eForensics Magazine

Computer

VOL.2NO.7

Computer Forensics JumpStart vol.1

A STEP BY STEP GUIDE TO BEGINNING COMPUTER FORENSICS

WINDOWS REGISTRY FORENSICS 101

UNDERSTANDING FILE METADATA

FTK IMAGER BASICS

BASIC APPROACH TO INVESTIGATE A DIGITAL CRIME

INTRODUCTION TO NETWORK FORENSICS USING WIRESHARK

Issue 07/2013 (11) July

What do all these have in common?

They all use Nipper Studio

to audit their firewalls, switches & routers

Nipper Studio is an award winning configuration auditing tool which analyses vulnerabilities and security weaknesses. You can use our point and click interface or automate using scripts. Reports show:

1) Severity of the Threat & Ease of Resolution

2) Configuration Change Tracking & Analysis

3) Potential Solutions including Command Line Fixes to resolve the Issue

Nipper Studio doesn't produce any network traffic, doesn't need to interact directly with devices and can be used in secure environments.

SME pricing from £650 scaling to enterprise level

evaluate for free at www.titania.com

2012 Computing Security Awards
WINNER
Enterprise Security Solution of the Year

2012 Computing Security Awards
WINNER
Network Security Solution of the Year

2012 Computing Security Awards
Runner-up
SME Security Solution of the Year

titania nipper STUDIO

CLOUD

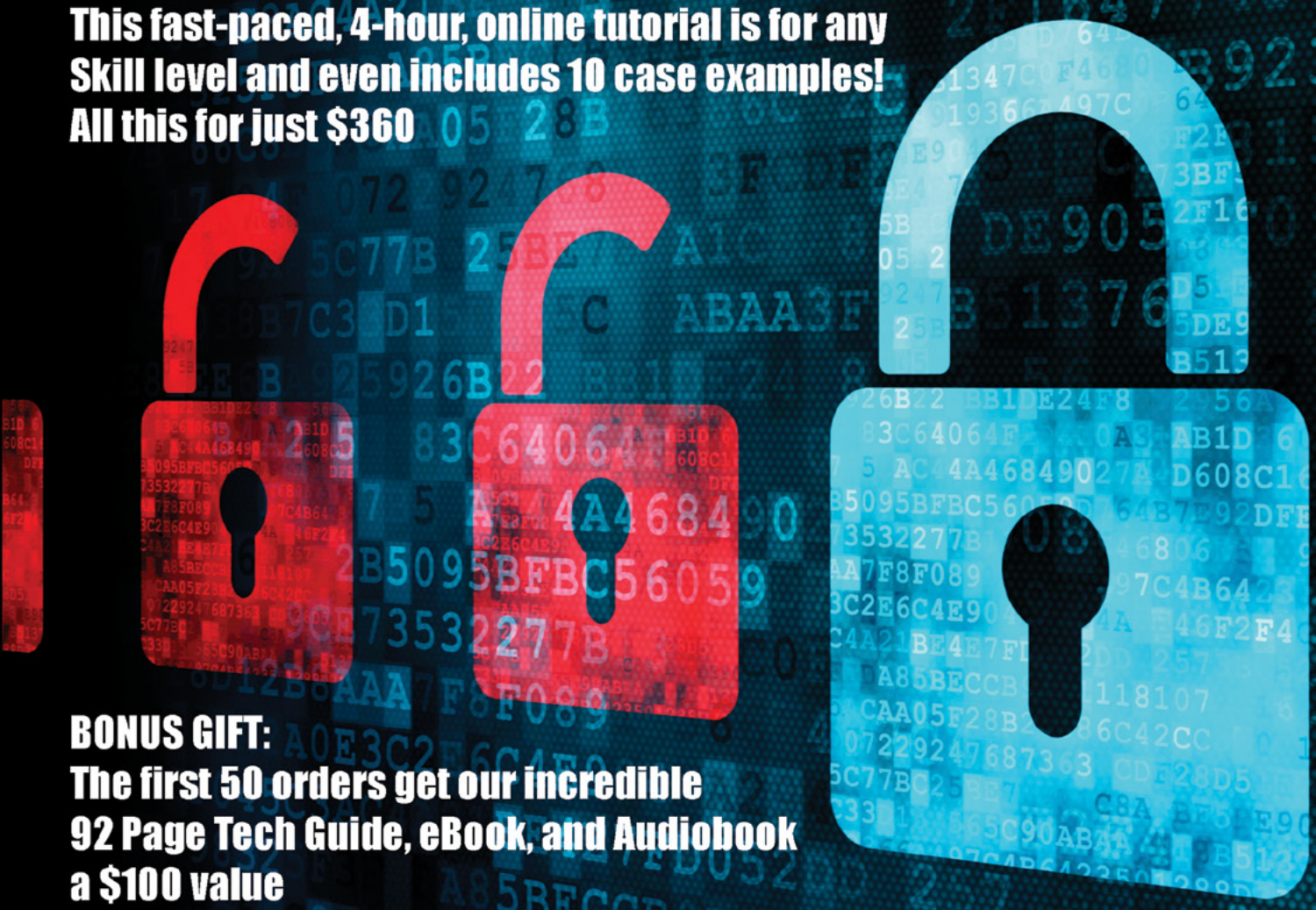
www.titania.com
T: +44 (0) 1905 888785

Make them hang on your every word...
Put them on the edge of their seats when you speak...

Leave them wanting more...
It can happen, but ONLY IF YOU GET THIS:

THE ELECTRONIC ADVANTAGE: 101 the Basics

This fast-paced, 4-hour, online tutorial is for any Skill level and even includes 10 case examples! All this for just $360



BONUS GIFT:
The first 50 orders get our incredible 92 Page Tech Guide, eBook, and Audiobook a $100 value

Go here now and order:
www.technologicalevidence.com

eForensics Magazine
TEAM

Editors: Artur Inderike
artur.inderike@eforensicsmag.com

Betatesters/Proofreaders:
TDL, Gabriele Biondo, Massa Danilo, Richard C Leitz Jr, Olivier.Caleff, M1ndl3ss, M.Younas Imran, Brett Shavers, Johan Scholtz, Nicolas Villatte, Johan Snyman, Pardhasaradhi C.H, Mubarak Al-Hadadi, Kishore P.V, Alex Rams

Senior Consultant/Publisher:
Paweł Marciniak

CEO: Ewa Dudzic
ewa.dudzic@software.com.pl

Art Director: Ireneusz Pogroszewski
ireneusz.pogroszewski@software.com.pl

DTP: Ireneusz Pogroszewski

Production Director: Andrzej Kuca
andrzej.kuca@software.com.pl

Marketing Director: Joanna Kretowicz
jaonna.kretowicz@eforensicsmag.com

Publisher: Hakin9 Media Sp. z o.o. SK
02-682 Warszawa, ul. Bokserska 1
Phone: 1 917 338 3631
www.eforensicsmag.com

Dear Readers!

First of all, I thank you for the attention you paid to eForensics Magazine and this Computer Forensics Jump-start issue in particular. Also, I would like to thank all the reviewers, and people who helped me to prepare this issue for you.
We noticed that Computer Forensics is an intensively growing field, and more and more people are interested in forensics or would like to become an expert in future. But, each newcomer in the digital forensics world has the same questions: "How to start?", "Where I can find all information needed?", "Who can help me?", "Where I can find an expert's opinion?" and finally "Is it possible to succeed without special courses and training?". That's why we decided to gather expert's experience in computer forensics, book review, useful internet sources and so on in one issue to facilitate your search and learning.
We tried to cover as many topics as we could. You will find articles about Forensic ToolKit, steganography, image acquisition, timelining, windows forensics, step-by-step guides etc. In other words, all what every beginner needs to start his or her forensics career or to raise his or her qualification.
Remember, that it's only Volume #1! We noticed that preorder of this issue was very popular and I hope our team will publish another Jump-start issue very-very soon.
One more time thank you very much for your interest and support. I believe you will find this issue very informative and useful. Don't hesitate to contact me though email.

Peace!
Artur Inderike
eForensics Team

BREAKING IN

by David Sekanic

"Warning, your computer's integrity has been compromised!" Well not exactly, but it could happen. When it does we, as end-users, will typically take action. We tend to freak out and consider trashing our computer, find someone to eradicate the dangers, or decide to fix the problem ourselves. Once we have determined our course of action, we are better suited to act upon it. We decide to protect our computers our way.

After doing some research and finally restoring our computer's integrity, we are pumped. The adrenaline is coursing through our veins. Thoughts begin to swirl around in our heads. What can we fix next? How can we keep problems like this from happening again? Someone could make a living taking care of security issues with computers. How can I find the source of the problem? Was it something I downloaded from the Internet? Maybe, it was from a rogue email that I may have opened? Perhaps, I could help other people...

Living in a digital age, we find that more crimes are being committed with electronic devices. Discovery and prevention of those crimes needs to happen, but how do we get involved? How do we "break in" to a somewhat new field in the way of electronics? How can I get started in electronic or Digital Forensics?

That is a great question. It is also a hot topic for several of the IT world listservs, blogs, and forums. Unfortunately the answer is not one we are prepared to hear. We have a desire to be on the forefront where the action is, not always wanting to go through the training, preparation, and experience gathering. We take one class or watch a few videos online and feel like we are ready to take on the biggest project to try to conquer all. That enthusiasm is definitely what we need, but we do need more than that.

Getting started in Forensics is not cheap, and it is not quick, but once in it can be fast-paced. There are several organizations like FEMA and ACCESSDATA that offer either free introductory courses into cybersecurity leading to forensics, or will provide a lite version of their software so users and potential forensicators can become familiar with their products and work towards certification, which is a big step in the right direction.

My recommendations, as a forensicator in training, to others are to start small. I know, you are thinking that you don't want to start small and feel ready to take on the world, but stay with me for a few minutes and understand that I am going through these things right now.

- Check out FEMA's "Domestic Preparedness Campus" through Texas A&M's School of Engineering (You will have to register for courses and as a user, but the courses are free, they are online, and you can do them at your leisure.)
- Go to AccessData's website, download the lite or trial versions of their software and become familiar with it. (Sitting for the ACE exam through AccessData is free and gives an industry cert that is recommended in the Digital Forensics world.)
- Start your own lab with old HDD's and some of the trial or free software available. Keep a log of the drives, the dates and times you accessed them, and what you may have decided to pull from them. (Log your hours and results. In doing so, you are building a portfolio that builds your experience and gets you into the paper trail mindset that allows record keeping and report writing to flow easier when you get established.)
- Set realistic goals for obtaining industry certs and training along with job placement to occur within the next 2-5 years.
- Stay passionate and enthusiastic about what you are pursuing!

Burgess Consulting and Forensics

Data Recovery Experts *Saving Data for Decades*

We can find what you thought was lost forever!

We pioneered the field of data recovery in 1985 and have successfully recovered data for thousands of clients since then.

From spilled frappuccinos to fires, floods and just plain drive crashes – count on us to **save your computer's data**, whatever disaster befalls it.

From personal laptops to smart phones and corporate databases, we pride ourselves on finding data that others can't – on all types of digital media.

With a 90% **success** rate, chances are we can save **your** data too.

Since 1985 we have extracted data from **more than 15,000** hard disks and digital devices.

We can save your valuable data from Windows, Macintosh, Linux, cameras, smart cards, smart phones and most other digital media.

In 2004, the Pine Grove School library in Orcutt, California **burned to the ground**. We **recovered all** of the insurance and inventory **data**, enabling the school to rebuild.

Let us save your data.

Computer Forensics
Expert Witness Services
Data Recovery

Office: 805-349-7676
Fax: 805-349-7790
info@burgessforensics.com
1010 W. Betteravia Rd., Ste. E
Santa Maria, CA 93455 USA

DIGITAL FORENSICS IN A NUTSHELL

by Barry Kokotailo

Before 1999, formal dedicated digital forensics toolkits did not exist. Then came the first free open source tool to perform digital forensics: The Coroners Toolkit created by Dan Farmer and Wietse Venema (http://www.porcupine.org/forensics/tct.html). This sparked a massive revolution in the science and art of digital forensics. This article will deal with the stages in a digital forensics examination, the tools used by most forensics people, and some final thoughts on the world of forensics.

What you will learn:

- Image media.
- Acquire memory.
- Analyze data.
- Present reports.

What you should know:

- Systems administrator
- Network administrator
- Programmer (Some popular languages in use: C, Assembly, C++, Java, Ruby, Python) before venturing into the world of digital forensics.

The first step in the process is the acquisition of evidence. There are two forms of evidence: volatile and non-volatile. I acquire all volatile data first. When the machine is turned off, all volatile data will be destroyed. Non-volatile data will survive when the machine is turned off. If the machine is on, we would like to acquire the volatile memory and save it to a file. If the machine is a Windows based system, the most often used open source tool used is MoonSols DumpIT (*http://www.moonsols.com/2011/07/18/moonsols-dumpit-goes-mainstream*) to actually acquire the image of ram. I usually run the MoonSols app from a large 128 gig USB drive and save the ram image to that drive. After the acquisition of ram, it would be advisable to check to see if the system disk is encrypted. A free tool such as Encrypted Disk Detector could be used to non-intrusively perform this examination (*http://info.magnetforensics.com/encrypted-disk-detector*). If it is and the machine is turned on, it would be wise to acquire an image of the hard drive. Ask the suspect for the encryption keys. If he fails to comply, see if it is possible to create a rescue key.

If there are any mobile devices at the scene (cell phones and tablets) acquire them. If the devices are on and unlocked, place a battery charger to the device to ensure there is enough power to get the device to the lab (*http://www.amazon.com/Enercell-Micro-Portable-Power-23-219/dp/B008PTXXBG*). Place the devices inside of Faraday bags for transport to the lab (*http://www.faradaybag.com/*).

All the tools examining a live system should be run from separate media (USB sticks). Ensure all tools on

your triage media come from reliable sources. Run MD5 and SHA1 hashes of all data acquired in the field. Take detailed notes and record the field triage. You might have to explain your procedure in court (Figure 1-3).

After the acquisition of dynamic evidence is complete, we then proceed to image any hard drives using a free application from AccessData called FTK Imager. With this tool we can preview the hard drive partitions, image partitions, make custom images, export individual files, create hashes, and grab memory. Another feature of FTK Imager is the ability to grab the registry files from a running system. I can also get the files that would allow me to decrypt any files that where BitLockered if this was a Windows target. I would have a very large target USB drive connected to the live system. Would then connect a USB drive with FTK Imager Lite on it. Then would proceed to image the suspect drive(s), registry files, and memory to the target USB hard drive. The memory acquisition is unique to FTK and seems to work with the FTK Forensics Toolkit.

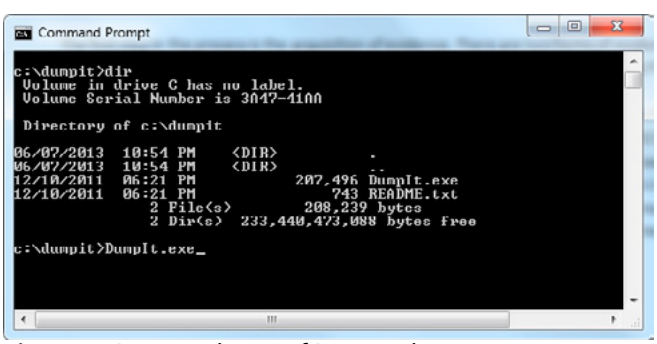


Figure 1. *Directory listing of DumpIT directory on my machine*

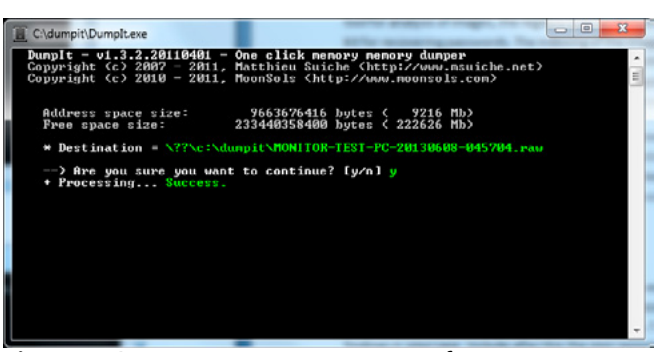


Figure 2. *DumpIt.exe acquiring memory from target machine*

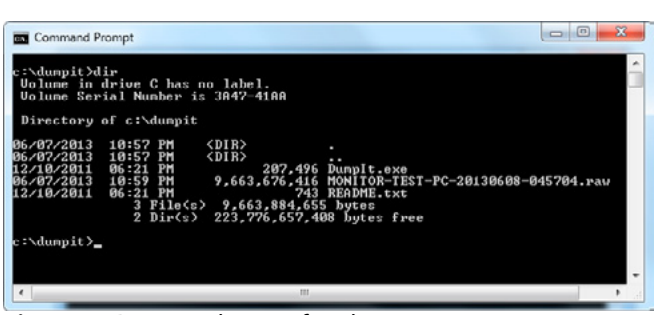


Figure 3. *Directory listing after the memory acquisition runs*

When you acquire images from a non-volatile disk, (it is not powered on), it is important to use a write blocker. The write blocker is placed between the drive you are acquiring and your imaging device. The write blocker ensures that that the forensics systems used cannot change the state of the

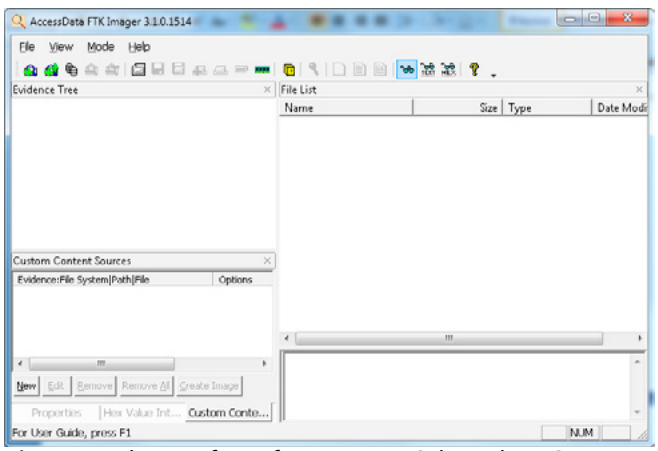


Figure 4. *The interface of FTK Imager. Select File -> Capture Disk Image to start the imaging process*

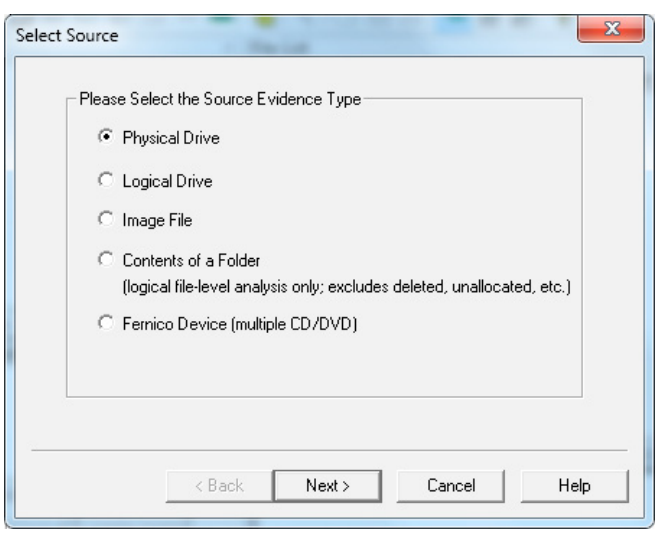


Figure 5. *Select the source to image*

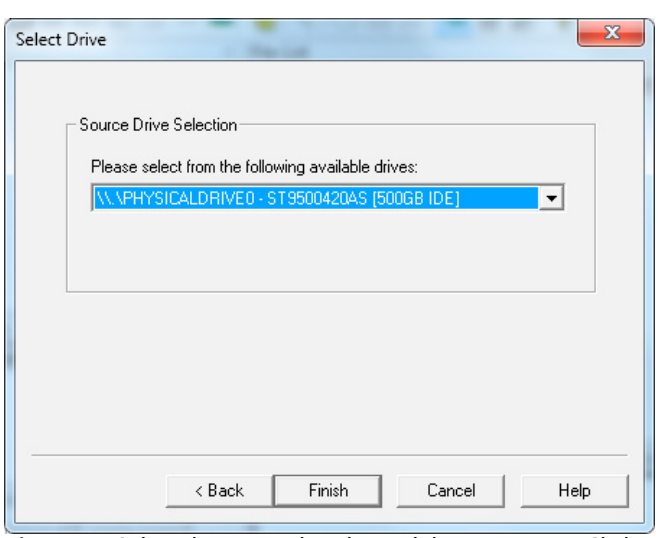


Figure 6. *Select the particular physical drive to image. Click on Finish*

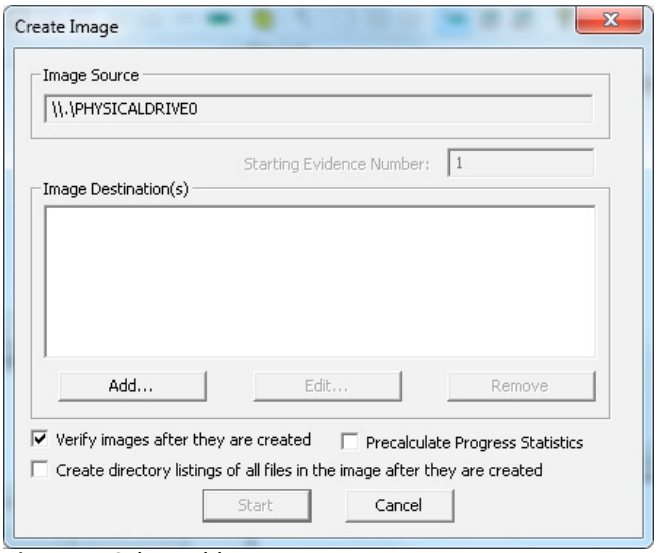


Figure 7. *Select Add*

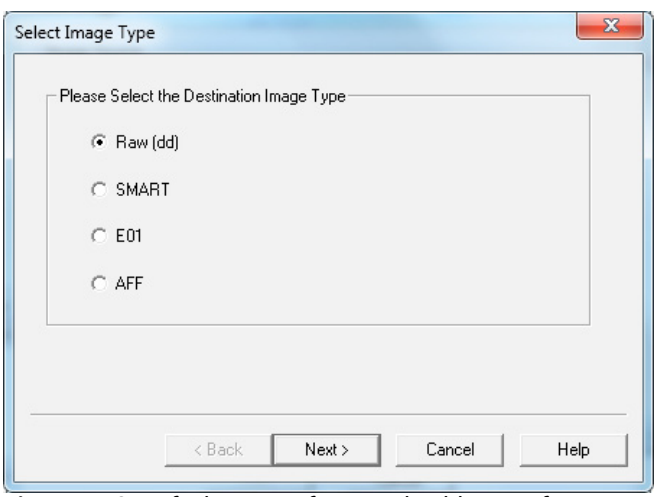


Figure 8. *Specify the image format. The dd image format is recognized by all forensic analysis tools. Others such as E01, for example, is specific to Encase and offers enhanced imaging features*

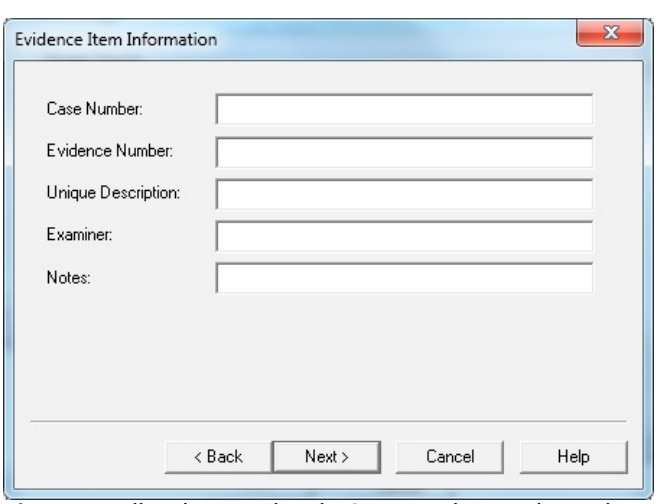


Figure 9. *Fill in the case details. Case Number used must be unique and describes the case. For example, if it relates to theft and the suspect is John Smith and possible title might be 2013-06-15-theft-John-Smith*

evidence. Important if you are going to go to court to prove that the original evidence was not altered. Check out nice write blockers at this URL (*http://www.forensicswiki.org/wiki/Write_Blockers*) Sorry, for this money is needed.

Software imaging is more time intensive then hardware imaging. If you are in the field at the suspects location and have a limited amount of time to acquire your evidence hardware based imaging solution would be in order. A tool such as Logicube's Forensics Dossier unit is a perfect choice for the job -> *http://www.logicube.com/shop/forensic-dossier/*. At capture rates of 7Gigs per minute this is the ideal solution for time sensitive acquisitions (Figure 4-10).

Final thoughts on imaging: usually it is a covert operation. You do not want your suspect to interrupt you. Ensure that this does not happen. Take pictures of the area. Leave the scene exactly the same as you left it. Make copies of anything that will help your case. Make sure it is legal.

SECOND STAGE OF FORENSICS – EVIDENCE ANALYSIS

I use Pelican cases for transport of all evidence and forensics tools (*http://www.pelican.com/canada/*). For transportation of cell phones and tablets I use Faraday devices (*http://www.paraben.com/stronghold.html*). I am back in the lab now and in a controlled and safe environment. If I physically had to take the evidence drives/machines with me, they would be starting the chain of custody process. The chain of custody process is simply a way to verify to a legal entity that the evidence was in your control from the time of seizure to the time it

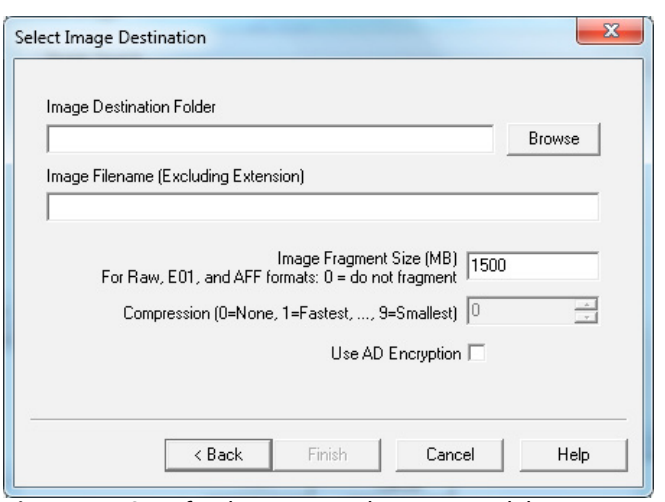


Figure 10. *Specify where to save the image and the image name. Ensure you have enough disk space to save the image. It will take anywhere from a few hours to a couple of days to image depending on the drive size and you machine. If the image is highly sensitive, you might want to save the image to an encrypted drive or encrypt the image itself. Do not forget the passphrase or lose all your keys. Assuming your using public/private cryptography*

went to trial and that its location and who had access to it can be proven. In essence, the original evidence was never tampered with.

We start the analysis by examining our ram image acquired in the field (Figure 11 and Figure 12).

After the recovery of any deleted evidence comes the analysis of the RAM dump.

The next process is to start to acquire all deleted material from the images. I would concentrate on pictures (jpg), documents (doc*), pdf documents (pdf), and email (Outlook, web based, and others). I use two tools to acquire deleted material. One is Recover My Files -> *http://www.recovermyfiles.com/*. This handles most Windows based file extensions. The other tool I use is R-Studio -> *http://www.r-tt.com/*. This is a more comprehensive tool. Both tools require some money, but are very affordable (Figure 13 and Figure 14).

The undelete process could take on the order of a few hours to several days to complete. After the process is completed comes the task of sifting through the recovered files looking for anything related to your case. Some are targeted. For example if I am working a porn case, obviously I would be looking for media files that contain porn material. I would be performing searches through the documents looking for keywords related to the case. All material that is identified as potential evidence is placed in my case folder for inclusion into the final report.

The Volatility Framework has made this process extremely easy. The first step is to identify the type of Windows architecture that this comes from. Then we use a set of pre-existing templates that were constructed to extract forensic evidence

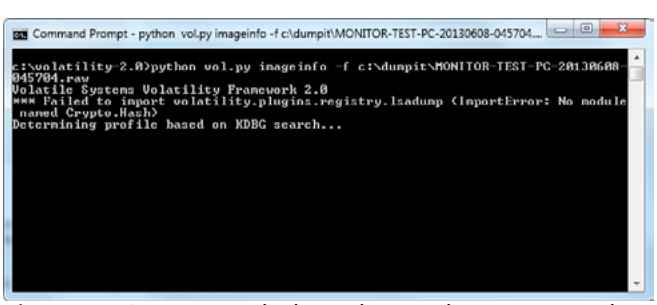


Figure 11. *Determine which windows architecture is used. This process is time intense. If you know the system use that profile. As you can see if the screen shot, the process is still working*

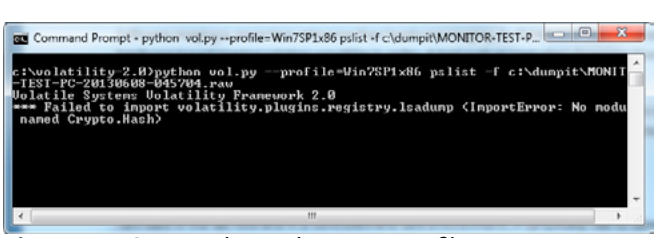


Figure 12. *Once we know the image profile, we can use the various plug ins to obtain forensics information from memory. In the above example I am getting the process list for examination*

of interest from the ram image. These templates can be acquired from here -> *http://code.google.com/p/volatility/* and *http://code.google.com/p/volatility/wiki/CommandReference*.

Now comes the time to examine and extract any evidence from the hard drive images acquired from the field. There are three tools that can be used for this. Two are commercial and the other is an open source tool. Guidance Software puts out a tool called Encase that allows analysis of a forensic s image. The other commercial tool is FTK tool kit from Access Data. Of the two commercial tools I would recommend the FTK product. Inexpensive, comes with several tools in the tool kit. The FTK Forensics tool for analysis of images, the registry tool for Windows registry files, and the Password Recovery Tool Kit for recovering passwords. The indexing of the image occurs first with FTK. With Encase, indexing takes place when you want to perform a search, which is more time consuming. EnCase changed this in their latest software so indexing takes place first as with FTK. The open

Figure 13. *The Recover My Files site*

Figure 14. *The R-Studio site*

source tools is the SIFT Tool Kit from the SANS organization. The major drawback with the SIFT tool is that its Unix based and mostly command line driven. Requires a bit more training and time to use (Figure 15-17).

Whatever tool you use, it comes to finding the evidence. I look in common areas where data is usually kept. I search for areas where applications keep their log files. I look for things that should not be there. I eliminate files that are known to be "good" files through the use of hashes. I do searching for keywords that are related to the case. Once these searches are completed any results are placed in the case folder for inclusion in the final report.

For the analysis of portable devices, there are number of commercial tools to acquire forensic evidence. They are:

Figure 15. *The EnCase site*

Figure 16. *The Access Data site*

- Cellebrite -> *http://www.cellebrite.com/*
- MPE+ -> *http://www.accessdata.com/products/digital-forensics/mobile-phone-examiner#.UbzkUJwmzPo*
- XRY -> *http://www.msab.com*

The procedure for cell phones or tablets is to ensure the device is protected by a Faraday device such as a Faraday tent. Connect the protected cell or tablet to the forensics workstation via the appropriate cable. You must have the pass codes or have the phone jail broken. Run the above software on the device. All three pieces of software are fairly intuitive to run. There are educations courses on their use. Problem is they are quite expensive. Basically they acquire the data from the cell or tablet, analysis the data and place it in a nice report. I am interested in the logical and physical acquisition of the device.

THIRD STAGE OF FORENSICS – REPORTING

I have acquired the data, parsed through the data to extract all evidence related to the case. Now it is time to place all this in a report. The sentence to keep in mind for the report is “keep it simple”. Start off with an executive summary that states your findings. Then explain the process used to discover the key findings in your case. Include after this the mini reports generated by the tools used. This is where I like FTK. It has report templates that simplifies and saves time in the detailed report generation. Anyway you would like to compose the report; it has to be understandable by any lay person reading the report. If called to court to testify, try to explain your findings in a way that any layman can understand but without landing yourself in contempt with the judge. A great resource for report writing and court presentations can be found at the Certified Fraud Examiners website:

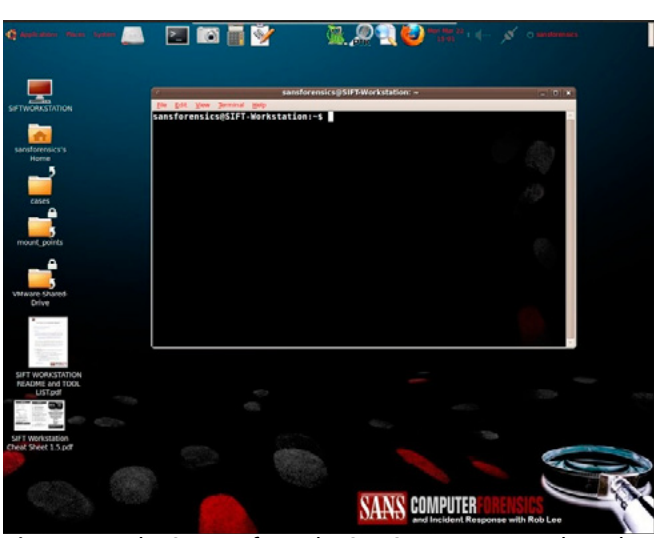


Figure 17. *The SIFT Kit from the SANS group. A UNIX based forensics system that is free*

- How to Become a Dangerous Expert Witness: Advanced Techniques and Strategies (*http://www.acfe.com/products.aspx?id=2502*)
- Writing & Defending Your Expert Report (*http://www.acfe.com/products.aspx?id=2508*)
- Depositions: The Comprehensive Guide (*http://www.acfe.com/products.aspx?id=2497*)
- Cross-Examination: The Comprehensive Guide for Experts (*http://www.acfe.com/products.aspx?id=2494*)

Although this is based on US legal systems, most of the material in these books can be universally applied.

CONCLUSIONS

Initially when digital forensics came about there was a large surge as people, professing to be forensics people, came into the field lured by the large dollar signs ($350/hr. USD). Time has vetted such people and more and more forensics is recognized and being standardized in the real world.

Forensics is time consuming and tedious work. It is meticulous work. Your explanations and procedures will be challenged in court. Remember, someone's freedom is on the line.

The above tools mentioned comprise the bare minimum. Always search for tools to augment your lab. Test them and ensure they can pass the court acid test. Ensure they give accurate reliable results.

Train, read, and constantly enhance your skills in all aspects of computing science. Programming, networking, and operating systems are key knowledge areas that you will have to be proficient in to survive and prosper in this business.

The reasons you are in forensics fall into the following categories: 1) You are working for the police. 2) You work for major consulting firms 3) You work for private firms. 4) You work for the bad guys. 5) Personal research.

Digital forensics is a focused market. Competition is tight and so is the demand for qualified people.

Hope you the best in the field of forensics.

About the Author

I have been working in the IT field since 1986. In that time I acquired knowledge and experience in Windows, MacIntosh, Unix, networking, programming, pen-testing, forensics and incident response. I have acquired several certifications: CISA, CISSP, EnCE, ACE, CSA, CSNA, RET, CEH. I currently work for the Edmonton Public School Board in Edmonton, Alberta, Canada and operate my own company Cerberus Security Integrators Inc. http://www.forensics-canada.com/ in my spare time. I teach classes at a local post secondary institute in forensics and Unix operating systems. When I have some free time I golf and fly fish. A more complete profile of me can be accessed over at http://www.linkedin.com/pub/barry-kokotailo/28/565/405.

HOW TO QUICKLY PROGRESS TO AN EXPERT FORENSICS CONSULTANT

by Andrew Bycroft

In our information economy we are deeply immersed in technology for a large portion of each day and by that very nature, we are all leaving behind clues about ourselves and our activities – so called digital fingerprints. Should someone step across the boundary into the realm of dubious or criminal activities, those digital fingerprints are the very evidence that forensic consultants are engaged to find. If the field of forensics is on your radar, then this article will teach you the fundamentals to launch your career as a forensics consultant.

What you will learn:

- The ten essential types of tools in a forensics consultant's toolkit
- Top six areas of skill and experience to focus on
- The most sought after forensics training and certifications
- Six steps to starting your first forensic investigation
- Twenty hot forensics tips to turn beginners into experts

What you should know:

- What digital forensics is
- How to run commands on a Linux system
- About different operating systems and file systems
- About TCP/IP networking

Unlike consulting in many other fields in information technology, a forensic consultant needs to invest in a toolkit that includes a number of pieces of software and hardware, and due to the rate at which technology changes that initial investment is growing in magnitude. You may be able to get by with some freely available pieces of open source software, but make no mistake, there will be some investments in hardware which will be much more difficult to escape. To reduce your costs of entry into forensics, for each tool type where software is available, up to three freely available tools will be listed:

IMAGING AND CAPTURE TOOLS

Disk or media imaging hardware or software will allow you to capture the entire disk of a compromised server or workstation, and in some cases, perhaps multiple drives and multiple systems. A hardware imager may allow you to capture images from multiple different operating systems, whereas if you take the software route you may need an imager for each operating system. It is also important to understand that most operating systems support multiple types of storage ranging from hard disks to USB flash to flash based memory cards such as SD, Micro SD and memory stick formats. In addition each operating system may support multiple file system types. Windows for example could be using FAT12, FAT16, FAT32, NTFS or ReFS, which you may have seen in Windows Server 2012. A Linux system may be using EXT, EXT2, EXT3, EXT4, ReiserFS or F2FS to name a few. You literally need to be prepared for just about

any operating system and file system. It could be a Solaris system; it could be a BSD system; or it could even be a mobile device running Blackberry, iOS, or Android operating systems. As an example of the diversity one may expect to find on the job, a former colleague once had to examine the hard disk of a Commodore Amiga computer as it was used as a backup medium for data that had been exfiltrated from a defence contractor. Some of the easiest disk images to collect are those of Virtual Machines in which you are simply copying a file which contains an entire disk image and sometimes you may also be collecting one or more files which contains state information showing what has changed from the default disk image. Some good examples of freely available disk imaging tools that run under Linux are:

- dc3dd;
- ewfacquire; and
- guymager

Memory imaging is also a fundamental requirement in digital forensics, especially if the attack is still in progress, it may be possible to capture information from running processes and great insight to the use of root kits or exploits that are being executed. To achieve this you will need memory imaging hardware or software. Like disk imaging you need to be prepared to capture the memory contents from almost any operating system. Due to extra security measures implemented in Windows Vista, Windows 7 and Windows 8, some software based solutions will not be able to capture memory images, in which case you may need to invest in one of the hardware solutions. Another thing to be mindful of is that if a machine is locked, performing a local login to capture the memory image will later the contents of the memory, so in that case, an external hardware based memory imager may be required. Freely available disk imaging tools that run under Linux are:

- fmem; and
- LiME

Packet capture hardware or software is useful for being able to capture attacks in progress and to replay these back. The analysis of network traffic for clues about an attack is referred to as Network Forensics. You can think of it as being a bit like a DVD recorder for the network. Network packet capturing tools should be able to capture the packet headers and data and should be protocol agnostic, just in case you have to analyse some legacy protocols such as DECNET, SNA, IPX, or AppleTalk. Many packet capturing tools will also have some built in analysis capabilities to make interpreting the network traffic much simpler, but these will typically be limited to analysis of TCP/IP traffic. It should be noted that although packet captures are of greatest use to capture details of an attack in progress, there is still some value in being able to capture traffic to and from a compromised system after the completion of an attack in case it is still attempting to contact any systems used by an attacker. Linux supports a number of free packet capture tools such as:

- tcpdump;
- Wireshark; and
- ettercap

ANALYSIS TOOLS

Once you have captured the evidence you will then need another set of tools to analyse it. Due to ever decreasing cost of storage the amount of effort you may need to expend looking for evidence could be compared to finding a needle in a haystack, so it is important that the tools you have at your disposal help you search and filter through a copy of the evidence collected.

Disk analysis tools help you piece together files that could be distributed across multiple blocks of a single storage device or even across multiple logical volumes of storage. The ability to search for specific characters or strings. Similarly memory analysis tools allow the same kind of organization and analysis of running processes at the time of capture in order to look for specific characters or strings of interest. Open source disk analysis tools that run under Linux include:

- Sleuthkit
- The Coroner's Toolkit (TCT)
- Digital Forensics Framework (DFF)

Disassemblers can help you look through memory contents and files at the instructions of execution and determine whether the instructions are malicious in intent. This can help determine analyse rootkits and malware to learn more about the sophistication of the attacker, what damage the attacker can manifest and may even provide clues to the location or identity of the attacker. The process of breaking executable code into a set of instructions is referred to as reverse engineering. Some freely available disassemblers under Linux are:

- udis86;
- lida; and
- objdump

File recovery tools, often referred to as file carvers, help recover files that have been deleted or corrupted and may give valuable clues to activities that were being performed by an attacker prior to the at-

tacker attempting to cover their tracks. A few popular and free file recovery tools for Linux include:

- scalpel; and
- Foremost

Another important file analysis toolset is that which can perform steganalysis to look for hidden data within files in case deceptive techniques to covertly conceal information were used, potentially to smuggle valuable content inside seemingly innocuous files. Steganography techniques are commonly used to hide confidential text within image files. Whilst the color within the image file is altered when steganography is used, it is often undetected by the human eye. Freely available steganalysis tools which run under Linux include:

- Stegdetect; and
- StegSpy

As each operating system and application conforms very loosely to a number of popular log formats, log file analysis tools are also valuable for normalizing data and making it easier to search for specific events that have occurred. Essentially, what you will want to do is search and filter out specific information from log files based on a regular expression match. A few freely available tools suitable for filtering out text under Linux include:

- grep; and
- awk

Network analysis tools will help you sequence the packets of network communications to be able to piece together any unencrypted IP based telephony conversations, web pages viewed, e mail communications sent and received, and even instant messaging conversations. You may be surprised at the little clues that may be discovered in network traffic. One former colleague was on a case attempting to determine the source location of three attacks with the same modus operandi launched on three separate targets in the banking and finance industry and piecing together the network traffic and whilst each of the attacks originated from different public hotspot locations the identity of the attacker was able to be revealed because he accessed his Facebook page shortly after the third attack. The most popular freeware tool under Linux for this is:

- Wireshark

Finally, you will need some kind of storage to collect the images and packet captures. Large portable hard disks are the most convenient as these can be used to capture large disk images. It is important to ensure that each new project is placed into a separate logical partition if at all possible, or at the very least in its own directory so as to not confuse evidence collected with that from previous forensics investigations. In order to conserve storage space you should perform a checksum of the contents copied to ensure that any collected data has not had its integrity compromised and you can then compress it and encrypt it.

TOP SIX AREAS OF SKILL AND EXPERIENCE TO FOCUS ON

Though forensics may be a specialist field, the broader the knowledge you have with various systems, devices and networks, the better equipped you are to deal with the breadth of cybercriminal activities. If you have a general love for technology, you will fare better than someone who knows the inner workings of OpenBSD, unless of course you want to focus on a niche within digital forensics that just deals with crimes committed on or using OpenBSD for example, which could pay well but be very limiting in terms of opportunities.

- In general it helps if you have at least had some exposure to various flavors of Windows, Unix and Mac OS and if you want to investigate the growing trend of crimes committed using mobile devices then some exposure to iOS, Android and BlackBerry.
- Experience with different file systems and knowing how to recover deleted files is important. Understanding that deletion typically involves writing over entries in file and directory tables on the file system rather than rewriting over each block used by a file is key to recovering information that may have otherwise been considered to be lost.
- Programming knowledge is useful, in particular assembly code if you want to analyse the behaviour of applications and either identify hidden malware within programs such as root kits or need to understand the outcomes of a particular exploit in order to examine the footprint of an attack. As more and more attacks are targeted at web applications understanding of one or more of the popular web coding languages such as HTML, ASP, JSP, Java, Flash, JavaScript, PHP, Perl, Ruby and Python will be valuable.
- An understanding of the seven layers of the OSI model and the functions performed by the presentation, transport, network and data link layers as well as having some exposure to packet capturing of TCP/IP traffic would be of extreme value.
- The ability to learn and be prepared to tackle new technologies would easily be the most valuable of skills to have, however, as technological advancements are not likely to slow

down any time soon, and, if anything, march forward at a growing pace.
- In almost all circumstances you will need to provide some form of report of your findings so it is important to have good oral and written communications skills and to be able to explain the results to non-technical audiences.

