```

This will save the output to the IRoutput.txt on the /mnt/sda1 device. The /mnt/sda1 must be mounted as read/write.

The Linux Static Binaries are from <http://www.e-fense.com>, courtesy of Drew Fahey.

Directory: \Static-Binaries\linux_x86

ald	env	kill	pathchk	strings
arch	ex	last	pcat	stty
arp	fatback	lastlog	pinky	su
bash	ffind	ldd	pr	sync
cat	file	lde	printenv	tail
chgrp	fls	less	procinfo	tee
chmod	fmt	link	ps	top
chown	foremost	ln	pstree	touch
chroot	fsstat	logname	pwd	tsort
cksum	fuser	ls	rarp	tty
clear	gdb	lsof	read_data	umount
cp	grep	mac-robber	readelf	uname
csplit	halt	mactime	readlink	unexpand
cut	head	md5deep	reset	uniq
date	hexdump	md5sum	rm	unlink
dcalc	hfind	memdump	rmdir	unrm
dcat	hostid	mkdir	route	uptime
dcgen	hostname	mmls	search_data	users
dd	icat	mmstat	seq	utmpdump
df	id	more	sha1	vdir
diff	ifconfig	mount	sha1sum	vi
disk_sreset	ifind	mv	sleep	w
disk_stat	ils	nc	sort	wc
dls	img_stat	netstat	sorter	whereis
dmesg	istat	nice	split	who
dstat	jcat	nohup	srch_strings	whoami
du	jls	objdump	stat	
echo	kern_check	od	strace	



Solaris

For an Intel-based Solaris x86 system, in the \Static-Binaries directory, there is a shell script called solaris-ir.sh. This is a simple incident response shell script that will collect system information using the static binaries. The output will be displayed on the console, but can be redirected using the standard Unix >> option.

```
solaris-ir.sh >> /mnt/sda1/IRoutput.txt
```

This will save the output to the IRoutput.txt on the /mnt/sda1 device. The /mnt/sda1 must be mounted as read/write.

The Solaris Binaries were obtained from <http://www.incident-response.org>, courtesy of Rob Lee.

Directory: \Static-Binaries\solaris_2.7

cat	echo	lastcomm	printenv	uname
chgrp	env	ln	pwd	uniq
chmod	factor	logname	rm	unrm
chown	file	ls	rmdir	uptime
chroot	gunzip	lsof	rmt	users
cksum	gzip	md5	sort	wc
cp	head	md5sum	split	who
cut	hostid	mkdir	su	whoami
date	hostname	mknod	sum	zcat
dd	icat	mv	sync	
df	id	nc	tail	
dirname	ils	od	tar	
du	join	pcat	touch	



FAQ

What is Helix?

Helix is a bootable CD originally based upon Knoppix, with an emphasis on Incident Response & Computer Forensics.

What are the minimum requirements to run Helix?

Helix needs lots of RAM and a x86 architecture (Intel, AMD, etc.). You might get it running on an a system with at least 48MB RAM, but don't expect much (like a GUI). You really need a Pentium class computer with at least 128MB RAM. The more RAM the better.

How do you install Helix?

In short you do not install Helix.

- 1) Burn the cd image (Helix.iso) to a CD.
- 2) Make sure that your machine can boot from a CD. (check the BIOS)
- 3) Reboot the machine with the CD in the CD-ROM drive.
- 4) Use Helix.

If you want a permanent installation of Helix you have the option of putting it on your harddrive. There is a script that will accomplish that called knx2hd. Keep in mind that this will destroy all existing data on the partition you install it on.

I get an error message "ERROR: Only one processor found"

This message doesn't matter. Just ignore it. The Helix kernel can handle multi-processor systems, and can in some situations think that your system may be multi-processor when it is not. This is not a problem.

I get the error Can't find Knoppix filesystem. then it drops me to a limited shell.

After isolinux starts up the first thing Helix wants to do is uncompress the filesystem. Helix probes for the CD on all SCSI and IDE buses. If it can't find it you'll get the error above. *This is fixed in all versions since 1.4

For Transmeta laptops and some Sonys with PCMCIA cd drives try: `helix ide2=0x180 nopcmcia`
You can also try `helix nodma` and/or `helix failsafe`

How is Helix licensed?

Helix is based off of the original Knoppix distribution and retains all of the original licenses from that distribution. All additions that I have made are covered under GPL or by the licenses of the prospective authors.

What's the root password?

There is no root password. This is built into the default Knoppix distribution that Helix is based on. If you need root access, you do one of the following:

- 1) run the command using 'sudo'
- 2) Run 'sudo su'

Does the Helix ISO has a virus or Trojan? My AV program picks them up.

Helix does not have any viruses or trojans. It is a false positive of your virus scanner. Your virus scanner is picking up tools on the CD (like Foundstones tools) which are security tools designed to find the viruses/trojans that are being picked up by your virus scanner. They have the same signature. Rest assured there are no viruses/trojans/backdoors on Helix.

Getting More Help

Since Helix is free, it comes without any support. If you have questions read the changelog (<http://www.e-fense.com/helix/changelog.php>), which will tell you what has been changed since previous versions, and check out the forums (<http://www.e-fense.com/helix/forum/index.php>) for more information.



Practice Labs

In this section, we provide the learner with several laboratories that they can use to practice their skills.

Lab	Title
1a	Create an Image of a suspect Floppy Disk (Windows, Live Acquisition, dd)
1b	Create an Image of a suspect Floppy Disk (Windows, FTK Imager)
2	Create a Floppy Disk from a suspect Image (Windows, FTK Imager)

Upcoming labs....

Lab 3 – Preview Image of suspect Floppy disk

Lab 4 – Create an Image of suspect hard drive using netcat

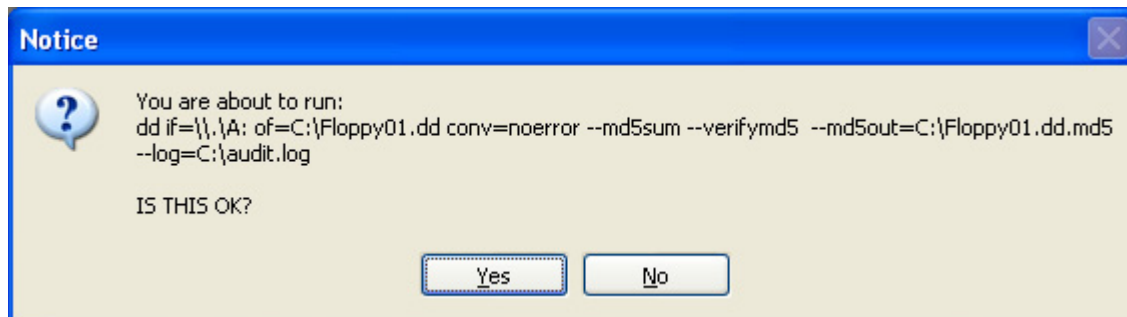
Lab 1a - Create an Image of a suspect Floppy Disk (Windows, Live Acquisition, dd)

You have been given a suspect's floppy disk, and you want create an image of it. On your system, insert the Helix CD, and once the menu comes up, select the icon for "Acquire a "live" image of a Windows System using dd". Before inserting the floppy disk in the drive, be sure it is write protected.

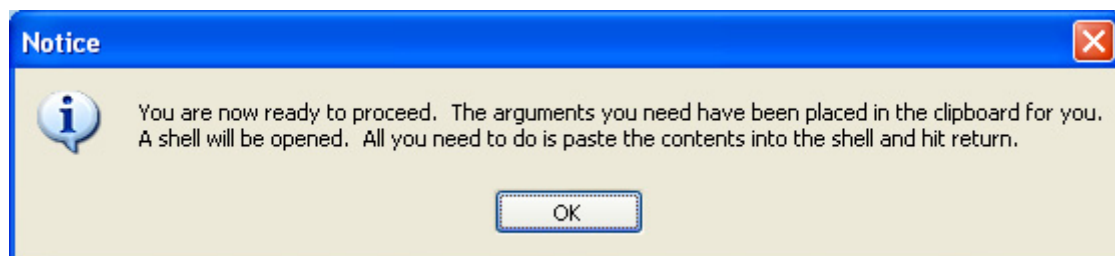
Our source will be the A:\ drive, our destination will be C:\, and our image name will be "Floppy01.dd".



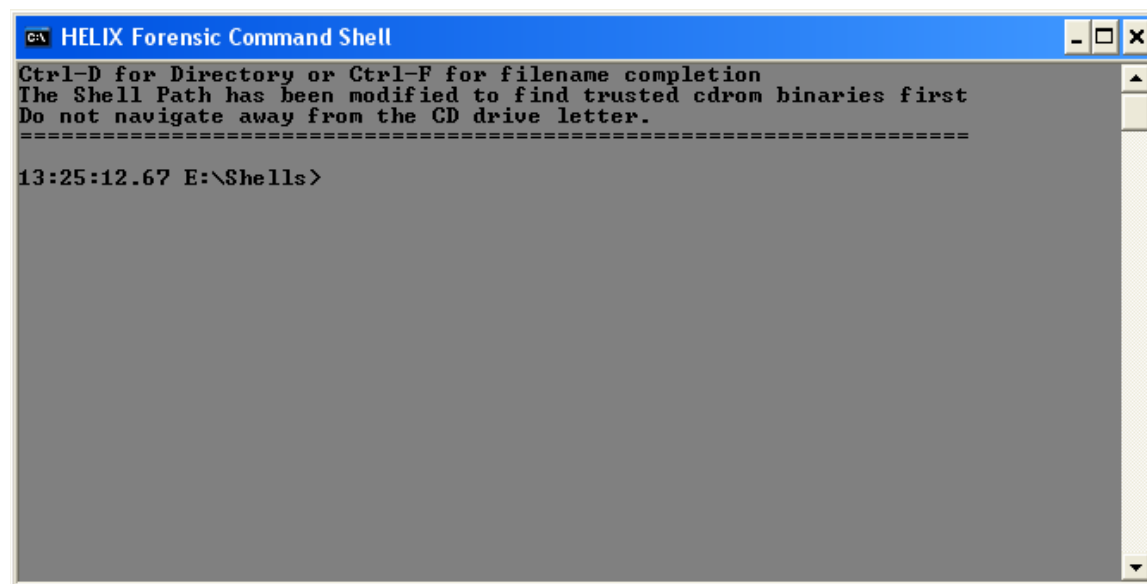
Press the "Start Helix Acquisition" button, and you will be presented with a command preview box



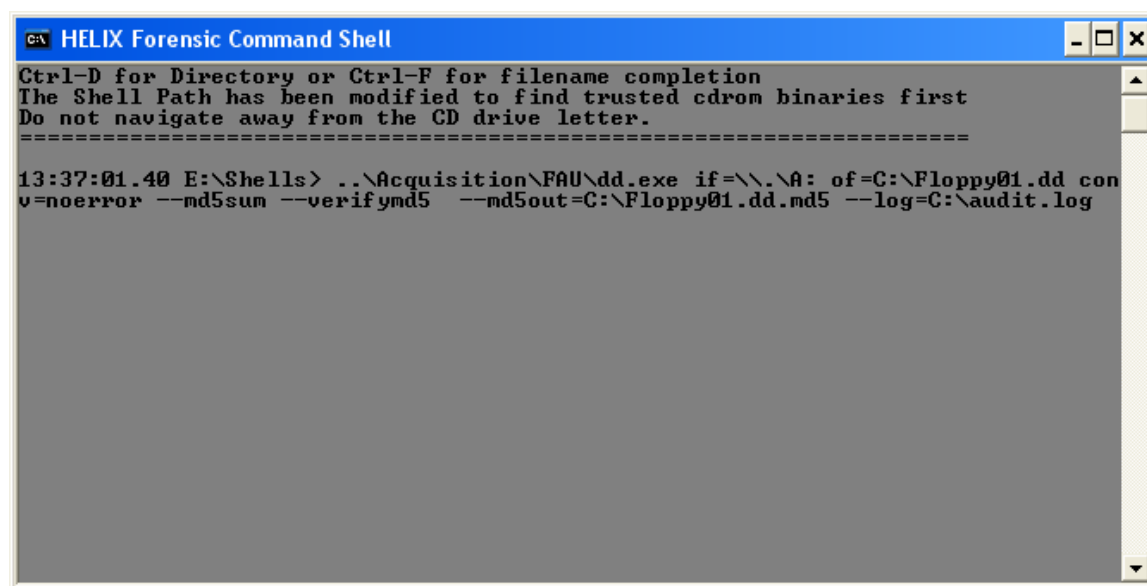
Click “yes”. You will now receive an instruction screen explaining what you should do next:



Click “Ok”. The forensic shell tool will open:



Once the shell opens, right-click inside the shell, and select “paste” from the context menu that appears. The command line will be pasted into the shell.



Press “Enter” to execute the command. After about a minute the command will finish.

There will now be 3 files in the destination directory:

- Floppy01.dd – the image of the floppy disk
- Floppy01.dd.md5 – a file containing the MD5 of the image file.
- Audit.log – a file containing the command and the output of the program.