THE MOST SOUGHT AFTER FORENSICS TRAINING AND CERTIFICATIONS

Whilst there is nothing like learning through experience and it is recommended you have at least 100 hours of practice conducting imaging and analysis from your own computers and packet capture and analysis from your own networks, undergoing formal training and obtaining a certification can quickly establish you as someone who is committed to the profession.

SANS (Systems Administration, Network, and Security) Institute offers three GIAC (Global Information Assurance Certification) certifications:

- GIAC Certified Forensic Analyst (GCFA)
- GIAC Certified Forensic Examiner (GCFE)
- GIAC Reverse Engineering Malware (GREM)

The Information Assurance Certification Board (IACRB) offers three certifications:

- Certified Computer Forensics Examiner (CCFE)
- Certified Data Recovery Professional (CDRP)
- Certified Reverse Engineering Analyst (CREA)

The International Association of Computer Investigative Specialists (IACIS) offers two certifications:

- Certified Forensic Computer Examiner (CFCE)
- Certified Electronic Evidence Collection Specialist Certification (CEECS)

The former American Society for Industrial Security (ASIS International) offers a certification:

- Professional Certified Investigator (PCI) – not to be confused with Payment Card Industry (which also uses the acronym PCI and is the body which governs the usage of credit cards)

The International Society of Forensic Computer Examiners (ISFCE) offers a certification:

- Certified Computer Examiner (CCE)

The Electronic Commerce Council (EC-Council) offers a certification:

- Certified Hacking Forensic Investigator (CHFI)

Mile2 offers a certification:

- Certified Digital Forensics Examiner (CDFE)

The Association of Certified E-Discovery Specialists (ACEDS) offers a certification:

- Certified E-Discovery Specialist (CEDS)

The CyberSecurity Institute offers a certification:

- CyberSecurity Forensic Analyst (CSFA)

The Digital Forensics Certification Board (DFCB) also offers a certification:

- Digital Forensic Certified Practitioner – Fellow (DFCP-F)

A number of forensic tool vendors also have training courses and certifications specific to their products.

If you are not sure which certification is the best one to start, the SANS courses are among the most well-known and respected with the GIAC Certified Forensic Analyst (GCFA) offering a strong syllabus and hands on exercises for acquiring, analysis and handling evidence. If you want to get into more advanced analysis of files including being able to reverse engineer code to determine whether specific applications were intended to cause harm, then following up the GCFA with the GIAC Reverse Engineering Malware (GREM) course and certification would help you gain knowledge about malicious code. If on the other hand, you were more interested in the legal side of forensics to better assist law enforcement with prosecuting cybercriminals, then the ISCFE Certified Computer Examiner (CCE) certification would be a great addition to the SANS GCFA.

All of these courses and certifications range in cost and content so it is well worth doing some research to find the ebst fir for you.

SIX STEPS TO STARTING YOUR FIRST FORENSIC INVESTIGATION

One of the most cost effective ways to start is to download the SANS Investigate Forensics Toolkit (SIFT) which comes as a virtual machine or as a bootable DVD image. It contains tools to help you capture disk images from the most common file systems used on Windows, Mac and Linux systems and supports common forensic image formats such as dd (disk dump), aff (advanced forensics format) and ewf (expert witness format). We will now explore a common set of operations for the typical forensics investigation of a compromised Linux system.

Once you have SIFT up and running the first step is to mount the disk you will use to capture the images to in read-write mode. Assuming you want to mount the disk connected to the USB interface to store your evidence and mount it to a subdirectory called “forensics” beneath the “mnt” directory you would issue the command:

```
mount /dev/usb /mnt/forensics
```

Next you will need to enter the forensics directory and create a subdirectory within it for storing evidence that you collect. It is useful to include the case number, client name and date in the directory name for easier referencing and searching in the future:

```
cd /mnt/forensics
mkdir case-001-xycorp-20130524
cd case-001-xycorp-20130524
```

Next you will need to identify the disk for which you need to create an image. If it is the primary disk of the compromised system then it will typically be referenced as hda (if it is an ATA or SATA disk. If it is a SCSI disk then it will be referenced as sda). It is important that you do not mount this disk as doing so many contaminate the evidence. It is also possible to capture a specific partition within a disk.

Then you can start to take an image using the dc3dd tool. This tools will take a complete replica of all the blocks of a filesystem which you can then later analyse. It is helpful to include the name of the compromised system and the disk reference the name of the captured image:

```
dc3dd if=/dev/hda1 of=financesystem-xy-001-hda1.img
                    conv=noerror,sync
```

The noerror option is useful in case the disk from which you are capturing the image has some bad sectors so that the capture continues regardless of the physical state of the disk.

Now you will want to mount the captured image and start analysis. Let’s assume you want to mount the image to the “analysis” subdirectory of the “mnt” directory which has already been previously created:

```
mount -t ext2 /mnt/forensics/financesystem-xy-001-
hda1.img /mnt/analysis -o loop,ro
cd /mnt/analysis
```

You will notice that the options include “ro” which means that the image is being mounted as read-only. This is important because you do not need to make changes, so mounting it as read-write runs the risk of contaminating the evidence. For an extra level of safety you can also make a backup of the captured image.

Now it is time to analyse the captured image and to do this you will use the Sleuthkit which comes with a number of convenient tools. One of the more useful ones allows you to search through the image for specific strings. For example if you wanted to look for evidence of the string “confidential”:

```
srch_strings -e l -t d /mnt/forensics/
financesystem-xy-001-hda1.img > /mnt/forensics/
financesystem-xy-001-hda1.str
grep - i confidential /mnt/forensics/
financesystem-xy-001-hda1.str
```

This allows us to perform a non case sensitive search for the word confidential and allows us to locate where within the image that string appears.

If the attack is still in progress or you want to analyse the compromised system to determine if there is any suspicious traffic being sent to it or being sent from it we can use the tcpdump command. Firstly, you would need to plug your forensics collection system into an available port on the same switch as the compromised system. You would then need to have the interface that the compromised system is connected to mirrored to the interface your forensic collection system is connected to. This technique is known as port mirroring or port spanning.

Assuming you just have one network interface in your forensics collection system, it will usually be referenced as `eth0`.

You would then issue the command:

```
tcpdump -i eth0 -s 65535 -w /mnt/forensics/
  financesystem-xy-001.pktcap
```

What this does is capture the entire packets rom the primary ethernet (network) interface of your forensics capture system which are a copy of those being sent from and received by the compromised system. The captured packets are being stored in the file named `financesystem-xy-001.pktcap`.

As this file can grow large in size quickly and it is not easily read by humans, it is helpful to use a graphical tool for analysis, known as Wireshark which allows you to color code different types of network traffic, filter out certain types of traffic and home in on what may be of interest. You can also use the streams option to piece together communications and view web pages that were browsed, email that was sent or received and various other types of communications that are not encrypted. It takes a lot of patience and lots of searching to find evidence within clear text packets, but Wireshark is one of the best freely available tools to help you complete this process.

If you can master these steps you will have taken your first steps to becoming an expert forensics consultant. Keep in mind that it takes a lot of practice to become a master, but every hour you spend working on forensics cases will bring you closer.

TWENTY HOT FORENSICS TIPS TO TURN BEGINNERS INTO EXPERTS

There are many useful tips which you will need in order to succeed in forensics. Here are twenty of the most important ones:

- Carry a portable computer such as a notebook with you for forensic investigations and use it specifically as a forensics collection and analysis system. IT is useful if the system has the following components often used by forensics consultants:
  - A wireless network interface (preferably to handle all of the 802.11 standards).
  - A wired network interface capable of at least gigabit speeds.
  - At least two USB 2.0 or better interfaces to connect storage.
  - A writeable optical drive in case you need to either load a forensics toolkit from CD-ROM or DVD-ROM or record evidence to a CD-R or DVD-R optical disc.
- Carry around any software or hardware collection systems, imaging tools or analysis tools and if any of those are hardware based, make sure you can keep those powered sufficiently for the duration of your work. Unless you can be absolutely certain you can capture what you need in an hour or less, relying on battery power is short sighted.
- Carry around some form of portable storage device. E-SATA or SCSI may be good choices, depending on the interfaces available on your collection system.
- Carry your mobile camera phone to premises where photographic cameras are permitted, as photographic evidence may sometimes be useful.
- Carry a set of screwdrivers with you in case you have to remove storage items from a device.
- Carry a set of cables, adapters and connectors with you including, but not limited to:
  - USB cable.
  - Mini USB cable.
  - Micro USB cable.
  - Serial cables with DB9 and DB25 interfaces
  - Null modem cable.
  - Category 5e or better RJ-45 to RJ-45 straight through cable.
  - Category 5e or better RJ-45 to RJ-45 crossover cable.
  - RJ-45 to RJ-45 roll cable.
- Carry a multi format card reader with you in case you need to gain access to Compact Flash, XD, SD, Micro SD or Memory Stick formats.
- Look for files that are marked as hidden by the operating system. These have the hidden attribute on Windows systems and typically begin with a period (.) on Unix systems (including Mac OS X)
- Remember that steganography may have been used to covertly hide data within other files. You will need to have a selection of steganalysis tools to look for evidence of this.

- Look at command history files on Unix systems.
- Look for deleted files and directories, remembering that file systems typically do not actually delete files but simply remove the entry for a deleted file or directory from the file and directory tables. Provided compromises are detected early, it may be possible to recover part or all of a deleted file and uncover incriminating evidence an attacker did not want you to see. Even remote wipe programs for most mobile platforms nowadays do not do a complete multi-pass rewrite of storage meaning that it may still be possible to recover some of the data.
- Parse log files to look for evidence of unauthorized changes. Don't rely on log files on compromised systems. If at all possible see if a copy also exists on a remote Syslog server just in case an attacker has attempted to cover their tracks and delete log files on systems they have compromised.
- Advise clients not to power down compromised systems as some of the best evidence is often resident in memory. If the client really must stop an attack in progress it is better to either disconnect compromised systems from the network or use firewall rules or IPS rules to block the source of the attack.
- You will need to connect your forensics collection system into the same switch as a compromised host in order to capture packets to or from a compromised system; and the switch must be capable of mirroring network interfaces. The alternative is to carry around a network tap or to use packets captured either by an existing client owned intrusion prevention system or network forensics tool, if the integrity of those packet sources can be established.
- Quickly ascertain whether the network communications are in clear text or encrypted. If encrypted the results will be of little value and may only tell you that there was some kind of communications occurring to or from a compromised system and the duration for which the encrypted session was active. You will not be able to see exactly what was being communicated by the attacker.
- Determine whether all of the sources of evidence are time synchronized using an NTP (Network Time Protocol) server, preferably with a stratum of 3 or less. This will make it easier to determine the sequence of events in an attack and should make it simpler to prepare the evidence for court and reduce the risk of it being dismissed as inaccurate.
- Does the client have a SIEM (Security Information and Event Management) platform that has already performed some correlation of multiple log sources to provide some interesting data? If the integrity of the SIEM and the logs it has received can be established, it may be able to provide some useful evidence and reduce the time spent conducting forensic analysis.
- Determine whether there are any other physical audit trails which may be of use. Not all attacks begin life as electronic attacks. Some may begin with a physical element such as installing a hardware based key logger or a wireless device or even disclosure of information in hard copy formats. In some cases the attackers may be sitting in the car park to carry out the attack or may have done some physical reconnaissance prior to launching the attack. Surveillance video feeds may help identify suspicious physical activity from strangers which can be correlated with digital evidence to help identify the attackers.
- Above all, make sure at all times that you maintain a chain of custody which involves careful handling and preservation of evidence such that it is submissible as evidence which safeguards your integrity and helps your client achieve a desirable outcome in court.

ON THE WEB

- *http://computer-forensics.sans.org/community/downloads* SANS Investigate Forensic Toolkit (SIFT) description, details and download
- *http://www.giac.org/certification/certified-forensic-analyst-gcfa* SANS GCFA certification details
- *http://www.giac.org/certification/reverse-engineering-malware-grem* SANS GREM certification details
- *http://www.isfce.com/certification.htm* ISCFE CCE certification details

SUMMARY

If you are the kind of person who knows a little about a lot and has a love of technology combined with a passion for solving mysteries then you have the makings of a good forensics consultant. The rest is up to you to obtain the tools you will need to do the job and obtain the knowledge and experience you will need to get the job done as journey into the world of uncovering the digital fingerprints that others have left behind.

About the Author

With close to two decades of industry experience, Andrew Bycroft is passionate about helping organizations create a more secure online experience and develop a competitive edge by educating and providing strategic direction to decision makers and influencers on taking a risk driven approach to information security rather than a technology, budget or compliance driven approach. Andrew is founder, director and thought leader at The Security Artist (http://www.thesecurityartist.com)

CYBER ATTACKS ARE ON THE RISE.

SO, YOU THINK YOUR SYSTEMS AND NETWORKS ARE SECURE?

Think again – you've already been attacked and compromised.

And, we should know because we did it in less than four hours. Here's the good news: we're the good guys. We can tell you what we did and how we did it, so you'll be prepared when the bad guys try it – and they will. We'll show you how.

- ✔ COMBAT CYBER ATTACKS
- ✔ ENSURE RESILIENCE
- ✔ MITIGATE RISK
- ✔ IMPROVE OPERATIONAL EFFICIENCY

Visit www.KnowledgeCG.com to learn how KCG's experienced, certified cybersecurity professionals help our government and commercial customers protect their cybersecurity programs by knowing the threat from the inside out.

TRUSTED CYBER ADVISOR

KNOWLEDGE consulting group KCG

A STEP BY STEP GUIDE TO BEGINNING COMPUTER FORENSICS

by David Biser

We live in an era of digital connectivity such as the world has never known. Each age has one symbol that seems to identify it to all other time periods, for instance Roman is known by the Imperial Eagle, the Industrial Revolution by the machines that were developed and used, our age can probably be symbolized by 1s and 0s.

What you will learn:

- Some tools, both software and hardware that you can use in conducting forensic examinations
- Various techniques that will aid you in your quest to become a computer forensic examiner.
- We will also be learning about file formats that a computer forensic examiner should be familiar with and how to store those digital files.
- We will also examine the imaging process and learn more about it, step by step, up until the analysis portion of the examination.

What you should know:

- Several different operating systems, such as MAC, Windows and Linux.
- You will have to be able to handle examining log files, slack space and then be able to explain all of your activities to a judge, jury or even to other investigators.
- You should also have a knowledge of basic computer hardware to aid in the dismantling of systems that you will be examining.

Nearly everyone is connected to the Internet in some form or manner, by smart phone, tablet or laptop. With such connectivity comes crime which brings the need for investigators with a specific skill set to be able to investigate, track and apprehend criminals in the digital world. This is where the exciting and ever changing world of computer forensics begins. As a computer forensic examiner you will find yourself tracking child pornographers, cyber thieves and terrorists, responding to the worst of crimes, all in an effort to deter and stop cyber crime. A very exciting field indeed!

Seem daunting? To many it is, but again, this is part of the excitement and sense of purpose that you can find in working as a computer forensic examiner!

WHAT IS A "FORENSIC" EXAMINATION

A very important point to remember here is that you are conducting a "forensic" examination. The use of the word forensic is important and you should know what it means and how it applies to what you are doing. Data recovery and computer forensics are two separate fields, with differing standards, but sometimes utilizing the same tools and processes. As a computer forensic examiner you are obtaining evidence that might be presented before a court of law and the word forensic is used to describe that process. Your work as a computer forensic examiner will be reviewed by many people so it must be of the highest quality, it must be repeatable and it must be accurate. Computer forensics can be defined as collecting

computer data securely and accurately, analyzing or examining that data to discover evidence, presenting the evidence to the court and doing all this within the scope of various laws. We will not cover these topics in this article, so go and explore these realms more fully!

When a crime occurs, a report is made and the investigation begins. Hopefully this investigation will lead to the issuance of a search and seizure warrant and the seizure of digital evidence pertinent to the case under investigation. It is extremely important to only conduct computer forensic examinations under the right legal conditions. Whether it is a search and seizure warrant that details specifically what evidence is going to be searched for, or consent from the owner or as part of a civil process, you must have the proper legal justification for examining the evidence. If you do not, then you open yourself up to liability, which you most certainly do not want to do. Protect yourself and ensure that you have adequate legal justification for the work you are going to perform.

THE START OF A COMPUTER FORENSIC EXAMINATION

So you are a computer forensic examiner, sitting in your laboratory/office, when the call comes in, "Hey, we are bringing you a computer to examine!" Your heart begins to race, your mind begins to run through a variety of scenarios and questions begin to form that demand answers! What kind of a case? What condition will the computer be in? What tools am I going to utilize to extract the pertinent data? And so your mind will go as you eagerly await the arrival of the evidence. It is here we must begin.

As a computer forensic examiner you must become an expert at all things technical, including the dismantling and reconstruction of computers. I have discovered over the past 10 years as a computer forensic examiner that every computer is different! It doesn't matter if it should happen to be two Dell Inspirons, you will find that the hard drives are never in the same place twice! Well, I might exaggerate a little bit here, but the truth of the matter is that nearly all computer manufacturers place their hard drives in different locations. Some are easy to remove and others not so easy! So, as you begin to study and train to conduct forensic examinations of digital equipment also begin learning all you can about the hardware construction.

If you are conducting a "dead box" examination then you have to know where to find the drive, how to remove it and how to put it back in place. You should invest in a good computer tool kit. These can be found in most computer stores and are worth every penny! A good tool kit should include several sizes of screwdrivers, torque wrenches, slotted bits and needle nose pliers. You should also invest in an anti-static wrist band. Since computers are electronic and you are going to be handling the interior parts of one, you do not want to inadvertently shock the system. You could accidentally fry a fragile electronic piece and have to replace it. So, make sure that you prepare yourself for such work because you are going to have to do it! (Figure 1)

WHAT TO DO FIRST

When you receive the evidence into your possession you should check several things immediately. First, review the chain of custody form that should be with it. This is a document that contains several valuable pieces of information. First, it should have a detailed description of the evidence that you are receiving down to the serial number. Second, it should have the names, dates and times, of each person who has handled the evidence. (This will become important shortly.) Third, it should have a place on it for other people to sign and date, to show that they have been in possession of the evidence. Digital evidence is fragile and can be easily modified or tampered with and any attorney worth their money will check this custody form to ensure that it is completed properly!

After reviewing the chain of custody move on to an examination of the evidence itself. Taking photographs is an excellent way to document the evidence's physical state. If it is damaged, document it. If it is in excellent shape, document it. Confirm that the power is turned off and that all ports and openings have been closed or covered with evidence tape to prevent tampering. This is often neglected by the person seizing the evidence so you should address it to make sure that nothing has been done to the computer since it was seized. In a case that I worked, the computer, which was a laptop, had been seized at the execution of a

Figure 1. *A good computer tool kit*

search warrant. The detective took the laptop back to his office, turned it on and tried to examine it himself. He encountered difficulties with the file system and subsequently turned to me for help. I didn't ask and he didn't tell what had happened to the laptop since it had been seized. Friend, learn from my mistakes! As I began the file system examination I began seeing time and date stamps on the files that showed they had been accessed AFTER the seizure of the evidence. I also observed that the network card had been accessing a wireless network in the department where I worked. These are bad things to have to explain before a jury. Make sure these questions are asked and answered so that you can conduct a more efficient and thorough examination!

DOWN TO WORK

Once you have photographed the evidence and debriefed the seizing authority, it is time to remove the hard drive. Now, in this article, we will focus on computers that have removable hard drives. In some systems, such as MAC computers, you will find that removal of the hard drive is nearly impossible! In these instances you will have to conduct the acquisition in another manner, again, we will not cover those techniques in this article. So you see, you something to look forward too!

After the hard drive has been removed it is time to record it. Photographing the hard drive's label is easy to do. This label contains much beneficial information for you and it is definitely worth your time to record it. The maker, serial number, logical block address information and size are usually all recorded on the label itself. This information can be used to verify the integrity of the system once you have obtained the forensic copy utilizing FTK Imager or some other imaging software. Remember, you are conducting a forensic examination so verification plays a tremendous role in what you are doing. Knowing that a hard drive is 320 gb in size with a LBA of 625142448 enables you to visually confirm that the entire hard drive was imaged and a host protected area or some other part of the disk was not missed.

Again, verify, verify, verify! Make sure that you verify every step you take, document every action you conduct as you examine the evidence. Whether it is by taking a photograph, writing a note or including it in your written report, you should in some way make it easy for another examiner to repeat the examination exactly as you have done and find the same evidence.

WHAT TO LOOK FOR IN A FORENSIC COMPUTER

Now let us take the hard drive and attach it to our examination machine. Oh, yes, we have to cover this aspect as well. Each and every examiner should have a computer that is designated as his/her examination machine. This should be a computer with a high speed processor, at least 8 gb of RAM, preferably more, and a very large hard drive. What many examiners fail to take into account is that forensic programs tend to utilize a lot of processing power and RAM. Many of the commercial forensic programs such as FTK and EnCase are known to hang and lock up if they do not have sufficient processor capabilities and RAM. This can cause you unnecessary heartache, so make sure that you take this into account when you are purchasing or designing your examination machine.

There are some commercially available forensic computers on the market as well. One of the most popular is known as "FRED" or forensic recovery of evidence device, produced by Digital Intelligence. There are other companies that sell computers specifically designed to conduct forensic examinations so take a little time, do a little research and learn more about them as well. A laptop with enough processing power and RAM can successfully handle most forensic programs. These can be purchased nearly anywhere and you can even specifically design one for your own needs! I would only counsel you to bear in mind your needs as a computer forensic examiner, which are specific and make sure the computer meets them!

Figure 2. *A Tableau write blocker with hard drive attached*

FORENSIC IMAGING

The hard drive, once removed from the computer and documented by you must now be forensically copied. In the field there are multiple terms used for this so in order to avoid confusion I will use the words copy or image. The hard drive needs to be reproduced so that it can be examined and this needs to be done in a way that does not change the original evidence! This is where write blockers come into play in the process.

In order to prevent the evidence hard drive from being written to you must utilize a write blocker of some sort. There are basically two types of write blockers available, software and hardware. Amongst forensic professionals there is a variety of opinion as to which is better and both do have their pros and cons. I prefer to utilize a hardware write blocker from Tableau. These are excellent portable devices that can be configured to match nearly any hard drive you will find. They provide excellent protection against possible contamination of the evidence, but as with all hardware they can break down (Figure 2).

There are many types of software blockers available also. Some you have to purchase in order to use and others are freeware. Two popular bootable CD's available to you are CAINE and DEFT. There was a third, HELIX, but it is now a commercial product and you must purchase the CD from the company. Anyone of these Linux based CD's allow you to utilize dd, dcfldd or a graphical tool to image a hard drive. They usually also allow you to choose which type of image you want to create as well, but more about that in a minute.

Utilizing a Linux bootable CD is a great way to forensically copy a hard drive without removing it! A caveat to this is that you should make sure that the operating system you are imaging is supported by the software or hardware tool you are going to use. This can be a problem and you need to be able to resolve the problem as a forensic examiner! So conducting a little research before hand can go a long way in preventing a snag in the imaging process.

FTK IMAGER AND OTHER IMAGING PRODUCTS

Using FTK Imager to conduct the imaging process is a smart and easy move. With all of the difficulties that can be encountered during the course of a forensic examination you should keep things as simple as possible. Imager is a great GUI tool that provides you with many excellent options when creating a copy of an evidence hard drive. With a GUI tool like FTK the examiner is given the option of pointing and clicking their way through an acquisition, simplifying the process greatly! Now, don't get me wrong, having a good working knowledge of command line processes is a skill that every examiner should have, in case of problems. So, learn the dd and dcfldd commands, or Linux command line so that you have a variety of options at your beck and call.

As I mentioned in the paragraph above FTK Imager gives you numerous options to choose from and it is some of these options we turn to now. When creating the copy of the evidence hard drive you must choose what type of file you are going to save it as. This can be an important decision as you move forward in your analysis of the hard drive image, so we will take a little time here to examine some of the formats available to you.

FILE FORMATS FOR IMAGING STORAGE

It is usually standard practice in the forensic world today to create a file copy of an evidentiary hard drive rather than a bit for bit copy. The bit for bit or disk to disk copy was utilized when moving the forensic copy to a similar hard drive for examination or re-installation purposes. With the advent of virtualization and programs such as LiveView this is no longer a necessary option and most forensic examiners will create a disk to file copy of the evidence hard drive. Basically all this means is that FTK Imager is going to re-image the evidence hard drive to a file that is then stored on your computer or storage device. This file is utilized when forensically examining the evidence. These files can be created in several different formats.

A new format that you can choose from is called the Advanced Forensic Format (AFF). This format is gaining recognition from computer examiners because it is open source and many vendors will be, if they aren't already, including this format in their tools. A second image format is known as the Raw format. This is an older, but very reliable format that can be chosen and it can be used across a wide base of forensic tools, both commercial and free. Raw has a fast rate of transfer, which is defi-

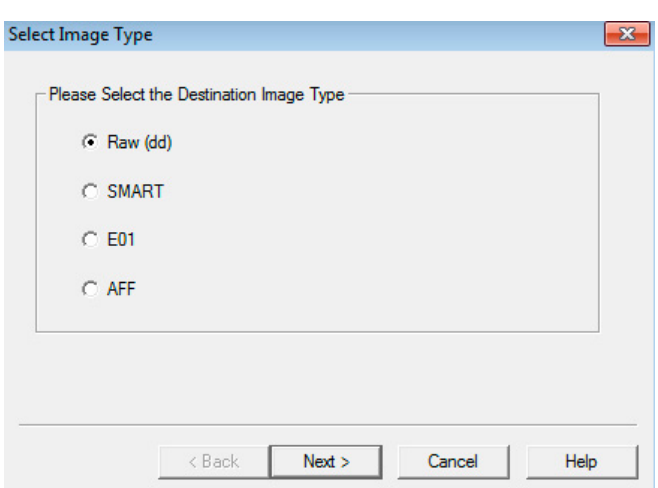


Figure 3. *FTK Imager providing you with various formats for imaging*

nitely a nice choice in a world with multi-terabyte hard drives. A draw back to the Raw format is that the image requires a disk that has as much storage space as the original. For some forensic examiners this can be problematic, especially since hard drives are getting larger and larger! These are some facts that you should keep in mind when deciding what format to create your image in.

There are also some proprietary formats that a forensic examiner can consider. The Encase format, otherwise known as .E01 files, are popular. The drawbacks to the proprietary image formats are that they are proprietary and may not be recognized by some forensic tools. Now, if you use the same forensic tool again and again, such as EnCase, this will not create a problem, but if you utilize several different tools you might encounter a problem, so consider and choose wisely!

One caveat for the .E01 files is that they are verified extensively. Within the image itself are CRC values for each image segment, which verifies that segment internally. A full hash of the entire image file is also created which hashes the entire file, so in essence you have multiple sources of verification, which can come in extremely handy if faced with a verification problem later in the examination and analysis portion of the case.

The FTK Imager provides you with the "select image type" check boxes, showing at least four different image formats to choose from; Raw (dd), SMART, E01 and now Access Data has included AFF as well. So, we can see that FTK Imager provides you with a great range of options when it comes to create a disk to image copy of the evidence hard drive.

STORAGE OF THE FORENSIC IMAGE

Now that you have removed your evidence hard drive, or otherwise prepared it for the digital copying that will now occur, you must have a place to store that copy. There are a number of different methods that you can choose, so consider carefully what choice you make. In fact, many organizations dictate in policies and procedures how the image will be created and how it will be stored. So, this is yet another consideration that a new examiner must be aware of. Do not do anything against your agencies policies and procedures and if you are working in a private capacity I would counsel you to choose a course and then use it every time you handle a case. This is another form of verification and if you encounter problems then you will find problem solving to be much easier than if you are always trying to use new methods.

Storage of forensic images can be problematic. Some agencies provide a new, wiped hard drive for each image to be made. This hard drive becomes the working copy of the evidence. All case files, recovered evidence and reports are also stored on this hard drive, which is then kept as another piece of evidence in the case. This is a valid method with many things to commend it to practice, however consider the growing size of hard drives and cost. You should also consider storage space. If you have a high case load you are going to produce a lot of these storage hard drives and they must be stored some where!

Another method of image storage is to utilize a server. This server is a stand alone server, with connections only to the lab and it serves as a hub for storage of all evidence images, reports and anything that might be attached to the case. This is a very practical method of keeping the evidence, but it is also a high cost method. Some agencies can afford this process whereas others can not. When starting a computer forensics lab, storage space is something that must be considered, lest it creep up on you later and you find yourself scrambling to find an answer.

OTHER CONSIDERATIONS FOR THE PROCESS

Now that you have obtained your evidence, set up your imaging device or process, selected your storage space, it is time to consider the actual analysis or examination. Here, as in the other steps we have looked at in this article, there is a wide range of possibilities for an examiner to choose from. There are many different ways that evidence can be examined so before moving on we need to consider a couple of things first.

What kind of case is it? This is extremely important when considering what type of forensic examination process or program you are going to use. If it is a child pornography case then you will want to use a program that provides excellent viewing capabilities. Since child pornography cases are often image based the forensic examination will also be image specific and your forensic program should be crafted with that end in mind. Or is it instead a credit card fraud case? In this type of case your forensic examination will be conducted by searching the image for credit card numbers, personal identifying information and other related items. So, you will need to have strong scripting capabilities and strong string search facilities ready to use. Or is it a harassment case in which the Internet played a strong role. Your forensic examination software should be able to examine Internet Explorer artifacts, browser history and other such related items. As you can see the forensic program you utilize may heavily depend upon the type of case that you are working and what evidence you are trying to recover.

FORENSIC ANALYSIS PROGRAMS TO CONSIDER

Using several different types of forensic tool is usually the best course to follow since each forensic

Figure 4. *A screen shot of the SIFT Forensic Workstation*

tool has different strengths. Some are excellent when it comes to handling graphics, whereas others are weak. Some can extract and reconstruct email messages so that they look as if they were hot off Yahoo or Hotmail and others struggle to put them into a text file that is readable. Some of these issues you will have to learn as you go, experience is the best teacher! In some cases I utilize both FTK and EnCase. EnCase does a fantastic job rendering graphics in a way that makes it easy to review photographs and video files for evidence. I will then fire up FTK and use it to check for emails and Internet related searches. So you can see that as you gain experience examining digital evidence you will gravitate towards certain tools that you have learned do a certain kind of job well, take this experience and run with it. It will save you time and frustration in the long run.

I mentioned FTK and EnCase, both of which are extremely popular and well built programs. Most forensic examiners are familiar with these products and use them extensively in their cases. A drawback to these products is that they cost! Neither program is cheap I am sorry to say. They run anywhere from $5 to 6,000 dollars per user license. Now some examiners can afford this price tag and if you can, then my suggestion is go for it! You will find these two programs hard to beat in the forensic world. Now if you can't afford that do not give up hope! There are many free tools out there that can do much the same job, if in a different way.

One program that I would highly recommend is free from SANS Institute. It is called the SIFT Forensic Workstation. This program can be downloaded from the SANS web site, after you create

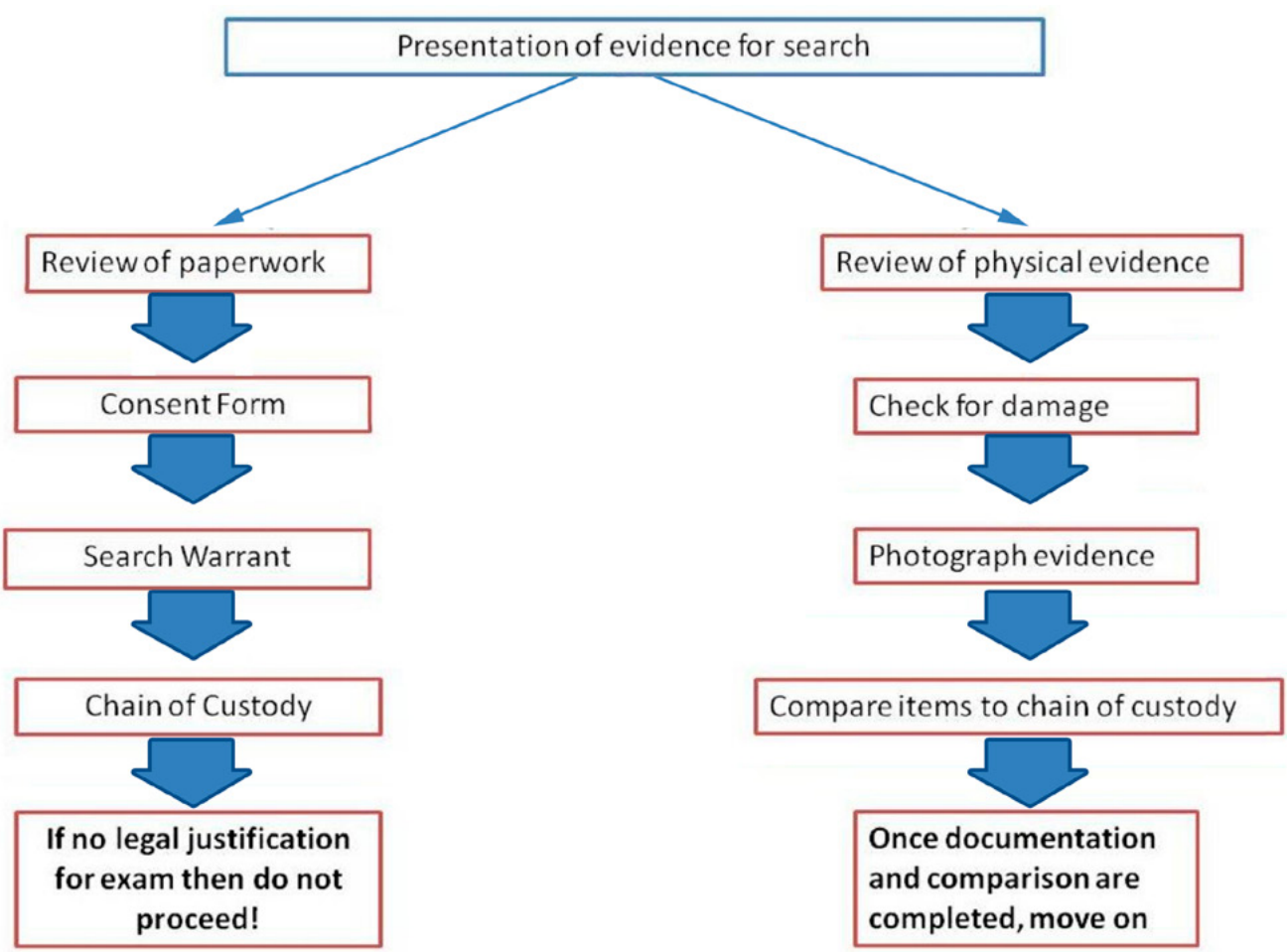


Figure 5. *Flow chart for Beginning Computer Forensic Examinations*

an account, and then run in a virtual machine. It was built with forensic examinations in view and has a multitude of different open source, Linux based tools that can be used to examine nearly any operating system. The only issue that the SIFT workstation brings with it is the need to know the Linux command line. Most of the tools found on the SIFT are Linux based and require the user to be able to navigate the CLI (Figure 4).

CONCLUSION

This was a short review of the forensic imaging process, including many issues that a new forensic examiner should consider. We looked at some tools that you can use, both free and paid, we briefly looked at some different operating systems you should become familiar with and we walked through the initial process of a computer forensic examination. None of this was too technical, which often leads to it being over looked in training! Now take this article as a starting point and run! Digital forensic examination is a growing field, with many new twists and developments arising daily, take the lead! Start your education now, learn all that you can and go on to become what SANS proudly calls a "lethal forensicator!"

About the Author

David Biser is a computer forensic examiner and ethical hacker. He has worked in the field for over 10 years and has obtained the Certified Ethical Hacking and Certified Computer Forensic Examiner certs from EC Council and the IACRB. He has attended training from SANS, the United States Secret Service and the National White Collar Crime Center. David has worked hundreds of computer forensic cases ranging from child pornography to credit card fraud to hacking cases and testified as an expert witness in state court. David enjoys pursuing new techniques in digital forensics and network security and spending time with his family. He is an avid reader and ethical hacker, constantly exploring new ways to help secure networks and investigate network related crimes.

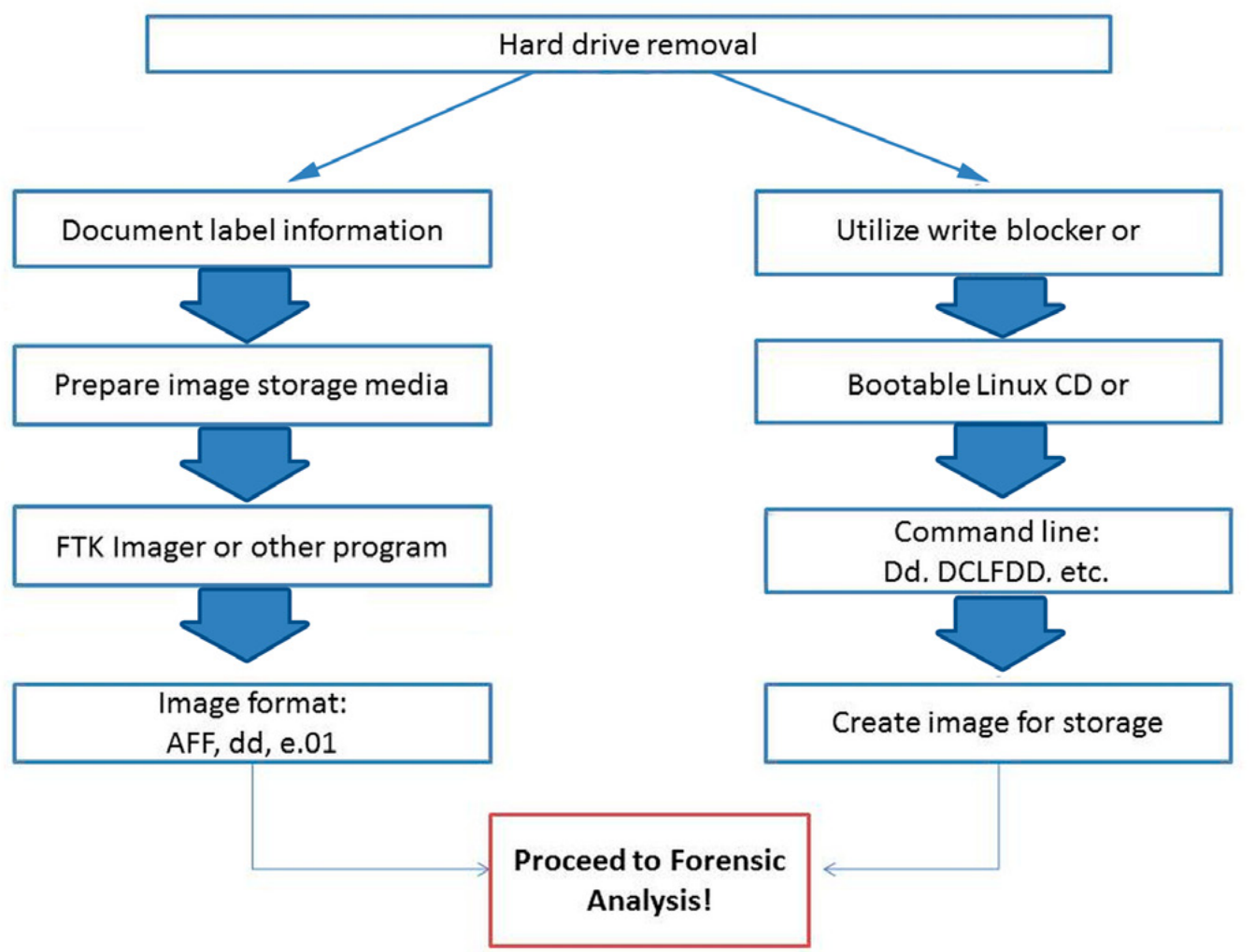


Figure 6. *Flow Chart for Beginning Computer Forensics part II*

DATACENTRE
TRANSFORMATION
CONFERENCE

9 July 2013
aql Conference Auditorium,
Salem Church, Leeds

Excellent Networking Opportunities!

Find your way through the data centre maze

The modern datacentre is transforming, impacted by efficiency demands, new design concepts, legislation and IT innovation, solutions that worked yesterday can soon be rendered uncompetitive or even obsolete.

The Datacentre Transformation Conference sets out to help steer you through the maze of requirements and developments.

Register today to secure your place

www.dtconference.com

Sponsored by:

3M adapt Airedale APPLIED THERMAL INNOVATION
APC by Schneider Electric ARISTA EMERSON Network Power
Gigamon RITTAL Server Technology Quality Rack Power Solutions

In Association with:

datacentrealliance The Data Centre Industry Association
UNIVERSITY OF LEEDS
datacentre solutions DCS EUROPE

AN ANGEL EVENT

WINDOWS REGISTRY FORENSICS 101

by Jason Stradley

This article is meant to serve as a very basic introduction to the Windows Registry and its usefulness as a resource for certain types of forensic investigations. Windows 9x/ME, Windows CE, Windows NT/2000/XP/2003 store configuration data in a data structure called the Registry. The Windows Registry contains lots of information that are of potential evidential value or helpful in aiding forensic examiners on other aspects of forensic analysis. It is a central repository for configuration data that is stored in a hierarchical manner.

What you will learn:

- Windows Registry Structure
- Windows Registry Forensic Analysis skills
- Windows Registry Analysis tools

What you should know:

- Basic Windows Operating System Skills
- Basic entry level forensics skills

System, users, applications and hardware in the Windows Operating System make use of the Registry to store their configuration information and it is constantly accessed for reference during operation. The Registry was introduced to replace most text-based configuration files used in Windows 3.x and MS-DOS, such as .ini files, autoexec.bat and config.sys. Due to the vast amount of information stored in the Windows Registry, it can be an excellent source for potential evidential data. For instance, the Windows Registry contains information on user accounts, typed URLs, network shares, and run command history. While the Windows Registry has many similarities across the various versions of Windows operating system, aspects discussed in this article are based solely on Windows 7 Service Pack 1 and tools that work in that environment.

REGISTRY STRUCTURE

The Windows Registry can be seen as one unified file system by invoking the Registry Editor (accomplished by typing regedit in the run window). The left-hand pane, also referred to as the key pane contains an organized listing in a folder-like structure. The five folder-like structures at the top of the hierarchy are called hives and begin with designation “HKEY” (an abbreviation for Handle to a Key).

Although five hives may be seen, only two are actually real, `HKEY_USERS (HKU)` and `HKEY_LOCAL_MACHINE (HKLM)`. The other three hives are shortcuts or aliases to branches within one of the two hives. Each of these five hives is composed of keys, containing values and subkeys. Values

designate the names of certain items within a key that identify specific values relating to the operating system, or to applications that are dependent on that value.

Figure 1 shows Windows Registry logical view from Register Editor (Windows default registry editor). Each folder in the left key pane is a registry key. The right panes show the keys value. Subkey is used to show the relationship between a key and the keys nested below it. Branch refers to a key and all its subkeys. Windows uses a symbolic link (i.e. similar to file systems shortcut) to link a key to a different path which allows the same key and its values to appear at two different paths. (Russinovich, 1999).

A common way of explaining the structure of the Windows Registry is through comparison to the Windows Explorer file system, given their similarity in structure. The structure of the key pane of the Windows Registry is very similar in nature to the left-hand pane in the Windows Explorer file system.

The keys and subkeys located within the five main hives are similar to the folder structure of Windows Explorer with key values being similar to files within a folder. Using this same analogy a value name in the right-hand pane of the Windows Registry is similar to a files name, its type is comparable to a files extension, and its data is akin to the actual contents of a file.

There are 5 root keys (i.e. starting points) in the Windows registry. Table 1 shows the root keys and the abbreviation normally used to represent each of the 5 root keys.

Table 1. *Root Keys*

Name	Abbreviation
HKEY_CLASSES_ROOT	HKCR
HKEY_CURRENT_USER	HKCU
HKEY_LOCAL_MACHINE	HKLM
HKEY_USERS	HKU
HKEY_CURRENT_CONFIG	HKCC

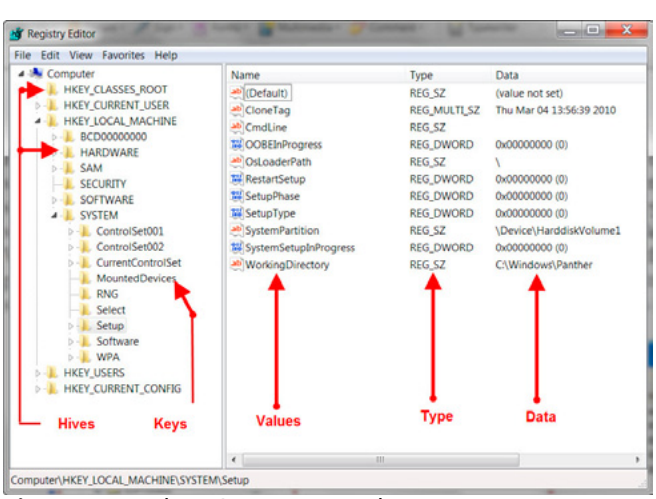


Figure 1. *Windows Registry Logical View*

Below are very basic descriptions for each of the five hierarchical hives listed in Table 1. Beside the root key is their commonly referred to abbreviation in parenthesis, which will frequently be referred to as appropriate throughout the remainder of this article.

HKEY_CLASSES_ROOT (HKCR)

This hive stores information that ensures that the correct program opens when it is executed in Windows Explorer. It also contains additional details on shortcuts, drag-and-drop rules, and information on the user interface. Alias for: `HKLM\Software\Classes`.

HKEY_CURRENT_USER (HKCU)

The HKCU hive contains configuration information for the current user of the system, including user's profile including; folders, screen colors, and Control Panel settings. The alias for a user specific branch in `HKEY_USERS`. Generic information usually applies to all users and is found in `HKU\.DEFAULT`.

HKEY_LOCAL_MACHINE (HKLM)

This hive holds machine hardware-specific information that the operating system runs on. Included in that hardware-specific information is a list of mounted drives and generic configurations of installed hardware and applications.

HKEY_USERS (HKU)

THE HKU hive contains information of all user profiles on the systems, including application configurations, and visual settings.

HKEY_CURRENT_CONFIG (HKCC)

This hive stores information about the systems current configuration. Alias for: `HKLM\Config\profile`.

VALUES

Each key has one or more values. There are 3 parts in value, which are Name, Type and Data, as shown in Table 2.

Table 2. *Value Parts*

Value Parts	Description
Name	Every value has a unique name in that particular key.
Type	Value's type determines the type of data value contains. The common value types in registry for instance are: `REG_BINARY` type contains binary data; `REG_DWORD` type contains double-word (32-bit) data; `REG_SZ` type contains fix-length string data.
Data	Value's data contains data which usually relates to the value's type.

ORGANIZATION OF REGISTRY ROOT KEYS

HKLM and HKU are the only root keys that Windows physically stores on files. HKCU is a symbolic link to subkey in HKU. HKCR and HKCC are symbolic links to subkeys in HKLM. (Honeycutt, 2003).

REGISTRY HIVES

From a forensic analysis perspective, an analyst does not generally interact with the Registry through the Registry Editor. An analyst will most likely interact with Registry hive files directly, through some type of forensic analysis application, or as a result of extracting them from a file system or from an acquired image. There are a number of such tools available, several of which will be discussed later in this article. However, it is important for the analyst to know where these files exist on disk so that they can be retrieved and analyzed. The main, core system Registry hive files (specifically, SAM, Security, Software, Default &System) can be found in the Windows\system32\config directory, as illustrated in Table 3 below.

Table 3. *Registry Hive Files in Windows-System32-Config*

Hive	File Location
HKEY _ LOCAL _ MACHINE \ SYSTEM	\system32\config\system
HKEY _ LOCAL _ MACHINE \ SAM	\system32\config\sam
HKEY _ LOCAL _ MACHINE \ SECURITY	\system32\config\security
HKEY _ LOCAL _ MACHINE \ SOFTWARE	\system32\config\software
HKEY _ USERS.DEFAULT	\system32\config\default

In addition there are some hives that don't have associated files due to their volatility. The system creates and manages these hives entirely in memory. These hives are consequently temporary in nature and are created at every system boot. Some examples of volatile hive are:

```
HKEY_LOCAL_MACHINE \HARDWARE
HKEY_LOCAL_MACHINE \SYSTEM \Clone
```

WINDOWS REGISTRY SLACK SPACE

Slack space is the colloquial term used to describe remnants of user activity, installed applications, etc. that have been deleted and left behind in Registry hive files that are not part of the active hive file itself. The ability to identify these remnants and analyze them is of great forensic significance to an analyst attempting to piece together all information relevant to an incident. Given that the methods and techniques to accomplish this are somewhat advanced, a more detailed examination of this subject is better left to another discussion.

WINDOWS REGISTRY FORENSIC ANALYSIS

Now that we have a basic understanding of the structure and layout of the Windows Registry it is time to start to discuss the meaning of all of this from the perspective of the incident responder and forensic analyst? What it means is that there is a significant amount of information in the Windows Registry that tells the operating system and applications what to do, where to put things, and how to react to certain stimulus.

While the Windows Registry is forensically significant, often it is not captured during the triage of a live system. Similarly, it is often overlooked during post-mortem examinations. On a regular basis, examiners are faced with many challenges: a lack of training to perform triage on a live system; examining multiple hard drives containing terabytes of data; dealing with pressures from management to complete an arbitrary, often unrealistic, quota of examinations per month; constantly juggling and prioritizing overwhelming caseloads; shortages of personnel; and until recently, limited tools for examining Windows Registry files. When faced with these challenges, it is easy to understand why the Windows Registry is not forensically examined to its fullest a great deal more often.

The Registry contains information that Windows constantly references such as the user profiles, the applications installed on the computer, hardware on or attached to the system, application icons, property sheet folder settings, the ports being used, and so on. From a forensic perspective, the Windows Registry is a veritable goldmine that can often provide probative information to an analyst. For instance, some of the information that can be seen in the Windows Registry includes:

- Auto run locations that list applications to automatically run when the computer is booted
- Lists of the most recently used files or applications
- URL's accessed from a system
- All USB storage devices that have been attached to the computer
- Internet Search Assistant
- Printers, Computers and People
- Remote Desktop – Connections
- MSPaint – Recent Files
- Mapped Network Drives -
- Windows Explorer searches
- WordPad – Recent Files
- Excel – Recent Files

ACCESSED URL'S

An example of the type of forensic data that can be extracted from the Windows operating system is provided through the use of Internet Explorer. Internet Explorer is the default web browser in Win-

dows operating systems. It makes extensive use of the Registry widely in the storage of data, like many applications. Internet Explorer stores its data in the `HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer` key. There are three subkeys within the Internet Explorer key that are of the paramount significance to the forensic analyst. The first is `HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\Main`. User's settings for Internet Explorer are stored in this key, and contains information such as search bars, form settings, start pages, etc.

The second and perhaps one of the most informative important subkeys to a forensic analyst is `HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\TypedURLs`. Figure 2 demonstrates the content of what the TypedURLs key shows.

From this data a forensic analyst could the following: (1) That someone may have deleted some entries from this subkey given that the entry numbers skip over some values (url9 – url14); (2) that they like to go to the playboy website, which may or may not be a violation of a given organization's corporate policy; and (3) the user has visited the pastebin site, which houses a lot of questionable things and is typically not allowed by many corporate browsing policies. Other things of interest include the fact that the user may have an eBay account, has accessed several financial institutions from the work system and has a LinkedIn account (Figure 2).

The third subkey that may interest a forensic analyst is `HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\Download`. This key shows the last directory used to store a downloaded file from Internet Explorer, giving the analyst a clue as to the location of where the user stores their files.

COMMON TOOLS & TECHNIQUES FOR WINDOWS REGISTRY ANALYSIS

There are many automated tools both commercial and open source that have the capability to automate the retrieval of registry contents. In addition there are two basic techniques for performing analysis activities. The first is a live analysis performed on systems while they are powered on and potentially still connected to a network. The second is the classic off-line analysis where most commonly in response to some type of incident a first responder has taken the system off the network and taken disk images on which the analysis will be performed.

For the purposes of this article three open source or freeware tool sets will be examined for the ability to automate the collection of Windows Registry information in a live analysis scenario. After the live analysis scenarios have been reviewed another open source tool will show one method for analyzing Windows Registry files that have been retrieved from a system offline on another system.

The tools sets that will be reviewed for the scenarios described include:

- Microsoft PowerShell Scripting – Live Analysis (Free / Open Source)
- Autoruns – Live Analysis (Free / Open Source)
- SysInternals – Live Analysis (Free / Open Source)
- Registry Decoder – Offline Analysis (Free / Open Source)

MICROSOFT POWERSHELL SCRIPTING

Windows PowerShell is Microsoft's task automation framework, consisting of a command-line shell and associated scripting language built on top of .NET Framework. PowerShell provides full access to COM and WMI, enabling administrators to perform administrative tasks on both local and remote Windows systems.

Using the previous example of looking for the TypedURLs a PowerShell command like that shown below will provide a listing of the TypedURLs. After starting PowerShell enter the command string shown below:

```
Get-ItemProperty "HKCU:\SOFTWARE\MICROSOFT\
                  INTERNET EXPLORER\TYPEDURLS"
```

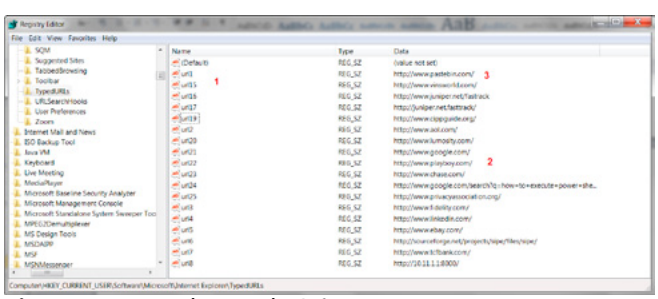


Figure 2. *Example Typed URL's*

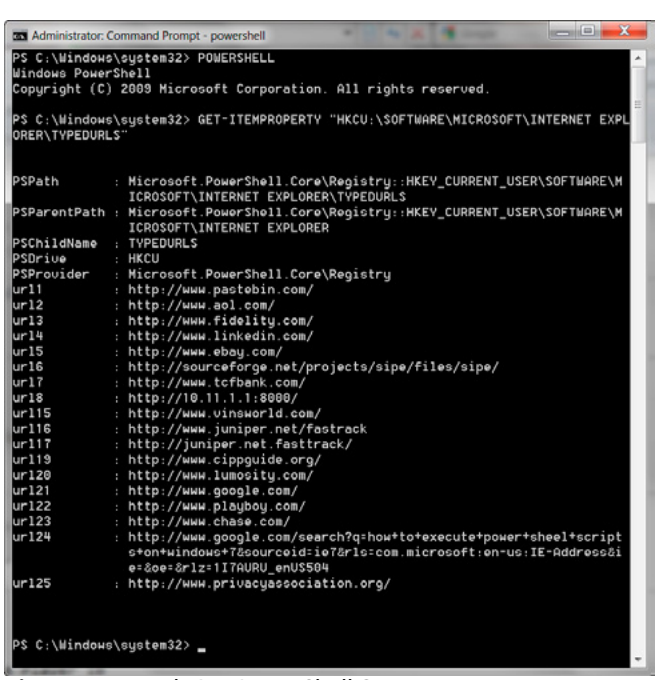


Figure 3. *TypedURLs PowerShell Output*

The output of that command yields a listing of the TypedURLs from that registry key as shown in Figure 3.

The ability to automate the collection of registry data from multiple registry locations using PowerShell scripting is a valuable addition to any incident responder's or analyst's tool kit.

AUTORUNS

Autoruns.exe is an excellent tool written by Mark Russinovich of Microsoft, part of the SysInternals tool set. Autoruns is a great GUI tool that allows you to see a lot of the various locations on a system, where various programs can be run automatically, with little to no user interaction.

Figure 4 shows the Autoruns GUI when the tool is run on Windows 7. The most notable addition to GUI is the available tab named SideBar Gadgets. Figure 4 also shows that there are a number of locations, many of which (albeit not all) are found in the Registry, that allow programs to start automatically, often with no more interaction from the user than booting the system or logging into the system. Autoruns is a very useful tool for troubleshooting systems, as well as for locating malware and suspicious applications, during incident response.

Autoruns comes with a command line companion tool called autorunsc.exe (note the addition of the "c" in the filename), both of which are intended to be run on live systems. Incident responders can include this tool in batch files used for collecting information from systems and gain a considerable amount of insight into what may be happening on the system.

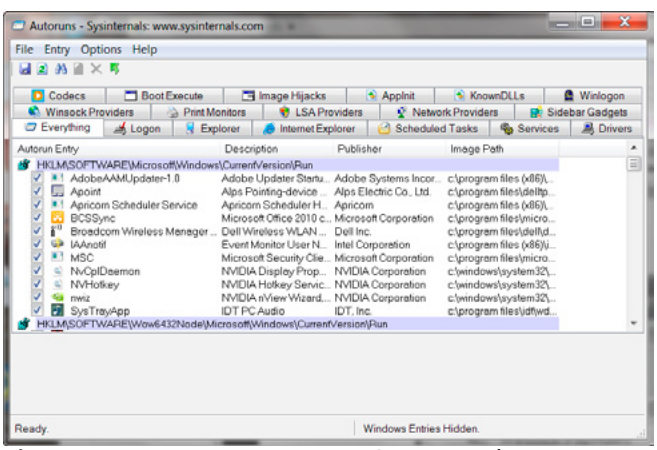


Figure 4. *Autoruns, Version 11.42, GUI on Windows 7*

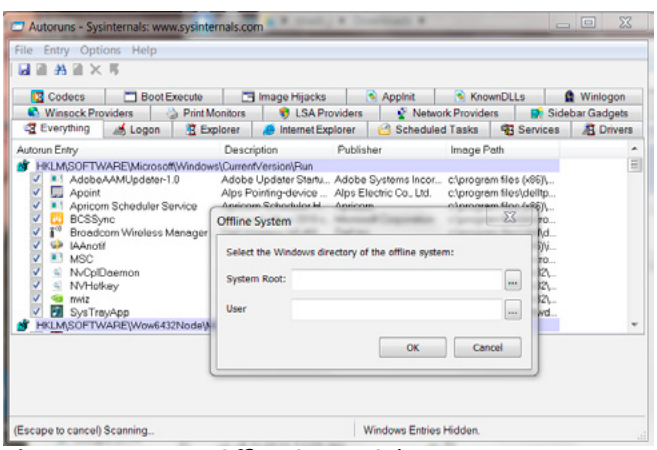


Figure 5. *Autoruns Offline System Selection*

ACCESSING THE WINDOWS REGISTRY REMOTELY

The Autoruns tool can also be deployed remotely by responders using the Psexec.exe (remote command execution tool) also available from Microsoft. As of version 10, Autoruns includes the capability to analyze off-line Registry files; the administrator simply selects the appropriate locations via the "Offline System" dialog box illustrated in Figure 5.

Another method of accessing the Windows Registry of a remote system is a capability that might be very useful for an incident responder or investigator to have. It is possible to access the Windows Registry of a remote system using regedit.exe or reg.exe. After starting the regedit program select the "Connect Network Registry" as shown in Figure 6.

Regedit will prompt the user to enter the name of the remote system using host name or IP Address as shown in Figure 7.

The user will be prompted for credentials to access the remote systems. Once those credentials are entered the Registry for that remote system will appear below the registry of the local system as shown in Figure 8.

The advantage of this method is that it is included in the operating system and is fairly easy to use. The disadvantage is that not all registry infor-

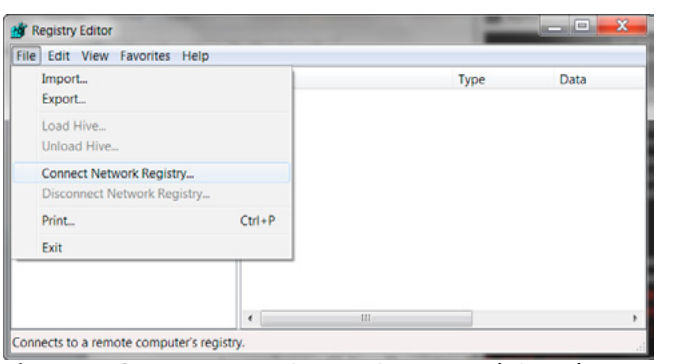


Figure 6. *Connecting to a Remote Registry with Regedit*

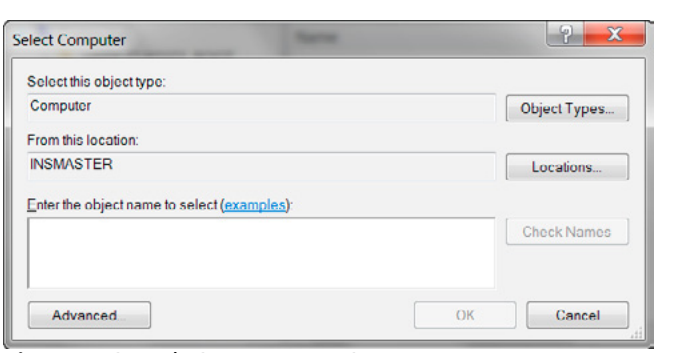


Figure 7. *Regedit Remote Login Prompt*

mation is available. As shown in Figure 8 only the `HKEY_LOCAL_MACHINE` and `HKEY_LOCAL_USERS` keys are directly accessible. The aliases or shortcuts to branches within one of the two hives are not available in the same way as when connecting to the Windows Registry locally.

Other methods to access the Windows Registry of a remote system that permit access to the entire registry include utilizing remote desktop connectivity to run, the Microsoft remote tools framework, PowerShell remote and of course WMI scripting.

REGISTRY DECODER FOR FORENSIC AMALYSIS OF THE WINDOWS REGISTRY

Registry Decoder is an open source project with funding from the *National Institute of Justice* (NIJ) and the *National Institute of Standards & Technology* (NIST). Its purpose is to help automate the acquisition, analysis and reporting of the contents of the Windows Registry. Registry Decoder consists of two components: A live data acquisition tool (Registry Decoder Live); and an offline analysis tool (Registry Decoder).

Obtaining the Windows Registry information using the *Registry Decoder Live* (RDL) tool is a relatively straightforward process. Registry Decoder Live has the ability to obtain Windows Registry data from either the current Windows Registry files on a running system or the backup Windows Registry files.

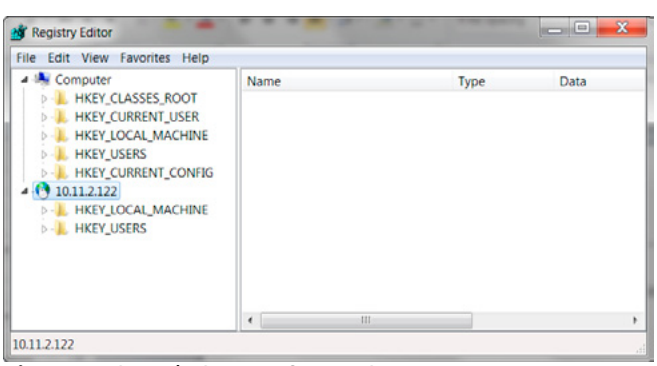


Figure 8. *Regedit Remote System Registry*

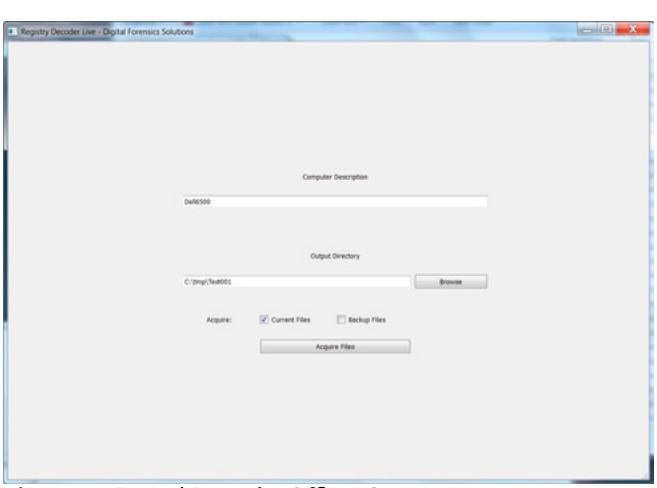


Figure 9. *Digital Recorder Offline Case Initiation*

The acquisition component using the RDP tool is very simple and contains only a single form as shown in Figure 9. Analysts simply need to input a description of the case, select an empty directory in which to copy acquired files, and indicate which registry files should be acquired – the current files, backup files, or both.

Once the acquisition options are chosen, it is just a matter of clicking the "Acquire Files" button and waiting for acquisition to finish. The acquired registry files will be written to the chosen output directory, along with a log file that lists the selected options and acquired files, as well as an SQLite database with information on each file obtained. This directory can then be imported into the offline analysis tool.

Once that data is imported, Registry Decoder can perform an offline analysis of Windows Registry. To begin the offline analysis a new case will need to be initiated using the offline tool. To initiate a new case, just run Registry Decoder and click "Next" on the first form. This will then bring you to the case information form as shown in Figure 10.

The case information form is simple and the only field needed is the directory to which case data will be saved. A case's files include a copy of the registry files analyzed and SQLite databases that store needed information. To proceed, just fill out the form and then click "Create Case".

The next form allows for adding of evidence to the case as shown in Figure 11 on the next page. To add evidence click the "Add Evidence" button and then choose the evidence type that is to be used for the offline analysis. Registry Decoder supports the following formats for this function:

- Databases from the live tool
- Individual (or groups of) registry files
- Raw dd disk images
- Split dd images
- Encase (E01) disk images (not the newest version)
- Encase split images

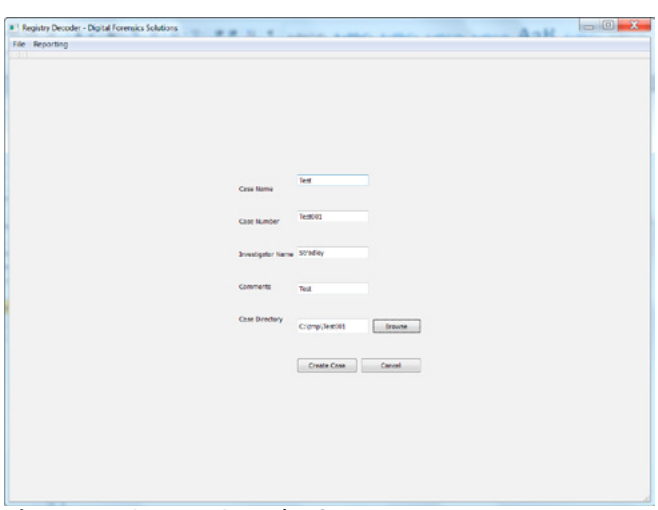


Figure 10. *Registry Decoder Case Initiation*

To give a certain piece of evidence an alias, such as the name of the machine from which a registry file came from or the name of the person to whom the disk image belongs, place it in the Alias field.

The “Add Evidence” has two options for disk images, the two checkboxes (current and backup) determine which files will be acquired. The Current files option includes those that would be active on the running machine:

- Everything under c:\windows\system32\config
- All ntuser.dat files

The Backups option will attempt to gather files from the Reg-Back folder of Windows 7.

When all of the evidence has been added to the case, click on the “Next” button to advance to the processing form. To start processing the evidence files click the “Starting Processing” button as shown in Figure 12 on the next page. The evidence will then process, and when completed, will open the investigation tabs.

Successful processing of the imported Windows Registry data results in a view of the Analysis Tab as shown in Figure 13.

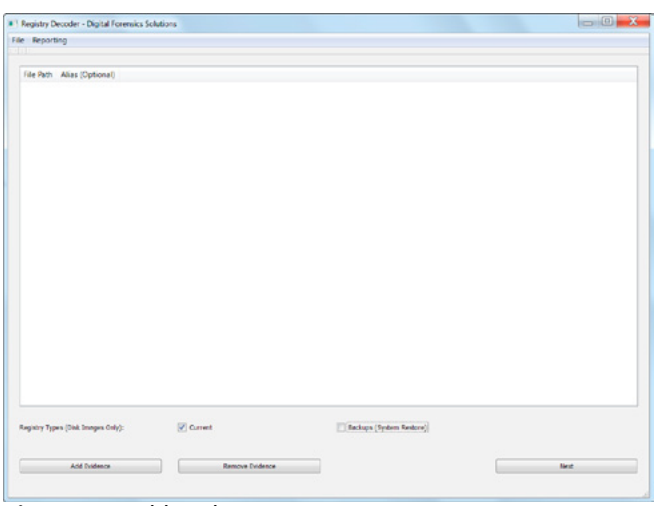


Figure 11. *Add Evidence Form*

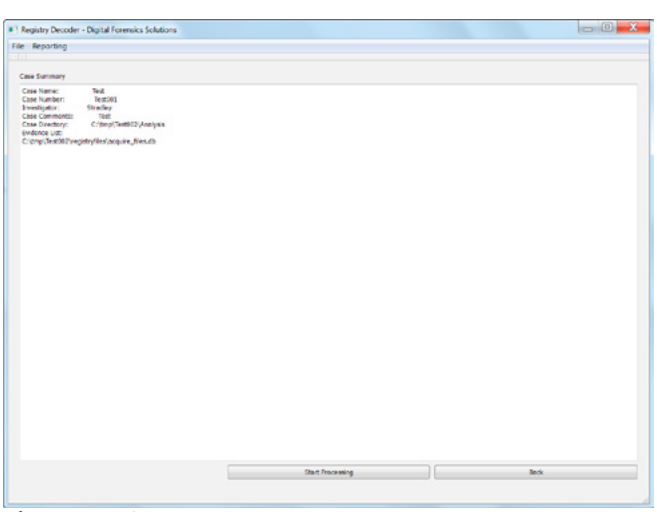


Figure 12. *Processing Form*

There are four general characteristics of the Analysis Tab with regard to how data is presented and its overall navigation.

- Each analysis tab shows a tree view of the loaded evidence in a given case file. Any files chosen for any type of analysis available in Registry Decoder will be selected from this tree structure. This tree structure supports a range of selection options including:
  - Selection of a single file.
  - Selection of a group of files, causing the analysis to execute on each file in the group and can include an entire disk image or groups within a disk image.
  - Selection of multiple files or groups throughout the tree structure.
  - Selection of all files will execute the analysis against each file in the case, which may produce a very large amount of data based on the number of evidence files added to the case.
- All tabs generated during analysis can be safely closed, but the initial set of tabs, as shown in Figure 10, may not be closed. Tabs can also be closely automatically using the CTRL+w shortcut.
- Backups of cases can be made through the “File” menu once a case has been loaded. The backup process will create a ZIP file with the chosen name of all files in a case directory. This directory can be later decompressed and opened on any machine running Registry Decoder.
- A case can be closed at any point in time by choosing “Close Case” from the File menu. Any case can be reopened from the initial form where you have the option to open a new case or create a new one.

Now that you have the evidence loaded into Registry Decoder a number of analyses and reporting tasks may be performed including:

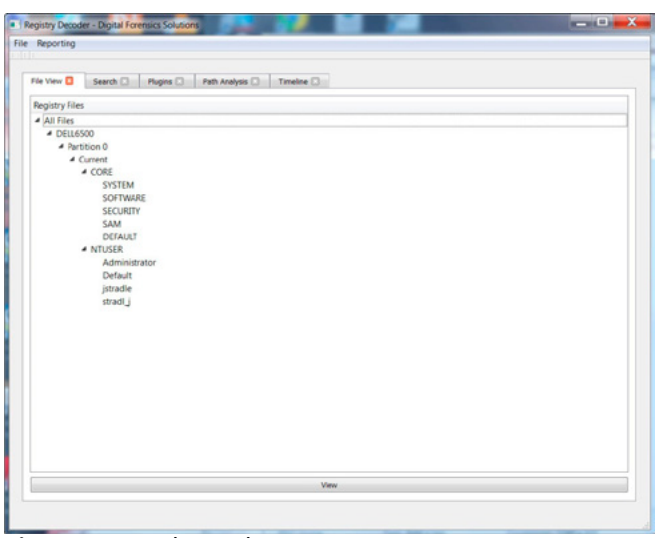


Figure 13. *Analysis Tab*

BIBLIOGRAPHY

- Registry Decoder – Instructions for Offline Analysis Component. (2011, 11 27). Retrieved April 15, 2013, from Registry Decoder: *https://code.google.com/p/registrydecoder/downloads/detail?name=RegistryDecoder-Offline-Analysis-Instructions-v1.1.pdf&can=2&q=*
- Registry Decoder Live – Instructions for Online Acquisition Component. (2012, March 21). Retrieved April 18, 2013, from *http://code.google.com/p/regdecoderlive/downloads/detail?name=RegistryDecoder-Online-Acquisition-Instructions-v1.1_ljp.pdf&can=2&q=*
- Farmer, D. J. (n.d.). A Forensic Analysis Of The Windows Registry. Retrieved April 18, 2013, from Forensic Focus: *http://www.forensicfocus.com/a-forensic-analysis-of-the-windows-registry*
- Honeycutt, J. (. (2003). Microsoft Windows XP Registry Guide. In J. Honeycutt, Microsoft Windows XP Registry Guide. Microsoft Press.
- Russinovich, M. (1999, May). Inside the Registry. Retrieved March 18, 2012, from *www.windowsitpro.com*: *http://www.windowsitpro.com/Articles/Print.cfm?ArticleID=5195*
- Thomassen, J. (2008, April 11). FORENSIC ANALYSIS OF UNALLOCATED SPACE IN WINDOWS REGISTRY HIVE FILES. Retrieved April 16, 2013, from *http://www.google.com/url?sa=t&rct=j&q=&esrc=s&frm=1&source=web&cd=2&cad=rja&ved=0CDsQFjAB&url=http%3A%2F%2Fwww.sentinelchicken.com%2Fdata%2FJolantaThomassenDISSERTATION.pdf&ei=TBF4UbcuiP-tAeHZgIAI&usg=AFQjCNGGzG0hqgnn74NrsOLwBv0yZ4wNLA&sig2=3soTEuMU6TAXB*
- Wong, L. W. (2007, February 1). Forensic Analysis of the Windows Registry. Retrieved April 17, 2013, from FOrensic Focus: *http://www.forensicfocus.com/index.php?name=Content&pid=73&page=1*

- Hive Browsing
  - Similar to regedit and AccessData‟s Registry Viewer®.
- Hive Searching
  - Performs full text searching across keys, values, and names
  - Creates tables of results
  - Provides automated reporting of the search term and matches
- Plugins
  - Similar to RegRipper
  - Registry Decoder currently has 30 plugins
  - Provides automated reporting of plugin results
- Hive Differencing
  - Can show the variances between two registry hives using either search or plugin results as the data source
- Timelining
  - Similar to regtime.pl from Harlan Carvey
- Path-Based Analysis
  - Allows exporting and viewing of paths and their key value pairs. Useful to identify if malware or other specific software pieces or events occurred on a computer
- Reporting
  - Searches and plugins can be individually exported to HTML, PDF, or XLS
  - “Bulk” Exports can be performed for all active analysis results tabs

CONCLUSIONS

Given the huge market share that the Windows operating system enjoys for personal and corporate use now and in the foreseeable future it is important for computer forensic analysts to understand the intricacy of the Windows Registry. The data and potential evidence that exists in the Windows Registry make it an essential forensic resource. The ability to consistently retrieve and analyze this data is crucial to any digital investigation. By understanding the fundaments of the Windows Registry from a forensics standpoint, an analyst can develop greater precision in the accounting of what actions transpired on the given system.

This report should in no way be considered a complete guide to all of the steps and methods required to perform a complete Registry Examination. Such detail will vary based on the type of incident that initiated the examination. It does presents some explanations and examples of what types of data can be found, how it can be found, and why it may be pertinent to an examination. As long as operating systems continue to use the Registry as a configuration database, and applications continue to use that database for storage, there will continually be new locations to discover that provide forensic support in digital investigations.

About the Author

Jason Stradley is a recognized thought leader in the area of information security. Acknowledged as a visionary security executive he combines an entrepreneurial spirit with the ability to execute against his vision. He has worked with many organizations to develop information security solutions to solve business issues in large complex environments. His organizational expertise and thought leadership combined with his strong communication skills allow him to communicate his vision to all levels in an organization, motivating others to succeed. Mr. Stradley is a published author and frequent speaker. Some of his works have been published in CSO Magazine and the Cutter IT Journal. He has presented at venues including EC Council, SANS, MISTI, Gartner, DRJ and others. Jason currently holds the CISSP, CGEIT, CBCP, CISM, SANS GSLC, CBCP, CRISC, CCSK and C|CISO certifications as well as several solution specific certifications. Jason may be contacted at jstrad@aol.com.

REVIEW OF

GUIDE TO COMPUTER FORENSICS AND INVESTIGATIONS

4TH EDITION

by Richard Leitz

Are you interested in computer forensics, or are required to take a class in it for college? The 4th edition of Guide to Computer Forensics and Investigations could be one of the many books you acquire for your growing collection. At 652 pages, it won't be a book that I would consider to be a light read.

As the authors Nelson, Phillips and Steuart state, their book is intended for someone new to the field of computer forensics; however, they should have a background in computer repair and networking. Because computer forensic investigations, have a good chance of being used in a court of law, it is important to know that the authors background and their work, is based on Federal and State laws of the United States. The laws and rules of evidence will vary based on your country.

The book begins with an initial history of computer forensics, and the basics of the United States legal process in chapter 1; while chapter 2 deals with the differences between an internal company investigation, and an investigation that will be tried in a court of law. The chapter also describes proper procedures that should be used in an investigation, to keep your evidence secured from attacks by opposing legal counsel. To help the reader reinforce the information and concepts learned in each chapter, the reader is given review questions, and hands on projects.

Chapter 3 helps you create a real world office and lab, to perform forensics as well as the requirements for certification by America Society of Crime Laboratory Directors (ASCLD). It was nice to know this information, even though as a novice you most likely wouldn't be setting up a lab. The real hands on information regarding the software and hardware that you use to clone a drive, or retrieve data from a live system, is reviewed in chapter 4. Popular forensic programs like EnCase, ProDiscover, FTK, and others are discussed, and used, in the chapter's hands on projects; further use of these software tools are also discussed in chapter 7.

Chapter 5 discusses how to deal with law enforcement in United States, and how to process the scene to avoid the data you recover from being thrown out of a court of law, because of improper procedures being followed. In chapter 6, Windows and DOS based systems are reviewed for file structures and how the Windows registry is set up; and in chapter 8, Macintosh and Linux system specifics are reviewed. I do wonder why

these chapters were split apart by chapter 7 on forensic tools, instead of having it following chapter 4. I would hope in the 5th edition, this lack of cohesion is resolved.

Finding important data will not always be easy, especially if the computer hacker or user is technically savvy. In chapter 9, you have the opportunity to learn techniques being used to hide data through the use of bit shifting to steganography. This chapter also covers processes on how to perform remote access to systems that cannot be shutdown. Chapter 10 expands on the use of steganography to hide critical data in a picture, sound or video file, and the steps you can use to find it.

Today's corporate world uses standalone and virtual servers to store their data, this can make capturing the data more difficult to retrieve. The authors cover some of the basics of virtual environments, and how to perform a live acquisition using Backtrack in chapter 11; while in chapter 12, they review over the processes that you will need to perform for recovering evidence from email servers. Chapter 13 is a short chapter, it discusses the basics of cell phones, and how to retrieve data from a SIM card. I hope in the next version of the book, the authors take the time to expand upon this chapter, because it was lacking in its coverage in forensics for iPhone and Android devices.

The final chapters review the process of writing a report that can be used in a court of law, as well as how to prepare to testify as an expert forensic witness in court. Additionally, they mention a number of professional certification organizations, which can further your investigation in enhancing your skill set in digital forensics.

As is common in many technical books, this edition being published in 2010, is starting to show its age, especially when it comes to some of the chapters; mobile phones and virtual machines, are good examples of chapters that need to be further expanded. Additionally there is no mention of performing forensics in today's cloud environment. So much of our data is being moved to environments like iCloud, Dropbox, Carbonite, or in the corporate world to a Software as a Service (SaaS), Platform as a Service (PaaS), or even the Infrastructure as a Service (IaaS) platforms. It is essential for a forensic investigator to understand how these services work, and the procedures necessary to retrieve critical forensic data from them.

I found this book was a good edition to my collection; but it wouldn't be enough by itself. I have recently read Master Windows Network Forensics and Investigation, which helped fill in many areas that were missing in this book. Even with its expensive price tag, I would still consider it a good choice for your collection; especially if you are new and just starting out in the computer forensics field.

About the Author

Richard Leitz is a security engineer with 20+ years of experience who works for the US Army. He has his Masters in Information Assurance as well as the following industry certifications GCSEC, MCSE, Network+ and A+. He is currently finishing up his MBA and deciding on where to attend for his PhD. rcleitz@gmail.com.

DIGITAL FORENSICS – OVERVIEW OF SEARCH & SEIZURE

by Patrick Ouellette

When we hear people talk about forensics, we typically imagine scenes from Crime Scene Investigation (CSI) or Crime Scene Unit (CSU) shows or movies so popularized in recent years. Although glamorized and using shortened time-frames for processes involved, these shows do adequately represent standard criminal and crime scene investigative and analytical processes.

What you will learn:

- Background information about what is Digital Forensics
- Basic principles and sub-specializations of Digital Forensics (or eForensics)
- The initial processes of Digital Forensics using a fictitious case

What you should know:

- A basic understanding of Windows Event Log
- A basic understanding of Windows Registry
- A basic understanding of Windows Operating System

However, the reality of a digital crime is a much more complex one and involves a much broader spectrum of knowledge and skills related to technologies, non-localized criminal element that may not even be human in nature, and potential theories.

In this article, we will discuss the basic processes of search and seizure as it applies to the investigative portion of a digital forensic case. This is intended to be the first in a series of articles in which the author will explore the different aspects and processes of digital forensics.

A BRIEF OVERVIEW OF FORENSIC SCIENCE

It is important to start by noting that Forensic Science is a very young science, when compared to other branches of modern sciences, with the earliest recorded application of forensics using scientific principles dating back to 1888. According to the "*Handbook of Forensic Pathology*", prepared by the College of American Pathologists, forensic science is defined as:

"The application of physical sciences to law in the search for truth in civil, criminal, and social behavioral matters to the end that injustice shall not be done to any member of the society."

Therefore, we can deduce that the main aim of any forensic investigation is to determine the *evidential value* of the crime scene and the related evidence. In laymen terms, establish the worth of any evidence found as a normal function of the investigative process for a civil, criminal or inves-

tigative case. Forensic scientists are tasked with properly analyzing the physical evidence, answering case-pertinent or legal questions, provide expert testimony in court as required, and furnish training in the proper recognition, collection and preservation of physical evidence

A NEW TECHNOLOGICAL FRONTIER REQUIRES A NEW FORENSIC SCIENCE

Digital Forensics is an even younger branch of the standard forensic sciences, intended to expand the range of forensic sciences into investigating the use of telecommunication and digital technologies to commit a multitude of criminal endeavors. This simply means that it is a budding science, adapting to an ever changing technological and telecommunication landscape over time.

It can encompass the recovery and investigation of potential evidentiary information pieces found in or on various electronic/digital devices, and is often done in conjunction to investigations related to technological crimes, or in conjunction with Incident Response. However, statistics show that more-and-more criminal investigations into crimes of a non-technological nature involve use of technology in some form or another – cell phones, tablets, laptops, e-mails, etc.

The early 1970's saw an increasing frequency of electronic crimes, especially in the financial sector. This forced regulatory agencies, investigators and law enforcement agencies to start exploring technologies as a source and tools of potential criminal activity. Unfortunately, most law enforcement officers lacked expertise in technology to ask the right questions or properly preserve electronic evidence for trial. Thus, out of sheer necessity, Information Technology and Information Sciences experts were recruited to help create the processes, procedures and tools for Digital Forensics.

Digital forensics, according to Steve Hailey of the CyberSecurity Institute (CSI), is defined as:

> *"The preservation, identification, extraction, interpretation, and documentation of evidence, to include the rules of evidence, legal processes, integrity of evidence, factual reporting of the information found, and providing expert opinion in a court of law or other legal and/or administrative proceeding as to what was found."*

Digital forensics therefore involves obtaining and analyzing information of various types from various devices and technologies, using proper evidentiary procedures, each piece of said evidence to be analyzed for the potential of being provided as evidence in civil, criminal, administrative or remedial/preventative action cases.

Digital Forensics has taken many names along the years, including "*Computer Forensics*", "*Digital Forensics*", "*IT Forensics*", "*IS Forensics*", and more recently "*eForensics*".

But if we go back and read the details of the definition of Digital Forensics above, some important key elements do stand out:

- Preservation of evidence
  - No possible evidence is damaged, destroyed, tampered with or compromised by the forensic procedures used to investigate the equipment
- Identification of evidence
  - Must be able to show supporting documentation of how the evidence was located, how and why.
- Extraction of evidence
  - Any extracted and possibly relevant evidence is properly handled and protected from later mechanical or electromagnetic tampering
- Interpretation of evidence
  - *Must provide quantifiable, unbiased, scientifically supported proof that the evidence is probative* (*Probative*: Serving or designed for testing or trial; affording proof or evidence) *to the case at hand.*
- Documentation of evidence
  - Proper documentation of when & how evidence was located, extracted, analyzed and interpreted.
- All above processes must also correctly follow the required *Rules of Evidence*, *Legal Processes*, *Integrity of Evidence,* and *Factual Reporting*
  - *A breach in these requirements may invalidate any evidence uncovered outright, or under review bring into question the validity of the evidence found*.

These days, Digital Forensics or eForensics are the accepted umbrella terms encompassing all of the more specific digital forensic specializations:

- *Computer Forensics*: investigates data that can be retrieved from a computer or other storage media. May also be tasked with recovering data hidden / deleted by users;
- *Mobile Device Forensics*: relating to recovery of digital evidence or data from a mobile device. It differs from Computer forensics in that a mobile device will have an inbuilt communication system (e.g. GSM) and, usually, proprietary storage mechanisms. Investigations usually focus on simple data such as call data and communications (SMS/Email) rather than in-depth recovery of deleted data
- *Network Forensics*: yields information about how a perpetrator or attacker gained access to a network and associated devices;

- *Wireless/Telecommunication/Network Device Forensics*: investigates data retrieved from wireless, telecommunication or network devices and associated storage devices. May help to yield information on how a perpetrator or attacker gained access to said networks.
- *Forensic Data Analysis*: examines structured data with the aim to discover and analyze patterns of fraudulent activities resulting from financial crime.
- *Database Forensics*: relates to the forensic study of databases and their metadata. Investigations can use database contents, log files and in-RAM data to build a timeline or recover relevant information.
- *Data Recovery*: Recovering information deleted by mistake, or during a power surge, server crash or deliberately;
- *Disaster Recovery*: Uses forensic techniques to retrieve information clients have lost due to similar incidents to Data Recovery; There are currently 3 *major* applications of Digital Forensics that use these specializations:
- *Forensic Analysis*, where evidence is recovered to support or oppose a hypothesis currently before a criminal court;
- *eDiscovery,* a form of discovery related to civil litigation;
- *Intrusion Investigation*, a specialized investigation into the nature and extend of an unauthorized intrusion as part of Incident Response within an organization;

THE INHERENT INTRICACIES OF DIGITAL FORENSIC

Forensics science is not a trivial endeavor that just anyone can dally in. As seen in any of the CSI TV shows and movies, the consequences of improper forensic examination can range from improper identification of evidence and loss of credibility of the examiner all the way to losing a court case and letting a criminal go free. This can be said to be a stronger reality when dealing with digital forensics due to the plethora of technologies in use and constantly being developed, the ever growing complexity and variety of operating systems these technologies use, and the global scope of telecommunication capabilities available to anyone.

As well as potentially identifying direct evidence of a crime, digital forensics can also be used to attribute evidence to specific suspects, confirm alibis or statements, determine intent, identify sources (i.e. copyright cases) or authenticate documents. One has to remember that investigations are much broader in scope than other areas of forensic analysis, often involving complex time-lines or hypothesis.

It is therefore critical that the forensic investigator /analyst be mindful of these complexities prior to, during and after the capture of evidence, as well as remember the specific scope of the theory or questions being answered.

THE PROCESS OF DIGITAL FORENSIC ANALYSIS

But how does one actually DO digital forensics, how do we find evidence of an intrusion/crime/event? Obviously, the answer will vary based on the specific technologies involved, whether a single or multiple devices and/or storage media were used, what techniques were used to breach the system(s), etc.

But the basic digital forensic principles of how to look for and obtain these tidbits of potential "evidence" or data containers and then identify whether they are probative to the situation, tend to be the same regardless of the above variations.

One of the harshest aspects of digital forensics is the understanding that any potential perpetrator or attacker must be assumed to be intelligent, have a good understanding the technologies used and/or breached, likely has at least a basic understanding of forensic principles, and is quite likely to want to hide what they have done as best they can. There is also the potential for multiple perpetrators being involved, and multiple devices used directly or indirectly involved in the process of committing the alleged criminal act or digital breach.