```
Forensic Acquisition Utilities, 1, 0, 0, 1035
dd, 3, 16, 2, 1035
Copyright (C) 2002-2004 George M. Garner Jr.

Command Line: ..\Acquisition\FAU\dd.exe if=\\.\A: of=C:\Floppy01.dd conv=noerror
--md5sum --verifymd5 --md5out=C:\Floppy01.dd.md5 --log=C:\audit.log
Based on original version developed by Paul Rubin, David MacKenzie, and Stuart
Kemp
Microsoft Windows: Version 5.1 (Build 2600.Professional Service Pack 1)

29/09/2005 04:39:53 (UTC)
Current User: TAL_MC\bj gleason

unable to display device infoCopying \\.\A: to C:\Floppy01.dd...
\1d32a686b7675c7a4f88c15522738432 [\\.\A:] *C:\Floppy01.dd

Verifying output file...
\1d32a686b7675c7a4f88c15522738432 [\\.\A:] *C:\Floppy01.dd
The checksums do match.

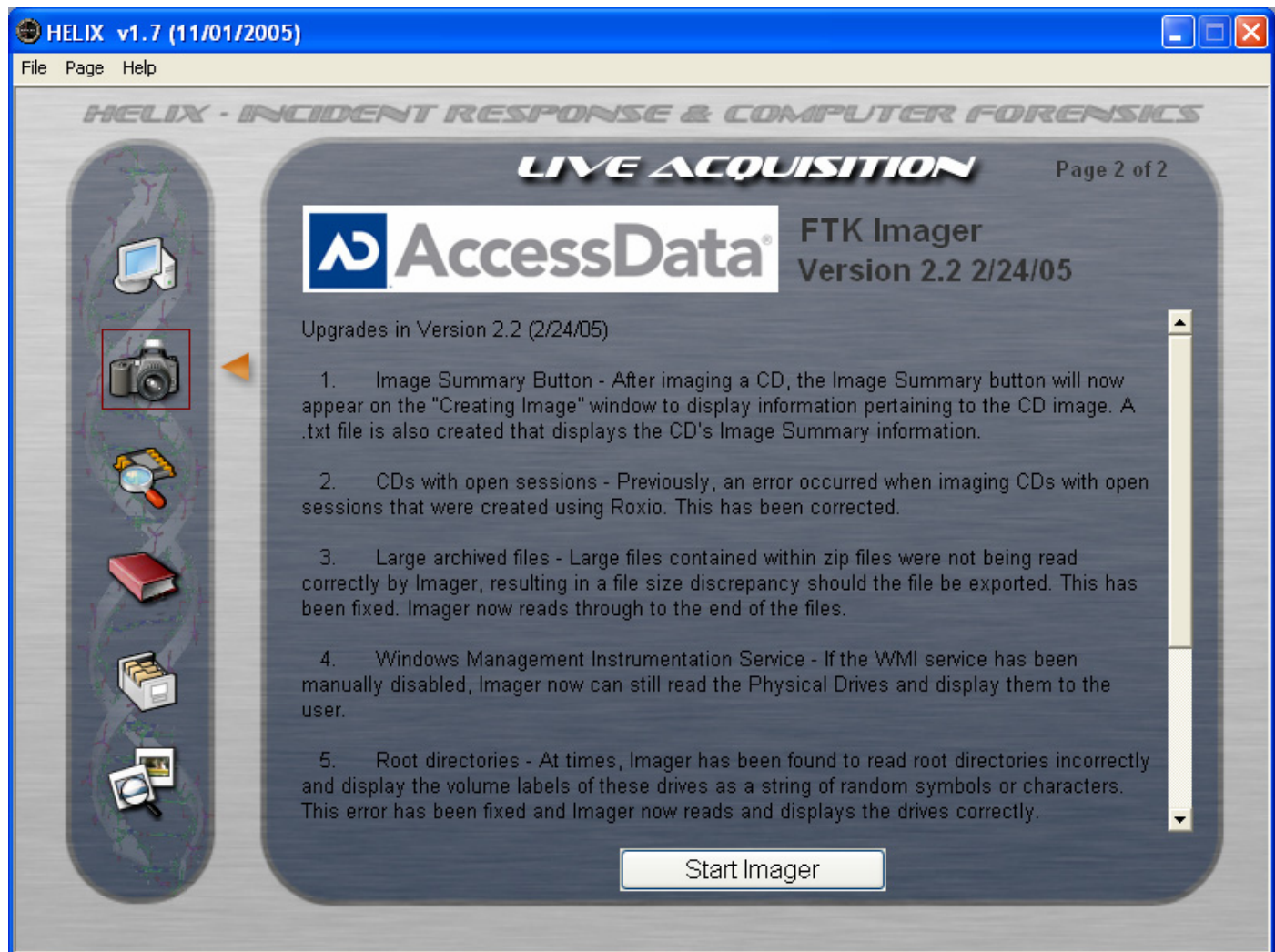
Output C:\Floppy01.dd 1474560/1474560 bytes (compressed/uncompressed)
360+0 records in
360+0 records out
```

You should examine the Audit.log file. If all went well, you should see that both the MD5 hashes match, and you will see the message “The checksums do match”. If they match, that means you have an accurate copy of the evidence. If they don’t match, that typically means you have a bad disk, and the drive had a problem reading the floppy.

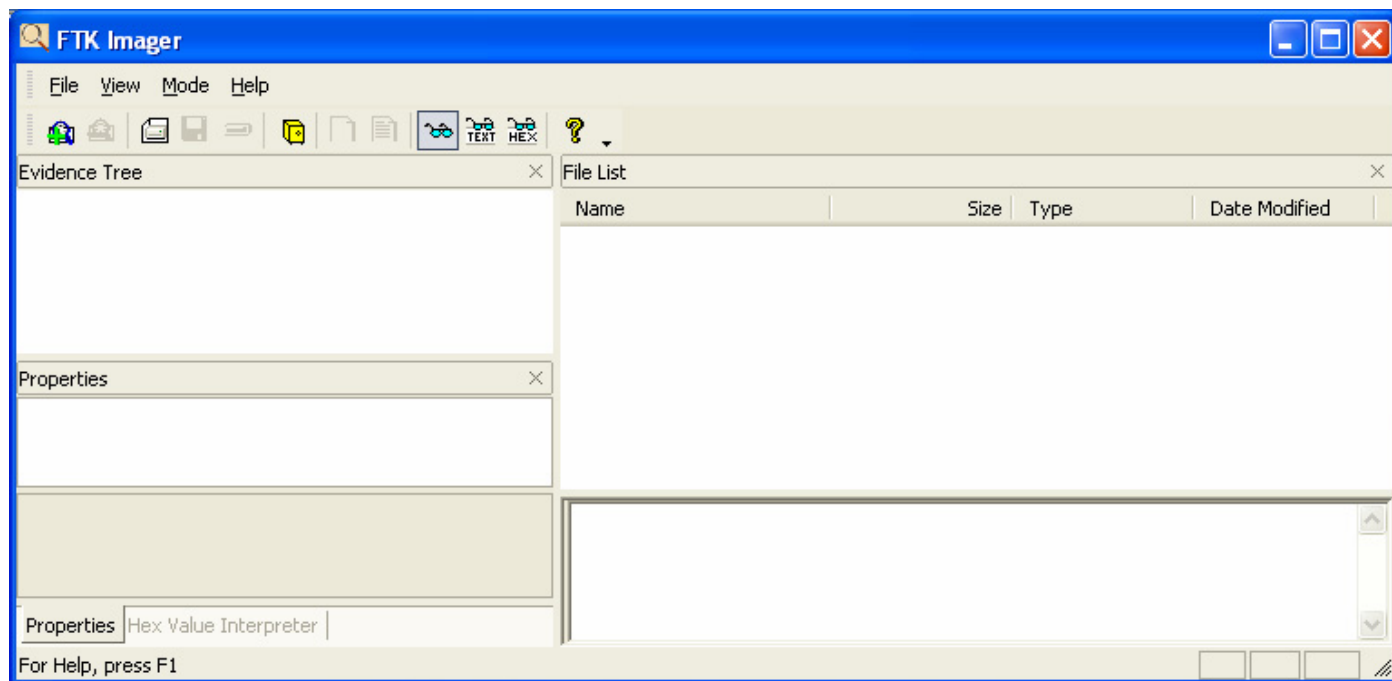
And that’s it. You now have an accurate copy of the suspect’s floppy. Print out the Audit.log file, put it in the evidence envelope along with the original floppy, update the chain-of-custody form, and return the evidence to the evidence locker.

Lab 1b - Create an Image of a suspect Floppy Disk (Windows, FTK Imager)

You have been given a suspect's floppy disk, and you want create an image of it. On your system, insert the Helix CD, and once the menu comes up, select the icon for "Acquire a "live" image of a Windows System using dd". Click on the triangle to go to the second page, the FTK Imager. Before inserting the floppy disk in the drive, be sure it is write protected.