Therefore the forensic examiner is left with asking a few key, yet maddening, questions:

- Did I find only what the perpetrator/attacker intended me to find OR is this real potential evidence?
- What did I find, exactly, and what does it mean, specifically, within the specific context of the investigation?
- Having found something and answered the above question, did I actually find everything OR did I miss anything in the process?

Left to his own devices, and without guidance, you can see that the forensic examiner could end up chasing his tail and second guess each-and-every detail found for quite some time. Fortunately, most investigations bring with them a set of pre-determined restrictive questions to be answered and/or hypothesis to be tested, created by the investigators and/or lawyers based on the scope of the alleged crime/breach.

The job of the forensic examiner is therefore to ensure that they have answered these questions to the best of their ability, based on the time/money allocated and the capabilities of the forensic tools available. If anything probative is found beyond the original questions/theory posed, it is then their responsibility to present the potential evidence in order for the investigators & lawyers

to determine whether it is probative to the case at hand or not, and whether the investigation/analysis needs to be expanded/extended based on this new information.

But when you consider that the average data set extracted from a simple server can span megabytes, gigabytes and – in recent years – as far as petabytes, this is a very challenging prospect and tedious process at best. This can be where most of the cost of forensic analysis are incurred – costs for analysis in the order of $1000-$1500 per Gig of data to be analyzed are not uncommon.

So one can quickly see that, as we explained earlier, this can readily lead to information, research and analysis overload, and costs overrun. Therefore, a necessary part of the digital forensic process involves sorting and filtering through all of the potential evidence available to find the smaller subset – also called data set – of evidence *relevant* to the specific case at hand.

From this smaller subset, the forensic analyst can then start to identify any evidence that might be *admissible* in court – initial determination will have to be reviewed by the lawyer(s), presented and argued in court. Ultimately, the final determination of admissibility and validity will be in the hands of the courts (see Figure 1).

Guidance must therefore be obtained prior to any forensic analysis from law enforcement in charge of investigations, lawyer(s) managing the court case or company executives/representatives guiding the internal investigation. This guidance can be in the form of key question(s) that need to be answered, theories that need to be proven/disproven, or specific tidbits of information being sought. The specific type of guidance will depend on the case specifics, results being sought and type of investigation being instigated.

Fortunately, the processes, procedures and practices of digital forensics borrow very heavily from hard lessons learned by the forensic sciences over time, both through trial-and-error during investigations and though prior court cases of a similar nature (*case precedents*).

- The forensic processes and procedures MUST be based on the *scientific method*
  - "*To be termed scientific, a method of inquiry must be based on gathering observable, empirical and measurable evidence subject to specific principles of reasoning. A scientific method consists of the collection of data through observation and experimentation, and the formulation and testing of hypotheses."*
- The forensic examiner MUST follow a code of *professional conduct – a strict code of ethics, confidentiality, morals and standards of behavior* – at all times during the examination.
  - *This is to ensure the examiner maintains unquestionable objectivity and sustains unbiased opinions, while ensuring the investigation retains all required credibility should it need to go to court.*
- The forensic examiner MUST follow the principles of *forensic soundness* at all times during the investigation and examination.
  - *The purpose of a forensically sound process is to support identification and authentication of evidence. This means the evidence*

Figure 1. *Filtering a massive evidence set*

is actually what you claim it to be and has not been altered or substituted since initial collection, accomplished through physical documentation, with time stamp and possibly supporting signatures, of each-and-every examination or investigative step taken along the way.
    - *Although this process starts with the initial identification of potential evidence, it must continue untarnished until the case is solved, processed in court OR dismissed.*
- All fact-finding processes used, potential evidence found, theories tested and how, MUST be clearly documented, along with a time stamp, in a forensic journal.
    - *This is to ensure that, should you need to refer back to your notes to help remind you of previous findings or conclusions OR to answer questions as a witness, you are clear and precise in what was done or found.*
    - *All assumptions, errors in procedures and abnormal findings should also be documented, to maintain credibility of the rest of the evidence found.*
    - *The preference remains written documentation, although audio recordings and electronic documentation have gained some ground in acceptance with the courts.*

So how does the actual process of forensic examination typically work? Assuming that you have been handed a scope (i.e. questions to answer, theories to examine) by law enforcement and/or lawyers, you will need to answer some fundamental questions relating to that scope, using the 5 W's

- Who/What/When/Where/Why …and maybe How:
    - What happened when (sequencing);
    - Who interacted with whom and/or what (linkage);
    - What was the motive/why behind the actions (intent);
    - The origin of a particular item (evaluation of source)
    - Who was responsible (attribution)

In other words, your goal is to reconstruct the crime/event to the best of your abilities to answer the question presented to you as accurately as possible (see Figure 2).

DIGITAL FORENSIC AMALYSIS – GATHERING EVIDENCE

In order to properly reconstruct any crime/breach, we need to gather and obtain information. Said information, depending on how it is obtained and processes, can fall into one of three evidence types:

- Lawfully obtained and admissible
    - This is the crown jewel of any forensic process – evidence that directly answers the question(s) or theory posed, whether it be in a positive or negative way
- Lawfully obtained and inadmissible
    - May still be useful in focusing the investigation and thereby obtain lawfully some other admissible evidence
- Unlawfully obtained and inadmissible
    - This kind of evidence cannot be used in any way or form, as it may attract civil or criminal sanctions and may jeopardize the investigation or any potential prosecution thereafter

Admissibility of the evidence depends on the manner it was obtained, processed/analyzed, the rules for the types of proceedings it is intended for (i.e. criminal, civil of administrative), and, most importantly, the legal right to access the environment/device that may contain evidence.

Most countries have Evidence Acts in place to help establish the rules of evidence gathering, based on the specifics of the situation and/or crime AND the intended outcome:

- Specific rules and requirements for ensuring all evidence is obtained in a lawful manner;
- How to determine/ensure & maintain admissibility of any evidence found; and
- How to document, store, sign-out and preserve any evidence obtained to maintain its admissibility as it is passed between environments, agencies and people (*rules of evidence management*)

The principles of Forensic Soundness, Rules of Evidence, and Evidence Management govern the types of evidence that are admissible, and well as the quality and quantity of evidence necessary to fulfill the burden of proof in a court of law. Failure to follow any of these principles/rules properly may automatically invalidate any evidence, and by association any conclusions obtained from said evidence, in court.

Accidents and/or mistakes do happen – no one's perfect. However, even if the facts are unpleasant and reflect badly on some, such events must be documented truthfully, genuinely and accurately, along with whatever was done – if anything – to rectify the situation. It will then be in the hands of the courts to decide the admissibility of the evidence.

As the evidence is rarely "deposited" right in the law enforcement and/or forensic investigator's lap, the need to go and find this evidence can be assumed. But the question of whether the investigator(s) have the legal "right" to gain access

to where the evidence is may / might be found is yet another hurdle that must be overcome BEFORE the evidence is gathered.

In some situations, the right to access the container that might contain probative evidence may freely be given by the person who it belongs to, typically based on a request from the investigators.

However, in other situations where that right is withheld or not available, it is required that a search warrant – sometimes also called a "*search & seizure warrant*" – is obtained from a local magistrate to ensure that the investigator(s) have the "rights" to gain access to the evidence container (*Evidence Container* – source where evidence was found, be it an actual container or a logical one) and the evidence within by the courts.

The process of and requirements for obtaining a search and seizure warrant will vary based on the jurisdiction(s) involved and laws therein. But it generally involves providing evidence based on direct or hearsay information, provided within the application for said warrant, the purpose of which is to support probable cause. The courts do not take breaking an individual's rights and privacy lightly.

If the potential culprit is already in custody, stopped by law enforcement for an infraction, or caught in the process of a crime, some jurisdictions will allow for a search of the person's premises without the need for a warrant.

In all cases where a warrant is needed, choosing the proper wording and scope of said warrants is critical to ensure that the scope of the warrant isn't to broad YET affords the investigators the required ability to gain access to the appropriate areas/devices/situations they are intending to search for the required evidence.

In some jurisdictions, the ability to seize any potential evidence and/or container may be implied. In others it is required to be included in the warrant OR may be driven by rules of probable cause.

DIGITAL FORENSICS – A FICTITIOUS CASE

Okay, let's use a fictitious example and try and follow it through using forensic processes and the search warrant processes. This example is entirely concocted, has no basis in any real-life event – but it is a very plausible one!

Let's assume that local law enforcement, Constable Ed, comes to you and are asking for your help as a forensic examiner – they've used you in the past as a consultant with success. An incident happened and they need your help to determine a more exact scope and specific details of what happened. They go on to give you the facts of the case, as they know them.

Constable Ed start by stating that the company wants this kept confidential and as quiet as humanly possible, for obvious PR reasons. The company's clients have NOT been notified of the po-

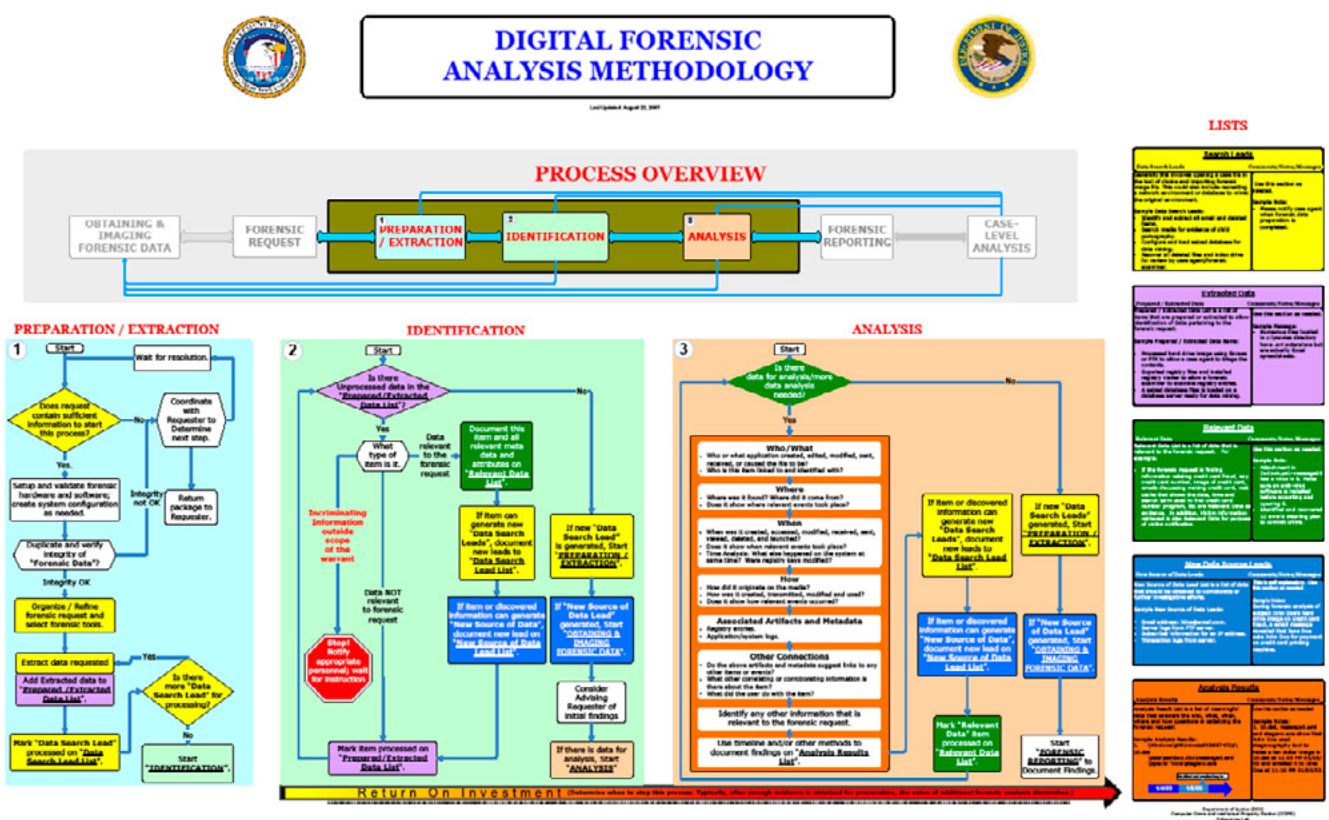


Figure 2. *Digital Forensic Analysis Methodology*

tential breach or the potential impact of the breach as of yet. However, the member of the legal team who is part of the company's Incident Response team, Tom, has been briefed, is involved in the investigation and is the one who called in law enforcement to investigate the case. Tom is calling the shots in this investigation, Constable Ed being there solely as investigators and, if needed, representative of the law.

The case is with the Ajax IT Company, which maintains a large corporate network and Data Center, on which they store confidential information and transactions for a list of Fortune 500 clients.

On Monday, June 13th, 8am, one of the company's log analyzers sent in a report of an atypical administrative access of the corporate gateway on Sunday evening June 12th. Paul, who is in charge of reviewing these daily reports as part of his responsibilities in the IT Support group, read the report. He then investigated the atypical access and found that, on Sunday, June 12th, at 11:30pm, a known administrative password was used to gain access to the network, said password having been recently changed. These passwords are only known to the Security group within the IT Support group, so only a very select group of people could have accesses the gateway at any time – the Security group only consists of 4 members.

No one from the Security group or the IT Support group is scheduled to be on-site at that time, but one member of the Security group is always on e-mail/pager/phone standby rotation, should anything go wrong or the company's security alarms go off.

When Paul checked with the member assigned to standby for June 12th, he stated he neither had a need or reason to access the network at the specified time. When they checked further, no internal alarms had been triggered, and no e-mail/page/phone call had gone out, confirming his statement.

When Paul investigated further, by interviewing the rest of the Security group, no one seemed to be using the gateway at the time. The alibies were then further solidified by looking up the IP of the system that connected to the gateway at the specified time – all of the Security group's home systems are using company paid Internet access with known static IPs for security & tracking purposes.

Having covered all the plausible explanations, Paul triggered the incident response request, which as forwarded to Tom for a response.

After discussing the situation with the Security group, Tom called in law enforcement, and Constable Ed was dispatched. Constable Ed reviewed the information made available by Tom, and decided to interview Paul to see if any critical detail might have been missed or forgotten. Satisfied that this was not the case, Constable Ed next turned his investigation to the Security group.

Constable Ed and Tom interviewed each member of the Security group to confirm the alibies and statements, and to also discover whether someone had inadvertently or carelessly passed on the password. After a discussion of the company policies, consequences of this kind of thing happening, and how they retake ethics training each year, Constable Ed was satisfied that this was a low likelihood.

Spurred by a hunch, Tom checked corporate records and found that, a short time after the password were last changed, one of the IT Support group employees, Marc, was let go from the company after multiple complaints of misuse of the corporate network and alleged hacking for personal gain. The issue was not pursued in court, for PR reasons, but the person was dismissed by HR on the basis of not following internal company policies, of which he was aware and trained on. A severance package was given out to the employee, as per HR policies. As of the termination date, his office computer seized and cleaned, and his account(s) were revoked. Marc was not happy with the incident and was quite verbal about a potential retaliation to anyone who would listen, all the while being escorted from the building by security.

Tom remembers the case, although he wasn't assigned to that particular case. Marc's file is well documented, thanks to good HR procedures, and can be made available to Constable Ed and yourself as needed.

The Security group was then asked to find the IP of the system that gained access to the corporate gateway, which they located rather easily on the gateway's log files. Using a search warrant obtained by Constable Ed, they went back to the ISP the IP belongs to and gained access to the account info using that IP at the time – it was a used called HackerM.

Out of curiosity, you ask the Constable what the perpetrator did once he got access to the corporate network and whether there was any data stolen. He smiles and simply states that the question is not relevant to the case at hand.

Your job is simply to *prove or disprove the hypothesis that Marc gained unauthorized access to the corporate gateway using forensic procedures and concrete evidence*.

Constable Ed then tells you that the company does not currently have concrete proof that Marc actually did the deed, but Tom, the lawyer, and Constable Ed deem he seems a likely suspect to pursue at this time, given the timing of the events and his status – i.e. disgruntled employee.

You then go on to accept to take on the case, confirming that you will act as the forensic examiner / analyst for the client, Ajax IT Company, with Tom as the lawyer and primary representative of the company for this case.

FORENSIC ANALYSIS 0 DISSECTING THE INFORMATION GIVEN

Wow – a lot of information. In fact, you'll find that, with a little bit of digging into the background of the story of the "event", and by using interviews of the people involved directly or indirectly, a fairly large amount of information/data can be readily be obtained about the case.

The forensic task, at this point, is NOT to dismiss or filter information out, but to accumulate & document ALL of the information available. The determination of validity of said information as potential evidence will only happen once further analysis and examination of the data can be done.

You'll notice some vague terminology used – as this is a case that might go to court, it is necessary to ensure that the legal attribution of guilt or responsibility isn't done without proper evidence. Even when just spoken, any improperly used words or assumptions can turn against you later on in court.

So what we know for *certain*, based on the information above (*remember, 5 W's*):

- An atypical administrative access was reported on Sunday, June 12th, at 11:30pm, said access being on the corporate gateway of the Ajax IT Company.
  - No one who has authorized access to the gateway was working at the time OR accessing the gateway remotely.
- At the time of the atypical access, none of the company's Security group, who has authorized access to the gateway, had a reason to use said access. Nor was any of their known static IPs used to gain said access to the gateway.
  - Unless something comes up later to prove otherwise, members of the Security group are not suspects.
  - The same can be said for the IT Support group and other existing employees: there simply isn't any evidence at this stage to suspect any of them, and it isn't likely they would have access to the password to do so.
- The IP that was used to gain access to the corporate router at that time was extracted, traced to the appropriate ISP and, with a warrant in hand, linked to a user account of HackerM.
  - The account info, when extracted, proved to be entirely fake, including the use of a known hacked credit card number.
- We know there is an ex-employee, indicated as disgruntled, who was let go for misuse of the corporate network and allegedly hacking for personal gain using the same corporate network.
  - In other words, he likely knows the network quite well, and seems to have a few skills that would allow him to gain unauthorized access.
- The ex-employee's dismissal is documented as being a short while after the gateway password was last changed.
  - We say allegedly because said activity was never proven in court!
- The ex-employee was let go within a short timeframe of the administrative password for the gateway having been changed
  - As he was an alleged hacker, it is within the realm of possibilities he figured out how to "capture" the new password.
  - And it's quite likely he had a good picture of the corporate network, including the gateway's IP and make/model information.

So we can identify that the hypothesis you are tasked with evaluating is based on:

- *Who*: Marc, disgruntled ex-employee, allegedly having used an administrative password on the corporate gateway to gain access [*potentially*]
- *What*: Unauthorized access to the corporate gateway [*known fact*]
- *When*: Sunday, June 12th, 11:30pm [*known fact*]
- *Where*: Company headquarters corporate gateway remote access port [*known fact*]
- *Why*: Disgruntled employee [*potentially*]
- *How*: Through Internet connection to corporate gateway, using ISP account with fake credentials [*known fact*]

So we have some known facts and some potential evidence/facts/information. Certainly not what a lawyer would consider a clear-cut case by any measure of the law.

But we are left with only two *known* hypothesis that need to be tested, evaluate and prove or refute, and that need to be supported by hard, concrete and legally admissible evidence either way:

- Either Marc gained unauthorized access to the corporate network through the corporate gateway at the date and time indicated above using the ISP account and IP found;
- Or he did not do the above and therefore is not the intended suspect;

Although many a of thoughts/ideas and questions would go through a good examiner's mind at this point, you have to remember the specific theory you were asked to prove or refute – nothing else matters for now.

FORENSIC ANALYSIS – FINDING POTENTIAL EVIDENCE CONTAINERS

The forensic examiner could spend time with the company's corporate gateway, possibly even with

the corporate network and data, all to determine if there is any further information to be extracted that could potentially point to the suspect's actions and intent.

So, at this point, you consult with Constable Ed to determine how best to proceed – Constable Ed has had considerable experience with these cases, so consulting with him makes sense.

In cases like these, law enforcement will generally want to determine whether they have enough evidence obtain a search warrant of the suspect's premises – the requirements will vary based on jurisdictions and situations.

- If a warrant is possible and/or can be obtained, it may simplify the situation.
- Otherwise, law enforcement may need to interview the suspect at their residence, OR bring them in to the local station for questioning, to potentially obtain more evidence to allow for a search warrant.

There's no easy way of determining this aspect of the process, nor is there any way to anticipate how it will turn out – too many variables. But it is primarily a law enforcement aspect of the investigation, and it is the logical next step.

So let's assume Constable Ed and Tom worked together and managed to convince a judge to allow a search & seizure warrant for the suspect's premises. As both Tom and Constable Ed have some previous experience with digital crimes, they are able to ensure that the wording of and instructions contained in the warrant, although necessarily restrictive, are appropriate for the intended need.

Remember, a search warrant is "*a court order authorizing the examination of a place for the purpose of discovering contraband, stolen property, or evidence of guilt to be used in the prosecution of a criminal action* (*http://legal-dictionary.thefreedictionary.com/search+warrant*)." Because of the nature of search warrants, the courts ensure that they are specific and restrictive, based on the scope and strength of the evidence presented. No one wants to just outright dismiss an individual's rights to privacy – people are, after all, innocent until proven guilty.

In this case, the search warrant is aimed only at examining and/or seizing computing device(s), networking equipment, any electronic storage devices and mobile phone(s) found within the suspect's primary residence, vehicle(s) or on his person. The warrant allows law enforcement to determine if potential evidence containers should be seized as potential evidence for further forensic analysis OR can safely be forensically examined on-site to determine whether they contain probative information and then seized as evidence.

With this warrant in hand, Constable Ed and yourself head for Marc's last known residence.

FORENSIC ANALYSIS – LOCATING POTENTIAL EVIDENCE SOURCES

From this point on, you need to be taking notes about EVERYTHING that happens, that you see/touch/examine, and every action you take as you go along. You can take notes of or record what is

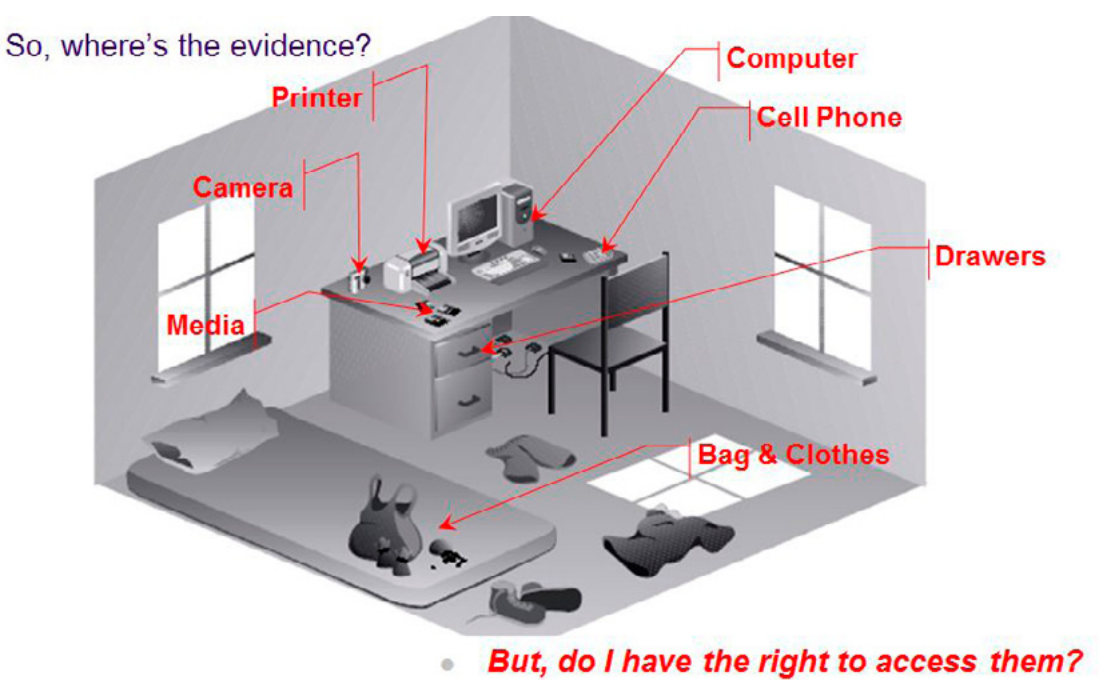


Figure 3. *The Crime Scene*

said at the scene – it is likely that law enforcement will also be doing this, but having a backup is never a bad thing.

Another very important part of the initial forensic discovery and documentation is taking pictures of everything found, in the original state it was found, BEFORE anyone touches or moves it (Figure 3).

Once you get to the suspect's residence, he is a little hesitant to open the door and only wants to talk through the crack of the door. Once Constable Ed explains shows him the warrant, explains the situation and the requirement for the suspect to allow both of you in. Marc seems started, a little hesitant, but in the face of Constable Ed 's practices self-assured manner, he allows you in.

Constable Ed quickly informs the suspect to remain calm & quiet, that he is not to touch anything and is to remain standing where he is, unless he is called upon by you or himself. Constable Ed confirms that the suspect understands the instructions. Next, you and Constable Ed examine the apartment. It is quite small, contains 2 rooms (*one bathroom, one living/bedroom)*, sparsely furnished, and looks like it has been well used. The only furniture in the room are a desk with 2 drawers, a chair, and a mattress on the floor (see Figure 3).

Marc is quick to state that this hovel of an apartment is all he can now afford, since the company unjustly laid him off. Constable Ed reminds the suspect of the instructions, and Marc shuts up but seems agitated.

At first glance, the room has a few obvious containers/devices:

- A computer on the desk, likely with some form of Internet connection
- A cell phone on the desk
- Some storage media visible sitting on the desk

Any one of these individual items could, in itself, contain information relevant to prove, or refute, the hypothesis you are investigating. But it is also possible that more than one may have probative information on it – not unusual to have data on multiple devices.

However, there are some less obvious choices we can't ignore:

- The digital camera
  - Most digital cameras are used to store images, and there might be something in those that could be probative in some way
  - But digital camera can also contain files, small applications, etc. The same storage media used for the pictures can be readily formatted to contain digital files and applications
- The printer
  - Modern printers almost always have storage available for processing large printing jobs, typically RAM.
  - However, some of these also use Solid-State Drives (SSD) or Hard Drives, and can therefore also be formatted and configured to be used as storage

We also have to consider potential hidden sources of further evidence.

- The desk has drawers
  - The search warrant allows for examining the premises to locate any probative devices listed in the warrant.
  - A visual search of the drawers uncovers a USB storage device, but nothing else of importance.
- There is a bag of clothes on the bed
  - Again, the search warrant allows for examining the premises, so this is allowed.
  - A search of the bag unveils clothes and a wallet.
  - Nothing further or probative is found in the bad OR the wallet.
- The bed itself, when inspected, proves to contain nothing of importance.

FORENSIC ANALYSIS – EXAMINING SOURCES FOR EVIDENCE

Now that we've located the potential sources of evidence, the next step is to determine how to proceed with these sources. This is critical as a wrong decision here can potentially alter, damage or even destroy potentially probative evidence – so everyone needs to proceed with great care and thoughtfulness.

However, the driving force behind the decisions is based on how to proceed to "seize" & collect the evidence for later analysis.

SoFor the sake of brevity, let's concentrate on the basics of the obvious device – the computer.

First, determine if the computer is turned on or not. This must be done without altering anything internal or external to the device. So look for key signs that it is on WITHOUT changing anything, if at all possible.

- If the computer is currently turned on, it is IMPERATIVE that a live forensic capture of the computer be done BEFORE it is turned off, moved or modified in any way.
  - There is a large amount of information contained in live RAM when a computer I running that will be lost entirely if turned off.
  - The operating system has a multitude of files that will either be lost OR modified by turning the computer off.

    - As crazy as it may sound, even taking a computer out of sleep or screen saver mode will modify information in RAM and on the storage.
    - All of the above information may be probative and must be captured, using live forensic capture techniques – the specifics of which are not a part of the scope of this article.
        - *N.B.: This is one of the areas where mistakes are routinely made – turning off a live computing device CHANGES the state of the information it contained.*
- If the computer is turned off, it is safe to pack up for further forensic evidence.
    - Again, turning it on will modify operating system files on the internal storage, potentially altering or destroying evidence.
    - At worst, the internal storage should be removed as evidence, to be forensically duplicated and analyzed in a controlled environment.
- The peripherals (keyboard, mouse, monitor, etc.) are not necessary, as they will not likely contain any probative information. However, documenting the key information about each of them, along with a photograph of their connection and setup, is important.

An important part of the process of evidence seizure is to use proper evidence gathering, identification, documentation and storage/protection processes – the specifics of which are beyond the scope of this article. This process is quite similar to standard law enforcement evidence gathering, seizure and management, and in fact uses the same principles with slightly modified tools and equipment geared to digital technologies.

A similar process of examination, determination and potential seizure is then made for each of the potential evidence containers found within the suspect's residence, as per the scope of the warrant.

Finally, having gathered all of the evidence containers, said containers are then brought to an appropriate evidence locker, where they are received, documented and labelled, signed in, and appropriately stored, waiting to be signed out for forensic analysis to determine if any probative information is contained within them.

WHAT'S NEXT

In the next article, we will look at the basics and complexities of evidence management, from finding the evidence all the way to presenting it in court.

SUMMARY

As you can see, the basic processes of digital forensic are very procedural, stringent and sometimes downright tedious compared to what we are presented in CSI TV shows and movies.

Digital Forensics requires a person with a scientific mind, a rigorous attention to procedures and details, a willingness to remain unbiased and focused, and an ability to keep well-ordered and clear notes and documentation. It requires years of professional training and field experience, and a willingness to spend hours upon tedious hours to filter through sometimes massive amounts of information … all to obtain a few small tidbits of information that could be the answer to a question or hypothesis.

This article is intended to give the readers a glimpse into the world of Digital Forensics. If the reader is interested in finding out more about the processes and specific tools, here are some resources to help guide them into this direction:

- *Handbook of Forensic Investigation*, by Eoghan Casey, Academic Press, ISBN 0123742674
- Evidence and Investigation – from the Crime Scene to the Courtroom, by Watkins/Anderson/Rondinelly, EMP
- *Computer Forensic series*, by EC-Council, EC-Council Press/Cengage Course Technologies (5 book series)
- *Guide to Computer Forensics and Investigations*, by Nelson/Phillips/Steuart, Cengage/Course Technologies, ISBN 1435428196
- *Incident Response & Computer Forensics*, 2nd Ed, by Mandia/Prosise/Pepe, Osborne, ISBN 007222696X
- *Scene of the CyberCrime*, 2nd Ed, by Shinder/Cross, Syngress, ISBN 1597492760

About the Author

The author has been working and experimenting with various digital technologies for the last 30+ years, and has also been doing consulting/corporate training in for 25+ years in areas of wireless/network/security. Currently, he is the Program Coordinator for Security programs in the Information Communications & Technology department, and a professor in the School of Advanced Technology at Algonquin College in Ottawa, Canada.

What do all these have in common?

They all use Nipper Studio

to audit their firewalls, switches & routers

Nipper Studio is an award winning configuration auditing tool which analyses vulnerabilities and security weaknesses. You can use our point and click interface or automate using scripts. Reports show:

1) Severity of the Threat & Ease of Resolution

2) Configuration Change Tracking & Analysis

3) Potential Solutions including Command Line Fixes to resolve the Issue

Nipper Studio doesn't produce any network traffic, doesn't need to interact directly with devices and can be used in secure environments.

SME pricing from £650 scaling to enterprise level

evaluate for free at www.titania.com

2012 Computing Security Awards
WINNER
Enterprise Security Solution of the Year

2012 Computing Security Awards
WINNER
Network Security Solution of the Year

2012 Computing Security Awards
Runner-up
SME Security Solution of the Year

titania nipper STUDIO

CLOUD

www.titania.com
T: +44 (0) 1905 888785

A PRACTICAL APPROACH TO TIMELINING

by Davide Barbato

When conducting forensics analysis it can be useful to have an understanding of the events that occurred on the system to correlate your analysis and gain a better focus on your investigation.

What you will learn:

- A basic understanding of NTFS structure
- What is a timeline and how to create one

What you should know:

- A basic understanding of Windows Event Log
- A basic understanding of Windows Registry
- A basic understanding of Windows Operating System

Sometimes we need to investigate a data breach, an identity thief, a program execution or, in a more general way, we need to know what happened on a system on a specific time: to accomplish that, we need to create a timeline of the system activities so we can add context data to our investigation.

As it is, timelining refers to the technique used to keep tracks of changes occurred in an operating system by creating a timeline of activities, pulled down from various data sources, as the investigation process requires.

DATA SOURCES

To create a timeline, we have to choose which information to retrieve and which kind of data we need to look at. On a Windows system, we have a lots of data sources: system events, prefetching, registry keys, web history and even file system changes. There are two basic approach to timelining:

- create one big timeline and then filter the data you might be interested in,
- create a timeline for each data source, filter the data of interest, and then merge into one timeline.

When creating a big timeline it can be difficult to search through the data, but it can give much more insight about what happened and when; furthermore, in my opinion it is easier to filter one big thing than filtering small piece of files and then put them all together.

However, it depends on how comfortable you are with, but in this article we use the first approach, the big timeline creation.

As said, there are lots of data sources to look at when creating timeline, and the choice to include one instead of another depends on the investigation requirements. We are presenting just a small sets of data sources, so you can create an high view of your system activities.

MFT TIMESTAMPS

First of all, let's talk about file systems: in this article we assume that we are dealing with NTFS file system because it's more interesting and common than the old fascinating FAT file system.

Each file system keep tracks of object changes, which it gives a timestamp to mark the temporal activities for the object that was involved.

NTFS uses four different timestamps to track temporal activities:

- Modified, when the object is updated and/or modified,
- Accessed, when the object is read,
- Changed, when the object metadata are updated and/or modified,
- Created/Birth, when the object is created.

Table 1. *$STANDARD_INFORMATION and $FILE_NAME timestamps changes*

$FILE_NAME	Rename	Local Move	Volume Move	Copy	Access	Modify	Create	Delete
Modification		X	X	X			X	X
Accessed			X	X			x	
Change (meta)		X	X	X			X	X
Born			X	X			X	

$STANDARD_INIFO	Rename	Local Move	Volume Move	Copy	Access	Modify	Create	Delete
Modification						X	X	
Accessed			X	X	X	X	X	
Change (meta)	X	X	X	X			X	X
Born				X			X	

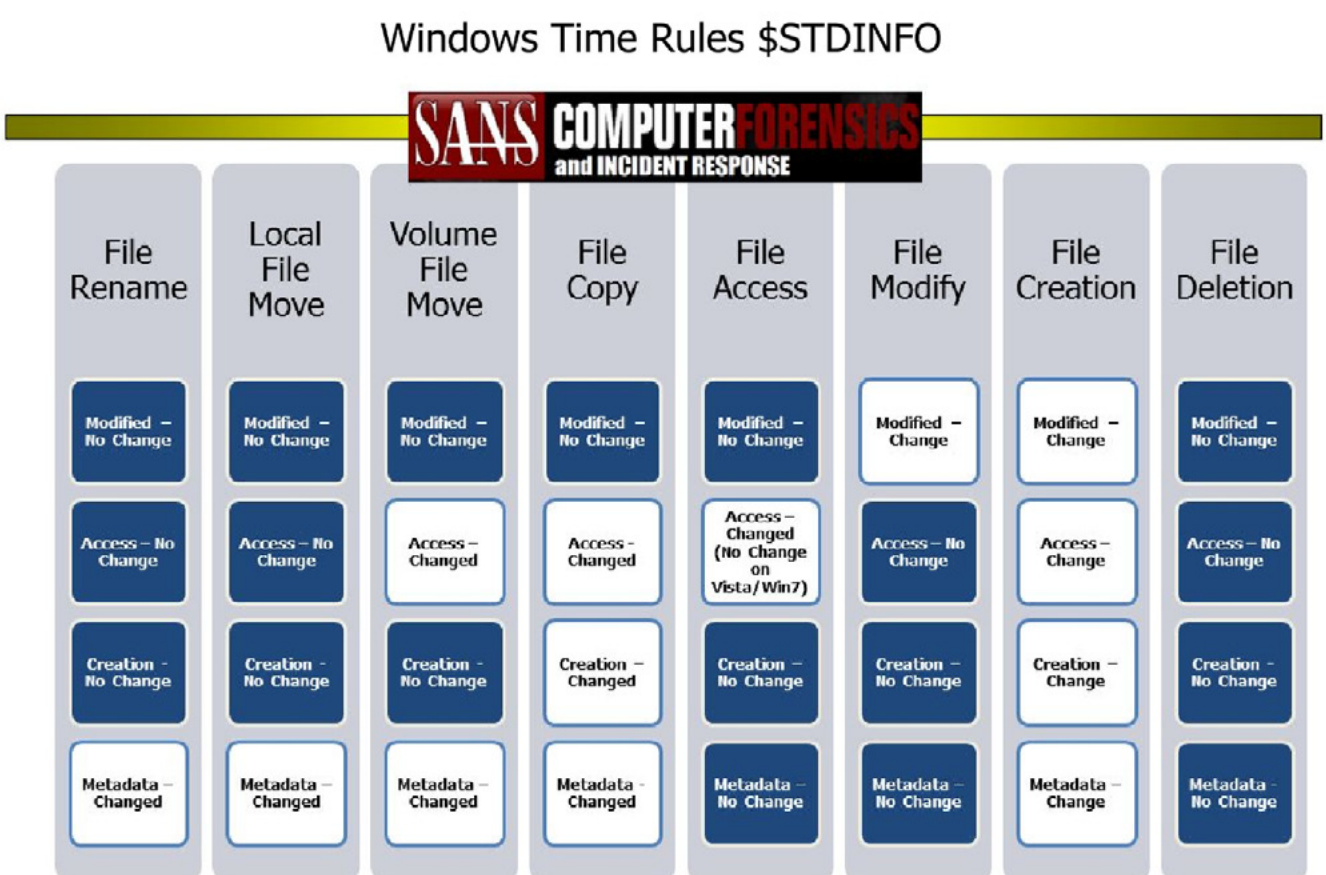


Figure 1. *$STANDARD_INFORMATION timestamps changes in Windows 7*

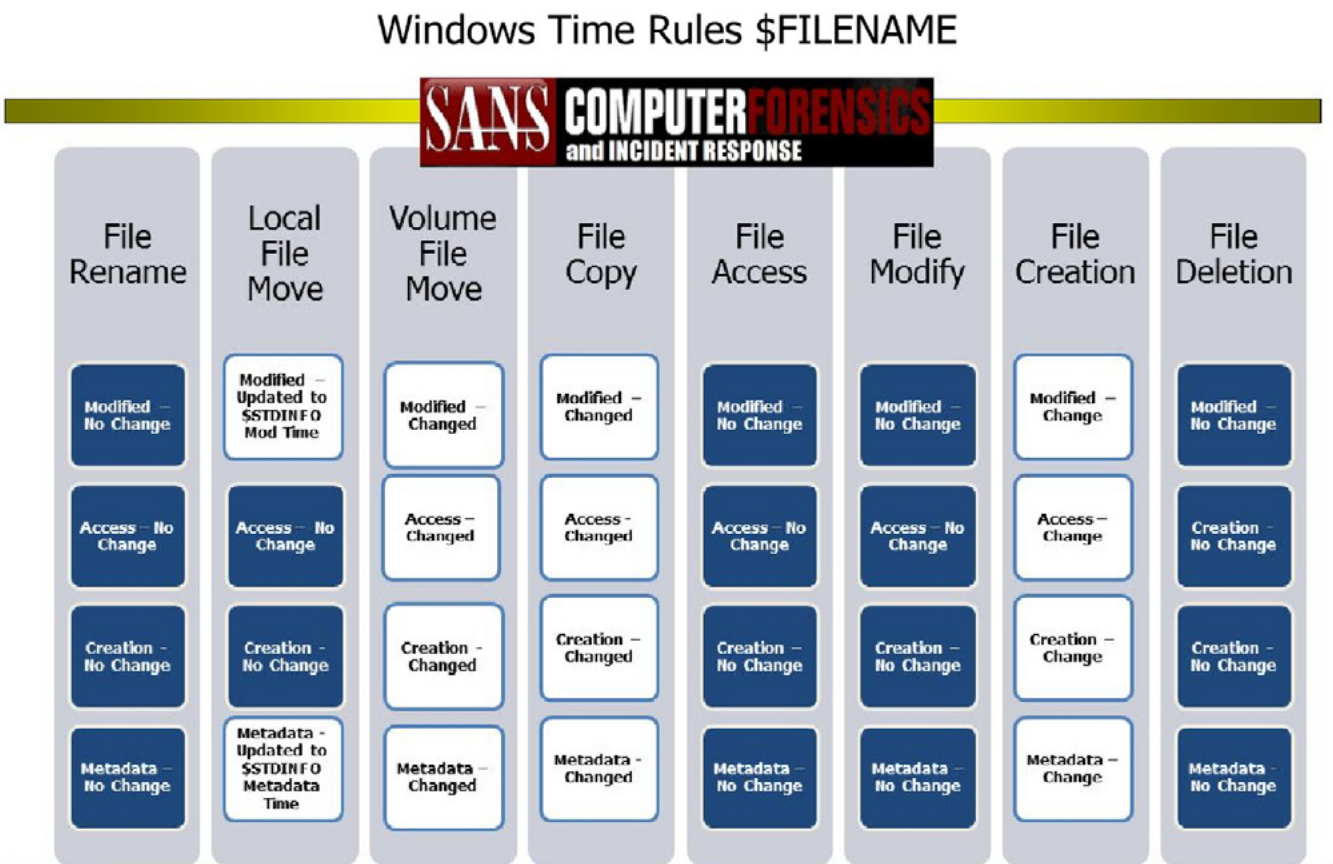


Figure 2. *$FILE_NAME timestamps changes in Windows 7*

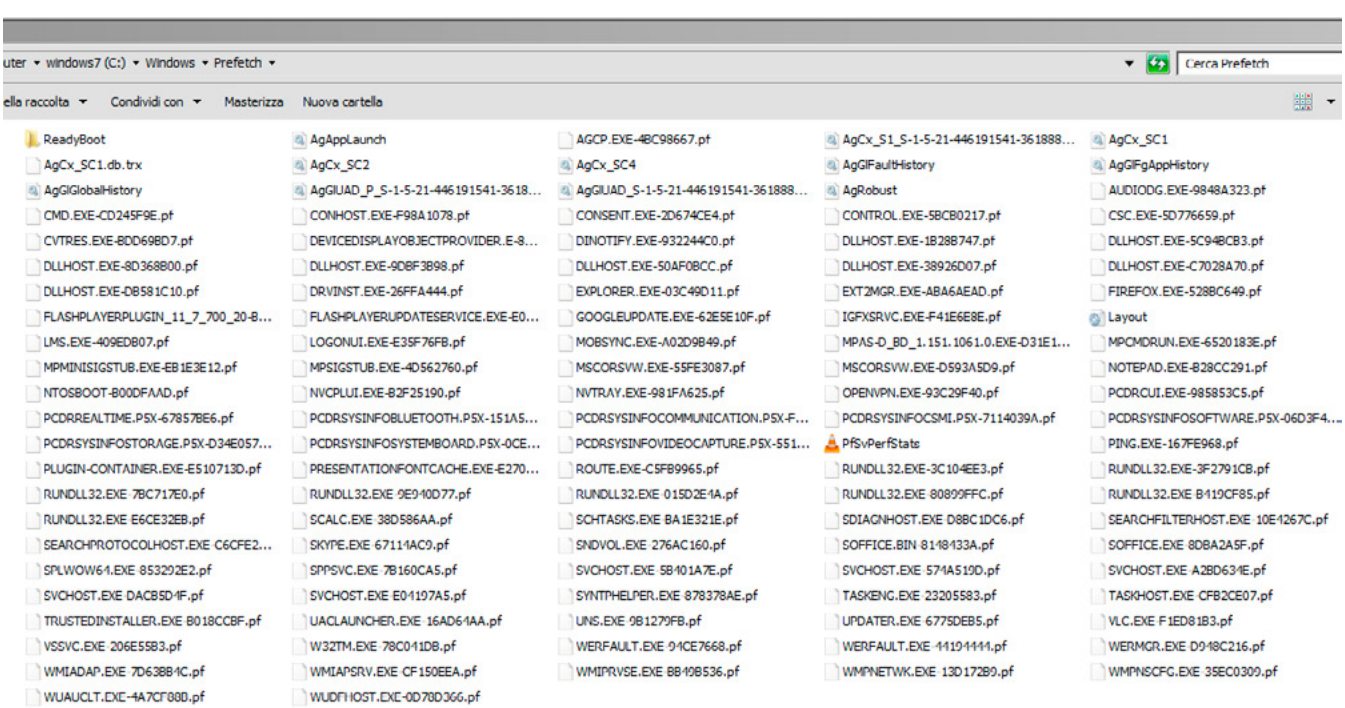


Figure 3. *Prefetch directory*

The above timestamps are grouped into the word *MACB*.

NTFS stores information in the MFT, the Master File Table: each entry in that metafile is an object and can be a file, a directory or another metafile.

Each object in an NTFS file system has two attributes: `$STANDARD_INFORMATION ($SI)` and `$FILE_NAME ($FN)`.

Both attributes stores the four timestamps listed above: modify, access, change, create (plus other information that we actually don't need).

The difference between `$SI` and `$FN` is that the values in `$SI` are updated frequently, as the user uses the GUI, because it is updated by the Windows API in user space, instead of `$FN`, that quite often reflect the object real timestamps, as it needs kernel space access.

Looking at Table 1, we can see which timestamps of which attributes change based on the action we take on Windows system prior to Windows 7.

And these are two cheat sheet from SANS that address the `$SI` and `$FN` changes in Windows 7 (Figure 1 and Figure 2).

Why show all that information? Because the timeline is a matter of time: all of the timeline is built and sorted around timestamps, to create a chronological view of events and activities, thus we can have a walk through of what was going on.

Before we start our tests, we need to keep in mind an important aspect of timelining, the "*temporal proximity*": it means that we can being close to an event in time, but we cannot get all the historical timestamps changes, so we can have a kind of snapshot of data states. Take as example the *LastWrite* object of Registry keys: it can hold only the last time it was written, not all the time it was.

PREFETCH FILES

To speed up performance, Windows has enabled by default "prefetching": briefly, when you run an executable, Windows records all the loaded modules, files and dependencies to a file called "prefetch", under the directory *C:\Windows\Prefetch*, and it holds only the last 127 software calls.

The file has .pf extension and has a binary format, so you need a specific tool to parse them: when done, you can read not only the loaded modules but also the run count (how many times the software ran) and the last run time. You can deduce, due to the file creation timestamp, the first run.

Here is a screenshot of the Prefetch directory (Figure 3).

As you can see, the file name is composed by: "NAME.EXT-HASH.pf", where "HASH" is an alphanumeric string based on the name of directory which the executable belongs.

Parsing and analyzing the prefetch files can tell you what software the system started, when and how many times, so you can have an overview of the system software runs.

If you cannot find any prefetch file, there are two possibilities:

- Windows is installed on an SSD drive, then Windows automatically disables prefetching,
- the registry key `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Memory Management\PrefetchParameters\EnablePrefetcher` is set to 0.

WINDOWS REGISTRY HIVE: NTUSER.DAT

A lots of user activities done through the GUI are recorded into the file *NTUSER.dat*, found under the user home directory.

Among the information the file records, there are a small set that can be used to retrieve user activities such as recent open files, opened windows and so on.

The type of information pulled down by *log2timeline* that you might be interested in is, but not limited to, as follows:

- *RecentDocs*, which hold the files recently opened,
- *FileExts*, which tell you what software opened a file,
- *MountPoint*, which shows you when a removable device was mounted (Figure 4).

20/02/13 23:11:39	UTC	MACB	REG	RecentDocs key	File opened	dab	DAB-PC	Recently opened file of extension: .NEF - value: DSC_1029.NEF
20/02/13 23:11:39	UTC	MACB	REG	FileExts key	Extension Changed	dab	DAB-PC	File extension .NEF opened by WLXPhotoGallery.exe
09/03/13 12:37:00	UTC	MACB	REG	NTUSER key	Last Written	dab	DAB-PC	Software/Microsoft/Windows/CurrentVersion/ActionCenter/Checks/{A5268B8E-7DB5-465b-BAB7-BDCDA39A394A}.check.100
10/03/13 21:09:17	UTC	MACB	REG	NTUSER key	Last Written	dab	DAB-PC	Software/Microsoft/WindowsPhotoViewer/Viewer
18/03/13 22:55:48	UTC	MACB	REG	MountPoints2 key	Drive last mounted	dab	DAB-PC	{1ddb5b65-95c0-11e1-9885-001bb1fffba4} volume mounted
26/03/13 17:26:09	UTC	MACB	REG	RecentDocs key	File opened	dab	DAB-PC	Recently opened file of extension: .pdf - value: storia-versi.pdf
26/03/13 17:26:15	UTC	MACB	REG	NTUSER key	Last Written	dab	DAB-PC	Software/NVIDIACorporation/Global/nvUpdate
26/03/13 18:30:50	UTC	MACB	REG	NTUSER key	Last Written	dab	DAB-PC	Software/Microsoft/WindowsNT/CurrentVersion/AppCompatFlags/CompatibilityAssistant/Persisted
26/03/13 18:31:33	UTC	MACB	REG	RecentDocs key	File opened	dab	DAB-PC	Recently opened file of extension: .tif - value: 1.tif
26/03/13 18:31:50	UTC	MACB	REG	FileExts key	Extension Changed	dab	DAB-PC	File extension .tif opened by DllHost.exe
26/03/13 18:34:10	UTC	MACB	REG	NTUSER key	Last Written	dab	DAB-PC	Software/Adobe/AcrobatReader/9.0/AVGeneral/cRecentFiles/c1
26/03/13 18:34:10	UTC	MACB	REG	NTUSER key	Last Written	dab	DAB-PC	Software/Adobe/AcrobatReader/9.0/RememberedViews/cNoCategoryFiles/c11/cViewDef
26/03/13 18:34:10	UTC	MACB	REG	NTUSER key	Last Written	dab	DAB-PC	Software/Adobe/AcrobatReader/9.0/AVGeneral/cRecentFiles/c2
26/03/13 18:34:10	UTC	MACB	REG	NTUSER key	Last Written	dab	DAB-PC	Software/Adobe/AcrobatReader/9.0/RememberedViews/cNoCategoryFiles/c11
26/03/13 18:34:10	UTC	MACB	REG	NTUSER key	Last Written	dab	DAB-PC	Software/Adobe/AcrobatReader/9.0/AVGeneral/cRecentFiles/c3
26/03/13 18:34:10	UTC	MACB	REG	NTUSER key	Last Written	dab	DAB-PC	Software/Adobe/AcrobatReader/9.0/AVGeneral/cRecentFiles/c4
26/03/13 18:34:10	UTC	MACB	REG	NTUSER key	Last Written	dab	DAB-PC	Software/Adobe/AcrobatReader/9.0/RememberedViews/cNoCategoryFiles/c11/cViewDef/cTopLeftView
26/03/13 18:34:10	UTC	MACB	REG	NTUSER key	Last Written	dab	DAB-PC	Software/Adobe/AcrobatReader/9.0/AVGeneral/cRecentFiles/c5
26/03/13 18:34:43	UTC	MACB	REG	NTUSER key	Last Written	dab	DAB-PC	Software/SmithMicro/StuffIt
26/03/13 18:34:43	UTC	MACB	REG	NTUSER key	Last Written	dab	DAB-PC	Software/SmithMicro/StuffIt/ContextMenu
26/03/13 18:41:48	UTC	MACB	REG	NTUSER key	Last Written	dab	DAB-PC	Software/Microsoft/Windows/CurrentVersion/Explorer/StartPage
26/03/13 18:41:48	UTC	MACB	REG	NTUSER key	Last Written	dab	DAB-PC	Software/Microsoft/Windows/CurrentVersion/Explorer/StartPage/NewShortcuts
02/04/13 18:19:30	UTC	MACB	REG	NTUSER key	Last Written	dab	DAB-PC	Software/Microsoft/MediaPlayer/Player/Skins/res://wmploc/RT_TEXT/player.wsz
02/04/13 18:20:10	UTC	MACB	REG	NTUSER key	Last Written	dab	DAB-PC	Software/Microsoft/MediaPlayer/Setup
02/04/13 19:12:24	UTC	MACB	REG	RecentDocs key	File opened	dab	DAB-PC	Recently opened file of extension: .jpg - value: 21 DSC_6506.jpg

Figure 4. *Excerpt from a Windows 7 timeline showing some NTUSER.dat events*

The last information, MountPoint, can be correlated with the ones extracted from the *setupapi.log*.

SETUPAPI.LOG

The file *setupapi.log* is very interesting: it hold the list of every devices connected to you system, along with device and volume serial number, GUID and other information that can be cross-checked with both *NTUSER.dat* and *SYSTEM* hives to get a complete view of every USB device connected to your system.

It is important to say that *setupapi.log* can be found only on systems running Windows XP, under *C:\Windows*. On a Windows 7 system, the file is now called *setupapi.dev.log*. By the way, *log2timeline* only support *setupapi.log* parsing.

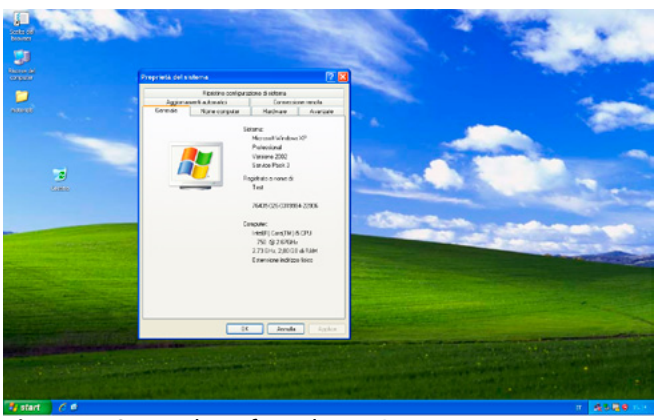


Figure 5. *Screenshot of Windows XP properties*

Figure 6. *Windows XP mounting*

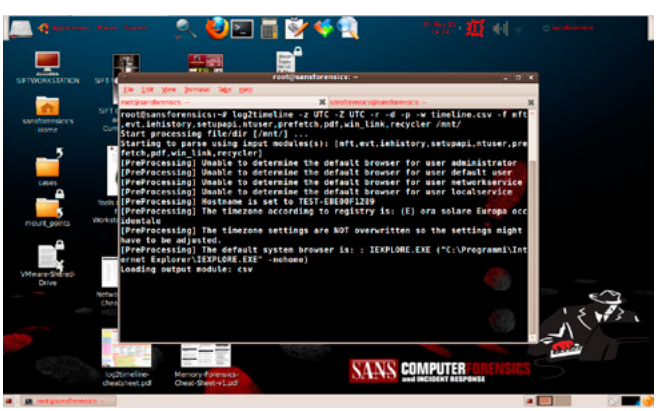


Figure 7. *log2timeline in action*

TIMELINING

So, for example, if you need to track user logon and logoff activities, you have to extract the associate events from the Windows Event Log system, write down the date and time of each events, and then create a simple timeline of such events.

Fortunately, there are many software programs out there that have the ability to create timeline in an automated way: *fls* and *mactime* from The SleuthKit can be used to spawn the timeline.

The tool used in our experiment is *log2timeline*, the perl script version 0.65, shipped with the Linux distribution SIFT Live CD made by the SANS Forensics community.

By specifying which sources to parse, *log2timeline* can extracts all the needed informations and build an effective timeline; the default output is a CSV file format that can be easily managed and imported into Microsoft Office Excel or OpenOffice Calc.

GETTING STARTED

Before getting started, just a foreword: since the goal of timelining is to get a trustworthy chronological events time line, it is important to make certain of computer and BIOS clock, so we can correlate the time found on the machine with the real time, using a reliable time source such as a NTP server, and write down the machine time delay.

Let's start our tests to check out Windows XP activities. To do that, we are using a virtual machine running Windows XP Service Pack 3, as the Figure 5 says.

After booting into SIFT, we ran `log2timeline` against `/mnt`, which was the mount point of our Windows XP installation, mounted in read-only: Figure 6.

As data source to parse with *log2timeline*, we selected the following:

- *evt,* Windows Event Log found under *Windows\system32\config* folder,
- *iehistory,* Internet Explorer history, all the *index.dat* found on the system,
- *prefetch*, as the name says, the .pf files found under *Windows\prefetch* folder,
- *recycler*, every items found into recycle bin folders,
- *win_link*, every LNK found,
- *mft*, as the name says, all the informations found into `$MFT` metafile,
- *ntuser*, parse *NTUSER.dat* to find user activities through GUI,
- *setupapi*, parse *setupapi.log* to find USB attached devices.

It could be useful to specify a timezone which the system belong or which timezone is used in the output file: it can be done with the *-z* and *-Z* flags.

The complete `log2timeline` command line was formed as follow (Figure 7):

```
log2timeline -z UTC -Z UTC -r -w timeline.csv
-d -p -f evt,iehistory,prefetch,recycler,
win_link,mft,setupapi,ntuser /mnt
```

After a while (it depends on how many source type you use on the command line and how much information it has to parse and grab), you have the *timeline.csv* file ready to be read.

READING THE TIMELINE

Now we need to read and to understand the informations extracted and sorted by timeline.

So, here the first lines of our timeline, in which you can read the columns meaning (Figure 8).

The most important columns to keep in mind when reading timelines are *date, time, sourcetype, type* and *short*, plus another one not included in the screenshot that you may take a look at, named *desc*, that holds a description of the related action.

With this information fixed in mind, we can start understanding what the timeline is trying to tell us.

The highlighted rows of Figure 9 show us two events parsed from Windows Event Logs: event ID 6005 and event ID 6009 both mean that the system has started the Event Log Monitor and then it shows the operating system information. Both the events are pulled down from the System Event

	date	time	timezone	MACB	source	sourcetype	type	user	host	short
2	05/31/13	14:16:17	UTC	.A..	FILE	NTFS $MFT	$SI [.A..] time	-	TEST-E8E00F1289	/System Volume Information/_r
3	05/31/13	14:16:17	UTC	.A..	FILE	NTFS $MFT	$SI [.A..] time	-	TEST-E8E00F1289	/WINDOWS/system32/ntdll.dll
4	05/31/13	14:16:17	UTC	..C.	FILE	NTFS $MFT	$SI [..C.] time	-	TEST-E8E00F1289	/System Volume Information/_r
5	05/31/13	14:16:17	UTC	MAC.	FILE	NTFS $MFT	$FN [MAC.] time	-	TEST-E8E00F1289	/System Volume Information/_r
6	05/31/13	14:16:17	UTC	MACB	PRE	XP Prefetch	Last run	-	TEST-E8E00F1289	NTOSBOOT-B00DFAAD.pf; NT
7	05/31/13	14:16:22	UTC	.A..	FILE	NTFS $MFT	$SI [.A..] time	-	TEST-E8E00F1289	/.

Figure 8. *timeline columns*

1111	05/31/13	14:16:29	UTC	MACB	FILE	NTFS $MFT	$FN [MACB] time	-	TEST-E8E00F1289	/System Volume Information/_restore{3EDD201A-674D-4829-AFD2-565E
1112	05/31/13	14:16:29	UTC	.A..	FILE	NTFS $MFT	$SI [.A..] time	-	TEST-E8E00F1289	/WINDOWS/system32/drivers/ndisuio.sys
1113	05/31/13	14:16:29	UTC	MACB	EVT	Event Log	Time generated/writt	-	TEST-E8E00F1289	EventLog/6005;Info;
1114	05/31/13	14:16:29	UTC	MACB	EVT	Event Log	Time generated/writt	-	TEST-E8E00F1289	EventLog/6009;Info;5.01. - 2600 - Service Pack 3 - Multiprocessor Free
1115	05/31/13	14:16:30	UTC	.A..	FILE	NTFS $MFT	$SI [.A..] time	-	TEST-E8E00F1289	/WINDOWS/system32/wbem/mof
1116	05/31/13	14:16:30	UTC	.A..	FILE	NTFS $MFT	$SI [.A..] time	-	TEST-E8E00F1289	/WINDOWS/system32/drivers/parvdm.sys
1117	05/31/13	14:16:30	UTC	..C.	FILE	NTFS $MFT	$SI [..C.] time	-	TEST-E8E00F1289	/WINDOWS/system32/wbem/Repository/FS
1118	05/31/13	14:16:30	UTC	.A..	FILE	NTFS $MFT	$SI [.A..] time	-	TEST-E8E00F1289	/WINDOWS/system32/logonui.exe.manifest

Figure 9. *user login*

1478	05/31/13	14:18:12	UTC	..C.	FILE	NTFS $MFT	$SI [..C.] time	-	TEST-E8E00F1289	/WINDOWS/explorer.exe
1479	05/31/13	14:18:12	UTC	MACB	REG	UserAssist key	Time of Launch	Administrator	TEST-E8E00F1289	UEME_RUNPATH:...[My Computer] VIRTUAL
1480	05/31/13	14:18:12	UTC	MACB	REG	UserAssist key	Time of Launch	Administrator	TEST-E8E00F1289	UEME_UISCUT
1481	05/31/13	14:18:15	UTC	MACB	REG	MountPoints2 key	Drive last mounted	Administrator	TEST-E8E00F1289	{d7b5e5a0-c9fc-11e2-bf3a-000c29a2309b} volume mounted
1482	05/31/13	14:18:15	UTC	MACB	LOG	SetupAPI Log	Entry written	-	TEST-E8E00F1289	DriverContextual Information. Contextual Information. Contex
1483	05/31/13	14:18:18	UTC	MACB	LOG	SetupAPI Log	Entry written	-	TEST-E8E00F1289	DriverContextual Information. Contextual Information. Contex
1484	05/31/13	14:18:26	UTC	.A..	FILE	NTFS $MFT	$SI [.A..] time	-	TEST-E8E00F1289	/WINDOWS/system32/diskcopy.dll
1485	05/31/13	14:18:26	UTC	MAC.	FILE	NTFS $MFT	$SI [MAC.] time	-	TEST-E8E00F1289	/WINDOWS/Prefetch/VERCLSID.EXE-3667BD89.pf
1486	05/31/13	14:18:26	UTC	MACB	PRE	XP Prefetch	Last run	-	TEST-E8E00F1289	VERCLSID.EXE-3667BD89.pf; VERCLSID.EXE was execut
1487	05/31/13	14:18:27	UTC	MACB	REG	NTUSER key	Last Written	Administrator	TEST-E8E00F1289	Software/Microsoft/Windows/CurrentVersion/Explorer/Stream
1488	05/31/13	14:18:27	UTC	MACB	LOG	SetupAPI Log	Entry written	-	TEST-E8E00F1289	DriverContextual Information. Contextual Information. Contex

Figure 10. *user clicked on "My Computer" icon*

1509	05/31/13	14:19:04	UTC	.AC.	FILE	NTFS $MFT	$SI [.AC.] time	-	TEST-E8E00F1289	/Programmi/Adobe/Reader 11.0/Reader/AcroRd32.exe
1510	05/31/13	14:19:04	UTC	..C.	FILE	NTFS $MFT	$SI [..C.] time	-	TEST-E8E00F1289	/WINDOWS/system32/shell32.dll
1511	05/31/13	14:19:04	UTC	MACB	REG	NTUSER key	Last Written	Administrator	TEST-E8E00F1289	Software/Microsoft/Windows/ShellNoRoam/MUICache
1512	05/31/13	14:19:04	UTC	MACB	REG	FileExts key	Extension Changed	Administrator	TEST-E8E00F1289	File extension .pdf opened by AcroRd32.exe
1513	05/31/13	14:19:05	UTC	..C.	FILE	NTFS $MFT	$SI [..C.] time	-	TEST-E8E00F1289	/Documents and Settings/Administrator/Impostazioni locali/Cronologia/History.IE5/MSHist012013053120130601
1514	05/31/13	14:19:05	UTC	..C.	FILE	NTFS $MFT	$SI [..C.] time	-	TEST-E8E00F1289	/Documents and Settings/Administrator/Impostazioni locali/Cronologia/History.IE5/MSHist012013053120130601/index.dat

Figure 11. *pdf opening*

1519	05/31/13	14:19:05	UTC	MAC.	FILE	NTFS $MFT	$SI [MAC.] time	-	TEST-E8E00F1289	/Documents and Settings/Administrator/Recent
1520	05/31/13	14:19:05	UTC	.A..	FILE	NTFS $MFT	$SI [.A..] time	-	TEST-E8E00F1289	/Documents and Settings/Administrator/Impostazioni locali/Cronologia/desktop
1521	05/31/13	14:19:05	UTC	MACB	FILE	NTFS $MFT	$SI [MACB] time	-	TEST-E8E00F1289	/Documents and Settings/Administrator/Recent/Mandiant.pdf.lnk
1522	05/31/13	14:19:05	UTC	MACB	FILE	NTFS $MFT	$FN [MACB] time	-	TEST-E8E00F1289	/Documents and Settings/Administrator/Recent/Mandiant.pdf.lnk
1523	05/31/13	14:19:05	UTC	MACB	FILE	NTFS $MFT	$SI [MACB] time	-	TEST-E8E00F1289	/Documents and Settings/Administrator/Recent/Install Ubuntu Gnome (E).lnk
1524	05/31/13	14:19:05	UTC	MACB	FILE	NTFS $MFT	$FN [MACB] time	-	TEST-E8E00F1289	/Documents and Settings/Administrator/Recent/Install Ubuntu Gnome (E).lnk
1525	05/31/13	14:19:05	UTC	MACB	WEBHIST	Internet Explorer	Last Visited/Last Visited	Administrator	TEST-E8E00F1289	visited file:///E:/Mandiant.pdf
1526	05/31/13	14:19:05	UTC	MACB	WEBHIST	Internet Explorer	Last Visited/Last Visited	Administrator	TEST-E8E00F1289	visited file:///E:/Mandiant.pdf
1527	05/31/13	14:19:05	UTC	MACB	WEBHIST	Internet Explorer	Last Visited/Last Visited	Administrator	TEST-E8E00F1289	visited file:///E:/Mandiant.pdf
1528	05/31/13	14:19:05	UTC	M...	WEBHIST	Internet Explorer	index.dat creation time	Administrator	TEST-E8E00F1289	visited file:///E:/Mandiant.pdf
1529	05/31/13	14:19:05	UTC	MACB	REG	UserAssist key	Time of Launch	Administrator	TEST-E8E00F1289	UEME_RUNPATH:C:/Programmi/Adobe/Reader 11.0/Reader/AcroRd32.exe
1530	05/31/13	14:19:05	UTC	MACB	REG	UserAssist key	Time of Launch	Administrator	TEST-E8E00F1289	UEME_RUNPATH
1531	05/31/13	14:19:05	UTC	MACB	REG	RecentDocs key	File opened	Administrator	TEST-E8E00F1289	Recently opened file of extension: .pdf - value: Mandiant.pdf
1532	05/31/13	14:19:05	UTC	MACB	PRE	XP Prefetch	Last run	-	TEST-E8E00F1289	ACRORD32.EXE-11DE888E.pf; ACRORD32.EXE was executed
1533	05/31/13	14:19:06	UTC	MACB	REG	RecentDocs key	Folder opened	Administrator	TEST-E8E00F1289	Recently opened file of extension: Folder - value: Install Ubuntu Gnome (E:)

Figure 12. *Adobe reader start*

1637	05/31/13	14:19:27	UTC	MACB	REG	NTUSER key	Last Written	Administrator	TEST-E8E00F1289	Software/Adobe/AcrobatReader/11.0/AVGeneral/cRecentFiles/c4
1638	05/31/13	14:19:27	UTC	MACB	REG	NTUSER key	Last Written	Administrator	TEST-E8E00F1289	Software/Adobe/AcrobatReader/11.0/AVGeneral
1639	05/31/13	14:19:27	UTC	MACB	REG	NTUSER key	Last Written	Administrator	TEST-E8E00F1289	Software/Adobe/AcrobatReader/11.0/AVGeneral/cRecentFiles/c2
1640	05/31/13	14:19:27	UTC	MACB	REG	NTUSER key	Last Written	Administrator	TEST-E8E00F1289	Software/Adobe/AcrobatReader/11.0/AVGeneral/cRecentFiles/c3
1641	05/31/13	14:19:27	UTC	MACB	REG	NTUSER key	Last Written	Administrator	TEST-E8E00F1289	Software/Adobe/AcrobatReader/11.0/AVGeneral/cRecentFiles
1642	05/31/13	14:19:27	UTC	MACB	REG	NTUSER key	Last Written	Administrator	TEST-E8E00F1289	Software/Adobe/AcrobatReader/11.0/AVGeneral/cRecentFiles/c1

Figure 13. *Adobe Reader recent documents key*

1737	05/31/13	14:20:20	UTC	MACB	EVT	Event Log	Time generated/written	-	TEST-E8E00F1289	VMTools/108;Info;
1738	05/31/13	14:20:20	UTC	MACB	EVT	Event Log	Time generated/written	-	TEST-E8E00F1289	EventLog/6006;Info;
1739	05/31/13	14:20:21	UTC	.A..	FILE	NTFS $MFT	$SI [.A..] time	-	TEST-E8E00F1289	/WINDOWS/system32/Microsoft/Protect/S-1-5-18/User/Preferred
1740	05/31/13	14:20:25	UTC	MAC.	FILE	NTFS $MFT	$SI [MAC.] time	-	TEST-E8E00F1289	/WINDOWS/WindowsUpdate.log
1741	05/31/13	14:20:26	UTC	MAC.	FILE	NTFS $MFT	$SI [MAC.] time	-	TEST-E8E00F1289	/WINDOWS/system32/config/software.LOG
1742	05/31/13	14:20:40	UTC	MAC.	FILE	NTFS $MFT	$SI [MAC.] time	-	TEST-E8E00F1289	/WINDOWS/bootstat.dat
1743	05/31/13	14:20:40	UTC	MA..	FILE	NTFS $MFT	$SI [MA..] time	-	TEST-E8E00F1289	/WINDOWS/system32/config/system
1744	05/31/13	14:20:40	UTC	MA..	FILE	NTFS $MFT	$SI [MA..] time	-	TEST-E8E00F1289	/WINDOWS/system32/config/software
1745	05/31/13	14:20:40	UTC	MA..	FILE	NTFS $MFT	$SI [MA..] time	-	TEST-E8E00F1289	/Documents and Settings/NetworkService/NTUSER.DAT

Figure 14. *User logout*

REFERENCES

- H. Carvey, Windows Forensic Analysis Toolkit 3rd Edition, Syngress, ISBN 978-1597497275
- K. Guðjónsson, Mastering the Super Timeline With log2timeline, *http://computer-forensics.sans.org/community/papers/gcfa/mastering-super-timeline-log2timeline_5028*
- Timestomp, *http://www.forensicswiki.org/wiki/Timestomp*
- log2timeline download page, *https://code.google.com/p/log2timeline/*
- H. Carvey, Windows Registry Forensics, Syngress, ISBN 978-1597495806
- B. Carrier, File System Forensic Analysis, Addison Wesley, ISBN 978-0321268174
- H. Carvery, Windows Incident Response Blog, *http://windowsir.blogspot.it/*
- SANS SIFT, *http://computer-forensics.sans.org/community/downloads*

Log. It is important to see that by default Windows XP disables the Security Event Log, so you won't find any event associated with the user login. But that is a different story that we will address later in the future.

The rows from 1481 through 1483 tell us that an USB device was inserted and what was the last time it was plugged into the system: you can see the events parsed from both the Registry and the *setupapi.log* (Figure 10).

Looking at rows 1478 and 1479 you can see two Windows Registry entries, parsed from the *NTUSER.DAT* file: the user assist key, which sets the `UEME_RUNPATH` value.

Looking at the fourth column, you can see the entry has all the timestamp values set up (MACB) which means, as we said on the previous paragraph, that the entry was created for the first time (Figure 11 and Figure 12).

The two images above tell us an important thing: the first, at row 1512, tell us that a pdf file was opened by *AcroRd32.exe*, which is the binary file of Adobe Reader. The second image can be read as follow:

- row 1519: the Windows Recent folder timestamp were updated, as the MAC values told us,
- row 1521 and 1522: the entry *Mandiant.pdf.lnk* was created under the Recent folder, as the MACB timestamps said, and you can see also the setting of `$SI` and `$FN`, that are set up both only on file creation,
- row 1525: the file Mandiant.pdf was opened from the E: drive,
- row 1528: the index.dat file was updated, as the above event was written (M timestamp),
- row 1529: user assist key created to include the Adobe Reader starts,
- row 1532: the creation of the Adore Reader prefetch file.

So, from the actions listed above, we can reconstruct the events: on 31 May 2013, at the 14:18 UTC, someone inserted an usb drive, which had the E: logical drive letter, then at 14:19 UTC the user opened a PDF file on the E: drive named *Mandiant.pdf* that was opened by Adobe Reader.

Figure 13 shows us the creation of Adobe Reader recent opened documents on the Windows Registry *NTUSER.DAT* file.

The last image represent the user logout, as the Windows Event ID 6006 asserts (Figure 14).

CONCLUSIONS

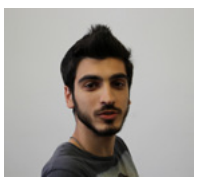
As we had seen, timelining is a powerful technique to reconstruct events that have occurred on a Windows Operating systems.

It is important to keep in mind that each Windows versions diverges from the others in some ways, so it is important to study, stay up to date and, most important, test and experiment Windows behavior with timelining.

In this case we had reconstructed a possibly malicious action, since the file could either have been stolen or could have contained malware. It is also important to keep in mind the goal of investigations, so you don't drive out of scope.

The tool presented, log2timeline, is a leading one on the open source digital forensics community, but you can use any tools you are comfortable with: the key concept is to be able to extract the information needed, read them, and present them so either a not technical audience can understand them.

About the Author



Davide Barbato has 10 years of IT experience, the last three in Digital Forensics and Incident Response. He is currently employed in an important DFIR national firm, in which his works as Chief Security Officer and DFIR analyst. He is also a teacher and speaker at national meetings and universities about Digital Forensics, IT Security and IT Privacy.
dav.barbato@gmail.com

Dr.WEB®
20 years

Dr.Web SpIDer is 8-legged!

New Version 8.0

Security Space and Dr.Web Antivirus for Windows

Get your free 60-day license under https://www.drweb.com/press/
to protect your PC and your smartphone with Dr.Web!
Your promo code: ***Hakin9***

Protect your mobile device free of charge!
https://support.drweb.com/free_mobile/

Doctor Web is a Russian anti-virus vendor with a software development record dating back to 1992.
www.drweb.com

UNDERSTANDING FILE METADATA

HOW TO VIEW & INTERPRET DATA ABOUT DATA

by Chris Sampson

Metadata exists throughout data storage systems, from the creation and modification dates stored within the file system, through to specific information embedded within the content of a file. Metadata can be hugely important to any forensic investigation, knowing how to extract this information and spot when it has been manipulated can prove very important.

What you will learn:

- What metadata is and how to access it
- Tools that allow the display of file metadata for multiple file containers
- The importance of researching and developing your own metadata extraction techniques
- Manipulation and modification of metadata

What you should know:

- You should be confident with at least one of the discussed operating systems (Windows 7, Mac OS X 10.8.3 and Ubuntu Linux 12.0.4)
- Basics of general operating system usage, file storage concepts, file properties and attributes
- Basics of your chosen operating system's command line

This article, aimed at those new to forensics, looks at various forms of metadata. It provides examples of the way in which we can manually retrieve this information using our operating systems and moving on to specific tools which can extract the metadata for us.

WHAT IS METADATA?

Metadata is a fairly broad topic, there are many different things that can accurately be described as being metadata. Metadata exists in many different forrms, each describes a specific feature or attribute of a file or directory item on a computer, some are common to all data, others are unique to a specific file or directory type.

Put simply metadata is data that describes other data. It exists everywhere, in your file system and (if supported) its journal, in email headers, within instant search databases, inside the Windows Registry, log files and Mac OS file resource forks to name but a few. For many file container types there is a huge amount of metadata that can also be found within the individual files themselves.

Metadata can be gathered from a large number of different resources, key to getting the most accurate picture of a file from its metadata is in understanding the system that created or used the data file, its quirks, peculiarities and indexing abilities. If you have a good grasp of this then you are off to an excellent start.

Good knowledge of the file type that you are investigating will help you to get the most complete picture of the metadata that it can store and where this data can be found. If you are investigating a new file type for the first time it may be wise to con-

duct a little research, information that may prove helpful could include:

- Documentation from the publisher of the software used to create the file type, the availability of this kind of information varies from publisher to publisher.
- If possible you should install and use the software, create files of the type that you are investigating then examine these to create familiarity with the file container.
- Find third party documentation regarding a file type. The open source community, particularly those who create tools to access or modify the specific data type, can be a great source of detailed information.
- Familiarize yourself with manually editing or manipulating metadata within the data container.

The more information that you are armed with prior to carrying out an examination, the better placed you will be to accurately and efficiently extract the information that you need.

FILE SYSTEM METADATA

The first place to look for metadata is within your computer's file manager. File managers are the most direct link between the computer's storage and the user interface. Lots of information that you will already be familiar with is available directly from the GUI of all modern operating systems. Examples of the types of metadata normally retrieved by your computer directly from the file system are:

- File Name
- File Path
- File Size
- Creation Date
- Modification Date

Whilst the above are pretty standard, the type of information available can vary significantly depending upon the specific operating system involved and the type and version of file system that is being used to store data for that OS.
Often additional information exists defining certain attributes that have been assigned to a file or folder by the OS or the user. This information can include permissions, author, character encoding, file version information and potentially much more. In many modern file systems additional streams or forks exist for each file entry which can contain additional information which is not normally displayed in the GUI.

We should all already be familiar with how to view most of this information, as users we have had to sort data in *Windows Explorer*, *Finder* or other mangers. We sort by date to find the file or version of a file that we need and by alphabetical order to quickly find a file when we know its name. We do this daily without a second thought but as computer examiners we often need more information than the operating system is going to provide us with by default.

File system metadata is just one source of information that is available about a file. Different file types often have attributes specific to their usage or purpose, often this data is not universally applicable.

WHAT KIND OF METADATA CAN BE STORED WITHIN A FILE?

The short answer is, anything, the limit exists only within the file specification, the format designer's creativity and practical usability limits. The kind of metadata that is stored in a file can vary from none, or only certain key attributes, through to many different features, properties, timestamps and more.

In the table below we look at five commonly used file types and describe some of the metadata which can be written to (and subsequently extracted from) a specific file container. With many of these file types being based upon different interpretations of standardized data containers, some or all of the supported metadata may be missing or implemented incorrectly. Sometimes a file specification is followed very closely but something additional is added within the metadata to serve the needs of a particular implementation.

It is important that metadata only be used as a guide rather than an absolute, later we will look at the output of different applications using the same file format as well as tools and methods that are available for the reading, extraction and manipulation of file metadata

Table 1. *Common internal file metadata*

File Type	Samples of Supported Metadata
.jpeg image files (EXIF)	Date and time, specific camera information including make, model and settings, thumbnail of the image, description, dimensions, copyright information, GPS information
MP3 files	Title, artist, album, genre, comments, copyrights, size, bitrate
PDF Documents	Version, File Size, No of Pages, Producer, Creator, Title Subject, Author, Pages, Keywords
DOCX files	CRC32, author, last modified date, version number, preview thumbnail, creation date, file size, word count, number of pages etc

Note The above information is not intended to be a complete list of available metadata for each file type, it is rather an example of some of the data that is commonly available within each file type. Researching of the internal structure of a file container and its supported metadata is required for a thorough understanding of the possibilities and limitations of metadata storage for a particular container type.

Much of this data is accessible directly from either your computer's file manager. There are also a plethora of tools available, many third party utilities enable direct viewing and often editing of the file's metadata.

EXAMINING FILE METADATA

The information displayed within the file manager window within a GUI based operating system is normally only a small subset of the metadata that exists and is accessible for any given file. Some is hidden, reserved for OS usage, some is considered to not be important and is therefore not displayed. But most information that is stored can be accessed in one way or another. Often, more modern OS features like versioning, journaling and instant search can hold more data than is available directly through the interface. In most cases there are tools, applications or techniques which can be used to display this data.

WINDOWS

Using *Windows Explorer* we can see a number of metadata elements from within an *Explorer* window. This information is configurable too and supports the metadata of many different file types. To discover what types of metadata can be viewed through *Explorer*, try the example given below:

- Use *Windows Explorer* to navigate to the folder that contains the file types that you wish to examine.
- Change the View type to 'Details'
- Right click the column headings to display the following contextual menu
- Click the More item at the bottom of the menu

This will open a new window within which you can choose the type of metadata that you want to display. If Windows supports the file type and that file type contains the metadata that you have selected, you will be able to see the meta contents directly within *Windows Explorer*.

UBUNTU

Linux systems are a little more limited in the information provide within the standard GUI, although this can easily be changed. Whilst no specific tool exists within Ubuntu for metadata viewing there is the *file* command.

There are some limitations to *file* though as it is not really intended as a metadata analyser, so although you can find out lots of detail about the meta content of a Microsoft Word OLE (.doc) document, there is no metadata available for Microsoft Open XML (docx) files.

Here is an example of using *file* to display the metadata for a newly created word document:

```
:~/Desktop$ file Sample\ Document.doc
```

Sample Document.doc: Composite Document File V2 Document, Little Endian, Os: MacOS, Version 10.3, Code page: 10000, Author: Christopher Sampson, Template: Normal.dotm, Last Saved By: Christopher Sampson, Revision Number: 2, Name of Creating Application: Microsoft Macintosh Word, Total Editing Time: 01:00, Create Time/Date: Sat Jun 8 11:39:00 2013, Last Saved Time/Date: Sat Jun 8 11:39:00 2013, Number of Pages: 1, Number of Words: 4, Number of Characters: 24, Security: 0.

The formatting of this data isn't the most readable, but there is a good amount of important information that is displayed. However, running the file command on one of the newer Office Open XML docx format files simply returns:

```
:~/Desktop$ file Sample\ Document.docx
Sample Document.docx: Microsoft Word 2007+
```

This shows that whilst *file* is able to determine the document type, it does not currently support metadata parsing from it. *File* is designed primarily to identify different file types and as such is not really best suited to the task of metadata analysis, with that having been said though it is possible to extend *file* with new file types.

There are other ways to get more information about lots of different file types on a Linux system using *ExifTool*, a free application that will also run on Windows and Mac. We will look at *ExifTool* once we have discussed the Mac OS X default options.

MAC OS X

File is a Unix utility which is available by default throughout most Unix and Unix like operating systems, as such Mac OS X (which borrows large portions of FreeBSD for its core) can also make use of *file*. As with the Ubuntu example above, *file* alone is probably not the best solution for metadata examination.

Similar to *Windows Explorer* the Mac OS X *Finder* is capable of displaying additional metadata for supported file types directly within the interface. The process for enabling this is almost identical to the way that we achieve this within Windows. First we need to set the *Finder* window to list view, right click (or ctrl+click) on the column header area where you can then select the data that you want

to view. It should be noted that a quirk of Mac OS X is to only display the available metadata options relevant to the data shown within the window that you are changing.

There is another powerful option if you are running a version of OS X that includes Apple's Spotlight application. Mac OS X systems that include Spotlight also include a fantastic tool for viewing metadata that has been captured and indexed by Spotlight, mdls. Mdls can be used to see extended information for supported file types and there are a lot of supported file types. The output of mdls is very comprehensive, to the right is the data that was extracted from a newly created Microsoft Word .docx file: Listing 1.

As you can see, whilst the output of mdls is very detailed, it is not formatted to make for easy reading. Despite the formatting it is still a pretty simple task to extract the required information from the output.

Spotlight presents several interesting possibilities for metadata analysis in general but we will not go into any further detail about that within this article.

THIRD PARTY METADATA TOOLS

Whilst it is useful to understand the possibilities of viewing file metadata within the operating system, without the need for third party tools, there are other more comprehensive options available. We will look briefly at some of the freely available tools before discussing the cross platform *ExifTool,* which will be used during our examples.

IMAGE FILES

Image files often contain a large amount of metadata, from camera type, to time stamps, geo location and more. Much of this information can be extracted by doing nothing more than opening the file using a text editor or hex editor. Free tools are plentiful, as are libraries and open source projects, which can be used to develop your own utilities.

For a quick and simple inspection of a supported file types e*xifviewer.org* has a web based tool that displays friendly, easy to interpret metadata. *Exifviewer* is built upon the *Exif2* library.

It is also important to note that many image file formats can contain a thumbnail of the original image. In most cases this thumbnail will mirror the full sized image. When it does not comparison can help to identify potential editing and manipulation.

Many operating systems also cache thumbnails within their File Managers for image previewing purposes. OS caches can prove an important source for metadata analysis.

PDF FILES

The metadata contained within a PDF document varies greatly and can depend on what tool creat-

Listing 1. *Using Mac OS X's mdls command to extract file metadata*

```
Command:
chrissampson$ mdls ~/Desktop/Sample\ Document.
docx

Output:
kMDItemAlternateNames     = (
  "Sample Document.docx"
)
kMDItemAuthors          = (
  "Christopher Sampson"
)
kMDItemContentCreationDate    = 2013-06-09
10:35:06 +0000
kMDItemContentModificationDate = 2013-06-09
10:35:06 +0000
kMDItemContentType        = "org.
openxmlformats.wordprocessingml.document"
kMDItemContentTypeTree      = (
  "org.openxmlformats.wordprocessingml.
document",
  "org.openxmlformats.openxml",
  "public.zip-archive",
  "com.pkware.zip-archive",
  "public.data",
  "public.item",
  "com.apple.bom-archive",
  "public.archive",
  "public.composite-content",
  "public.content"
)
kMDItemDateAdded         = 2013-06-09 10:35:06
+0000
kMDItemDisplayName        = "Sample Document"
kMDItemEditors          = (
  "Christopher Sampson"
)
kMDItemFSContentChangeDate    = 2013-06-09
10:35:06 +0000
kMDItemFSCreationDate      = 2013-06-09
10:35:06 +0000
kMDItemFSCreatorCode       = "MSWD"
kMDItemFSFinderFlags       = 16
kMDItemFSHasCustomIcon      = 0
kMDItemFSInvisible        = 0
kMDItemFSIsExtensionHidden    = 1
kMDItemFSIsStationery      = 0
kMDItemFSLabel          = 0
kMDItemFSName          = "Sample Document.docx"
kMDItemFSNodeCount        = 22944
kMDItemFSOwnerGroupID      = 20
kMDItemFSOwnerUserID       = 501
kMDItemFSSize          = 22944
kMDItemFSTypeCode         = "WXBN"
kMDItemKind           = "Microsoft Word
document"
kMDItemLogicalSize        = 22944
kMDItemOrganizations       = (
  "TRC Data Recovery Ltd"
)
kMDItemPhysicalSize       = 24576
```

Listing 2. *An example of metadata extracted using ExifTool*

```
chrissampson$ exiftool ~/Desktop/Sample\
Document.docx
ExifTool Version Number      : 9.31
File Name                    : Sample Document.docx
Directory                    : /Users/chrissampson/
Desktop
File Size                    : 22 kB
File Modification Date/Time  : 2013:06:09
11:35:06+01:00
File Access Date/Time        : 2013:06:09
12:13:07+01:00
File Inode Change Date/Time  : 2013:06:09
11:35:06+01:00
File Permissions             : rw-r--r--
File Type                    : DOCX
MIME Type                    : application/
vnd.openxmlformats-officedocument.
wordprocessingml.document
Zip Required Version         : 20
Zip Bit Flag                 : 0x0006
Zip Compression              : Deflated
Zip Modify Date              : 1980:01:01 00:00:00
Zip CRC                      : 0xb01051e9
Zip Compressed Size          : 397
Zip Uncompressed Size        : 1474
Zip File Name                : [Content_Types].xml
Preview Image                : (Binary data 9500
bytes, use -b option to extract)
Title                        :
Subject                      :
Creator                      : Christopher Sampson
Keywords                     :
Description                  :
Last Modified By             : Christopher Sampson
Revision Number              : 1
Create Date                  : 2013:06:09 10:34:00Z
Modify Date                  : 2013:06:09 10:35:00Z
Template                     : Normal.dotm
Total Edit Time              : 1 minute
Pages                        : 1
Words                        : 4
Characters                   : 24
Application                  : Microsoft Macintosh
Word
Doc Security                 : None
Lines                        : 1
Paragraphs                   : 1
Scale Crop                   : No
Company                      : TRC Data Recovery Ltd
Links Up To Date             : No
Characters With Spaces       : 27
Shared Doc                   : No
Hyperlinks Changed           : No
App Version                  : 14.0000
```

ed the document as well as the settings for that application. There are also a number of different PDF specifications that govern the file format, with varying metadata support and implementation. One of the simplest approaches to extracting the metadata from a PDF document is to open the file in your text editor or hex editor.

Some metadata from a PDF is also available within the operating system or via specific tools like *Adobe Acrobat* and *Acrobat Reader*. For a more thorough examination or a custom implementation, *Xpdf* can be considered. *Xpdf* is open source under the GPL. You can find many pre-compiled versions of this tool for different systems.