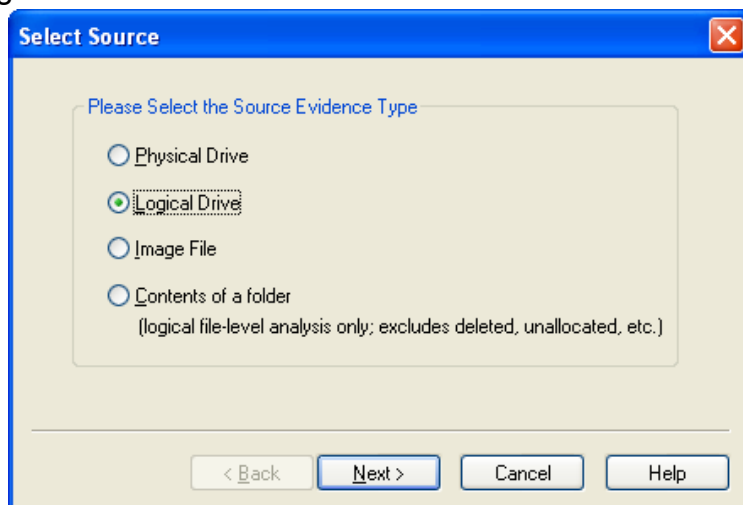


Press the "Start Imager" button, and the FTK Imager will load.

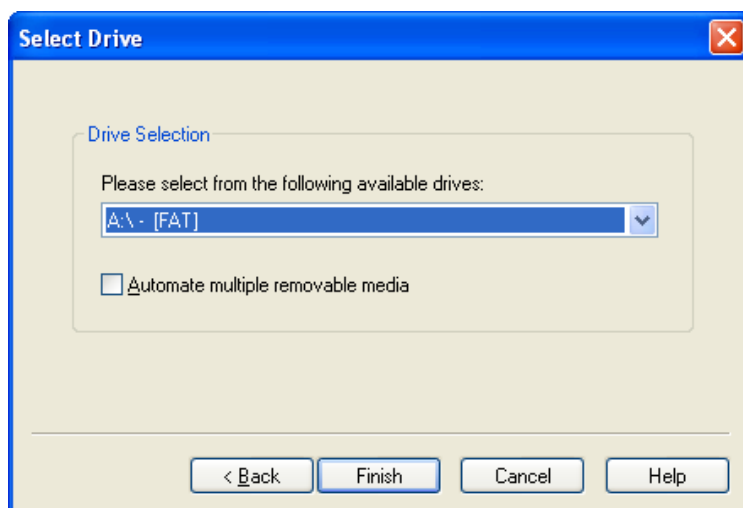


From the menu, select File / Create Disk Image.

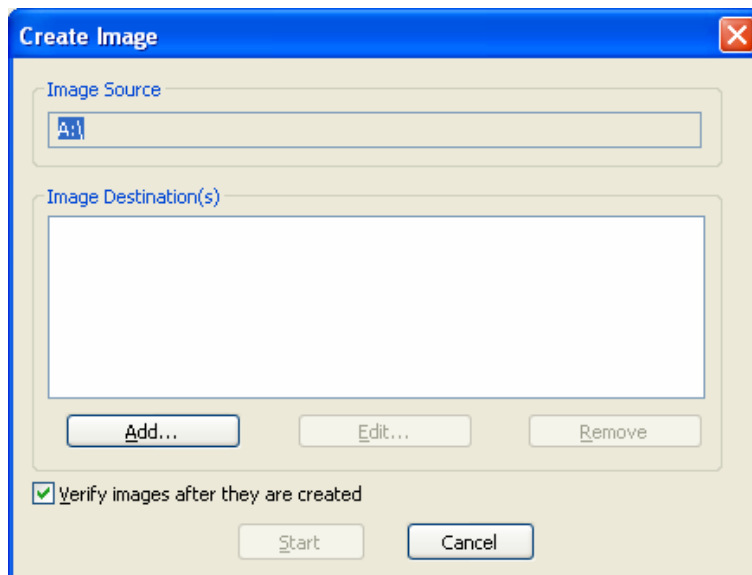
Select Logical Drive, and click Next.



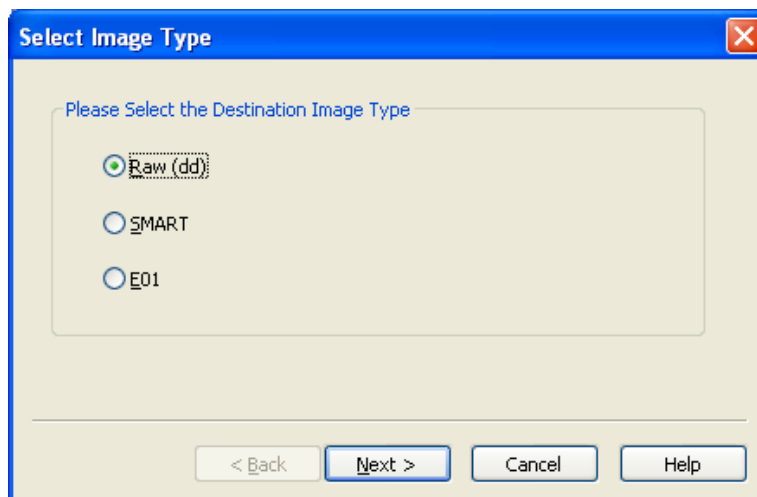
Select A:\ from the drop down menu, and click Finish.



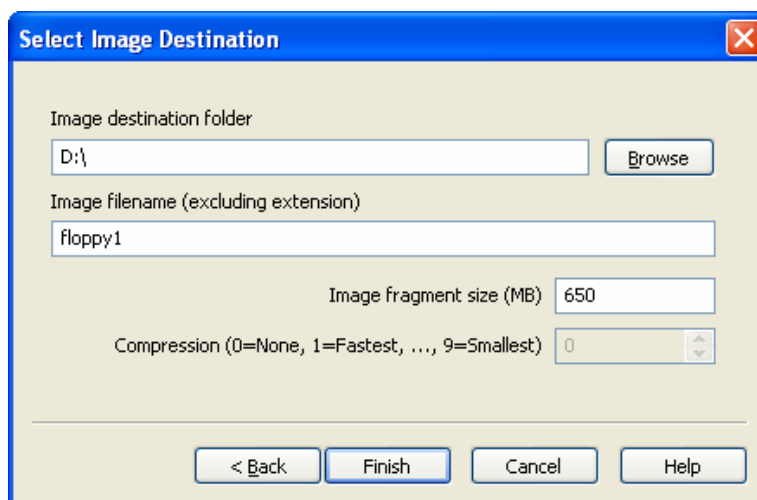
Now you are to select the destination drive. Click “Add...”



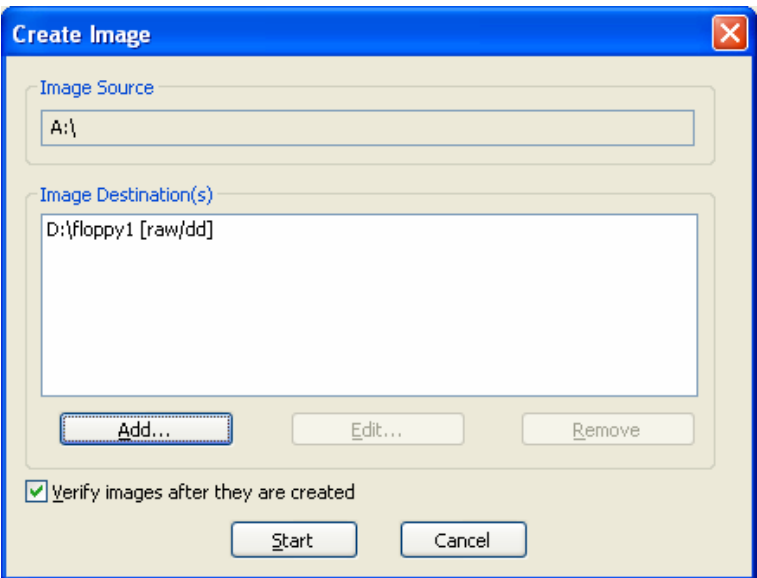
You can choose any of these Image types. Raw (dd) is the same format as created by the dd command, and is the most universal format. Smart is for the SMART forensic tool from ASR Data, and E01 is the format used by EnCase. Be sure that the “Raw (dd)” option is selection, and click Next.



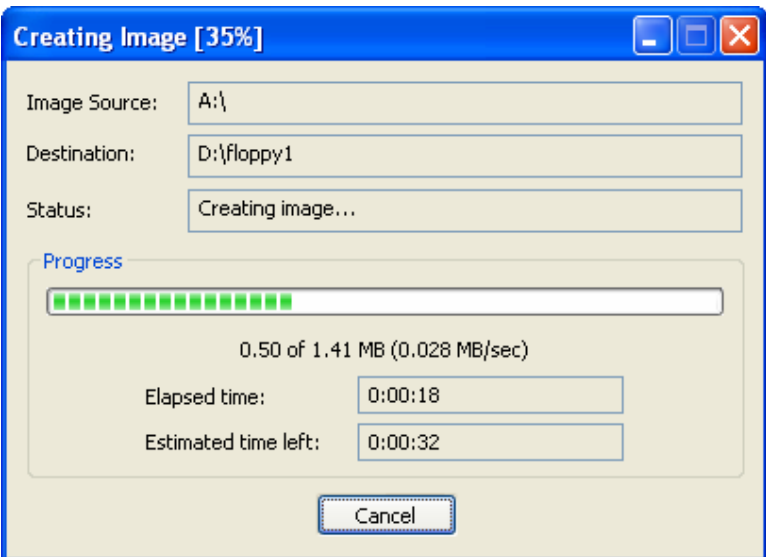
You can use the browse button to find the folder you wish to create the image in. Include an image filename, but not the extension – it will be added automatically. The Image fragment size is used to split large images in to chunks that can fit onto removable media, such as in this case, for a 650 MB CDROM. Click Finish.



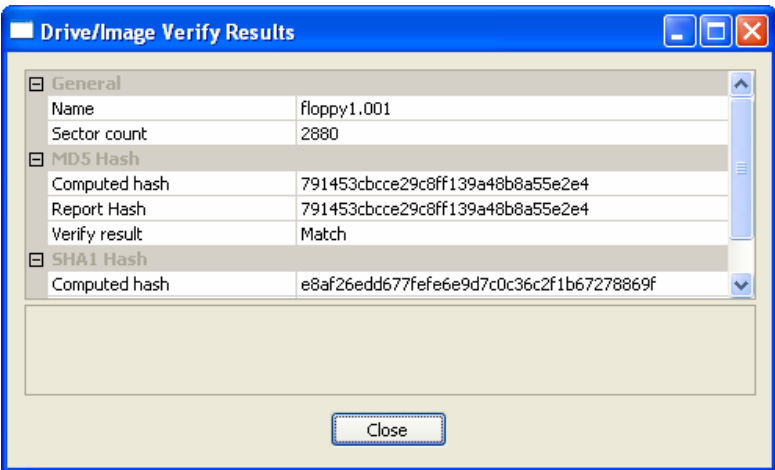
You are returned back to this screen. Click Start.



It typically takes about a minute to duplicate a floppy disk.



Once it is finished, it will display the Image Verify Results, and if all went well, you should see that both the MD5 and SHA1 hashes match. If they match, that means you have an accurate copy of the evidence. If they don't match, that typically means you have a bad disk, and the drive had a problem reading the floppy. Click Close. You can click Close again on the Creating Image screen.



If you now look in the destination folder, you should see two files:

floppy1.001	– this is the image of the floppy disk, and should be 1,440 KB in size.
floppy1.001.txt	– this is a copy of the Imager Verify Results screen.