MULTIPLE FORMAT APPLICATIONS

A particularly useful application for metadata analysis is *ExifTool*. *ExifTool* is written in Perl and as such is available for most operating systems, giving a consistent command line interface across each. *ExifTool* has support for a huge number of different file types (which are also expandable) and is an excellent tool for extracting metadata from common file types. Just like *mdls*, the output of *ExifTool* is extremely detailed, but unlike *mdls, ExifTool* can also be used on Windows and Linux as well as Mac OS X. To the left is the output of *ExifTool* on our *Sample Document.docx* file: Listing 2.

Below is the same output from *ExifTool*, but this time we have broken it down by metadata source. As you will see some of the information displayed by *ExifTool* is file system metadata and not simply file metadata: Listing 3.

ExifTool should be used in conjunction with your own examination and validation techniques. We often use *ExifTool* at TRC Data Recovery when we are examining a new file type.

EXAMPLE: MANUALLY EXAMINING THE METADATA OF A DOCX FILE

Microsoft Office has a massive installed base. Most examinations of computers will turn up docx files, many will also call for an interpretation of specific attributes and metadata of these files. For this article we are going to be looking at a sample docx file in 2 ways, first we will see what our operating system can tell us about the file, then we will dive in to the file's content to see where this information came from and whether it is possible to extract anything more or even differently interpreted data from the file.

Perhaps the simplest way to find out information about our Sample Document.docx file is to use our Windows system. Simply by right clicking on the file and selecting properties from the contextual menu provides us with quite a bit of useful information. Microsoft Word does not need to be installed for this information to be displayed, however, if it is we can get more details: Figure 1.

Listing 3. *An example of metadata extracted using ExifTool broken down by category*

```
chrissampson$ exiftool ~/Desktop/Sample\ Document.docx

ExifTool Information

ExifTool Version Number     : 9.31

File System Metadata

File Name            : Sample Document.docx
Directory            : /Users/chrissampson/Desktop
File Size            : 22 kB
File Modification Date/Time    : 2013:06:09 11:35:06+01:00
File Access Date/Time      : 2013:06:09 12:13:07+01:00
File Inode Change Date/Time    : 2013:06:09 11:35:06+01:00
File Permissions        : rw-r--r--
File Type            : DOCX
MIME Type            : application/vnd.openxmlformats-officedocument.wordprocessingml.document

Parent zip container details

Zip Required Version      : 20
Zip Bit Flag          : 0x0006
Zip Compression         : Deflated
Zip Modify Date         : 1980:01:01 00:00:00
Zip CRC             : 0xb01051e9
Zip Compressed Size       : 397
Zip Uncompressed Size      : 1474
Zip File Name          : [Content_Types].xml

Open XML Metadata Extracted from the Document

Preview Image          : (Binary data 9500 bytes, use -b option to extract)
Title              :
Subject             :
Creator             : Christopher Sampson
Keywords            :
Description           :
Last Modified By        : Christopher Sampson
Revision Number         : 1
Create Date           : 2013:06:09 10:34:00Z
Modify Date           : 2013:06:09 10:35:00Z
Template            : Normal.dotm
Total Edit Time         : 1 minute
Pages              : 1
Words              : 4
Characters           : 24
Application           : Microsoft Macintosh Word
Doc Security          : None
Lines              : 1
Paragraphs           : 1
Scale Crop           : No
Company             : TRC Data Recovery Ltd
Links Up To Date        : No
Characters With Spaces     : 27
Shared Doc           : No
Hyperlinks Changed       : No
App Version           : 14.0000
```

If Word is installed you can find a document's metadata through the File menu by selecting the Info tab. Towards the right hand side of the resulting tab you will find the file details and a toggle link that enables more or less information to be displayed: Figure 2.

This information is particularly important as it introduces us to the first example of ways to edit the file's metadata. In this case we can only set attributes that are not already set during the creation of the document, yet it does show that direct manipulation is possible.

Manual extraction of the content of a Microsoft Office XML file relies upon an understanding of how the file format is structured, luckily enough the format and structure of the docx file is standardized and relatively straightforward. A simplistic way of looking at a docx file is to consider it as a folder, this folder, just like any other folder on your computer has a hierarchy and somewhere within that hierarchy is the data that we are looking for.

The docx folder structure is zipped to save space. The first step involved in delving into the contents of the docx, without resorting to specialist tools, is to rename the file extension from docx to zip. Doing so will allow extraction of the file's content using any standard application that can extract a zip file. All of our operating systems can do this natively but there are also many third party tools available that can also carry out this task.

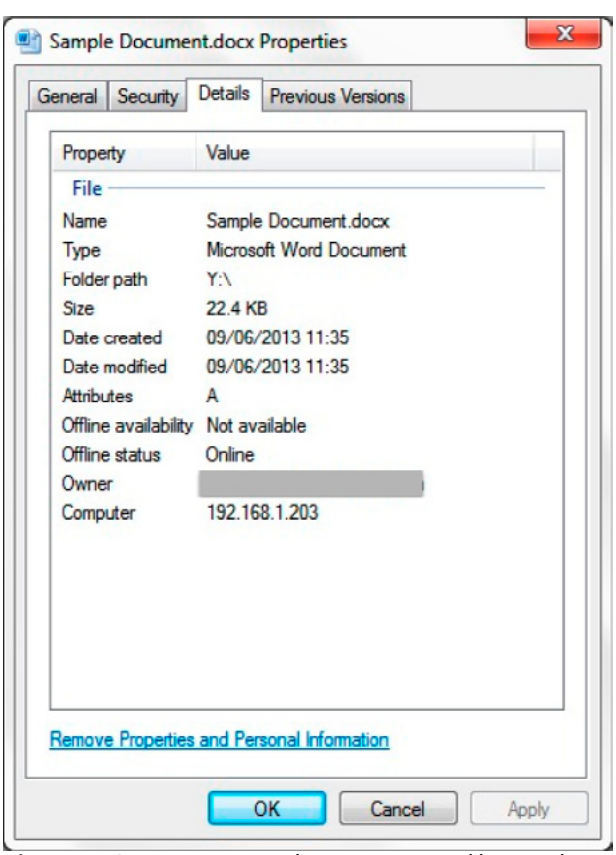


Figure 1. *Document metadata as presented by Windows 7 within the Properties window*

Once extracted we can see how a docx file is really made up, providing that your file was not encrypted or damaged, you should be presented with a structure that is very similar to the structure of our 'Sample Document.docx', here is an *ls* output: Listing 4.

Please note, the structure of other *Microsoft Office files*, such as those created with *Powerpoint* and *Excel*, follows the exact same structure as the docx file, but some of the internal structure is slightly different. Why not also rename your xlsx and pptx files to zip and see how they differ.

The internal contents of a docx file are based upon many XML files (this is why the letter 'x' was appended to the original doc format file extension, it also why the standard is known as Office Open XML). XML files are not all that a word document can contain, it is possible to have images and oth-

Listing 4. *Recursive output of the ls command on our unzipped docx file*

```
chrissampson$ ls -R
[Content_Types].xml  docProps
_rels       word

./_rels:

./docProps:
app.xml     core.xml thumbnail.jpeg

./word:
_rels       settings.xml      theme
document.xml      styles.xml     webSettings.
xml
fontTable.xml     stylesWithEffects.xml

./word/_rels:
document.xml.rels

./word/theme:
theme1.xml
```

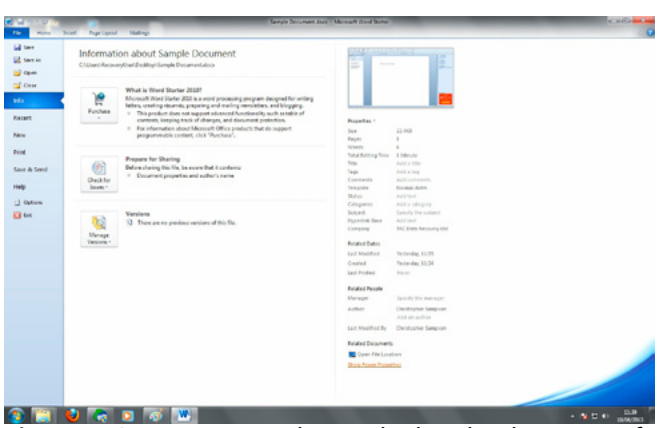


Figure 2. *Document metadata as displayed within Microsoft Word*

er files that are available as individual items embedded within the file. These items can also be extracted from your renamed docx file.

So, this article is all about metadata, and having used tools like ExifTool and mdls we already know that our sample file is full of metadata, how do we find it? Well there are a few different locations, but the most important metadata exists within the ./docProps directory as below:

```
./docProps:
app.xml
core.xml
thumbnail.jpeg
```

The core.xml and app.xml contain the metadata that has been extracted by ExifTool in an XML format, the output of these files from our Sample Document.docx is reprinted Listing 5 and Listing 6.

So now lets compare the information extracted manually with the metadata displayed within *ExifTool (Table 2)*.

So, as we can see, there is no great mystery to determining the metadata for the docx file type. The same applies to most other file types, all that is required is a basic understanding of how the file is structured and what metadata can be contained within it. Once we have a clearer understanding about the inner workings of the file, we can compare the results of tools like *ExifTool* or *mdls* with what we can actually see within the file.

Note: for an even deeper understanding about metadata extraction, the source code for *ExifTool* and Unix *file* is freely available and may represent

Listing 5. *Sample Document.docx App.xml content*

```
<?xml version="1.0" encoding="UTF-8"?>
<Properties xmlns="http://schemas.openxmlformats.org/officeDocument/2006/extended-properties"
xmlns:vt="http://schemas.openxmlformats.org/officeDocument/2006/docPropsVTypes">
  <Template>Normal.dotm</Template>
  <TotalTime>1</TotalTime>
  <Pages>1</Pages>
  <Words>4</Words>
  <Characters>24</Characters>
  <Application>Microsoft Macintosh Word</Application>
  <DocSecurity>0</DocSecurity>
  <Lines>1</Lines>
  <Paragraphs>1</Paragraphs>
  <ScaleCrop>false</ScaleCrop>
  <Company>TRC Data Recovery Ltd</Company>
  <LinksUpToDate>false</LinksUpToDate>
  <CharactersWithSpaces>27</CharactersWithSpaces>
  <SharedDoc>false</SharedDoc>
  <HyperlinksChanged>false</HyperlinksChanged>
  <AppVersion>14.0000</AppVersion>
</Properties>
```

Listing 6. *Sample Document.docx App.xml content*

```
<?xml version="1.0" encoding="UTF-8"?>
<cp:coreProperties xmlns:cp="http://schemas.openxmlformats.org/package/2006/metadata/core-
properties" xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:dcterms="http://purl.org/dc/
terms/" xmlns:dcmitype="http://purl.org/dc/dcmitype/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
  <dc:title />
  <dc:subject />
  <dc:creator>Christopher Sampson</dc:creator>
  <cp:keywords />
  <dc:description />
  <cp:lastModifiedBy>Christopher Sampson</cp:lastModifiedBy>
  <cp:revision>1</cp:revision>
  <dcterms:created xsi:type="dcterms:W3CDTF">2013-06-09T10:34:00Z</dcterms:created>
  <dcterms:modified xsi:type="dcterms:W3CDTF">2013-06-09T10:35:00Z</dcterms:modified>
</cp:coreProperties>
```

an excellent research opportunity for those wishing to learn more about metadata.

MANIPULATING METADATA

We will not discuss methods and techniques for manipulating metadata in depth, manipulation of this information is beyond the scope of this article. Below there is a single example of a very simple method of manipulating our Sample Document. docx. For this example no specialist tools are required beyond a text editor (all of our systems include text editors).

We have already renamed our Sample Document.docx file to .zip in the example above. To follow this yourself, please do the same and extract the contents. We are going to modify just one of the metadata fields changing the name of the person identified as the last modifier.

Using the table above we are able to see that the Last Modified By metadata is stored within the `./docProps/core.xml` file, between the tags `<cp:lastModifiedBy>` and `</cp:lastModifiedBy>`. If we again examine the content of the core.xml file we can see that the current value for this tag is 'Christopher Sampson', highlighted in yellow : Listing 7.

Next we are going to use our text editor to change the field contents to something different,

Table 2. *Comparison of ExifTool output and actual metadata location*

ExifTool Output	Actual File Location	XML tags	Content
Preview Image: (Binary data 9500 bytes, use -b option to extract)	./docProps/thumbnail.jpeg	N/A	9,500 bytes JPEG Image
Title:	./docProps/core.xml	<dc:title>	N/A
Subject:	./docProps/core.xml	<dc:subject>	N/A
Creator: Christopher Sampson	./docProps/core.xml	<dc:creator>	Christopher Sampson
Keywords:	./docProps/core.xml	<cp:keywords />	N/A
Description:	./docProps/core.xml	<dc:description />	N/A
Last Modified By: Christopher Sampson	./docProps/core.xml	<cp:lastModifiedBy>	Christopher Sampson
Revision Number: 1	./docProps/core.xml	<cp:revision>	1
Create Date: 2013:06:09 10:34:00Z	./docProps/core.xml	<dcterms:created xsi:type="dcterms:W3CDTF">	2013-06-09T10:34:00Z
Modify Date: 2013:06:09 10:35:00Z	./docProps/core.xml	<dcterms:modified xsi:type="dcterms:W3CDTF">	2013-06-09T10:35:00Z
Template: Normal.dotm	./docProps/app.xml	<Template>	Normal.dotm
Total Edit Time: 1 minute	./docProps/app.xml	<TotalTime>	1
Pages: 1	./docProps/app.xml	<Pages>	1
Words: 4	./docProps/app.xml	<Words>	4
Characters: 24	./docProps/app.xml	<Characters>	24
Application: Microsoft Macintosh Word	./docProps/app.xml	<Application>	Microsoft Macintosh Word
Doc Security: None	./docProps/app.xml	<DocSecurity>	0
Lines: 1	./docProps/app.xml	<Lines>	1
Paragraphs: 1	./docProps/app.xml	<Paragraphs>	1
Scale Crop: No	./docProps/app.xml	<ScaleCrop>	False
Company: TRC Data Recovery Ltd	./docProps/app.xml	<Company>	TRC Data Recovery Ltd
Links Up To Date: No	./docProps/app.xml	<LinksUpToDate>	false
Characters With Spaces: 27	./docProps/app.xml	<CharactersWithSpaces>	27
Shared Do: No	./docProps/app.xml	<SharedDoc>	false
Hyperlinks Changed: No	./docProps/app.xml	<HyperlinksChanged>	false
App Version: 14.0000	./docProps/app.xml	<AppVersion>	14.0000

Listing 7. *The unmodified content of ./docProps/core.xml*

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<cp:coreProperties xmlns:cp="http://schemas.openxmlformats.org/package/2006/metadata/core-
properties" xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:dcterms="http://purl.org/dc/
terms/" xmlns:dcmitype="http://purl.org/dc/dcmitype/" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance"><dc:title></dc:title><dc:subject></dc:subject><dc:creator>Christopher
Sampson</dc:creator><cp:keywords></cp:keywords><dc:description></dc:description><cp:lastMod
ifiedBy>Christopher Sampson</cp:lastModifiedBy><cp:revision>1</cp:revision><dcterms:created
xsi:type="dcterms:W3CDTF">2013-06-09T10:34:00Z</dcterms:created><dcterms:modified
xsi:type="dcterms:W3CDTF">2013-06-09T10:35:00Z</dcterms:modified></cp:coreProperties>
```

Listing 8. *The modified content of ./docProps/core.xml*

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<cp:coreProperties xmlns:cp="http://schemas.openxmlformats.org/package/2006/metadata/core-
properties" xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:dcterms="http://purl.org/dc/
terms/" xmlns:dcmitype="http://purl.org/dc/dcmitype/" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance"><dc:title></dc:title><dc:subject></dc:subject><dc:creator>Christop
her Sampson</dc:creator><cp:keywords></cp:keywords><dc:description></dc:description><cp:la
stModifiedBy>Somebody Else</cp:lastModifiedBy><cp:revision>1</cp:revision><dcterms:created
xsi:type="dcterms:W3CDTF">2013-06-09T10:34:00Z</dcterms:created><dcterms:modified
xsi:type="dcterms:W3CDTF">2013-06-09T10:35:00Z</dcterms:modified></cp:coreProperties>
```

Listing 9. *Output of ExifTool after manual modification of a docx file*

```
chrissampson$ exiftool ~/Desktop/Sample\
Document\ Modified.docx
ExifTool Version Number      : 9.31
File Name                    : Sample Document Modified.
docx
Directory                    : /Users/chrissampson/
Desktop
File Size                    : 14 kB
File Modification Date/Time  : 2013:06:10
12:18:17+01:00
File Access Date/Time        : 2013:06:10
12:21:54+01:00
File Inode Change Date/Time  : 2013:06:10
12:18:51+01:00
File Permissions             : rw-r--r--
File Type                    : DOCX
MIME Type                    : application/vnd.
openxmlformats-officedocument.wordprocessingml.
document
Zip Required Version         : 20
Zip Bit Flag                 : 0
Zip Compression              : Deflated
Zip Modify Date              : 1980:01:01 00:00:00
Zip CRC                      : 0xbc27c2c7
Zip Compressed Size          : 251
Zip Uncompressed Size        : 735
Zip File Name                : _rels/.rels
Template                     : Normal.dotm
Total Edit Time              : 1 minute
Pages                        : 1
Words                        : 4
Characters                   : 24
Application                  : Microsoft Macintosh Word
Doc Security                 : None
Lines                        : 1
Paragraphs                   : 1
Scale Crop                   : No
Company                      : TRC Data Recovery Ltd
Links Up To Date             : No
Characters With Spaces       : 27
Shared Doc                   : No
Hyperlinks Changed           : No
App Version                  : 14.0000
Title                        :
Subject                      :
Creator                      : Christopher Sampson
Keywords                     :
Description                  :
Last Modified By             : Somebody Else
Revision Number              : 1
Create Date                  : 2013:06:09 10:34:00Z
Modify Date                  : 2013:06:09 10:35:00Z
Preview Image                : (Binary data 9500
bytes, use -b option to extract)
```

in this example I have chosen 'Somebody Else' as the string to insert. So now our modified file looks like this: Listing 8.

Now we need to zip back up the modified contents to a standard zip file. Mac OS X users should note that the file .DS_Store will be created at least once within the zipped structure. Windows and Linux users will not see this behavior. There are Mac utilities that prevent this from happening but we will not discuss those here.

Once our document has been zipped up again we will need to rename it to create our docx file. For this example the file was named Sample Document Modified.docx. once complete we can once again examine this file using ExifTool: Listing 9.

There are other ways to modify these files and even specialist tools designed purely for that task, however, we will not discuss these here as we have not investigated any of these tools.

As you can see from this rather basic example manipulation is relatively straightforward once you understand how it is recorded within a file. This also raises the question of identification of manipulated metadata.

Whilst a detailed discussion is beyond the scope of this article, we will make a few observations that may help you to determine your own process for ascertaining if the metadata has changed from that originally written to the file:

- Inconsistency: Our document shows Revision number 1 and having been created by 'Christopher Sampson', yet the Last Modified By attribute shows that 'Somebody Else' was the last to make a change. This should not be possible for a file that is at revision 1.
- Whilst there are exceptions and changes that can be made to file system metadata when a file is moved to a new location, in most circumstances the file system 'Modified' date would remain in sync with the 'Modify Date' from the file's metadata. As stated there are a number of exceptions to this and this cannot solely be relied upon.
- Many modern operating systems cache supported metadata and some offer file versioning, this may represent a very useful avenue of pursuit whilst trying to ascertain whether or not file metadata has been manipulated.

There are many other routes to explore and these will vary depending upon the operating system, file type and various other external factors.

SUMMARY

This document presents the reader with a very brief and non-specific overview of some of the types of metadata that are available. The aim of the article is to present the reader with an introduction to ideas and concepts for viewing, understanding and getting the most out of file metadata. The article is written as a guide only with the intent of promoting the benefits of understanding and accessing this information without a reliance on any one specific operating system or software application. By understanding the way in which binary files are created, updated and used the examiner is empowered with the ability to spot inconsistencies and reveal evidence that was not immediately available for a far wider variety of file types than the samples offered within this article.

We also aim to impress upon the reader that metadata, as with all data that resides upon a storage device, or within a file can be modified and 'spoofed' by those with the knowledge of the inner workings of the file type. In fact, this is so easy to carry out in many cases that verification and validation of any and all metadata that is extracted should always be carried out.

ON THE WEB

- *https://developer.apple.com/library/mac/documentation/Darwin/Reference/ManPages/man1/mdls.1.html* – Mac OS X developer Man page for the mdls tool,
- *http://www.sno.phy.queensu.ca/~phil/exiftool/* – ExifTool, a utility to extract, view and manipulate the metadata for many different file types
- *http://unixhelp.ed.ac.uk/CGI/man-cgi?file* – Unix 'file' man pages
- *http://www.exifviewer.org* – ExifViewer.org, free online image metadata viewer
- *http://www.exiv2.org* – Exiv2 library
- *http://www.foolabs.com/xpdf/* – XPDF, GPL open source utility for extraction of metadata from PDF documents, the site contains a number of pre-compiled binaries for different systems as well as third party links to other pre-compiled binaries.
- *http://trcdatarecovery.com* – Author's website

About the Author

Chris Sampson is director of UK based data recovery company TRC Data Recovery Ltd. Chris has worked within the area of data recovery for over 10 years, producing tools and techniques for the recovery of lost information from all manner of different operating systems and file types. TRC Data Recovery Ltd primarily provide data recovery services but also produce software tools for Microsoft Windows and Apple Mac OS X to aid in data recovery and related matters. Chris is actively involved in research and development projects based upon the indexing of file types for the purpose of examination, repair and retrieval of these items from deleted and otherwise missing states. Current projects include research into the recovery of fragmented multimedia and document files where no file system information relevant to the file's location or fragmented status exists. chris@trcdatarecovery.com | +44 114 241 9309

MALWARE ANALYSIS: DETECTING AND DEFEATING UNKNOWN MALWARE

by Kevin McAleavey, The KNOS Project

Cyber-attacks against control systems are considered extremely dangerous for critical infrastructure operation. Today, the protection of critical infrastructures from cyber-attacks is one of the crucial issues for national and international security. Over the past ten years, intrusion detection and other security technologies for critical infrastructure protection have increasingly gained in importance.

What you will learn:

- How to locate suspicious programs and how to determine if they're malware or benign
- Using file hashes in order to verify the origin and authenticity of suspicious programs
- How to locate the startups for malware, all of the associated components and locations where malware can hide
- How to locate rogue services and unkillable malware processes and regain access to the system
- Useful tools to aid in the diagnosis and mitigation of malware

What you should know:

- Familiarity with use of the REGEDIT Windows Registry editor
- Familiarity with use of the Windows CMD Command line
- Familiarity with the normal layout of a Windows file system

According to antivirus vendor Symantec, over 1 million new pieces of malware are created every day. In 2011, Symantec saw over 403 million new malware samples according to Kevin Haley, Director of Security Technology and Response at his "Symantec Security Awareness" presentation in October 2012. The samples received by Symantec and other antivirus and antimalware vendors are analyzed primarily by automated systems, and occasionally by human analyst's to become "signatures" placed in their detection databases on a daily basis.

Despite all these known samples, it is common for malware to slip right past security solutions undetected and unmitigated, leaving more system's infected with each passing day despite these efforts. Those of us who have been fighting the "war on malware" know full well that we've been losing the battle badly, and so do Information Technology Administrators and Managers.

When malware strike's a computer, the obligatory response by IT Department's is to remove the machine, wipe and reformat the hard drive, reload a fresh copy of the operating system and then reimage the drive, with the normal compliment of authorized applications and configurations, whereupon it is returned to the victim in a known and "trusted" state. While this causes great inconvenience to the victim, who has now lost whatever work that may have been on the machine, it is the only practical mean's to remove malware, given that the antivirus and antimalware industry has been losing the battle against malware for years now.

In some situations however, the importance of being able to collect evidence or critical data from a machine which has been infected with unknown and undetected malware, requires technician's with forensics experience to collect the valuable data and preserve it without spreading the malware on the infected machine further. It becomes quite challenging when malware has caused the machine of interest to become inoperable or inaccessible.

In this article, I will explain numerous ways by which malware can be located and circumvented sufficiently in order to be able to access and recover critical data where required. I caution in advance that being able to successfully restore the machine to operable condition does not mean that there isn't additional malware which remains hidden and continues to pose a risk. I will also refer to "security software" packages from various vendors generically as "antivirus" for the sake of simplicity.

Once a machine has been compromised it can no longer be "trusted", since it is impossible to be certain that any cleanup is ever complete. Attempting to access data from an infected machine must be carefully considered, as to its values and risk's, before proceeding.

Each type of malware is different, and will have different effects on different machine's. Some are easily located and mitigated, some are highly complex and difficult to disable, and some are downright destructive. Worse, there are so many locations in which they can hide within' the Windows operating system, protecting them from easy discovery.

The one thing that all malware have in common is finding a means to plant a payload onto the victim's machine, and then trying to remain hidden from detection. It is the payload that must be found, and in most cases, the file or means by which the payload was "dropped" onto the system in the first place, in order to prevent the payload from reappearing. Additionally, that payload may also consist of "helpers" in order to keep the malware hidden.

We will assume of course that all security and antivirus software is in place, and fully up to date on any machine in question. And in the case of a forensic's investigation that a proper image of the original content's of the machine has already been completed.

It is commonplace for even the best security software to fail to perform its duties and detect all intruders. Security software depends on specific "signatures" in order to identify malware, and changing just one digital bit in a malware file will cause it to slip right past ordinary antivirus/antimalware "signatures".

Malware author's carefully test their code in order to ensure that it does not match, and therefore " trigger" AV "signature's" before they release each version, in order to take advantage of this inherent design weakness. Thus, in order to find "unknowns" it is necessary to use the same skill set that malware analyst's use to hopefully locate and defeat them.

MALWARE TYPES

The computer security industry has a history of confusing definitions for the various classes of malware, often based on the limitation's of a particular vendors choice's or design limits in their coverage of malware. As a result, there is much confusion as to the specific meaning's of various definition's, and proper classification of specific threat's. Therefore, I'd like to explain what those of us in malware analysis labs define specific malware types as, and their expected capabilities, regardless of individual vendor naming policies. For those who are already certain of the definitions I'm about to present, please feel free to skip to the next section.

VIRUSES

Viruses are a self replicating program that infects individual files on a computer. They modify the contents of one or more file's, and usually hide inside legitimate files. In the old days, they would append themselves to the end of a file, making it larger than the original file. In more modern versions of Windows, compilers leave a lot of dead space inside legitimate files, making it easy for viruses to copy their payload into those empty spaces, thus leaving the size of the file itself unchanged and harder to detect. Many viruses will also avoid modifying the date and time stamp, further complicating forensic detection. They will change the entry point address in the file header to point to the viral code, so in order that this new code will run first before calling the original content's.

A practical way of checking on whether a file contains a virus is to use an MD5 or SHA1 "hash" against a known good copy of the original file. File hashing is the "secret sauce" of the antivirus industry, its how most AV signatures are created and how samples are managed and shared within the industry. This hashing method is extremely useful for efficiently investigating malware and will be detailed later in this article.

WORMS

Though closely related, worms and viruses are two entirely different types of malware. Both have the ability to self-replicate and propagate by attaching themselves to files although not all worms are "file infectors." However, while viruses copy themselves from machine to machine through media such as a USB device, worms replicate through networks directly. Worms can travel through the internet or local networks and inflict mayhem ranging from deleting files to creating backdoors or botnets that provide remote control of a system. Worms are designed to perform this function autonomously without human assistance, through network's or through physical media.

BOTS

"Bot" is derived from the word "robot" and are automated malware that controls network communications or services. Bots often automate tasks and provide information or services that would otherwise be conducted by a human being. A typical use of bots is to gather information, and then interact automatically with *instant messaging* (IM), *Internet Relay Chat* (IRC), or other web interfaces. They may also be used to interact dynamically with websites, and recent ones have featured highly customized protocols of their own.

Bots are self-propagating malware designed to infect a host and connect back to a central server or servers that act as a *command and control* (C&C) center for an entire network of compromised devices, or "botnet." With a botnet, attackers can also launch broad-based, coordinated "remote-control," and flood-type DDOS attacks against their target(s).

TROJAN HORSES

Trojan horse malware are programs which appear harmless, and may also appear to be useful and completely functional. However, they will contain other components which will install malware onto the machine while keeping the victim busy with a diversion, or appearing to fail, causing the end user to think the trojan was merely defective in some way.

They might arrive as an email attachment, or present themselves as a useful application on a website enticing the user to download and install them. Because it does not have the ability to self replicate, trojans are a completely different animal from viruses or worms. Trojans require human assistance in order to spread, usually through “social engineering” means. They often deliver destructive payloads and can also install other types of malware.

RATS AND “BACKDOORS”

RATs are “Remote Access Trojans” also called “Backdoors”, and is the choice for espionage and exfiltration of data. They provide a "Backdoor" into the system through which external actor’s can remotely control a computer, running other malicious code if he/she chooses. They can even use these hijacked systems called “zombies”, to launch attack’s on other’s. RATs provide all the capabilities of legitimate remote access tools, and then add numerous other capabilities depending on the design.

These are the preferred tools of cyber criminals, and APT operation’s, and are designed to remain stealthy when in operation and often install other malware components as part of the overall infection. Their primary purpose is remote control of the individually infected machines through these “servers” which can be remotely accessed manually. RATs differ from bots in that RATs must be connected to individually whereas bots normally communicate collectively with other bots or controllers autonomously as a distributed network. Some RATs also offer bot capabilities, thus blurring definition’s among some vendors.

SPYWARE

Spyware, also called “adware” by some vendors, is the most commonly found malware on machines. Spyware is any program that tracks and reports your computing activity without consent. While it is not designed to inflict damage, spyware can seriously affect the performance of machines. Spyware is often bundled with free software, and automatically installs itself with the program you intended to use. Sign’s of spyware include sudden modifications to your web browser, unwanted additional searchbar’s or “toolkits”, or redirects of your search attempts and the frequent displaying of pop-ups.

Spyware and “adware” are often defined interchangeably, but “adware” simply displays advertisement’s, whereas spyware returns more detailed information than just a cookie to the site, which displays the ad’s such as browsing history detail’s, or other personally identifiable information about surfing habits. Spyware should not be confused with more serious malware like keyloggers, RATs or other exfiltration tools, these are largely a privacy issue.

Many antiviruse’s do not detect all spyware, given that numerous purveyors of spyware will sue in order to have the "false positives" removed from detection databases in security products. When investigating strange, undetected program’s on machines, spyware is often found and can prevent analysis from continuing further when actual malware exists. Therefore, when spyware is found, diagnosis should not end just because some spyware was found on a victim’s machine, it is often a sign that the hunt has only begun.

KEYLOGGERS

Keyloggers are a particularly nefarious type of malware, in that their purpose is to surreptitiously collect critical information such as; logins, passwords or other sensitive data which is manually entered from a client machine. Keyloggers will then record all keystrokes entered on the victim’s machine into a file, which can be transmitted back to the source of the infection or to a botnet controlled by that source. Some more advanced keylogger-type malware will also index and collect for transmission, critical files on the victim machine or the network’s to which it connects as part of a larger espionage campaign.

DROPPERS

Droppers are a compressed package of malware. Their design dictates that they be as small as possible, in order to not raise suspicion of a surreptitious download in progress. They are often hidden inside email, document’s, or by other means and

are quite compact. Their only purpose is to gain entry into a system, and once installed they will download other component's quietly in the background, or they will contain various file's needed for a successful infection. Whereupon, they will install all of the other pieces that malware requires to carry out its function. Droppers are usually compressed with a specially obfuscated "wrapper" that is designed to elude detection by antiviruse's by means of encryption and other techniques.

ROOTKITS

Rootkits are the most difficult of all malware to detect, as their purpose is to completely hide malware from the user as well as security software intended to detect malware. Rootkits operate at the system level and are designed like other system and hardware drivers, using special code in order to hide the malware installed at the user level. Some rootkits are almost impossible to detect, even with forensics, and run as system services, device drivers, or can even be written to firmware in the system's hardware such as; video cards, network cards, BIOS, or any other device which allows updating of the hardware's "firmware." Their entire function is to hide malware, and the only visible symptom of their presence is unexplained reboots, or blue screen crashe's, if they have any bugs in their code. Absent programming bugs however, are extremely difficult to detect. Rootkits run at the lowest levels of a particular operating system for which they are designed.

BOOTKITS

Bootkits are the latest concern, although despite the hype, their practical application is still largely theoretical and impractical. However, when the specific hardware configuration in the potential victim's computer is fully known, it is practical to construct one. Bootkits consist of custom code designed to function within a system's "firmware" such as; BIOS, printers, video cards, ethernet and WiFi device's, and any other hardware that is capable of having its firmware "flashed" or upgraded. Bootkits are differentiated from rootkits because a bootkit does not require the services of an operating system they are completely self-contained independently of the operating system, whereas rootkits are designed to work after the OS has been booted.

As a computer boots, BIOS (or in newer machines, EFI or UEFI data on the local hard disk) contains pointers to the various boot firmware in modern components. By placing malware in the hardware itself, a "bootkit" can be started before the operating system and maintain control throughout a session. However, any such "bootkit" has to be written specifically for that hardware, and will not work on another version of that hardware, owing to the variety of hardware designs. The term "bootkit" is also often referred to as, "boot sector infectors" though they're still technically considered a "virus", and because they run independently of the operating system, antivirus software is highly likely to never detect them at all.

Bootkits can function regardless of the operating system that is booted, and therefore a very valuable tool for APT and other situations where the expense of such custom malware justifies the effort. It was one of many considerations in our own KNOS design despite the current rarity of this threat.

Bootkits can only be defeated by clearing and reflashing firmware with tools provided by the manufacturer of the hardware itself. Even then, the hardware should no longer be considered "trustworthy" since most "bootkits" are placed following the last byte of firmware code in the device, and many "reflash" tools fail to overwrite the infection.

PSEUDO-ROOTKITS

I coined this term back around 2000 with our BOClean product to describe perfectly legitimate software which has been installed onto a machine in order to act as malware. Pseudo-rootkits are legitimate program's which were installed without authorization and when audited, will turn out to be completely legitimate. They include things such as FTP software, Remote desktop software, keyloggers used by parent's and corporation's to monitor use of machine's, chat software, torrent software and numerous other "name brand" products that security software will not detect because it's a "legitimate tool."

However, they are not installed in the normal location where they would be installed legitimately and have been modified with other "legitimate tools" to remain hidden while they're in use by remote actors. Pseudo-rootkits will not be listed in "Add/Remove software" since they weren't installed by legitimate means. Our former BOClean product was the only security product which set off warnings when this was the case, warning the user that such was operating without their knowledge and gave them the option to remove them. Most security vendors will not even attempt to detect these at all even if they are "out of place."

EXPLOITS

Exploits are malware which are designed to exploit weaknesses and design errors in existing software or operating systems and are often embedded in web sites, documents and file servers which consist of scripts such as; Active X, Java applets, Javascript, PDF files and multimedia file's which will, thanks to poorly written code, perform unauthorized actions on the victims machine. Persistent exploits require that a copy be stored locally, usually in the internet browser's cache file's, or in the TEMP space. There are numerous other examples of exploits as well, which are triggered each time a malware page is visited.

WARNING SIGNS THAT UNDETECTED MALWARE MIGHT BE PRESENT

The majority of malware is poorly written and will often noticeably affect the computer. When malware fails to be detected by your antivirus or other security software, the following symptoms merit further investigation:

- Degraded computer functionality.
- Antivirus, firewall or other security software has been disabled.
- Odd behavior, such as unexpected reboots or icons no longer responding.
- Popup windows appearing when not browsing the internet.
- System errors, blue screens, or reliable programs suddenly crashing.
- Strange traffic on the network, as well as slow browsing.
- Disabled functions such as Task Manager, Regedit, User switching, login, logout or shutdown.
- Files or programs on your PC that you do not recognize.
- While surfing the internet, certain sites such as *www.microsoft.com* or sites with antivirus, or other security software vendors do not work.
- There are folders in your Windows Explorer, but clicking on them doesn't open them.
- After a reboot, Windows reports a Data Protection Violation in "Windows Explorer", and shuts down Explorer to restart it right away.

HUNTING FOR MALWARE

Unplug your network cable and manually turn your computer off. Reboot your computer into "Safe Mode with Command Prompt". As the computer is booting tap the "F8 key" continuously, which should bring up the "Windows Advanced Options Menu", as shown below. Use your arrow keys to move to "Safe Mode with Command Prompt" and press Enter key.

Make sure you log in to an account with administrative privileges (login as admin).

Once the Command Prompt appears you have only a few seconds to type in *explorer* and hit Enter. If you fail to do it quickly within 2-3 seconds, Malware on the system is likely to take over and not let you type anymore. Keep trying if necessary until you succeed.

If you managed to bring up Windows Explorer you can try to run System Restore with the following commands:

- Win XP: C:\windows\system32\restore\rstrui.exe and press Enter
- Win Vista/Seven: C:\windows\system32\rstrui.exe and press Enter

Follow the steps to restore your computer into an earlier day prior to the infection if you have any idea as to when it first started showing symptoms. System restore often disables recently added malware and lets you get back into the machine. However, a large amount of malware disables your ability to get at many parts of your system. If this is the case, read below where I describe how to re-enable deliberately disabled functions.

If you were successful, then you can proceed with hunting for the malware. A system restore will only hopefully prevent malware from starting. It will still be present on the machine, hopefully inactive at least temporarily in order to allow you to find and remove it.

Now the hunt can begin. This can become quite involved, but generally you'll want to proceed in this order:

- Clean all temp folders and internet caches from all browsers.
- Perform an Antivirus scan while running in Safe Mode.
- Audit running processes.
- Audit the 'startup' files and registry entries.
- Audit network connections.
- Check the file system for files that have been hidden and REALLY hidden files.
- Check for unusual services.
- Audit the HOSTS file and Windows TCP/IP Settings (redirects to incorrect sites are often done by modifying these).

If the above doesn't solve the problem, then it's time to dive into the system manually. This is the hard stuff, and requires an absolute need to access the data on the infected machine since it can become quite labor and time intensive now.

MALWARE HUNTING THE HARD WAY

Look for odd processes such as normal service names which are misspelled. svchost.exe is expected, scvhost.exe is NOT. Random character file names are almost always guaranteed to be malware, and they change each time the system is rebooted in most cases.

Another frequent indicator of malware are filenames with double extensions such as program.pdf.exe where the double extension serves to trick victims into believing that the file in question is something other than an executable like in this case where it might appear to be a PDF file if "show file extensions" is disabled in the file explorer view.

Incorrect icons for files are also a reason for further investigation. With "show file extensions" enabled in the file explorer, executables which display a file folder icon, or an archive file icon or perhaps a document icon are highly suspicious. Legitimate programs will provide an icon that shows a product logo or an icon logically associated with that program and won't mislead the user visually as to what the icon represents.

Another clue which should be followed is to check for valid signatures for any executable which claims to have originated from a large vendor. Most major enterprises now sign their code and the absence of a signature, or misspelled “properties” information is also an important clue. Here, examining the “properties” for unknown programs is quite useful.

Unusual processes with high CPU utilization with seemingly legitimate process names which are unexpected are also a sign of possible infection by malware. The Windows Task Manager is usually circumvented by malware to prevent malware from appearing in the task list, however you might get lucky and spot some this way.

Searching the web using process names you are unsure about will give clues to the legitimacy of individual processes. There are a few sites that provide databases which will give a detail listing of processes, including file sizes and verdict as to whether they're legitimate or suspicious.

Installing a tool which will allow you to take an MD5 or SHA1 "hash" of any suspect files will make searches for those processes a whole lot saner and will make it easier to confirm whether the files are legitimate or suspect. Most malware search sites will use MD5 and/or SHA1 hashes in order to confirm their verdicts and thus being ready to determine the MD5 or SHA1 hash of suspect files is strongly recommended to save time. These tools include:

- Microsoft File Checksum Integrity Verifier (generates MD5 and SHA1 hashes) which can be downloaded here: *http://www.microsoft.com/en-us/download/details.aspx?id=11533*
- I prefer this tool myself: HASHCHECK, which installs as a shell extension right into Windows' file explorer: *http://code.kliu.org/hashcheck/*

WHY DO WE NEED FILE HASHING?

As I indicated earlier, "file hashes" are, the "secret sauce" of the antimalware business. By performing an MD5 or SHA1 hash on a file, a unique long number is generated that is unique to that specific file. It's a means of generating a quick and dirty "signature" for a known file whether it's legitimate or it is malware. It permits security software vendors to feed a file to their signature database without so much as looking at a sample that's been flagged as malware. No effort, no time, put it in the blender, out pops a new "definition." It's cheap, it's fast and it's lazy for the security vendors.

The downside with using hashes for antivirus signatures is that if one single BIT of code changes, due to encryption, a different packer, even adding blank characters to the end of a file making it one byte larger, that hash will no longer match the "known malware" signature and thus it will go undetected until the new variant gets hashed and added to that signature database. THIS is why your antivirus failed you! They probably saw a different copy than the one that just got installed on your machine and the signature will only spot that other identical copy, not necessarily yours. This is also the reason why the "body counts" are so high for malware and so many variant versions of the same exact malware in their definitions.

Years ago, our operation made a product called "BOClean." We actually examined each and every piece of malware in memory and then created a memory-based definition for malware that would always detect the malware no matter how it was repacked, encrypted, polymorphed or modified since all programs must shed all that once they are ready to run in memory on a computer.

As a result of this design, we could detect any variant knowing that there is a very limited number of coders who produce malware and we made it a point to study the authors themselves rather than their code and zeroed in on specific means of detecting the author. As a result, we detected everything new that they wrote without the need to add individual definitions. Even in the most sophisticated of “APT” malware, the authors always manage to leave some unique “signature” in their work that can be used to detect their next move.

When I was last directly in charge of antimalware labs two years ago, the number of "unique authors" was in the vicinity of only about 1,500 coders. I'd bet that the number today is less than twice that. But my premise was that each individual author had their quirks, and you could count on those showing up in the code they released. It was the secret of our reputation with BOClean. But it took a lot of work in the face of an ever expanding number of samples. Today you only see the security companies expending that level of money and effort on the likes of Stuxnet.

Since file hashes are the industry standard though, that's what they use for sample and definition sharing within the industry, as well as databases of malware publicly available to the public which you can use to research whether or not unknown files are already known to malware analysts or other vendors in the industry. MD5 hashes are the older standard, SHA1 hashes are the currently popular exchange information. So you will want to obtain both MD5 and SHA1 hashes of any files which you suspect and then use those values to perform internet searches at places like virustotal, jotti, or similar to determine if a file is legitimate or suspicious depending on the results of your search. Doing those hashes will save you a _lot_ of time in your hunt!

RELEASE THE HOUNDS, ON WITH THE HUNT!

In addition to the task manager that comes with Windows, these tools give far more detailed information for you to examine them and are less likely

to be fooled into hiding malware unlike the Windows task manager:

- GUI Process Explorer from Sysinternals: *http://technet.microsoft.com/en-us/sysinternals/bb896653*
- GUI Autoruns from sysinternals: *http://technet.microsoft.com/en-us/sysinternals/bb963902*

Lacking those tools, the next step is to look for odd entries in the "startup" sections of the registry using Windows' built-in tools. Keep in mind that legitimate programs are seldom found in multiple start locations, such as run, `runOnce`, `runOnceEx` and runservices, but it's quite common for malware to copy its startups to multiple locations in order to ensure that if one is found, others will successfully start the malware each time the machine is rebooted. If you are unable to access the registry or built-in tools, or are unable to run any programs at all, I will explain down below how to get around disabled functions and services.

TO LOOK FOR THE NORMAL STARTUP LOCATIONS WHERE MALWARE MIGHT BE STARTED

```
winkey + R | msconfig | Startup Tab
```

or

```
winkey + R | regedit | check the following
```

Audit the 'startup' locations. The most commonly used startup locations in the registry are:

- `HKLM\Software\Microsoft\Windows\CurrentVersion\Run`
- `HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnce`
- `HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnceEx`
- `HKLM\Software\Microsoft\Windows\CurrentVersion\RunServices`
- `HKCU\Software\microsoft\windows\CurrentVersion\Run`
- `HKU\.DEFAULT\Software\Microsoft\Windows\CurrentVersion\Run`

AUDIT NETWORK CONNECTIONS

WINKEY + R | CMD | NETSTAT –NAO

Connections continually in 'SYN sent' state on 445, 139, or other established connections on odd ports, such as 6667(IRC) established connections on typical but unexpected ports such as FTP, TELNET, even 80(http) where inbound connections aren't normally expected are a sign of malware running surreptitiously. Inbound connections to websites and other services normally occur on ports higher than 1024, but outbounds on ports below 1024 are highly suspicious when a server is not being deliberately run on the machine in question.