```
Information for D:\floppy1:

Physical Evidentiary Item (Source) Information:
[Drive Geometry]
  Bytes per Sector: 512
  Sector Count: 2,880
  Source data size: 1 MB
  Sector count: 2880
[Computed Hashes]
  MD5 checksum: 791453cbcce29c8ff139a48b8a55e2e4
  SHA1 checksum: e8af26edd677fefe6e9d7c0c36c2f1b67278869f

Image Information:
  Segment list:
    D:\floppy1.001

Tue Jan 31 13:03:46 2006 - Image Verification Results:
  MD5 checksum: 791453cbcce29c8ff139a48b8a55e2e4 : verified
  SHA1 checksum: e8af26edd677fefe6e9d7c0c36c2f1b67278869f : verified
```

And that's it. You now have an accurate copy of the suspect's floppy. Print out the floppy1.001.txt file, put it in the evidence envelope along with the original floppy, update the chain-of-custody form, and return the evidence to the evidence locker.

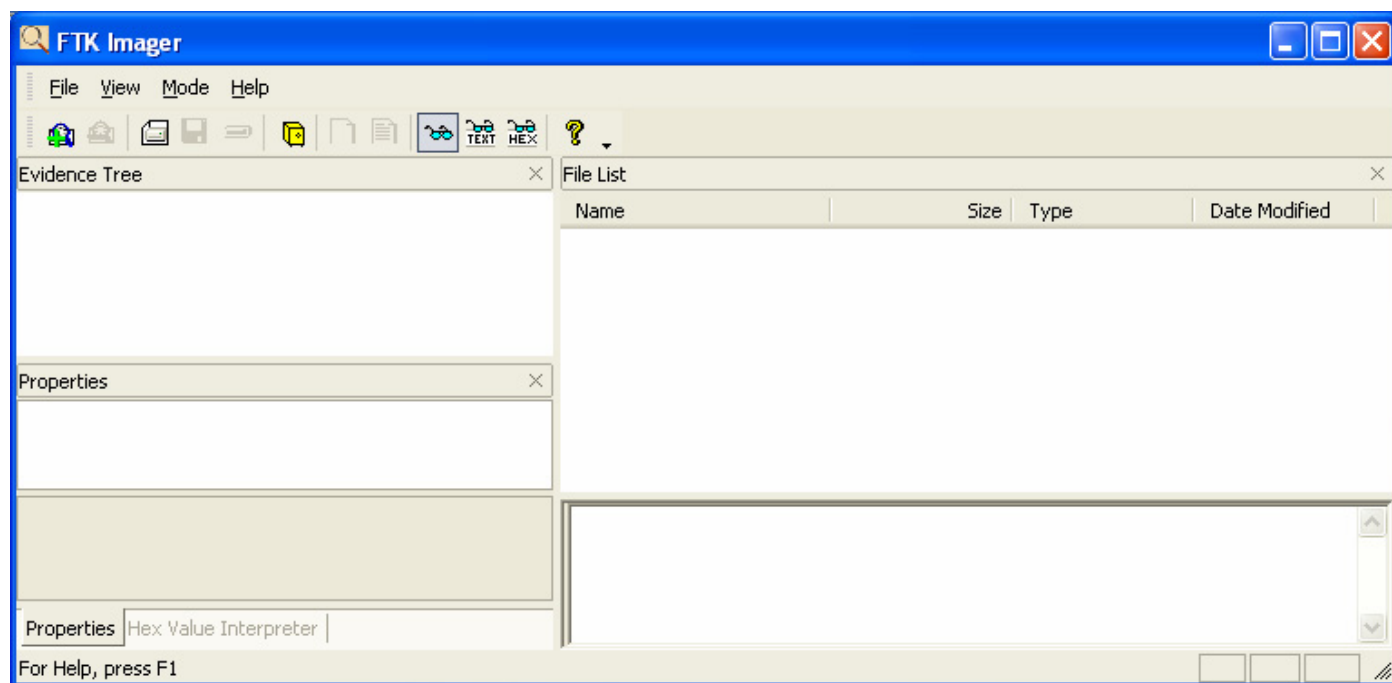
Lab 2 - Create a Floppy Disk from a suspect Image (Windows, FTK Imager)

After you have created an image of a suspect's floppy disk, it might be a good idea to make several duplicate physical copies of the disk, just in case.

For this lab, you will need to download the suspect image, which is one of the "Scan of the Month Challenges" from the HoneyNet Project (<http://www.honeynet.org>). The url for the image is <http://www.honeynet.org/scans/scan24/image.zip>.

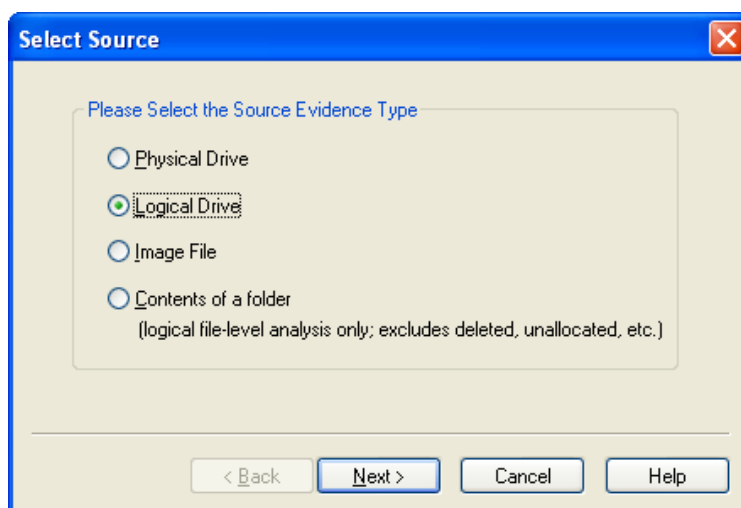
Download this file to a folder called "Lab2" on your forensic workstation.

On your system, insert the Helix CD, and once the menu comes up, use the "Quick Launch" menu option to start the FTK Imager. Before inserting the floppy disk in the drive, be sure it is write protected.

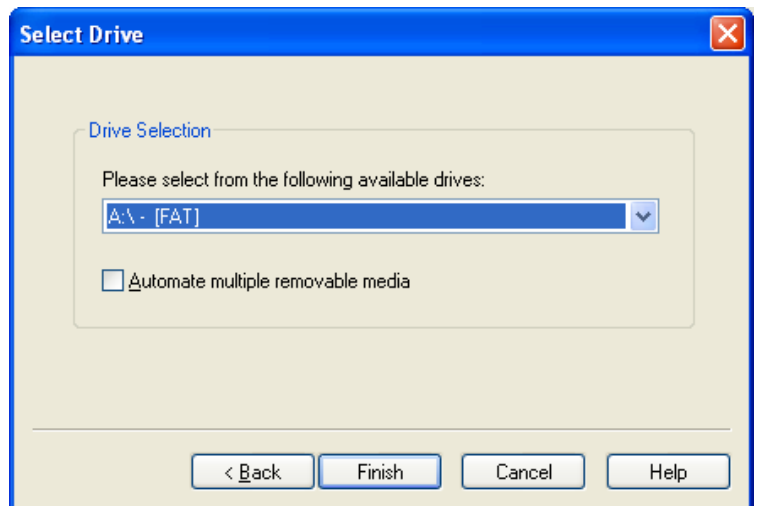


From the menu, select File / Create Disk Image.

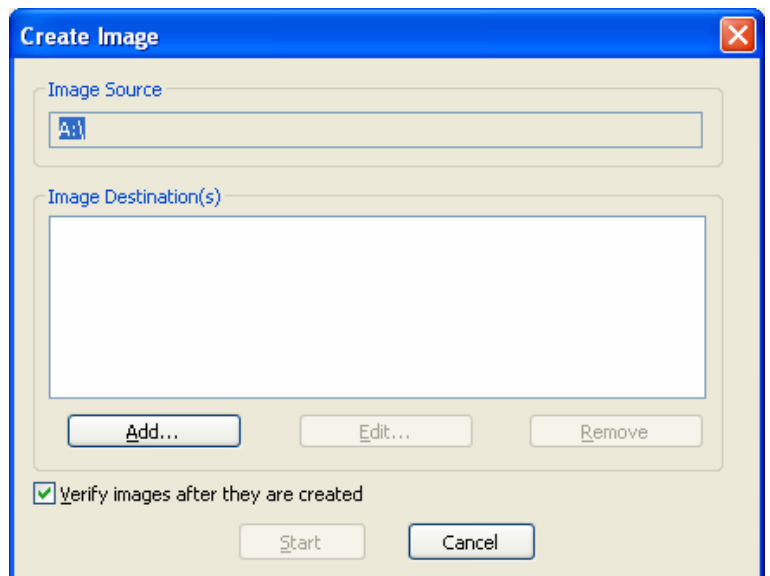
Select Logical Drive, and click Next.



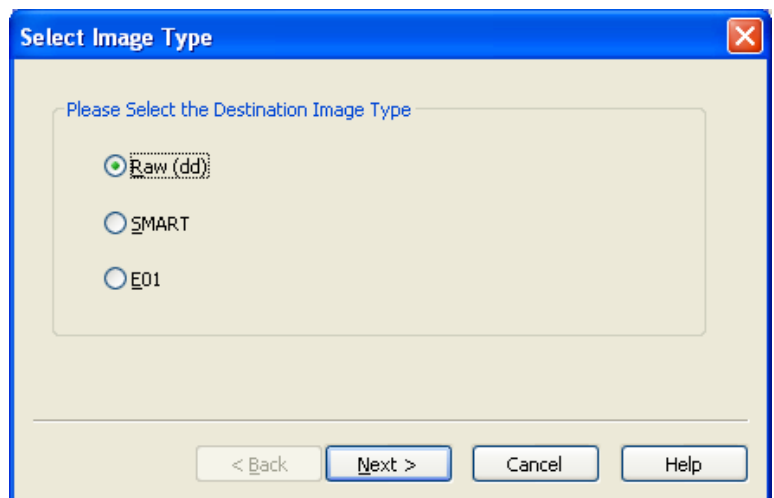
Select A:\ from the drop down menu, and click Finish.



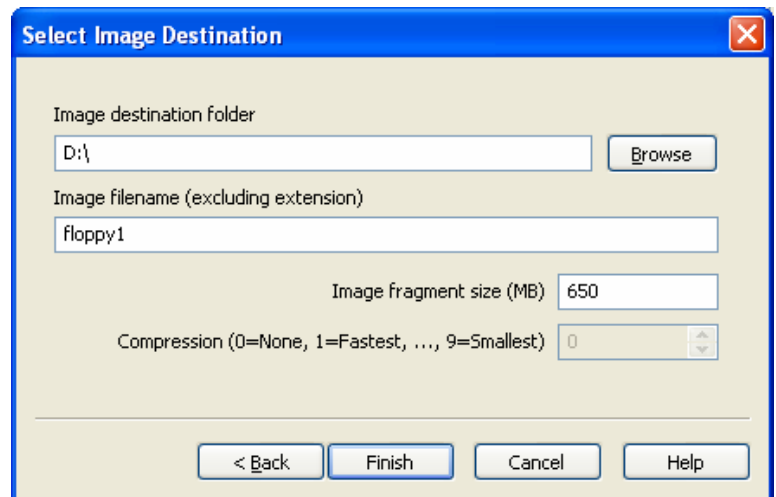
Now you are to select the destination drive. Click "Add..."



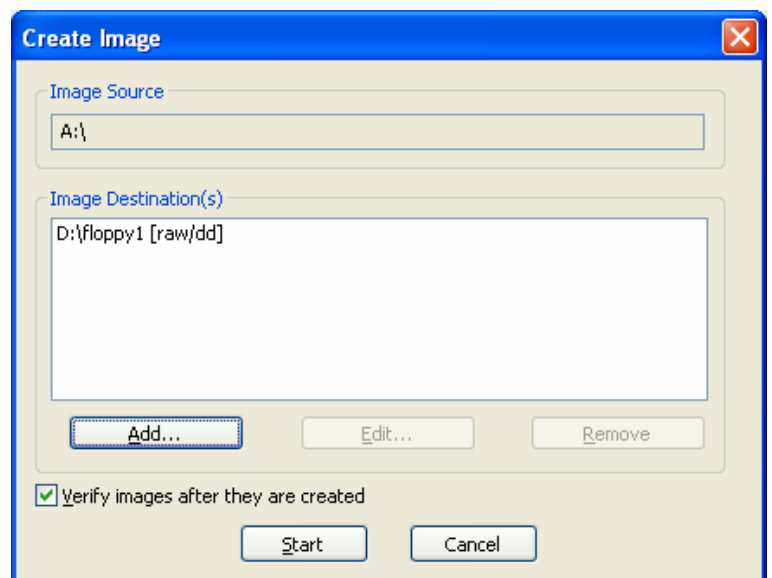
You can choose any of these Image types. Raw (dd) is the same format as created by the dd command, and is the most universal format. Smart is for the SMART forensic tool from ASR Data, and E01 is the format used by EnCase. Be sure that the "Raw (dd)" option is selection, and click Next.



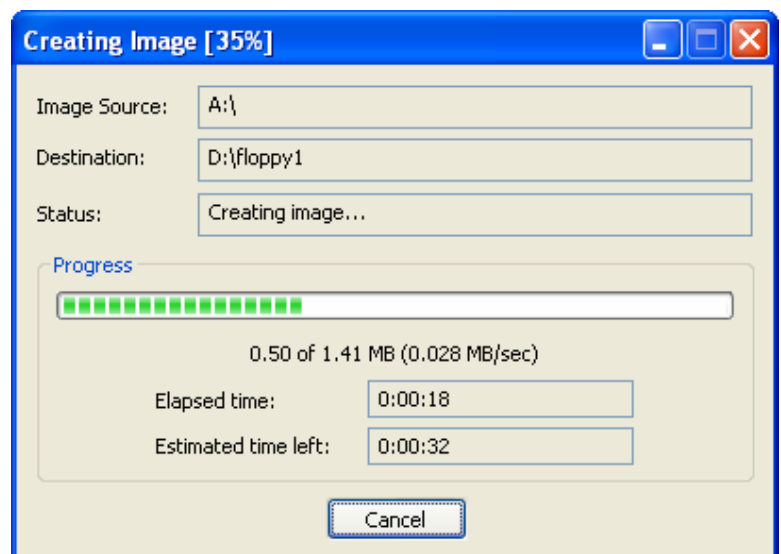
You can use the browse button to find the folder you wish to create the image in. Include an image filename, but not the extension – it will be added automatically. The Image fragment size is used to split large images in to chunks that can fit onto removable media, such as in this case, for a 650 MB CDROM. Click Finish.



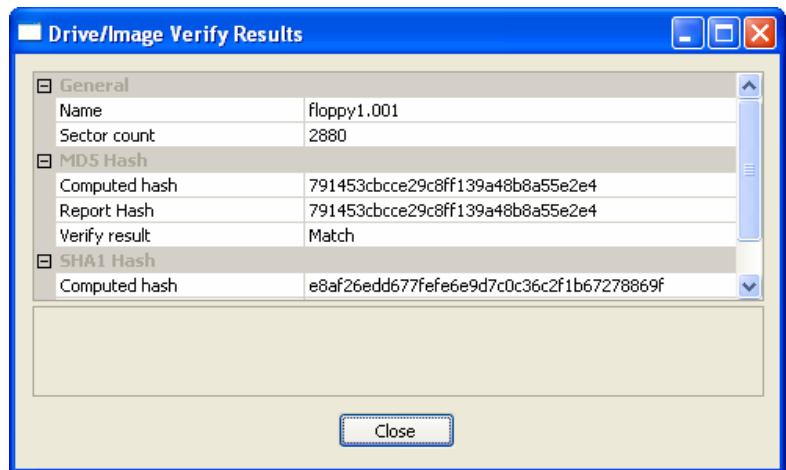
You are returned back to this screen. Click Start.



It typically takes about a minute to duplicate a floppy disk.



Once it is finished, it will display the Image Verify Results, and if all went well, you should see that both the MD5 and SHA1 hashes match. If they match, that means you have an accurate copy of the evidence. If they don't match, that typically means you have a bad disk, and the drive had a problem reading the floppy. Click Close. You can click Close again on the Creating Image screen.



If you now look in the destination folder, you should see two files:

- floppy1.001 – this is the image of the floppy disk, and should be 1,440 KB in size.
- floppy1.001.txt – this is a copy of the Imager Verify Results screen.

```
Information for D:\floppy1:

Physical Evidentiary Item (Source) Information:
[Drive Geometry]
  Bytes per Sector: 512
  Sector Count: 2,880
  Source data size: 1 MB
  Sector count: 2880
[Computed Hashes]
  MD5 checksum: 791453cbcce29c8ff139a48b8a55e2e4
  SHA1 checksum: e8af26edd677fefe6e9d7c0c36c2f1b67278869f

Image Information:
  Segment list:
    D:\floppy1.001

Tue Jan 31 13:03:46 2006 - Image Verification Results:
  MD5 checksum: 791453cbcce29c8ff139a48b8a55e2e4 : verified
  SHA1 checksum: e8af26edd677fefe6e9d7c0c36c2f1b67278869f : verified
```

And that's it. You now have an accurate copy of the suspect's floppy. Print out the floppy1.001.txt file, put it in the evidence envelope along with the original floppy, update the chain-of-custody form, and return the evidence to the evidence locker.

Appendix 1 – Samba Forensic Config File

```
# Samba Forensic Config File
# Global parameters
[global]
    log file = /var/log/samba/log.%m
    remote announce = 192.168.1.1/efense
    max log size = 500
    domain master = yes
    interfaces = 192.168.1.1/255.255.255.0
    dns proxy = No
    preserve case = Yes
    passwd program = /usr/bin/passwd %u
    preferred master = yes
    encrypt passwords = yes
    server string = efense Forensic Server
    workgroup = EFENSE
    hosts allow = 192.168.1.
    update encrypted = Yes
    passwd chat = *New*UNIX*password* %n\n *ReType*new*UNIX*password* %n\n
passwd:*all*authentication*tokens*updated*successfully*
    unix password sync = Yes
    netbios name = FORENSICS1
    socket options = TCP_NODELAY SO_SNDBUF=8192 SO_RCVBUF=8192
    local master = yes
    security = share
    os level = 65
    name resolve order = host lmhosts bcast

# Forensics Share information
[images]
    comment = Image Acquisition Share
    path = /mnt/images
    read only = no
    public = yes
    Browseable = yes
```

Appendix 2 – Linux Commands

adduser	<i>adduser dstoneil</i>	This command will automatically add a new user to the system The Bash script can be found in /usr/sbin if it needs to be changes
alias	<i>alias help=man</i> <i>alias long=ls -al</i>	The alias command allows you to substitute a new name for a command An alias can also contain command line options Unless the alias definition is included in your .login file it is only temporary
apropos	<i>apropos keyword</i>	Display command names based on keyword search
at	<i>at 1:23 lp /home/index.html</i> <i>at 1:50 echo "lp Job Done"</i> <i>at -l</i> <i>at -d 5555</i>	The at command runs a list of commands at a specified time (e.g. print @ 1:23) This uses the echo command to send a message at 1:50 saying a print job is done Lists all scheduled jobs; an alias for the atq command This will cancel job number 5555; an alias for the atrm command
batch	Example:	Temporarily blank
cat	<i>cat /etc/filename</i> <i>cat file.a > file.b</i> <i>cat -n file.a</i> <i>cat /proc/scsi/scsi</i>	Prints specified file to the screen Moves file.a to file.b This will show the contents of the file with line numbers (-b number only no blank lines) This will list all the SCSI processes running on your system
cd	<i>cd /home/dstoneil</i> <i>cd ~username</i>	Changes directories to the specified one This will move you to the users specified home directory
chattr	<i>chattr +i /etc/passwd</i>	makes the named file immutable. Attributes are not shown by ls; use lsattr
chfn	<i>chfn dstoneil</i>	This will allow you to change finger information on that user As an example it will allow you to change dstoneil to Darcy S. O'Neil
chmod	<i>chmod 666 filename</i> <i>chmod 777 filename</i> <i>chmod a=rwx file</i>	This command will give a file Read - Write permission for everyone This command gives Read - Write - Execute permission to everyone This gives Read - Write - Execute permission to all users
chown	<i>chown dso /home/html</i> <i>chown dso /home/file.a</i>	This command will change the owner of the specified directory to dso This command will change the owner of the specified file to dso
clear	<i>clear</i>	This will clear your screen
cmp	<i>cmp -s file.a file.b</i>	Compares 2 files of any type. The -s option will return nothing in the files are the same
cp	<i>cp file.a file.b</i>	This will create a duplicate of file.a under a new file name, file.b
cpio	<i>ls /home cpio -o > /root</i> <i>cpio -it < /root > bk.indx</i>	This will copy the files of /home to the directory /root This will extract all of the files to /root and creates an index file called bk.indx
cpkgtool		Graphical front end to installpkg, removepkg, makepkg that uses ncurses.
cron	<i>crontab -e</i>	Edit your personnel crontab file The main crontab files can be found in the /etc directory
date	<i>date</i> <i>date --date="2001-3-15"</i> <i>date --date="2001-3-15 11:59 AM"</i>	Will output the present date to the screen This will set date to 2001-Mar-15 This will set the date as well as time
df	<i>df -hT</i>	Displays the total size, used and available space on all mounted file systems
dmesg	<i>dmesg</i>	Prints out the bootup messages so you can locate errors
du	<i>du -k /home/html</i> <i>du -k /home/html/file.a</i>	Provides a summary of the disk space usage, in kb, within the specified path Provides a summary of disk space used by a particular file
e2fsck	<i>e2fsck /dev/fd0</i> <i>e2fsck /dev/hda1</i>	To "scandisk" a floppy (run while the floppy is unmounted) Also can be used to scan for disk errors on hard drive partitions
fc	<i>fc -l</i>	Lists your recent commands (Beware that fc is dangerous w/o -l because it will run commands)
fdformat	<i>fdformat /dev/fd0</i> <i>fdformat /dev/fd0H1440</i>	low level format of a floppy device in drive fd0 This will format a "Double Sided High Density" disk
fdisk	<i>fdisk -l /dev/hda</i> <i>fdisk /dev/had</i>	List all partitions on drive had, with out mounting n: P, primary: 1: t, c (Fat 32 LBA)

file	<i>file file.a</i> <i>file -z file.a.tar</i> <i>file -L file.a</i> <i>file -k file.a</i>	This command will try to determine what type of file file.a is. (exec, text, etc.) Looks inside a compressed file to determine it's type. Follows symbolic links to be followed to determine file type <i>Do not stop at the first matched test</i>
find	<i>find /path -name passwd</i>	Locates the specified string (passwd), starting in the specified directory (/path) All filenames or directories containing the string will be printed to the screen
finger	<i>finger</i>	This will list all users currently logged into the UNIX system
free	<i>free -t -o</i>	Provides a snapshot of the system memory usage
fsck	<i>fsck /hda</i>	file system check and repair
git		This is a file system viewer (Use F10 to exit)
grep	<i>cat /etc/passwd grep dso</i> <i>grep -i "Sample" /home/dsoneil</i>	This searches for and limits the command output to the pattern specified In this case all instances of dso from the /etc/passwd file are printed The -i option makes the search indifferent to case (e.g. sample or SAMPLE)
groupadd	<i>groupadd sudos</i>	Create a new group called sudos on the system
groups	<i>groups</i>	Shows which groups you are in
gzip	<i>gzip file.a</i> <i>gzip -d file.a.gz</i> <i>tar -zxvf file.a.tar.gz</i>	This will zip file.a and give it the extension file.a.gz This will unzip the file file.a.gz The z flag allow you to decompress the tar file on the fly
history	<i>history grep sneak</i> <i>history -d 1061</i>	To retrieve your recent commands with "sneak" somewhere in them. 6/00 To delete history entry 1061, which may be a password in cleartext.
hostname		Get or set hostname. Typically, the host name is stored in the file /etc/HOSTNAME.
ifconfig	<i>ifconfig eth0</i> <i>ifconfig eth0 up</i> <i>ifconfig eth1 192.168.0.2 up</i>	This will display the status of the currently defined interface (.e.g Ethernet Card 0) This flag causes the interface to be activated (To deactivate an interface use <i>down</i>) Makes eth1 active with IP address 192.168.0.2
insmod		used (by root) to install modular device drivers
installpkg	<i>installpkg -r packagename.tgz</i>	This will install a Slackware package with the name you specify (-r option)
ipchains	<i>ipchains [-A -s -d -j] [Input / Output]</i> <i>ipchains -A input -s 24.1.50.25 -j DENY</i> <i>ipchains -A output -d 24.1.50.2 -j DENY</i>	This command is used to ACCEPT or DENY access to your system This will block the IP address 24.1.50.25 from accessing your system This command will DENY your system from accessing this IP address [-A append] [-s source] [-d destination] [-j join]
jobs	<i>jobs</i>	This will list all jobs presently running on your system
kernelcfg		GUI to add/remove kernel modules (as root in X terminal).
kill	<i>kill 2587</i> <i>kill -9 2587</i>	Kills the process specified by the Process ID Number (2587) The -9 flag forces the process to die
last	<i>last -300</i> <i>last -5 username</i>	Prints to the screen the username, location, log-in and log-off times of the last -x logins to the system. The username will select the last x time that person has used the system. The last command is not traceable.
lastlog	<i>lastlog</i>	Displays a list of the login attempts / times of all users on the system (security check)
ldd	<i>ldd ./test.exe</i>	Display shared library Dependencies
less	<i>less /html/index.html</i>	Less displays information a screen at a time, you can also page back and forth
lilo	<i>lilo -v</i>	To write or correct boot config to disk. Use this command after modifying /etc/lilo.conf Do this before rebooting (to avoid "LIL-" on startup) if it's been a while <i>lilo -b /dev/fd0</i> This command will make a boot disk
ln	<i>ln -s /usr/dso /home/html</i>	Creates a "symbolic" link from the first directory or file to the second. A user changing into /home/html will actually be directed to the /usr/dso directory.

locate	<i>locate wordperfect</i>	The locate command will locate the file specified and output a directory path (see “ <i>updatedb</i> ”)
lpr	<i>lpr /home/html/index.html</i>	This command will print the file index.html to the printer
lprm	<i>lprm 12</i>	This command will cancel print job 12 in the printer queue
lpq	<i>lpq</i>	This will show the contents of the print queue
ls	<i>ls -al</i> <i>ls -F</i> <i>ls /proc/sys/net/ipv4</i> <i>ls -alRu</i>	Lists all information on all files (-a) in the current directory in single line format (-l). Includes permissions, owners, modification time, file size and name Marks (directories with a trailing /) - (executables with an *) (symbolic links w/ @) This will list all IP4 (masquerading) entries in the system processes directories List files by last access time
lsmod		used (by root) to show kernel modules currently loaded
lsof	<i>lsof grep :<port number></i> <i>lsof -i</i> <i>lsof -t</i>	This will show you what program has that port open. This will show Internet connections like netstat. This will show processes
lspci	<i>lspci</i>	Lists your PCI devices
make	<i>make mrproper</i> <i>make menuconfig</i> <i>make dep</i> <i>make clean</i> <i>make bzImage</i> <i>make lnx</i> <i>make install</i>	Cleans up junk accidentally left behind by the development team This will ask you a series of questions about your system and drive requirements This will use dependencies The clean command will clean up any unnecessary files left lying around This will begin the process of compiling your new kernel This specified that the source will be compiled under a Linux system After the make command this will install the compiled binaries to their directories
<i>make modules</i>		This will compile all the necessary modules
	<i>make modules_install</i>	This will install modules into the /lib/modules directory
man	<i>man vi</i>	Prints the manual page on the specific topic (vi) to the screen. To scroll down the page use the Space Bar, to scroll up use the letter b, to exit press q.
md5sum	<i>md5sum filename.tgz</i> <i>md5sum -b</i> <i>md5sum -t</i>	To ensure a copy between machines went perfectly Reads files in binary mode Reads files in text mode
mkdir	<i>mkdir pascal</i>	This will create new directory (pascal) in the present directory
mkfs	<i>mkfs -t msdos -c -v /dos-drive</i> <i>mkfs -t xfs -c -v /home</i> <i>mkfs.vfat -v -F32 -n DATA /dev/hda1</i>	Formats a partition and builds a new filesystem on it -t specifies filesystem type, -v produces verbose output, -c checks bad blocks -n Labels volume
more	<i>more /home/html/index.htm</i>	Paginates the specified file so it can be read line by line (using Enter key) or screen by screen using the Space Bar. Use b key to move back and q to quit.
mount	<i>mount -t msdos /dev/hda5 /dos</i> <i>mount -t iso9660/dev/sr0 /cd</i> <i>mount -t msdos /dev/fd0 /mnt</i> <i>mount -a /etc/fstab</i>	Mounts the msdos partition on the Hard Drive (hda5) to the directory /dos Mounts the CD-ROM under the directory /cd Mounts the floppy drive with an msdos file system to /mnt Attempts to mount all file systems located in the /etc/fstab file
mv	<i>mv ./home/file ./dso/file</i>	Moves the specified file to another directory
netstat	<i>netstat -tap grep LISTEN</i> <i>netstat -t -u -a</i>	This will give us a list of all currently running TCP servers that are LISTENing on a port This will list all UDP and TCP ports that are open
nice	<i>nice -5 sort one.a > two.b</i>	This command adjusts the priority of a process before it starts The higher the number the lower the priority. All process start at 10
nmap	<i>nmap localhost</i> <i>nmap -sT -sU -p 1-65535 localhost</i>	This will port scan the ‘localhost’ server to determine open ports This will thoroughly check all ports on the system (both UDP and TCP)
nohup		This command allows a process to continue after you log out
passwd	<i>passwd</i>	Launches the password program so the user can change their password

ps	<i>ps</i> <i>ps -ef grep dsonail</i>	Lists all current running processes, their corresponding pids, and their status This will find all of the processes for user dsonail
pstree	<i>pstree -p</i>	Provides a list of running processes in a tree structure
pwd	<i>pwd</i>	Prints the current working directory
quota	<i>quota</i>	Lists the user's quotas for both ada (/home/ada/a#/username) and amelia (/var/spool/mail/username), indicating the number of blocks used and the users quota.
renice	<i>renice -5 6641</i>	Adjusts the priority of the running process 6641 (The 5 lowers the priority to use less resources)
removepkg	<i>removepkg -copy packagename</i>	This will remove the named package but make a copy in the /tmp directory
rm	<i>rm file.a</i> <i>rm -i file.a</i> <i>rm -r /home/dso</i>	Removes the specified file in your current directory Removes specified file but prompts for confirmation before deleting Removes the specified directory and all files in that directory
rmdir	<i>rmdir pascal</i>	Removes the empty directory specified, if not empty you will receive an error
<i>rmdir -r pascal</i>		Removes the directory and all files in that directory (if supported)
route	<i>route -n</i> <i>route add -net 192.168.0.0 eth0</i> <i>route add default gw 192.168.0.5 eth0</i>	Displays the Linux Kernel IP routing table This will tell other systems what network to route your system on This will tell the your system where the Internet gateway is located This information can be added to you /etc/rc.d/rc.local system files
rpm	<i>rpm -i file.2.0-i386.rpm</i> <i>rpm -U file.2.0-i386.rpm</i> <i>rpm -i --force file.rpm</i> <i>rpm -e file.2.0-i386.rpm</i> <i>rpm -i --nodeps file.rpm</i> <i>rpm -qa</i> <i>rpm -qa grep gtk</i> <i>rpm -qi file.2.0-i386.rpm</i> <i>rpm --rebuild file.2.0.rpm</i>	This will unpack an RPM file. This is the most basic method of installation This will install an upgrade to a previous RPM package. The --force option will force the package to re-install This will remove and RPM package. (You do not need to use the complete name) This command uses the "no dependencies" flag. This will give a screen print out of all packages installed (q is query) This will print out all of the rpm packages will gtk in the file name This will provide information on the package you are about to install This will rebuild a package if it has been corrupted by another installation process
rpm2targz	<i>rpm2targz filename.rpm</i>	This will convert an RPM file to a Slackware .tgz package
service	<i>service smb status</i> <i>service --status-all</i>	This will check the status of the smb service Display status of all services
sha1sum	<i>sha1sum filename.tgz</i> <i>sha1sum -b</i> <i>sha1sum -t</i>	To ensure a copy between machines went perfectly Reads files in binary mode Reads files in text mode
shutdown	<i>shutdown -t 10.00</i> <i>shutdown -r -t 20.00</i> <i>shutdown -t +10 good day</i> <i>shutdown -f</i>	This will notify all logged in users that the system will shut down at 10:00 AM This will reboot the system at 8:00 PM This will shutdown the system in 10 minutes with the message "good day" sent The -f flag will cause Linux to do a fast reboot
sort	<i>sort myfile</i>	To sort files. (Options -r Reverse normal order, -n Sort in numeric order)
strings	<i>strings myfile</i>	Displays printable characters
su	<i>su username</i>	This will allow you to access the Superuser privileges. Type exit to revert back to normal
tar	<i>tar -cf /usr/dso.tar /home</i> <i>tar cvf /backup.tar /dso</i> <i>tar -xvf file.a.tar</i> <i>tar -tvf file.a.tar more</i> <i>tar -zxvf file.a.tgz</i>	This command copies the directory /home to the file /user/dso.tar This will create a tar archive of everything in the directory /dso This command will extract the tar archive This will allow you to check whether the tar archive starts with a directory This command will unzip and extract the file in one step as opposed to using gzip
top	<i>M for memory usage information</i> <i>P for CPU information</i>	This program shows a lot of stuff that goes on with your system. In the program, you can type: q to quit

touch	<i>touch file.a</i>	Creates an empty file in the current directory with the name file
updatedb	<i>updatedb</i>	This will update the “locate” database
upgradepkg	<i>upgradepkg packagename.tgz</i>	This will upgrade a Slackware package and remove any old or no used files
umask	<i>umask -S u=rw,g=,o=</i> <i>umask 022</i>	Specify the permission for files when files are created for owner(u), group(g), and others(o) you can use 022 for read only file permission for others and 077 for read and write permission
uname	<i>uname -a</i>	This will print to the screen the Linux Kernel in use on your system
uptime	<i>uptime -a 192.168.0.100</i>	Shows system uptime and includes a list of users who have been idle for more than one hour
userdel	<i>userdel -r dsoneil</i>	This will delete the user dsoneil from the system, the -r option will delete the users /home
w	<i>w</i>	Lists all users currently logged into the UNIX system. Provides information such as username, login time, idle time, and current action
whatis	<i>whatis cat</i>	Provides a one-line summary of the command
which	<i>which -a filename</i>	This will search through all directories in your current path and find all files named filename
who	<i>who</i>	Lists currently logged on users username, port, and when they logged in
whoami	<i>whoami</i>	Tells the user who they are acting as; usually their own username.

FORENSIC TOOLS : ANALYSIS

CONTENT LAYER TOOLS:

dcat	<i>dcat -f linux-ext2 -h /*.img 357 less</i>	Display the contents of a disk block (–h hex view)
dcalc	<i>dcalc -u 345 /*.img 644</i>	Maps between dd images and dls results
dls	<i>dls -f linux-ext2 /*.img > *.img.dls</i>	(unrm in TCT) Lists contents of deleted disk blocks
dstat XXX	<i>dstat -f linux-ext2 /*.img 357</i>	List statistics associated with specific disk block from an image
debugfs	<i>debugfs [options] *.img</i> <i>debugfs: lsdel</i> <i>debugfs: write</i> <i>debugfs: cat <inode#></i> <i>debugfs: dump<inode#> /mnt/file.txt</i> <i>debugfs: dump -R “stat <inode#>” *.img</i>	View and edit ext2 files FROM A DEBUGFS COMMAND PROMPT: List deleted Inodes Write a file out from the img. Read the contents of an Inode Dumps the contents of an Inode. Provides greater data about the Inode.

FILE SYSTEM LAYER TOOLS: File Types: linux-ext2, linux-ext3, solaris, openbsd, freebsd, ntfs, fat, fat12, fat16, fat32

ffind	<i>ffind -f linux-ext2 /*.img 2060</i> <i>-a</i>	Determine which file has allocated to an inode in an image Finds all occurrences of that file
fls	<i>fls -f linux-ext2 hda2.dd 33</i> <i>-a</i> <i>-d</i> <i>-u</i> <i>-D</i> <i>-F</i> <i>-r</i> <i>-p</i> <i>-m</i> <i>-l</i> <i>-z</i> <i>-s</i>	Displays file & Directory entries in Directory Inode Display ‘.’ and ‘..’ directories Display deleted entries only Display undeleted entries only Display Directory entries only Display File entries only Recurse on Directories Display full path when recursing Display in timeline import format Display long version (all times and info) Specify the timezone (for ‘-l’ listing) Clock skew in seconds (‘-l’ and –m only)
fsstat XXX	<i>fsstat -f linux-ext2 /*.img less</i>	Displays details about the file system
fsgrab	<i>fsgrab -c 1 -s 57 *.img > *.txt</i> <i>-c #</i>	Can grab blocks directly off of the disk Number of blocks to grab

	<code>-s #</code>	Number of blocks to skip
lazarus	<code>lazarus -h /*.img.dls</code> <code>-h</code>	Takes unallocated space from raw data and puts it in order -h directs the output to html
mac_daddy.pl		Standalone MACTIME analysis, writes STD-OUT to netcat.
mactime	<code>mactime -b /*.mac > /*.all</code>	Used to correlate data files; from fls/ils, grave-robber, mac-robber
mactime	<code>mactime -b 01/01/2001 /*.mac > /*.all</code>	Run mactime STARTING at 01/01/2001
	<code>-p</code>	Password file location (to replace UID)
	<code>-g</code>	Group file location (to replace GID)
	<code>-y</code>	Dates use the year first
	<code>-z</code>	Specify timezone

META DATA LAYER TOOLS:

icat	<code>icat -hf linux-ext2 /*.img 2060</code> <code>-h / -H</code>	Displays contents of a disk block allocated to an Inode -h DO NOT Display file holes, -H DO Display holes
ifind	<code>ifind -f linux-ext2 /*.img 2060</code>	Determine which Inode has allocated a block in an image
ils	<code>ils -of linux-ext2 /dev/hda1</code> <code>-o</code> <code>-e</code> <code>-r</code> <code>-f</code> <code>-m</code>	Display inode information, even those deleted Open but unlinked files on a live system, the process is running but the file is deleted List Every Inode Removed files, by default Tells ils which file system structure to read Extracts data from deleted inodes
istat	<code>istat -f linux-ext2 /*.img 2060</code>	Displays information about a specific inode

XXX

sorter	<code>sorter -f linux-ext2 -d /*.img /sorter</code> <code>-e</code> <code>-i</code> <code>-l</code> <code>-s</code> <code>-c</code>	Sorter runs both 'fls -r' and 'icat' to identify content Extension mismatch only Category Indexing Only List details to STDOUT only (for IR) Save the data to category directories Config file
---------------	--	---

FORENSIC TOOLS: LIVE

grave-robber	<code>grave-robber [options] > STD-OUT</code> <code>-E</code> <code>-i</code> <code>-P</code> <code>-s</code> <code>-t</code> <code>-l</code>	Data Capture Tool Grabs Everything Collects Inode data from unallocated space Run Process commands Gather network and host info Grabs all trust information Run lstat() on all files to get Inode information
mac-daddy		Run on a live system to create a timeline.
mac-robber		Run on a live system to create a timeline.
pcat	<code>pcat process_ID</code>	Copy process memory from live system

DATA LOCATION

<code>/etc/issue</code>	Operating System and Version.
<code>/tmp/install.log</code>	Operating System installation date or file list.
<code>/etc/timezone</code>	System timezone.
<code>/var/log/boot.log</code>	Boot Dates.
<code>/etc/fstab</code>	Partition Information.

Other Useful Commands

Ctrl-Alt-F1: installation dialog
 Ctrl-Alt-F2: shell prompt
 Ctrl-Alt-F3: install log
 Ctrl-Alt-F4: system related messages
 Ctrl-Alt-F5: other messages
 Ctrl-Alt-F7: X graphical display

References

- Access Data. (2005). FTK Imager Help File.
- Carvey, H. (2005a). *First Responder Utility*. Retrieved 31 Jan, 2006, from <http://www.windows-ir.com/tools.html>
- Carvey, H. (2005b). *Windows forensics and incident recovery*. Boston: Addison-Wesley.
- Chuvakin, A. (2002). *Linux Data Hiding and Recovery*. Retrieved 4 Feb, 2006, from <http://www.linuxsecurity.com/content/view/117638/>
- Computer Hope. (2006a). *3.5 Floppy*. Retrieved 31 Jan, 2006, from <http://www.computerhope.com/jargon/h/headslot.htm>
- Computer Hope. (2006b). *5.25 Floppy*. Retrieved 31 Jan, 2006, from <http://www.computerhope.com/jargon/h/headslot.htm>
- Digital Intelligence. (2006a). *Forensic Card Readers*. Retrieved 31 Jan, 2006, from http://www.digitalintelligence.com/products/forensic_card_readers/
- Digital Intelligence. (2006b). *Ultrablock*. Retrieved 31 Jan, 2006, from <http://www.digitalintelligence.com/products/ultrablock/>
- Digital Intelligence. (2006c). *USB Write Blocker*. Retrieved 31 Jan, 2006, from http://www.digitalintelligence.com/products/usb_write_blocker/
- Erickson, L. (2004). *NCFS Software Write-block XP - 5 Step Validation*. Retrieved 25 Jan, 2005, from www.ncfs.org/fleet/block/index.htm
- Frisk Software International. (2006). *F-Prot Antivirus for Linux Workstations - for home users*. Retrieved 4 Feb, 2006, from http://www.f-prot.com/products/home_use/linux/
- Gregg, B. (2004). *Chaosreader*. Retrieved 4 Feb, 2006, from <http://users.tpg.com.au/adsl4yb/chaosreader.html>
- Harbour, N. (2006). *dcfldd*. Retrieved 4 Feb, 2006, from <http://dcfldd.sourceforge.net/>
- Hurlbut, D. (2005). *Write Protect USB Devices in Windows XP*. Retrieved 31 Jan, 2006, from http://www.accessdata.com/media/en_us/print/papers/wp.USB_Write_Protect.en_us.pdf
- Kornblum, J. (2006). *Foremost*. Retrieved 3 Feb, 2006, from <http://foremost.sourceforge.net/>
- McDougal, M. (2005). *Windows Forensic Toolchest*. Retrieved 31 Jan, 2006, from <http://www.foolmoon.net/security/wft/index.html>
- McLeod, J. (2005). *Incident Response Collection Report (IRCR2) readme file*. Retrieved 31 Jan, 2006, from <http://ircr.tripod.com/>
- MemoryStick.com. (2006). *MemoryStick Write Protect*. Retrieved 31 Jan, 2006, from <http://www.memorystick.com/en/ms/features.html>
- Owen, H. (2004). *ECE 4112 Internetwork Security Lab: Data Protection*. Retrieved 4 Feb, 2006, from <http://users.ece.gatech.edu/~owen/Academic/ECE4112/Fall2004/Projects/Data%20Protection.doc>
- SecReport. (2005). Retrieved 31 Jan, 2006, from <http://members.verizon.net/~vze3vkmg/index.htm>
- The Living Room. (2006). *MMC/SD Card*. Retrieved 31 Jan, 2006, from <http://www.livingroom.org.au/photolog/pretec-4gb-sd-memory-card-1.jpg>
- Vincent, L. (2006). *Hardware Lister (lshw)*. Retrieved 5 Feb, 2006, from <http://ezix.sourceforge.net/software/lshw.html>
- Wikipedia. (2006a). *Knoppix*. Retrieved 7 Feb, 2006, from <http://en.wikipedia.org/wiki/Knoppix>
- Wikipedia. (2006b). *Linux*. Retrieved 7 Feb, 2006, from <http://en.wikipedia.org/wiki/Linux>
- Wikipedia. (2006c). *Xfce*. Retrieved 7 Feb, 2006, from <http://en.wikipedia.org/wiki/Xfce>
- Zalewski, M. (2002). *Fenris*. Retrieved 4 Feb, 2006, from http://www.bindview.com/Services/RAZOR/Utilities/Unix_Linux/fenris_index.cfm