A list of TCP and UDP ports and their expected purposes are listed here: *http://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers*.

Once again, sysinternals to the rescue: *http://technet.microsoft.com/en-us/sysinternals/bb897437*.

CHECK FOR HIDDEN FILES

There are files that are required for your system to run, many of these are hidden from accidental deletion. Malware takes advantage of such file attributes to hide from standard file searches, but the Windows folder view menu will allow you display any HIDDEN, SYSTEM and READONLY file settings on your system so that you can browse almost all of the files on the system being examined. There are also files known as "super-hidden files" which require the extra step of going into the View tab of the file explorer and specifically unchecking "Hide protected operating system files (recommended)" as well as ensuring that all other file types are made visible. Please be aware that malware can still hide their files from the File Explorer through the use of rootkits.

When performing this search pay particular attention to directories in the `%PATH%` variable such as C:\windows\system32, most malware tends to be placed in the system "path" environment setting and loaded via the registry without an absolute path. DO NOT delete any of the files listed by this command unless you are positive they are malware, you can easily hose a completely functional system by doing so. To delete files, you will need to UNSET hidden, system and read-only attributes first.

Be mindful also, that there are "super-hidden" files that Microsoft will just not let you access which begin with a $ sign as the first character of the filename. These are reserved for the system only and have been used to hide malware rootkits. One of the most infamous rootkits which I tracked down many years ago was the SONY rootkit, installed by DRM contained on their commercial music CD's. I wrote up a set of manual instructions for those who didn't use our BOClean product here: *http://www.dslreports.com/forum/remark,14817570*.

Malware can also hide in *Alternate Data Streams* (ADS), which hides files inside other files. One clever way of hiding malware, as well as purloined files in the act of espionage is to write the content's to an ADS, whereupon they're almost impossible to find without knowing the name of the ADS itself. These files are usually written to important system files and sometimes buried inside folder entries themselves rather than as files so as to elude detection. They will contain a colon (:) between two filenames such as for example winstart.exe:malware.exe or similar. The colon is the clue that you're dealing with an ADS file that will not appear in a directory listing. Most antiviruses do not look for these by default.

An excellent tutorial on how ADS works can be found here: *http://windowssecrets.com/top-story/hide-sensitive-files-with-alternate-data-streams/*.

And an even more useful description of what would be involved in deleting them can be read here: *http://www.bleepingcomputer.com/tutorials/windows-alternate-data-streams/*.

The following tool can locate them: Sysinternals Streams: *http://technet.microsoft.com/en-us/sysinternals/bb897440*.

Some malware will install as a system service. While removal of services is more difficult, you can usually at least disable them for temporary clean up:

```
winkey + R | msconfig | Services Tab
winkey + R | services.msc |
From the Command shell -> tasklist, taskkill,
tasklist /svc
```

Once malware has been identified, it's best to remove it while in safe mode. Some malware will have additional processes and DLL's (and possibly rootkits) that can prevent you from removing them by means of “injecting” their code into legitimate system processes. Deleting startup locations and files while in safe mode can usually restore a system to working condition but remember, there's no such thing as a "trusted machine" once it's been owned.

Listing 1. *Windows registry keys*

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunServices
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnceEx
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunServicesOnce
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunServices
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnceEx
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SafeBoot\Option subkey
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\*service* >ImagePath
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\Currentversion\explorer\User Shell Folders
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\SharedTaskScheduler
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Explorer\Browser Helper Objects
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Explorer\ShellExecuteHooks
HKEY_LOCAL_MACHINE\Software\Microsoft\Internet Explorer\Toolbar
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Active Setup\Installed Components
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Shell
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Notify
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Userinit
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\ShellServiceObjectDelayLoad
HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Run
HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\RunOnce
HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\RunOnceEx
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Shell Extensions\Approved
HKEY_LOCAL_MACHINE\Software\CLASSES\batfile\shell\open\command
HKEY_LOCAL_MACHINE\Software\CLASSES\comfile\shell\open\command
HKEY_LOCAL_MACHINE\Software\CLASSES\exefile\shell\open\command
HKEY_LOCAL_MACHINE\Software\CLASSES\htafile\Shell\Open\Command
HKEY_LOCAL_MACHINE\Software\CLASSES\piffile\shell\open\command
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager\BootExecute
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Terminal Server\Wds\rdpwd\StartupPrograms
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Shell
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Userinit
HKEY_CLASSES_ROOT\exefile\shell\open\command
HKEY_CLASSES_ROOT\comfile\shell\open\command
HKEY_CLASSES_ROOT\batfile\shell\open\command
HKEY_CLASSES_ROOT\htafile\Shell\Open\Command
HKEY_CLASSES_ROOT\piffile\shell\open\command
```

Finally, Rootkits: Rootkits are almost impossible to find although many aren't all that sophisticated. There are numerous tools which will attempt to detect a rootkit's files, but they're more prone to false positives than actually finding genuine rootkits. They include:

- Sysinternals Rootkit Revealer *http://technet.microsoft.com/en-us/sysinternals/bb897445*
- Hitman Pro (I recommend this comprehensive scanner which uses multiple AV engines to scan, but doesn't remove them) *http://www.surfright.nl/en*

Bear in mind that just because a "rootkit detector" fails to find any rootkits doesn't mean that there aren't any. Rootkits operate at the operating system level and can easily hide themselves completely since most security software operates at the user level, which is segregated from the operating system level. In Win7 and later, that isolation is even more profound than in earlier versions of Windows almost guaranteeing that rootkits will never be detected unless they're caught in the act of being installed.

LOCATING MALWARE WHICH MIGHT BE LOCATED IN STARTUP FOLDERS

The most ordinary autostart locations in Windows are in folders actually named "Startup." On all Windows computers, there is an individual Startup folder for every user (account) who has ever logged on in addition to a Startup folder shared by all users of a particular system. These folders can often be managed through the paths above or Windows Start button → All Programs → Startup → Right click. In XP they are:

- C:\Documents and Settings\<USERNAME>\Start Menu\Programs\Startup
- C:\Documents and Settings\Default User\Start Menu\Programs\Startup
- C:\Documents and Settings\All Users\Start Menu\Programs\Startup
- C:\windows\tasks

Whereas in Vista and later they've been changed to

- c:\ users\<USERNAME>\appdata\roaming\microsoft\windows\start menu\programs\startup
- c:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup

Startups can also be tucked away in these locations:

- C:\autoexec.bat
- C:\Windows\Win.ini
- C:\Windows\System.ini
- C:\Documents and Settings\Administrator\Local Settings\Temp\
- C:\windows\system32\
- C:\WINDOWS\Prefetch

In these file folders, if you opt to sort by date, you can sometimes see what has recently been touched, added or changed so long as the malware didn't change the particular file's time and date stamps. Malware will usually be listed with DLL, OCX, SCR, VBX, BAT, CMD or EXE file extensions, sometimes with random file names and in most cases, you should be able to search the internet for details by filename to determine if it's a known threat or not. This is when it's helpful to also know the exact file size in bytes as well as having an MD5 or SHA1 hash of the file in question to make matching your mystery easier.

If the filename doesn't turn up in an internet search, it's likely to be rogue, but again, you can't count on that either. Sometimes malware will use filenames similar to genuine system files such as rundll32, but will be located in the wrong directory. If you see something called rundll33 then it's a pretty safe bet that it's malware and should be deleted. System files are usually in SYSTEM32 and below, its presence in WINDOWS or TEMP is also suspect.

However, the vast majority of malware is started by entries in the Windows Registry, and here it gets dicey because there are just so many places that malware can be started from such as these particular Windows registry keys (see Listing 1).

The most difficult part of the registry hunt is that there are so many entries in each of them which culminates in an executable, DLL file or other library file being loaded and started. Unfortunately, each and every one has to be examined to determine whether it is legitimate or not. As I indicated at the outset, there are just too many places for malware to hide in the Windows operating system.

And these only account for EXE files. DLL's and other potential malware have other locations from which they can be started. Be particularly cautious when you spot a startup entry which begins with "RunDll" or "RunDll32" in front of a DLL, OCX or other file. RunDLL is Microsoft's "run a DLL as an executable" function which will allow malware to run as a library instead of an executable and thus loads it into the operating system itself instead of running it as a separate program. Any of those files must also be examined and verified as legitimate as well.

DLL's can also be started using entries called "ImagePath" or "ServiceDll" in the registry, and these are even harder to find because they're associated with "UUID" and "CLSID" entries buried in the registry. These can be searched for in the regedit utility and there are a LOT of them. These keys will directly call DLL's, OCX's and other library type files which can also contain malware.

To further complicate matters, Windows will not allow you to simply delete malware while it is run-

ning. Running processes and threads are protected by the operating system, and any attempt to delete it will be met with one form or another of an "access denied" message and any attempts to remove the rogue file will fail. The only way to delete a running file is to either kill the process (and even here, many protect themselves from this possibility) or to remove the corresponding startup entry in the startup folder and/or the registry and then reboot in hopes that there aren't other elements of the malware that will simply put the entry right back into the startup location and then restart after a reboot.

And it gets even worse with rogue DLL's and libraries which are used to "inject" code into other legitimate, running processes. Code injection has been the preferred method of infecting computers since DLL's cannot be killed, they must be "unhooked" by means of "object dereferencing" instead. The only other alternative to unhooking DLL's is to kill the process to which they've attached. Malware authors solved that problem as well by attaching them to numerous processes including core operating system processes themselves. Kill these processes and the machine remains infected, but spontaneously reboots or freezes in order to spank you.

Hopefully, the entire malware "system" is started by an executable which then loads its other components. If you find the main startup, you can hopefully defeat the rest of the chain upon a reboot by preventing the starter application from running. Pay particular attention to any startups however which include that "RunDLL" command in front of a library file - that may be the startup as well. If you can keep the rogue from starting after a reboot, that's the major part of the battle won!

SHUTTING DOWN ROGUE SERVICES

Since a Windows service can be turned off, Microsoft hasn't felt the need to let users delete services outright from the Services window. But services can cause all sorts of problems, whether they're unwanted add-ons to otherwise useful software, left behind by buggy uninstallers, or inserted surreptitiously by malware. So here's how to remove a service completely:

Open the Services window (services.msc) and double-click the service you want to remove. Highlight the text next to Service name (the first entry under the General tab) and press Ctrl-C to copy the name to the clipboard.

Next, open a Command Prompt window in Administrator mode and type the following at the prompt:

```
sc delete ("Rogue Service")
```

where (Rogue Service) (in quotes) is the name of the service you just copied. Press Enter, and if the removal was successful, you should see this message:

```
[SC] DeleteService SUCCESS
```

Return to the Services window and press F5 to refresh the list, and confirm the service is now gone. SC (or "Service Control") is an often neglected, but useful way to control running services, and can often stop rogue services.

UNKILLABLE PROCESSES

Antivirus and other security software require extraordinary means in order to kill rogue programs. Some methods involve the abrupt "TerminateProcess" function in code, while others require an even more extreme step in providing a kernel device driver that will literally unhook the file from the system and then delete it. Absent special utilities designed to terminate unkillable processes, or unhook injected threads, if the task manager doesn't allow you to kill a rogue program, or it keeps coming back after an apparently successful "kill" there are a few methods which I've found can do the job without these specially written utilities. They include:

Use the "ntsd" command to kill any process that thinks it's special and can't be killed using the task manager:

```
ntsd -p [pid] -c
```

If that fails, then try this trick using the following exact command line in a command prompt window:

```
reg add "hklm\software\microsoft\windows nt\
currentversion\image file execution options\
malware.exe" /v Debugger /t REG_SZ /d "C:\Windows\
System32\bogusfile.exe /F /IM malware.exe"
(the quotes are necessary)
```

Where the name "bogusfile.exe" is a nonexistent filename, what this does is force Windows to attach a debugger that doesn't exist and that will cause the program (named "malware.exe") to begin to start but not run because it's waiting for a debugger program that doesn't exist. A reboot should keep the program from running.

Another potential malware killer is "RKILL" which can be found here: *http://www.bleepingcomputer.com/download/rkill/*.

FINDING MALWARE

Many malware programs try to keep users and administrators from interfering with their operation and possibly shutting them down by disabling numerous programs such as regedit, task manager, and in extreme cases, not allowing you to run anything nor even shut down the computer. This slight of hand is accomplished by changing file associations in order to ensure that any time you try to start a program, the malware will be started first, and then call the program you wanted.

They will also use permissions to disable access to the system whereupon you will see a message indicating that the "Administrator" has denied access to you. One of the methods of complete denial to any resources is "Association hijacking" which can prevent programs from running as well as being yet another automatic startup source for malware.

To check for association hijacking, you'll want to check these registry keys:

HKEY_CLASSES_ROOT\.exe\PersistentHandler
- (Default) value should equal: {098f2470-bae0-11cd-b579-08002b30bfeb}

HKEY_CLASSES_ROOT\exefile\shell\open\command
- (Default) value should equal: "%1" %*
- IsolatedCommand value should equal: "%1" %*

Hijacking can occur for OTHER file associations such as HTML, PDF and many others. You will see those association keys in the same HKEY_CLASSES_ROOT areas as ".exe" and "exefile" but their entries can often be more complex. It helps when checking to have the same registry entry open for cross-checking with a machine that is uninfected to confirm whether or not to edit the entry.

A cute little brute-force method that also works when programs cannot be started is to copy the original utility file and then rename its file extension. For example, if you can't start any EXE files, try renaming the copied ".exe" to ".com" or ".scr" which are likely to work. If that gets regedit or your antivirus program running again, you're home free.

Some additional tips to work around disabled functions can be read here:

- *http://dottech.org/11980/re-enable-critical-windows-components-disabled-by-malware/*
- *http://malwaretips.com/Thread-Remove-malware-when-traditional-tools-fail*

PREPARING FOR THE INEVITABLE

Install these tools on your machines BEFORE you need them!

Sysinternals complete suite of system forensics utilities: *http://technet.microsoft.com/en-us/sysinternals/*.

Being prepared in advance to be able to generate MD5 or SHA1 "hashes" on suspicious files against "known good ones" or to identify specific malware in your searches with the Microsoft File Checksum Integrity Verifier is highly recommended: *http://www.microsoft.com/en-us/download/details.aspx?id=11533*.

Once again, I prefer this tool myself: HASHCHECK, which installs as a shell extension right into Windows' file explorer: *http://code.kliu.org/hashcheck/*.

Making a recovery USB stick - look for USB memory sticks that feature a physical write inhibit switch so they don't get infected too. Having all of your tools handy on a CD, DVD, or USB stick allows you to bring your arsenal to the machine in question without risking installing from the internet on an infected machine. The "write inhibit" switch on USB devices equipped with one will prevent any malware from the machines you're working on from spreading via that USB device. Of course, having a bootable CD or DVD is even better because there's no possible way to write to it at all.

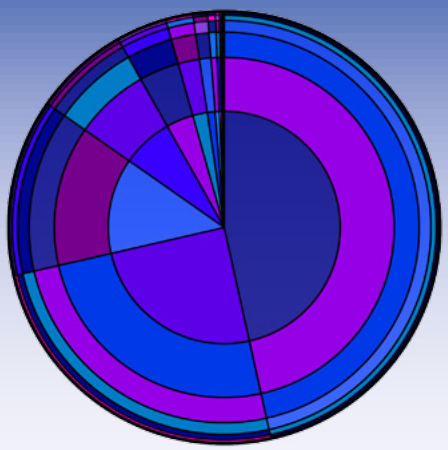
Most antivirus companies offer a downloadable "rescue disk" which you can download and burn to a CD, DVD or USB stick as well. Check with your security vendor to see if they have one available. The only downside is that each vendor has their own and therefore you depend on that vendor to detect and deal with any malware on the machine based on their detection database. And given that you'll be doing this only if they've failed, that might not help. If you're unable to boot the suspect machine, this will at least let you gain access to it.

In my situation, I use a custom build of our own product on a bootable USB stick, the KNOS Secure Desktop. On it I have all of the necessary tools, including antivirus, network and malware scanners as well as other tools that will allow me to investigate an infected Windows machine using KNOS itself in GUI mode. It even permits me to locate and examine super-hidden items such as the SONY Rootkit I mentioned earlier without difficulty. Some Linux live cd distributions will also suffice for accessing Windows machines although they have limits if there's no other means available to gain access to the infected machine.

Having the tools you need beforehand though cannot be emphasized strongly enough. Locating and defeating undetected malware is a formidable challenge, and I hope this lengthy dissertation has been helpful in proceeding with the task successfully. Good hunting!

About the Author

Kevin McAleavey is currently the co-founder of The KNOS Project, located in Voorheesville, New York (US) and serves as its chief architect. The KNOS Project manufactures a malware-resistant, secure desktop operating system based on BSD known as the "KNOS Secure Desktop" for use by client end users which delivers a familiar, user-friendly and functionally complete desktop environment that can be run from a DVD, USB memory sticks or installed onto a hard drive and ensures privacy and security in its operation. Since it doesn't use the host computer's hard drive, it can be run separately from the existing operating system with no risk or damage to the original desktop's contents, nor will it perform any writes to its host system.



The KNOS Project

The unique design of KNOS provides a secure "lockbox" which prevents tampering or abuse of any of its components and operates completely in memory so as to leave no forensically traceable contents when run on a computer. KNOS is delivered on read-only media with the ability to retain that full level of security when installed onto writable media. It is designed for full time desktop use on military, corporate and individual computers in order to properly secure client desktops with guaranteed "lockdown." It can also serve as a complete replacement of the existing Windows, Linux or OSX operating system already installed on the existing computer. We also work with OEM's if a factory install is desired on new equipment.

The KNOS Project also offers a prefabricated, fully complete desktop version with a complete suite of applications known as our "KNOS Secure Desktop, version 9" for use by the general public and is available for purchase directly from our site in both 32 bit and 64 bit versions. We also have an "Internet LITE" version which provides only internet surfing capabilities without the full complement of end user applications provided with our full version. The KNOS Project also provides a specialized version specifically for forensics use with a complete complement of investigative tools plus the capability of examining original evidence without any risk of alteration in order to quickly determine if further forensic investigation of a machine is warranted.

The KNOS Project's primary mission however is to deliver highly customized, specifically built versions of the KNOS system and user environment to exact specifications and requirements and can also provide a custom toolkit which permits external institutions the ability to develop their own secure desktop operating environment with our tools under their complete control, construction, and distribution independently. Our construction toolkit permits institutions to fully acquire and control the sources of all components which will result in the finished product under their complete control if preferred.

Prior to creating the KNOS Project, Kevin spent his career in the privacy and antimalware industry developing the first anti-cookie privacy software in 1997 known as NSClean and IEClean, with the first antitrojan software known as BOClean released in 1998. Prior to that, Kevin worked for the State of New York as an electronics engineer and network administrator for several state agencies. Kevin's experience in malware research and countermeasures goes back to 1985 and he remains fully active and current in antimalware research in order to ensure that the KNOS product cannot be impacted by viruses and other malware.

Kevin has also written numerous white papers and articles on computer security over many years and still serves as an expert technical resource in recent articles in computer security and other technical publications. He has also provided extensive technical consultatation to numerous government, military and law enforcement entities over the years, as well as assisting in investigations and prosecution as an expert witness. He continues to do so under the auspices of The KNOS Project.

THE INTERVIEW WITH

JAMES E. WINGATE

VICE PRESIDENT OF BACKBONE SECURITY

by Gabriele Biondo and Kishore P.V.

James E. Wingate, CISSP-ISSEP, CISM, CRISC, CHP, CSCS, NSA-IAM is Director of the Steganography Analysis and Research Center (SARC) and Vice President of Backbone Security. He is leading efforts to develop state-of-the-art digital steganalysis tools for use by digital forensics examiners and network security personnel in the public and private sectors. He is a member of (ISC)2, ISACA, HTCC, and HTCIA and regularly gives presentations on the threat from criminal or insider use of digital steganography to conceal evidence of criminal activity or steal sensitive information at major conferences across the United States. He retired from the US Air Force after more than 24 years of service as a Communications and Information officer. He holds a B.S. in Computer Science from Louisiana Tech University, Ruston, Louisiana, and an M.S. in Computer Engineering from the University of South Florida, Tampa, Florida.

In addition to being the leading provider of digital steganalysis tools, Backbone is a Payment Card Industry Data Security Standard (PCI-DSS) Approved Scanning Vendor (ASV) that conducts automated PCI-DSS compliance assessments with their industry leading 1Stop PCI Scan service. Backbone also provides real-time intrusion monitoring, vulnerability assessment, penetration testing, and business continuity and disaster recovery planning services. For more information: www.backbonesecurity.com.

The SARC is a Center of Excellence in digital steganography research and development within Backbone Security. The SARC has established the world's largest commercially available repository of digital steganography applications, fingerprints, and signatures and has developed industry leading computer forensics and network security steganalysis tools for detecting and extracting information hidden with digital steganography applications. For more information: www.sarc-wv.com.

Dear Mr. Wingate,
Thanks for giving us the time for this interview. I notice you are into Steganography, which is one of the most interesting ways to hide information. Can you please explain our readers what is it?

Throughout history man has sought ways to communicate secretly. One of the earliest methods for doing this was the use of wax tablets by the Ancient Greeks.

In 480BC, Demaratus used wax tablets to warn King Leonidas of King Xerxes I plan to lead his army into Greece prior to the historic Battle of Thermopylae. Because the danger of being discovered was great, Demaratus hid his warning by scraping the wax off the tablets and scribing his message directly onto the wood. Then he recoated the tablets with wax and sent the tablets via messenger to Leonidas. Interestingly, when the tablets were delivered, no one could figure out why that had received wax tablets with nothing written on them. According to *The Histories* written by Herodotus, widely acclaimed as the Father of History, Queen Gorgo, Leonidas' wife, said "If they would scrape the wax off the tablet, they would be sure to find the writing upon the wood." Thus, the warning was delivered but the Spartans got massacred at Thermopylae in one of history's greatest last stands.

Demaratus' use of wax tablets is one of the earliest and most widely known uses of steganography.

Steganography is derived from the Greek words "steganos", which means "covered" or "protected" and "graphein" which means "writing." When the two words are combined, the result is literally "covered writing" or "protected writing."

Essentially, steganography is a means of communicating secretly, or covertly. Here are a few interesting ways that steganography has been used in the past:

- The Chinese hid secret messages on slips of paper and baked them in moon cakes
- Mary, Queen of Scots, hid encrypted information in the bunghole of beer barrels
- Gaspar Schott hid information in musical symbols used to write sheet music
- George Washington used invisible ink to communicate secretly
- Microdots, the size of a period, were used in World War II to conceal information

For a comprehensive history of secret communication from Ancient Times to the present, the interested reader should read *The Code Breakers* by David Kahn.

In the Internet era, steganography has evolved from an analog form of information hiding to a digital form of information hiding. Accordingly, when talking or writing about steganography today, it is generally presumed the speaker or writer is referring to *digital* steganography.

Digital steganography is essentially about hiding a file in, or appending a file to, another file, called the *carrier file*, such that the carrier file is not altered enough to raise suspicion that something may be hidden within it or appended to it.

Other techniques for hiding information include a technique called *spam mimicry* where information is hidden by disguising it as spam (*www.spammimic.com*) or disguising the information as a nonsensical but often humorous one-act play as does Sam's Big Playmaker.

Another technique involves hiding information in the unused fields of communication protocols such as IPv4 and IPv6. A tool called v00d00n3t (VooDooNet) that hides information in unused IPv6 fields encapsulated in IPv4 packets was introduced at DEFCON in 2006. The tool effectively creates a tunnel for funneling hidden information through current generation network security appliances because most have yet to be programmed to inspect IPv6 packets.

Yet another very new technique is emerging. Information can be hidden in digitized voice streams generated by the growing number of Voice over Internet Protocol (VoIP) systems being deployed. Information is hidden by modifying the low order bits of digitized voice signals ever so slightly that the hidden information does affect the quality of the digitized voice signal.

How mature is nowadays the steganography technology? Can you please enlighten us about the state of the art?

Steganography technology is continually evolving. There exists a practically infinite number of ways to manipulate the 0's and 1's in any given file. Digital steganography researchers are diligently working to advance the state-of-the-art by finding innovative ways to manipulate bits. Some of the more advanced techniques are Cover Modeling, Quantization Index Modulation (QIM)-based Embedding, Spread Spectrum Embedding, embedding information in the unused fields of network protocols such as IPv6, and embedding information in executable files.

With the Operation Shady RAT attacks, we've now seen the first *steg bot* which exfiltrates information using steganographic techniques when instructed to do so by the Command and Control server.

What is the weakness in steganography techniques that help softwares detect them?

Many steganography applications leave meta-data in the carrier file along with the hidden payload.

The meta-data is typically used to indicate the byte offset where the payload begins and the length of the payload. The application uses the meta-data to extract the hidden payload. The meta-data can be used as a uniquely identifiable *signature* for the associated application that left the meta-data in the carrier file.

The challenge is to find the meta-data and possibly other information in the carrier file that may be needed to accurately identify the steganography application used to embed the hidden payload.

Steganography application signature discovery is a tedious, labor-intensive process that involves the use of a hex editor to compare a file, typically an image, that contains a known hidden payload with the identical file that does not contain the hidden payload. The steganalyst compares the two files to determine how the steganography application embedded the known payload. This process can take anywhere from a few hours to a month, or longer.

Who are using Steganography, nowadays? I mean, both "good" and "bad" guys. Is there any legal utilization, by the very moment? Is Steganography forbidden by the Patriot Act, or other US (European, International) laws?

As can be expected, both "good guys" and "bad guys" are using steganography.

Steganographic techniques are widely used in digital watermarking to embed identification information in files. So, the good guys use steganography to identify a variety of different types of files such as documents, images, audio files, and video files to determine the source or origin of documents or to protect the intellectual property rights of artists.

The bad guys are using steganography for many reasons. Insiders are using steganography to steal sensitive information such as intellectual property. Criminals, including terrorists are using steganography to conceal evidence of criminal activity such as child pornography, drug trafficking, weapons trafficking or to establish a covert channel for communication.

Digital watermarking is an example of "legal" utilization of steganography. We are not aware of steganography being forbidden by any US, European, or other international laws.

However, in the US, there are restrictions often placed on individuals convicted of certain crimes such as child pornography. In the US, individuals on parole or probation for a child pornography offense or certain other types of sexually oriented offenses, are often explicitly forbidden to use steganography. The rationale is the individuals could continue the behavior for which they were convicted by concealing images or other prohibited information, from their probation/parole officer.

We are interested in understanding how mature is Steganalysis (steganography analysis). What rate of success can be achieved? Given an encapsulating image, what is the likelihood to achieve the hidden data?

Given the large number of algorithms for detecting steganography developed by mostly academic researchers, a specific rate of success cannot be readily identified.

Steganalysis can be divided into two distinct areas each with a different approach to detecting and extracting hidden information.

The *blind detection* approach involves attempts to process suspect files to determine if information may have been hidden in the file without any prior knowledge of the steganography technique or program that may have been use or the payload that may have been hidden in the file.

The *analytical detection* approach involves detecting the fingerprints, or hash values, of file artifacts associated with the steganography application, detecting Windows Registry® keys or values associated with a particular steganography application, and detecting the *signature*, or meta-data, usually in the form of a hexadecimal byte pattern, associated with a particular steganography application. The main objective of analytical detection is to accurately identify the steganography application used to embed a hidden payload.

Many special purpose blind detection algorithms for detecting specific steganography techniques and/or for processing specific classes of file types such as images, audio files, and video files, etc. have been developed. The algorithm developers claim varying degrees of success when processing specific carrier file types to detect specific steganographic techniques.

The important thing to remember is that a general purpose steganography detection algorithm that can process a large variety of carrier file types to detect a wide range of steganography techniques has not been developed.

And, the result of running a blind detection algorithm, even "best in class" algorithms, is typically expressed as a probability that a hidden payload may exist in a given file. The main shortcoming of the blind detection approach is the probability of being able to extract the hidden payload is typically low even if the probability a hidden payload exists is very high.

On the other hand, the result of using tools based on the analytic detection approach is often the accurate identification of the steganography application used to embed the hidden payload and the capability to extract the hidden payload without having to use the application used to embed the payload to extract it! This is possible because a

very beneficial by-product of the research required to identify the meta-data (i.e., the signature) is the steps required to extract the hidden payload.

I notice you have been working with the NIST, in order to update their SP800-53. Besides being a huge achievement, how is working with the National Standard Institute?

It is very easy to work with the US National Institute of Standards and Technology Computer Security Resource Center to contribute to the development of Special Publications that address national level cybersecurity issues. Issuing drafts of publications under development for public comment is standard practice. It is highly noteworthy that Revision 4 of SP 800-53, Security and Privacy Controls for Federal Information Systems and Organizations, released on April 30, 2013, contains references to steganography in three separate security controls.

I notice you started the Certified Steganography Examiner certification. How recognized is it? Any plan for the future?

Our Certified Steganography Examiner (CSE)® course is a comprehensive two-day course designed to give students an understanding of the threat posed by the use of steganography to steal sensitive information or to conceal evidence of criminal activity in today's interconnected digital world. Students that pass the end-of-course examination receive the CSE certification from Backbone Security. Due to growing demand, we would like to create a computer-based version of the course in the future to make the course and certification available to more people around the world.

How harsh is, for attorneys, judges, juries, and law-people to understand Forensics, and, especially, Steganography? Do you have any interesting tip for us? Any story to tell us?

The legal community has become acutely aware of the value of digital forensics because practically every criminal investigation involves at least one computer or some other electronic device, and possibly several. Proof of this can be seen in the fact there are private sector digital forensics companies in the majority of countries around the world.

However, awareness of steganography and perception of the threat from use of steganography for nefarious purposes is not what it needs to be. Consequently, prosecuting or defending a case involving the use of steganography could be problematic.

A key precept of digital forensics is the preservation of evidence. This is typically accomplished

through creation of a bit for bit copy of the original evidence in order to show the original evidence has not been altered.

The key objective of digital steganography is to manipulate the bits in a file in order to hide information.

Thus, when presented with a case involving steganography, either the prosecution or the defense, or both, might wonder if the evidence was manipulated by the other side.

It will take some effort by the prosecution and/or defense to explain how digital steganography works so that it does not appear to be magic or a deliberate alteration of evidence.

We all know that the chain of custody has a key relevance when it comes to present evidences to the court. What is the way one should adopt to bring steganalytic evidences to a jury?

As alluded to above, it will take some effort to explain how digital steganography is used to manipulate the bits in a file. And, this would also place much more significance on proof of the chain of custody to remove any doubt that someone manipulated the evidence after it was originally collected.

What is the future of Forensics, in your opinion? Do you happen to see something like a merge between pentesting, forensics, and other related disciplines?

The future of digital forensics is exceedingly bright as more and more criminal and insider investigations will require use of digital forensics to find information of evidentiary value. But the bright future is accompanied by significant challenges in terms of the rapidly expanding volume of storage devices and the continually expanding number of mobile digital devices.

There is a convergence taking place between cyber security and digital forensics because it is more and more commonplace for digital forensics techniques to be used to either detect certain classes of attacks or to determine when and how a given cyber-attack took place.

Where is steganography going now? Is it for the good or bad?

Steganography is used for both good and bad purposes. Digital watermarking techniques are improving, which is good. However, the use of digital steganography for nefarious purposes is likely continuing to grow. No one really knows how much evidence of criminal activity is being hidden with steganography because hardly anyone is looking for it because they don't really believe anyone is using it because there's no large body of empirical evidence to prove that it is being used. All that exists is, for the most part, anecdotal evidence.

To state the obvious, that which is not looked for will never be found.

More proof that steganography is, in fact, being used by insiders to steal sensitive information and intellectual property and by criminals and terrorist to conceal evidence of criminal activity and to communicate covertly will not be available until attempts to detect use of steganography become much more prevalent.

Can you suggest ways for the everyday user how to suspect stegnographic trails without the use of software, if possible?

Well one way is simply looking at file and directory names. Many steganography applications have "steganography," or some variation thereof, in the file name.

Another way to suspect someone is using steganography is when they have multiple copies of the same image file with a slightly different name or a .jpg file that should be under 100K, for example, but the file size is multiple megabytes.

What would you suggest to a Forensics newbie? Any other suggestions for our readers?

Detecting the presence or use of steganography to conceal evidence of criminal activity and then attempting to extract the hidden information, if it exists, requires specialized tools and experience with digital forensics concepts.

Before attempting to perform steganalysis, I would highly recommend those interested first pursue a digital forensics certification such as the Certified Computer Examiner (CCE) certification offered by the International Society of Forensic Computer Examiners (www.isfce.com) or the Certified Forensic Computer Examiner (CFSE) certification offered by the International Association of Computer Investigative Specialists (*http://www.iacis.com*).

Lastly, I suggest that digital forensics investigators and examiners consider including steganalysis as a routine aspect of their forensics examinations. There could be information of evidentiary value hidden with a steganography application but it will not be discovered unless specialized tools are used to scan for the presence or use of a steganography application and attempt to extract hidden information, if any is detected, from suspect carrier files.

Thanks for your time.

Thorough . Insightful . Precise

The Global Leader in Digital Steganalysis

Are you equipped with the proper tools to detect data hiding applications and information concealed by steganography?

Computer Forensic Field Triage

NEW!

StegAlyzerFS
Steganography Analyzer Field Scanner

Perform rapid field triage for steganography artifacts and signatures!

Computer Forensic Lab

StegAlyzerAS
Steganography Analyzer Artifact Scanner

Detect files and registry entries associated with steganography applications!

StegAlyzerSS
Steganography Analyzer Signature Scanner

Detect files containing steganography and extract the hidden information!

Enterprise Network Security

StegAlyzerRTS
Steganography Analyzer Real-Time Scanner

Detect steganography artifacts and signatures in real-time over a network!

- Detect leakage of sensitive information and intellectual property through insider use of steganography
- Detect use of steganography to conceal evidence of criminal activity that would have otherwise gone unnoticed
- Enforce organizational policy prohibiting insiders from having or using steganography or other data-hiding applications
- Discover evidence of covert communications
- Detect artifacts associated with data-hiding applications using the largest commercially available steganography application hash set
- Detect signatures of steganography applications
- "Point-click-and-extract" capability to quickly and easily retrieve hidden information

SC MAGAZINE AWARDS 2008 NOMINEE Honored in the U.S.

EDN INNOVATION Honoring Excellence in Electronics FINALIST 2010

Discover hidden evidence that other tools miss!

BACK BONE SECURITY

STEGANOGRAPHY ANALYSIS AND RESEARCH CENTER
SARC
RAISING THE THRESHOLD OF PERCEPTION

877-560-7272

www.sarc-wv.com

STEGANOGRAPHY: THE ART OF HIDDEN DATA IN PLAIN SIGHT

by Priscilla Lopez

Steganography is the art of hiding messages in plain sight. Different forms of steganography have been used for many years throughout history. Nowadays just about any data type can be embedded with a secret message and the common passerby wouldn't even notice.

What you will learn:

- What is steganography.
- The difference between cryptography and steganography
- Techniques to obscure the data.
- Types of detection techniques used in the discovery process.
- A recent criminal case involving the use of steganography.
- Historical examples of steganography.
- How to detect a file.
- How to extract internal hidden data.

What you should know:

- Be familiar with common graphics terminology.
- Understand the concept of a bit.
- Know what cryptography or encryption is.

Steganography is the art of obfuscation. Criminals can hide *small* messages, files, and other data *inside* plain files. These plain files are commonly images, although steganography can be used in video files, audio files, emails, unallocated hard drive spaces and much more. These files can be easily communicated using the common means we use to transfer data today. A criminal can embed a hidden text file with account numbers or passwords into an image, put it on a thumb drive, go to an internet café and email the image to another criminal. No one would think twice about emailed images, Figure 1, because it's done every day. It is very difficult to detect steganography unless you suspect it. One possible cause of suspicion is a file seemingly not the right size for the media. Like a small pixel image that would normally be 20 kilobytes is seen as 400 megabytes. However, this is not common in steganography because normally the differences in a normal file and a steganography file is usually miniscule. Luckily there are tools that are available that can search a hard drive and detect these files. Another cause for suspicion is a distorted appearing image such as in Figure 2. Many times criminals will try to hide a large file in an image and the image becomes distorted.

Steganography should not be confused with cryptology. Cryptology involves actually changing the data into an unreadable format unlike steganography, which is hiding in plain sight. On that note, data to be hidden can be encrypted with a tool such as TrueCrypt then embedded with another tool such as OpenStego therefore combining cryptography

with steganography. This makes it much harder for the investigator to extract the file.

AN ANALOGY

An analogy of the concept is steganography is like trying to view a 3-D image like the ones in The Magic Eye books and looking for the hidden image in the pattern. These books, originating in the 90's, asked the reader look at a distant object and then attempt to view the hidden 3-D image. In Figure 3, if you attempt to look at a distance object and then bring your eyes back the image almost focusing through the image after several attempts you may or may not see a raised 3-D image. Figure 4 demonstrates what the raised 3-D image embedded in Figure 3 would look like. The same concept applies. A normal person may just see a normal image but the receiver knows to focus their eyes on the image to reveal the hidden message.

TECHNIQUES FOR OBSCURING

According to EC-Council, the most common techniques to obscure data, used by tools, are least significant bit, algorithms, filtering and masking.

Figure 1. *Original image on top left, suspect file on top right and Suspect file zoomed 400%. Adapted Image from http://imrannazar.com/*

When the suspected files are created by certain tools the file creates a sort of signature as part of the file. The signature is detectable by steganography detecting tools.

- In the least significant bit method, some of the original files bits are replaced by the bits of the file to be hidden. If the original file is `1100 1001 1000 1111` and the data to be hidden is `1001` then the new file would be `1101 1000 1000 1111`.

Figure 2. *Image distorted. Adapted from http://instructables.com*

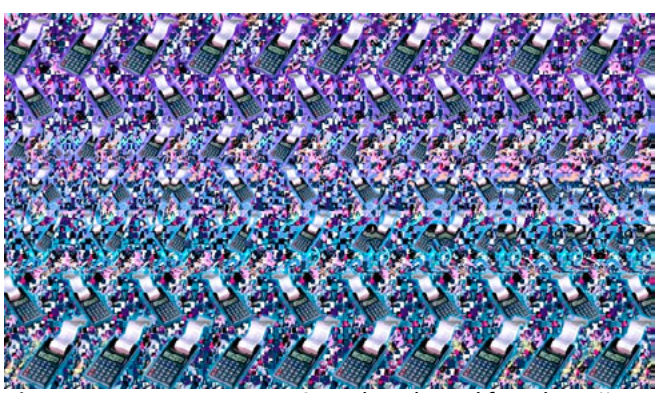


Figure 3. *Magic Eye Image Sample. Adapted from http://www.magiceye.com/*

Figure 4. *Magic Eye Image Solution Sample. Adapted from http://www.magiceye.com/*

The bold represents the last bit replaced by the bits from the file that are hidden (Figure 5). A sample tool that does this is Steganography Studio.

- The filtering and masking method is much like using watermarks, but the image is not compressed, cropped or processed. The image is clearer and less detectable (Figure 6). A tool that uses this method is Red JPEG.
- Mathematical algorithms are used to compress the original file in order to embed data. In order to detect an attack, the investigator would need to determine how to analyze the file or use several tools to analyze the file. A tool that uses this message ImageSpyer G2.

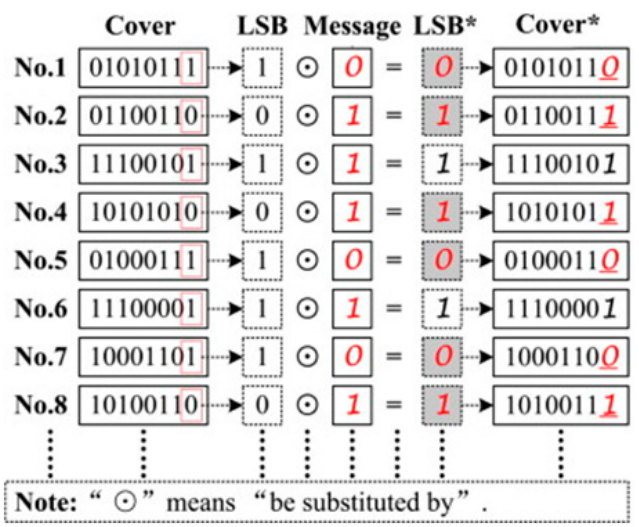


Figure 5. *Sample Least Significant Bit process. Adapted from http://www.sciencedirect.com/*

Figure 6. *Watermarking process. Adapted from http:// bit599.netai.net/*

TYPES OF DETECTION USED IN THE DISCOVERY PROCESS

According to EC-Council, there are several processes that can be used during an investigation.

- In *Known-message*, only embedded file is known, the suspected file needs to be found.
- In *Known-cover* only the original file and suspected file are known.
- In *Known-stego* the algorithm or tool used to embed the file and the suspected, hidden and original files are known.
- In *Stego-only*, only the suspected file is known.
- In *Chosen-stego* the tool used and suspected file is known.
- In *Chosen-message* a wanted message and a tool is used to create the steganography file. The investigator looks for the same patterns of the steganography file to look for other suspected files.
- An investigator can also modify and/or manipulate the suspected file in an attempt to extract the data.

CRIMINAL CASE USING STEGANOGRAPHY

Chet Hosmer, co-founder and chief scientist of the company that created the steganography tool StegoHunt, in 2009 stated that there are more than 1,000 steganography programs available for free. This makes it easily accessible for terrorists and spies to communicate illegal activity such as in the legal filings by the US Department of Justice on June 28, 2010. An 11-person Russian spy ring living in America was charged with conspiracy. For many years the ring attempted to develop relationships with key individual to send back intel and secrets to Russia. This case is an example of how criminals can use steganography and can communicate undetected for years.

HISTORY OF STEGANOGRAPHY

The concept of steganography have been around for many centuries. A few common historical examples are a person would be shaved and then tattooed with a message on their scalp. The hair would grow out and that individual would be sent the message receiver. The receiver would shave that individuals head to reveal the hidden message underneath. Another historical example is invisible ink. The ink would be written on an inconspicuous letter, allowed to dry then sent to the receiver. The receiver was the only one who would know how to view the invisible ink and reveal the message. Microdots, as seen in Figure 7, were used in wars, relayed and then the dots were viewed under a magnifying glass, to review the message to the receiver. Other common examples used in history are pictures when viewed at different angle reveal different images and

words used in the paragraphs from a normal handwritten/typed letter revealing a message that only the receiver would know to look for. All these historical methods all have one thing in common. There is a message hidden within a normal appearing media that a common viewer would not find suspicious and only the receiver would know to look for.

HOW-TO DETECT

- Step 1: Download X-Steg from *http://www.outguess.org/download.php* as seen in Figure 8.
- Step 2: Download and save a few sample files from *http://datahide.com/Bpcse/stego-images1-e.html* and any other searched website. Make sure to download "k-webpage.emb.jpg" to be used in the next section. Save to the folder of your choice.
- Step 3: Click on File then browse to the folder you saved the images. Then click on Open to start the scan.
- Step 4: The window displays files that were detected by Xsteg.

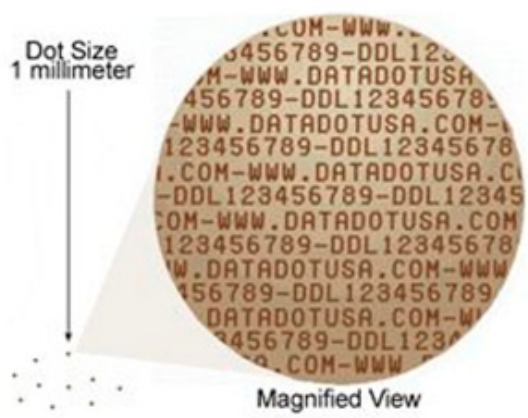


Figure 7. *Adapted from http:// carinsurance.arrivealive.co.za*

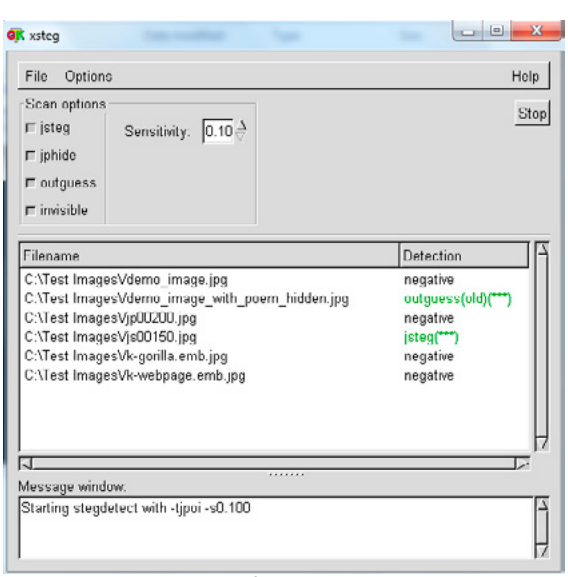


Figure 8. *Xsteg, a tool for detecting steganography*

Xteg did not detect the file "k-gorilla.emb.jpg" used in the next section but did detect other files due to the fact that the purpose of this tool is to only to detect jsteg, jphide, outguess, invisible signatures. Proprietary tools search for many more types of signatures.

HOW TO EXTRACT EMBEDDED FILE

- Step 1: Download qTech Hide and View from *http://datahide.com/BPCSe/index.html*.
- Step 2: Start the program.
- Step 3: Click on Information Extracting (Figure 9).
- Step 4: Click on Select Embedded image (Figure 10).
- Step 5: Navigate and open the image "k-webpage.emb.jpg" downloaded from the previous section.
- Step 6: Click on Access Key. The Access Key for this image is "k-webpage.emb.jpg". Other suspected file may or may not have a key or password. If you do not know the password you will need to crack it, which is beyond the scope of this article.
- Step 7: Enter the Access Key and click OK.
- Step 8: The extracted file pops up in a temp file created by qTech Hide and View as shown in Figure 15.
- Step 9: The extracted file can be saved and opened for further analysis or collected as evidence as displayed in Figure 16.

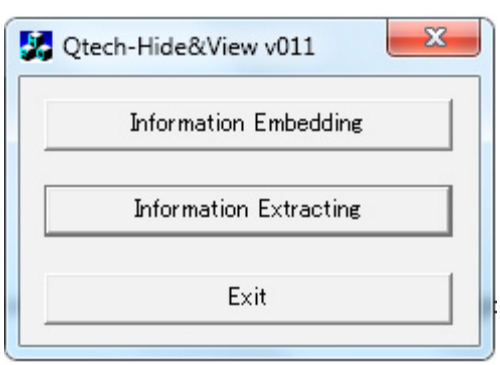


Figure 9. *qTech Hide and View Tool Menu*

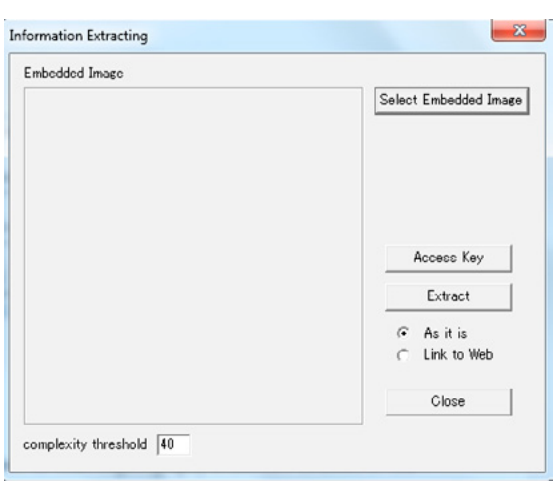


Figure 10. *qTech Hide and View Information Extracting Window*

ON THE WEB

- *http://datahide.com/BPCSe/index.html*, Site contains tool list, test images and detailed explanations used in this article.
- *http://www.garykessler.net/library/fsc_stego.html*, Site contains popular forensics author and professional information, links, tools and more.
- *http://www.cs.bham.ac.uk/~mdr/teaching/modules/security/lectures/Steganography.htm*, Site contains good explanations.
- *http://en.wikipedia.org/wiki/Steganography_tools*, Site list of a few tools.
- Information Hiding: Steganography and Watermarking : Attacks and Countermeasures By Neil F. Johnson, Zoran Đurić, Sushil Jajodia.
- *http://www.jjtc.com/Steganography/stego.html*, Site list of a few tools and explanations.

The file that is embedded is a text file containing 3 kilobytes of data. The image file "k-webpage.emb.jpg" is 11.9 kilobytes and 388 x 342 pixels.

SUMMARY

Hopefully you have learned the basics of what steganography is, how to detect it and how to extract data from suspicious files. The take away from this is to remember that files can be hidden into plain files unnoticeable to the public but retrievable by the recipient. Steganography is not illegal but can be part of a criminals arsenal. Practice using other tools and embedding different files into different media, then email to a friend. These skills will come in handy during an investigation.

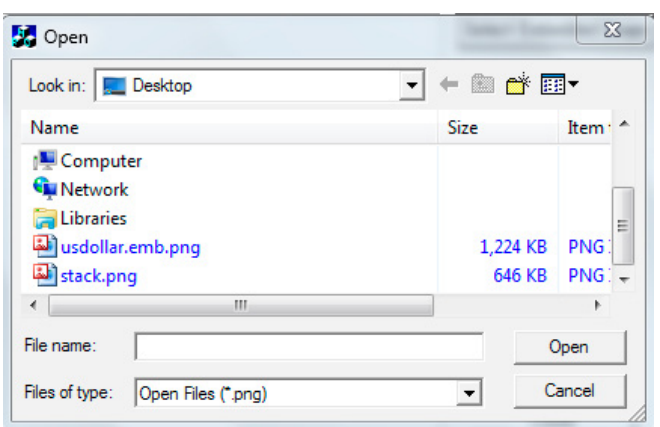


Figure 11. *Browse for the file to extract*

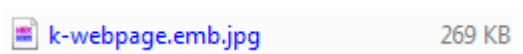


Figure 12. *Suspicious file*

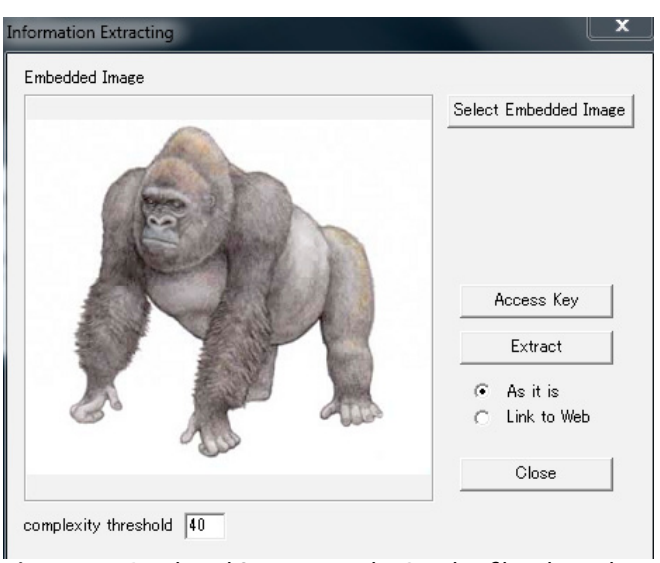


Figure 13. *Displayed Suspicious File. Graphic file adapted from http://datahide.com/BPCSe/index.html*

About the Author

Priscilla Lopez has earned M.S. in Information Security Assurance from WGU and B.S. in Computer Information and Technology with Minor in Business from UMUC. She holds five active computer certifications: CCNA, GIAC ISO Specialist, CEH, CHFI and CompTIA Network +. For over ten years she has been working with technology in her community, workplace, family and church. She is continuously expanding her knowledge and experience in the computer industry and enjoys sharing with students and those around her.

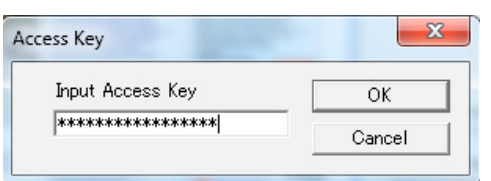


Figure 14. *Key Entry Window*

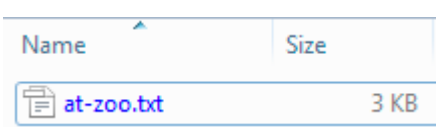


Figure 15. *File and Size of Extracted from "k-webpage.emb.jpg"*

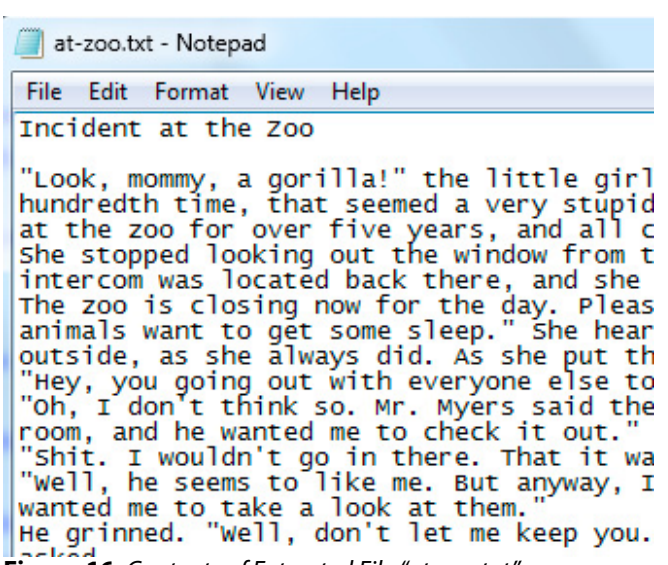


Figure 16. *Contents of Extracted File "at-zoo.txt"*

TrustSphere

Global Reputation

TrustCloud

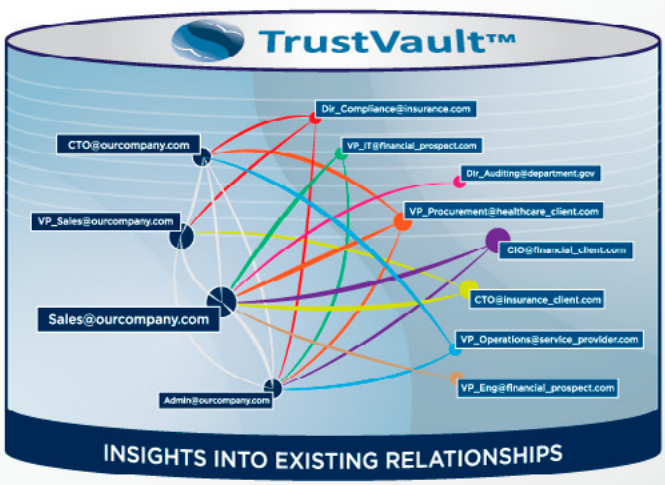
Industry's Most Comprehensive Real Time Dynamic Reputation List

+

Local Relationships

TrustVault™

Restoring Security, Integrity & Reliability to Messaging Systems



TrustSphere
Tel: +65 6536 5203
Fax: +65 6536 5463
www.TrustSphere.com

3 Phillip Street
#13- 03 Commerce Point
Singapore 048693

DIGITAL IMAGE ACQUISITION – STEP BY STEP

TOOLS AND TECHNIQUES FOR BEGINNING THE DIGITAL FORENSICS PROCESS

by Thomas Plunkett, CISSP, EnCE, MSIS

Proper digital image acquisition is key to any forensics practice. Accurate and thorough documentation along with rigorous adherence to procedures and established best practices lead to a successful acquisition process. This article will help the beginner learn what is necessary to successfully accomplish this important part of digital forensics.

What you will learn:

- General forensic documentation guidelines
- General digital image acquisition steps
- Digital image acquisition using FTK Imager

What you should know:

- Basic knowledge of computers
- Basic knowledge of disk drive interfaces
- General interest in digital forensics

If you are just getting started with digital forensics or even just have a passing interest in the subject, you may be wondering exactly where to get started. The answer is with proper image acquisition. Digital Image Acquisition is the process of identifying and documenting original physical evidence such as a hard-drive or computer and then preserving the digital evidence stored on the physical evidence.

Proper image acquisition is the foundation of the digital forensics process. The success of a digital forensic investigation is fully dependent on the proper acquisition and documentation of the digital evidence. If the acquisition is done improperly or is poorly documented, any evidence derived from the digital image may come under undue scrutiny, have its validity questioned, or even be completely disregarded or dismissed. This article is intended to provide new forensic examiners with the knowledge necessary to perform proper digital image acquisition.

DIGITAL IMAGE ACQUISITION

What is a digital image? A digital image in forensics, not to be confused with a photograph taken with a digital camera, is a bit for bit duplicate of the data stored on a given piece of digital media. It is verifiable, meaning that other people can look at the digital image and say without a doubt, “yes, this is the same data as on the original”. It can hold up in court as unaltered evidence. As you can see, the process used to acquire digital evidence can be quite important.

The process of digital image acquisition can be broken into six basic steps:

- Documentation of the acquisition process,
- Identification of physical evidence,
- Preservation of physical evidence,
- Documentation of physical evidence,
- Preservation of digital evidence,
- Documentation of digital evidence.

DOCUMENTATION OF THE ACQUISITION PROCESS

Note that of the six steps above, three begin with "Documentation". Documentation is the forensic examiner's number one tool and is the primary part of the acquisition process that can make or break a case. This is true even for the overall process.

Whether you work alone or for a forensics firm, your acquisition process should be documented to show the standard steps an examiner would take during a normal digital acquisition. The documentation should cover at a minimum:

- chain of custody,
- standard tools to be used,
- number of images to be made,
- whether or not encryption will be used and what type,
- the type of image to be created,
- what will be documented and how,
- naming conventions.

The process documentation can be as simple as a Word document that is maintained by the forensic staff. It should be tested to make sure that all general activities are covered and all examiners in the firm should know and follow the process. It is possible, and I have personally seen it before, for the process documentation to be reviewed by either a client or opposing party to make sure you are following your own processes.

Deviations from a documented process tend to make people upset but it is impossible to foresee all future situations you may encounter. So, make your process documentation flexible enough to allow examiners to document any necessary deviations to the process in case they have to improvise.

IDENTIFICATION OF PHYSICAL EVIDENCE

Identifying physical evidence may seem self-explanatory, and it is to a degree. However, consider a case I had recently in which I needed to get an image of a smart phone. I have a tool called a CelleBrite that allows me to plug in most any phone and preserve a digital image of the phone. I knew that the phone had 16 GB of storage space but my image kept ending up at less than 1 GB. So I read up on the device and found that a 16GB MicroSD card is mounted to the main board of the phone, inside the case.

Once I opened the case and removed a few pieces and some tape, I was able to get to the MicroSD card, remove it, document it as a new piece of physical evidence, and acquire an image of it. Had I not gone through the trouble of researching the phone specifications, I may have very well missed a key piece of evidence.

So, the identification of physical evidence involves understanding the devices you receive and finding other relevant devices. Ask questions like: Does it have more than one hard drive? Is there a CD-ROM in the CD player? Is there a flash drive plugged in to the device? Are there any storage devices such as a backup drive connected or near the evidence? You get the idea. There may be a drawer full of CDs that were used to backup important information just sitting there out of sight.

DOCUMENTATION OF PHYSICAL EVIDENCE

Once you identify a piece of physical evidence it needs to be documented. This means to label and photograph it and enter important information about the evidence into an evidence form. Important information to track for any piece of physical evidence is:

- case name,
- physical evidence number,
- date it was identified or date which you received custody,
- who identified or received it,
- who it was received from,
- location where it was identified or received,
- owner or custodian name,
- type of device (workstation, laptop, phone, flash-drive, etc.),
- manufacturer,
- model,
- serial number,
- asset tag,
- service tag,
- notes (like "found with giant scratch on the side" or "powered off by pulling power cable")
- BIOS information such as date and boot order

For the labels, I like to include case name, case number (if applicable), custodian name, evidence number, and current date. Write it all on a Post-it note and take pictures of the device with the label visible. Take a picture of the overall device, each side, any identifying tag or number, any external defect or damage, then in the case of a workstation, open it and take a picture of the inside with particular focus on the placement of hard drives.

Once the physical evidence is thoroughly documented, create a more permanent label with the same info as is on the photo label and affix it to the device. You'd be surprised how many Blackberries, iPhones, or Dell Laptops you end up storing. They all look alike and the only thing you have left to easily distinguish one from another will be the tag you just created.

PRESERVATION OF DIGITAL EVIDENCE

Everything discussed up to this point is done to support the preservation step. Just to be clear, acquisition or imaging is the process used to preserve digital evidence and I use the two terms synonymously.

Once the physical evidence is preserved and documented, hard drives can be removed, labeled, photographed, and imaged. Similar documentation must be done for the hard drives from a computer as we did for the actual computer. For now, make sure that you create a Post-It label with case name and number, custodian name, evidence number (the first drive in a computer that is evidence number 1 may have evidence number 1A and the second 1B), date, and examiner name or initials. Then, photograph the drive with the label next to it so the make, model, and serial numbers are all visible. More documentation will be done later.

There are two basic methods for obtaining a digital image, static acquisition and live acquisition. Static acquisition is the used whenever the original evidence can be powered off. It typically involves the removal of a digital media from a computer and that media being connected to an imaging device.

Live acquisition is used in cases where a device cannot be powered-off or the media cannot be removed from the computer. It is typically used when acquiring an image of a production server or a remote networked computer.

There are two types of digital acquisition, logical and physical. Logical acquisition involves the preservation of active content (as opposed to deleted content) stored on digital media. For instance, you may have several folders stored on a flash drive but you only need to preserve the “Secret Documents” folder and the contents therein. This is known as logical acquisition.

Physical acquisition is the process of preserving all content, active or deleted, that is stored on digital media. Every bit on every sector of the drive is preserved exactly as it is on the original media. The end result is what most people know as a digital image or just an image when talking to forensics people.

One key to creating a successful image is to always protect the original media. This means that the physical hard drive, flash drive or what have you is handled with special care. It is never powered on without write protection. Write protection can be accomplished with either hardware such as a Tableau T35 bridge or software such as System Acquisition Forensic Environment (SAFE) Block XP.

To begin the acquisition, you will need the following tools and equipment:

- original media (source or suspect),
- two (preferably blank) target media with more free storage capacity than the original media,
- one write blocking device or software that works with the original media,
- one imaging device or computer with imaging software,
- appropriate cables to power and connect the media to the imaging device or computer.

The basic steps for the preservation of digital evidence when using a computer as an imaging device are:

- Write-block the original evidence and connect it to the computer
- Determine the physical disk identifier of the original evidence
- Attach the target media to the computer
- Using forensic imaging software, create digital image of original evidence on the target media
- Verify by hash that the digital image matches the original evidence

One final point to note here is that not all disks are in perfect condition. An examiner will occasionally come across damaged, unreadable, or inoperable disk drives. In minor cases, such as a bad sector error, most imaging tools will continue to acquire an image by either skipping a bad sector altogether and padding the image with zeroes, or re-reading the bad sector a fixed number of times until it either gets data or skips the bad sector. Some devices like the ICS ImageMaSSter have the option to skip bad sectors altogether or stop the imaging process when it encounters bad sectors.

In the case of an inoperable or unreadable disk, the disk can often be repaired, read, and imaged by trained technicians with a clean-room facility and specialized tools.

DOCUMENTATION OF DIGITAL EVIDENCE

While I have documentation of the digital evidence listed as the last step, it can mostly be done while the image is being created. I use and suggest a spreadsheet, one per case, which can be combined with the physical evidence documentation and has the following columns:

- case name
- case number
- digital evidence number
- physical evidence number
- owner or custodian name
- type of device it came from (workstation, laptop, phone, flash-drive, etc.)
- physical evidence manufacturer
- physical evidence model
- physical evidence serial number
- physical evidence asset tag
- physical evidence service tag
- digital evidence manufacturer

- digital evidence model
- digital evidence serial number
- storage capacity in gigabytes
- number of sectors
- hash value of image
- type of image
- tool used to create the image
- tool used to verify the image
- date and time the image was created
- person who created the image
- OAD and WD inventory or serial number
- OAD and WD storage capacity
- date and time image was transferred to OAD and WD
- encryption information
- imaging notes (such as “logical image” or “bad sectors on original media”)

It is also necessary to label the target drives. For each of the drives I have in stock, I add a label (Figure 1) with a form printed on it that has an inventory number and asks for the following bits of information:

- case name,
- OAD, WD, DEL (deliverable), or IOD (interoffice drive),
- evidence number,
- custodian,
- date,

This label works great for quick identification of what is stored on the disk since you can and often do have multiple images from the same case stored on a single drive,.

Once all the documentation is complete, the original computer is re-assembled, and the OAD and WD are labeled and stored, then the acquisition process is complete.

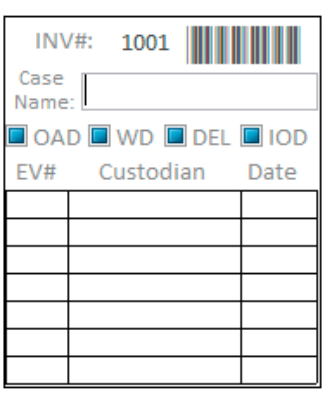


Figure 1. *Sample target drive label*

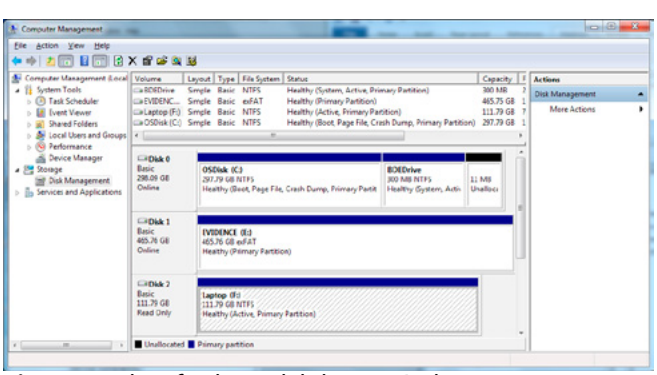


Figure 2. *Identify physical disk using Disk Management*

REAL LIFE EXAMPLE

The scenario is that you received a laptop suspected to contain digital evidence relevant to some litigation. Your job is to get a valid physical image of the hard drive, return the laptop to its owner, and perform some analysis at a later date as directed by the attorneys. You have already done all of the evidence intake and documentation and are ready to create the image.

The laptop hard drive is a 120 GB Corsair Solid State hard drive with a SATA interface. Since it is a forensics best practice to create two images, one for preservation and the other for analysis, you will need two 160 GB or larger hard drives for your target media. Note that if you plan to create an EnCase EX01 compressed image, you can get away with smaller target drives, but since storage is cheap and you can have multiple images on a single drive, then bigger is usually better.

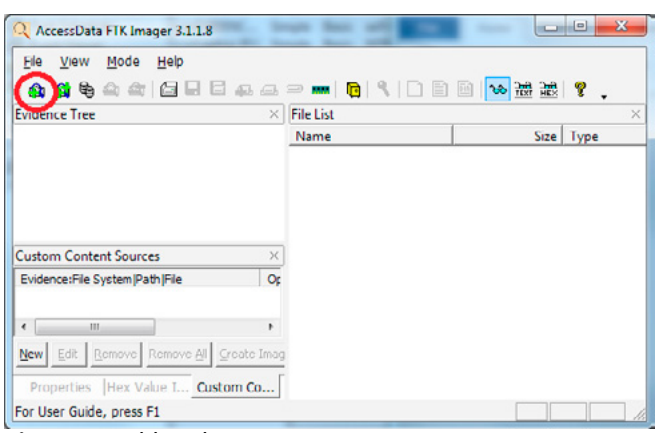


Figure 3. *Add evidence item in FTK Imager*

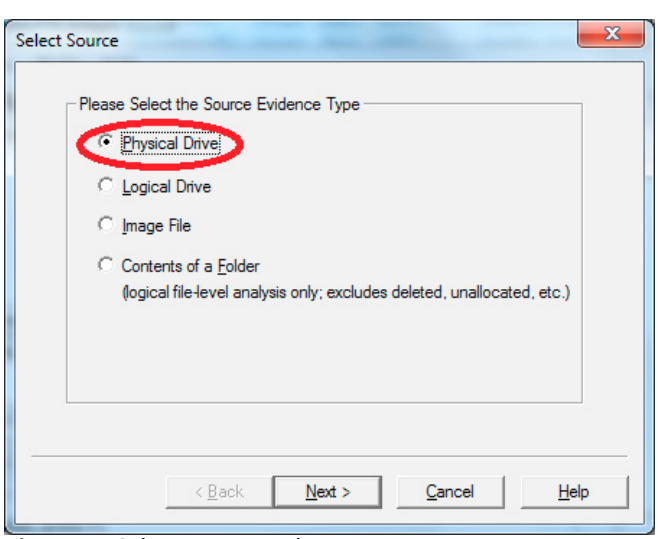


Figure 4. *Select source evidence type*

Since you are a new examiner and haven't purchased a lot of expensive imaging equipment and software, you are going to use your Windows laptop as your imaging device, a Tableau T35es-R2 eSATA Forensic Bridge for write-blocking, and FTK Imager 3.0 as the imaging software.

The first step is to connect the original media to the write-blocker and the write-blocker to the computer then power on the disk. Once it is up and running, you need to figure out which *physical disk* identifier Windows has assigned to the original media. Do this by right-clicking on the My Computer icon and selecting "Manage". Then select "Disk Management" from the left-hand column of the window. A list of mounted volumes and physical disks will be shown in the middle of the window. In the screenshot below "Laptop (F:)" is the Volume and "Disk 2" is the physical drive of the original media (Figure 2).

Now that we know which physical disk to image, we need to connect one of the target disks to the computer. This first drive will be the one for preservation also called the Original Acquisition Disk or OAD and will be handled much like the original media once the image is acquired. The second drive, called the Working Disk or WD, will be used for analysis and report storage. In Figure 1 above, the OAD is connected as "Disk 1". I formatted it as exFAT (though any format will work) and named it "EVIDENCE".

With both the source and target disks identified it is now time to start FTK Imager 3, which is free and available for download from AccessData software. First, add the suspect disk to FTK Imager by clicking the "Add Evidence Item" icon that looks like a single green plus sign "+" (Figure 3). From the menu, select "Physical Drive" and click the "Next" button (Figure 4). Now, in the "Source Drive Selection" window, select the correct physical drive from the drop-down menu. In our case it is `\\.\PHYSICALDRIVE2` and click the "Finish" button (Figure 5).

At this point the drive is mounted and you can navigate the file system on the target disk if desired (Figure 6).

To start the acquisition, click "File > Export Image" from the top menu. Then, in the "Create Image" window (Figure 7), click the "Add…" button.

You now have to decide on the "type" of image to create. The choices are Raw (dd), SMART, E01, and AFF. The differences between the formats can be found at *http://www.forensicswiki.org/wiki/Category:Forensics_File_Formats*, but generally either Raw (dd) or E01 are the best choices. Raw images are not compressible and do not have built-in error checking but are non-proprietary and can be mounted with any forensic tool. E01 (or EX01) images can be compressed and encrypted

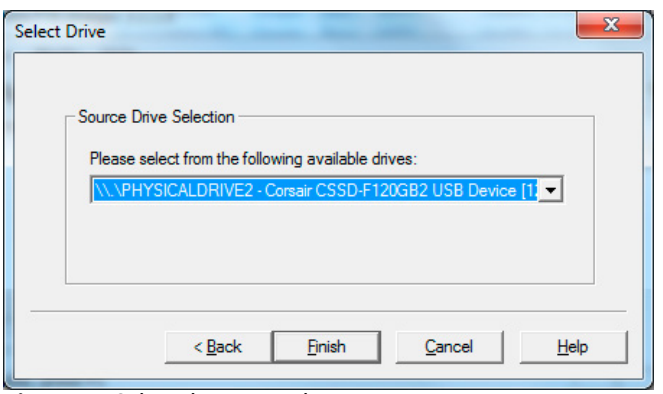


Figure 5. *Select the source drive*

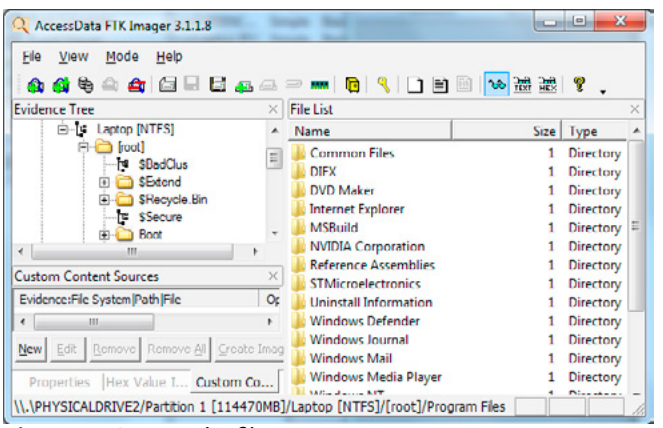


Figure 6. *Browse the file system in FTK imager*

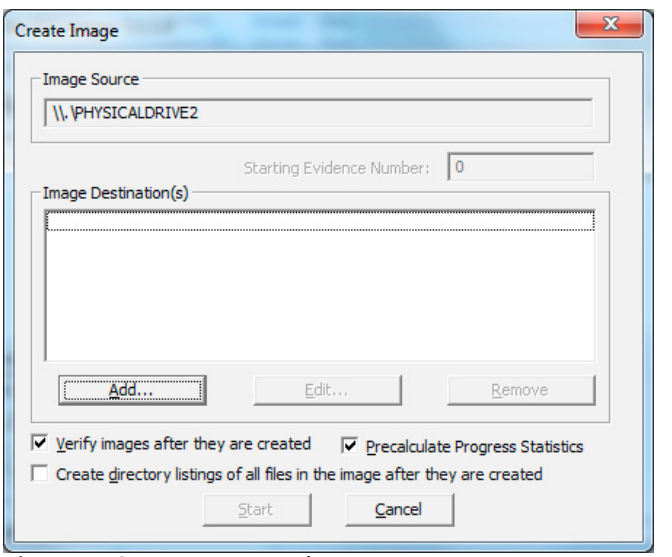


Figure 7. *Create Image window in FTK Imager*

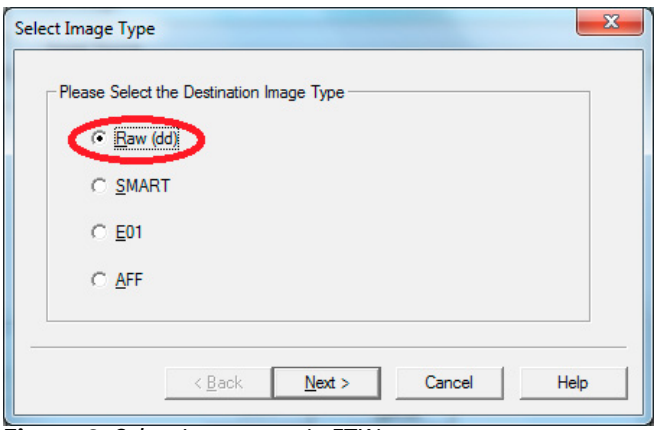


Figure 8. *Select image type in FTK imager*

and have built in error checking but are proprietary and the tools that can use them are somewhat limited. For the purposes of this article, choose Raw (dd) as the image type and click "Next" (Figure 8).

Next you will be presented with the Evidence Item Information window (Figure 9). Fill in the form with the proper case information. For the "Evidence Number", I like to like to keep it consistent with the "Image Filename" used in the previous step.

In the next window, titled "Select Image Destination", select or create a folder on the target drive to store the image (Figure 10). Then give the image a filename. I tend to use something simple like "EV001" for the first piece of evidence in a case. For the image fragment size, the default is 1500 MB. I always change it to 2048 MB (2 GB) due to the default size of other tools I use. I would suggest picking a size and sticking with it and do not go over 4 GB due to limitations of some file systems.

Finally, make sure that the "Verify Images after they are created" and "Precalculate Progress Statistics" options are checked (Figure 11). These options will let you know that you're image is good and will give an idea how long the process will take and how much time is remaining before completion. Click the "Start" button and the imaging will begin (Figure 12). As the image is created you can see the 2GB image fragments being created on the target drive (Figure 13).

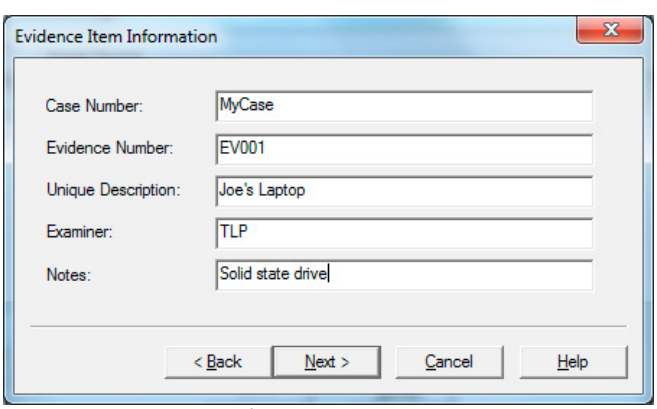


Figure 9. *Enter case information in FTK Imager*

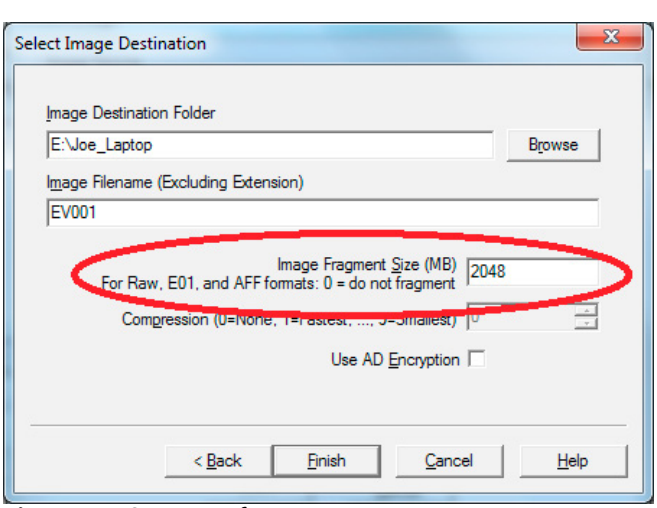


Figure 10. *Set image fragment size in FTK Imager*

Once complete, FTK Imager will display a window indicating 100% complete (Figure 14) as well as a window displaying the MD5 and SHA1 hashes of the source (Computed Hash) and target (Report Hash) (Figure 15). The two should match, indicated by the "Verify Result", and if they don't, FTK Imager will make it obvious.

In Figure 13 there is a button labeled "Image Summary…". Clicking this button opens a report that is created and stored with the image. It will be named like "ImageName.001.txt". In this case it is named "EV001.001.txt". The report contains the case information entered in Figure 8, important source and target disk details, as well as the calculated and reported hashes of the image. This report should always be delivered along with the image anytime the image is shared. Once the image is complete and verified it is time to create the second copy of the image on the working drive.

- Remove the source drive from FTK Imager (Figure 16) and power off the source and target media.
- Then move the OAD to the write-blocker and attach the second target drive to where the OAD was connected.

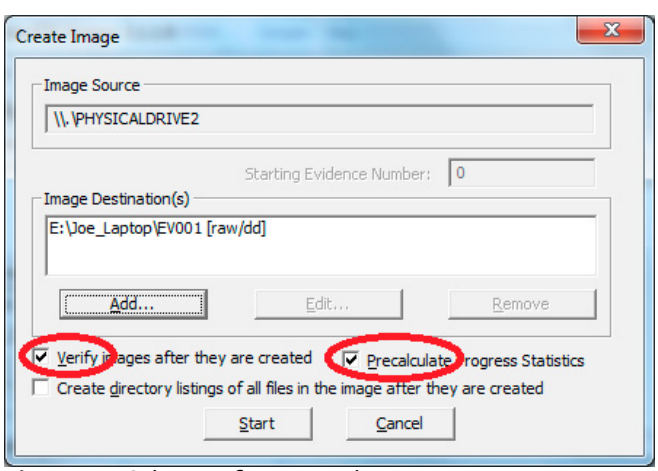


Figure 11. *Select verification and progress options in FTK Imager*

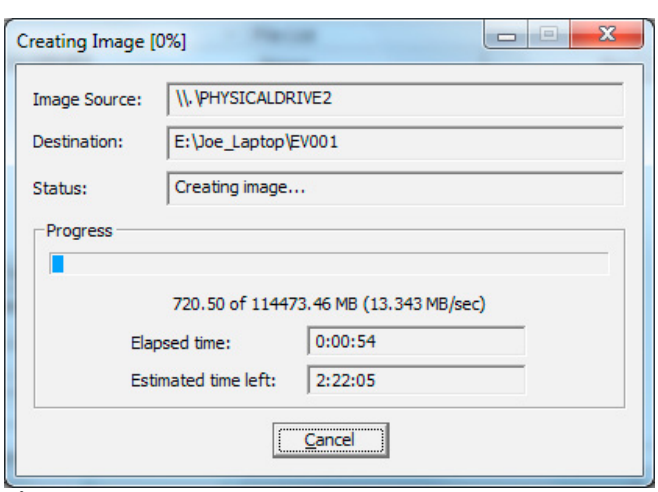


Figure 12. *Imaging progress in FTK imager*

- Format and name the WD with the same format and name as the OAD.
- Add the OAD as physical evidence in FTK Imager
- Navigate to the root of the OAD file structure in FTK Imager where you will see the image folder you created
- Right-Click on that folder and select “Export Files”
- In the “Browse for Folder” select the root of the WD and click “OK”
- Once the copy process is complete, the copied image must be verified. In FTK Imager:
- Click the “Add Evidence Item” icon.
- Select “Image File” (Figure 17).

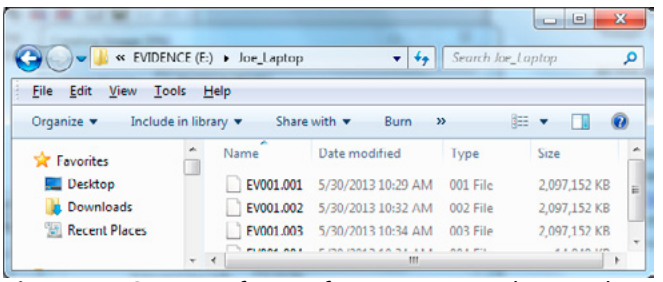


Figure 13. *Creation of image fragments in Windows Explorer*

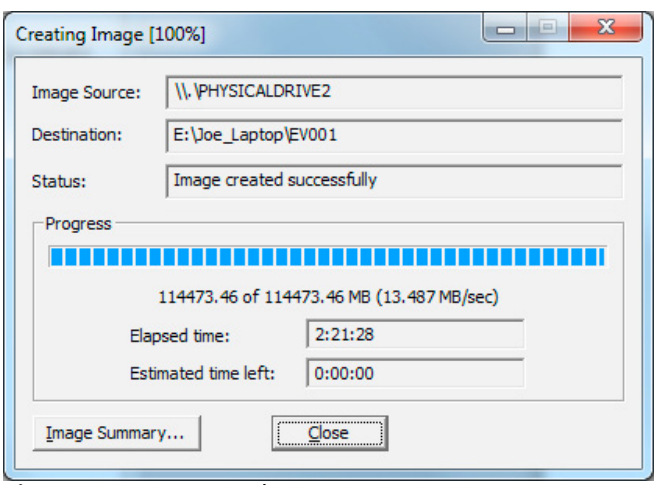


Figure 14. *Image completion in FTK Imager*

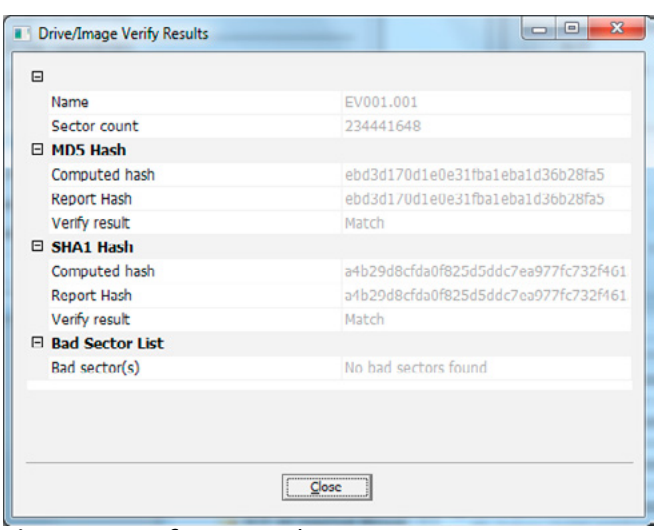


Figure 15. *Verification results in FTK Imager*

- Browse to the folder where the image is stored on the WD and Select the file EV001.001 (Figure 18).
- Click the “Finish” button.
- Right click on the image in the “Evidence Tree” column (Figure 19).
- Select “Verify Drive/Image…”.

You can browse the file system and even see the contents of files within FTK Imager while it is verifying (Figure 20).

Once the verification process is complete, compare the MD5 hash to that of the hash in the report file that is stored with the image. They should match. I like to take a screen-shot of the verification message and save it with the image on the WD. The screenshot documents the fact that I verified the working drive image. Finally, remove the image from FTK Imager, power-off the two target drives, and the acquisition is complete.

OTHER HANDY TOOLS

In this article I have focused on using FTK Imager. However, there are a number of other tools out there, both software and hardware, which will do the job:

- Forensic Tool Kit (FTK) from AccessData – Popular commercial forensic software suite

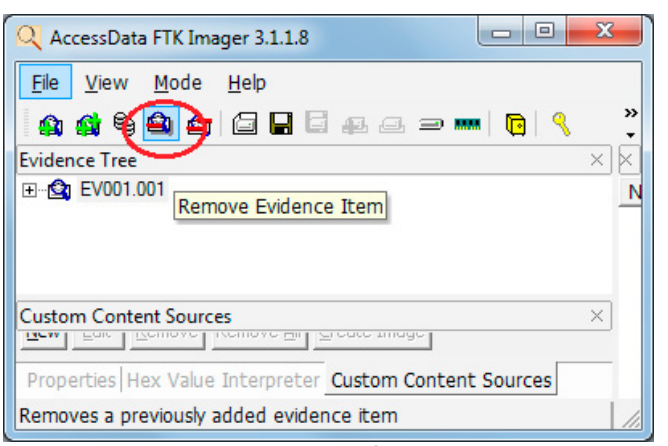


Figure 16. *Remove evidence item from FTK Imager*

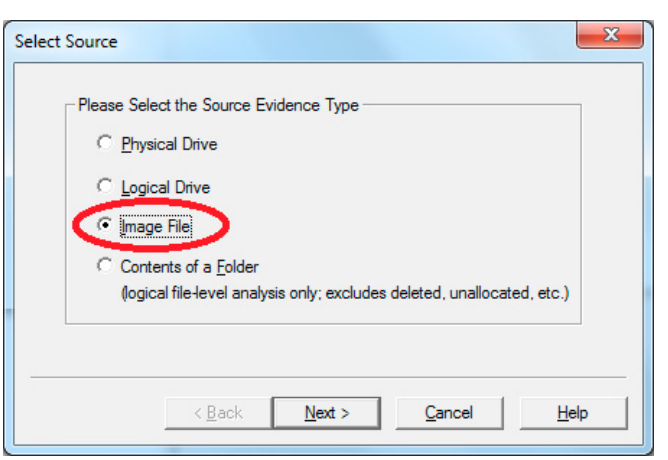


Figure 17. *Add image file to FTK Imager*

- EnCase made by Guidance – Popular commercial forensic software suite
- ImageMaSSter from Intelligent Computing Solutions – Commercial hardware solution. Allows simultaneous creation of an image to multiple target drives
- F-Response from F-Response software – Commercial remote acquisition tool
- dd – Open source command line image creation tool for *nix
- dcfldd – Open source command line image creation tool for *nix. Similar to but has more forensics features than dd.
- dc3dd – Open source command line image creation tool for *nix. dc3dd is a patched version of GNU dd to include a number of features useful for computer forensics. Many of these features were inspired by dcfldd, but were rewritten for dc3dd.
- CelleBrite – Commercial hardware and software for imaging and examining mobile devices

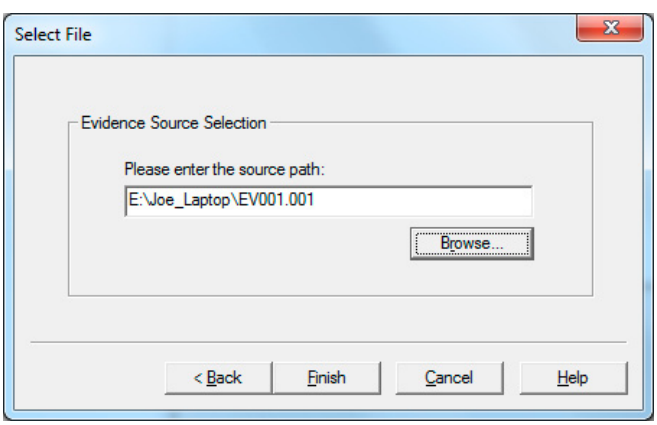


Figure 18. *Select source path in FTK Imager*

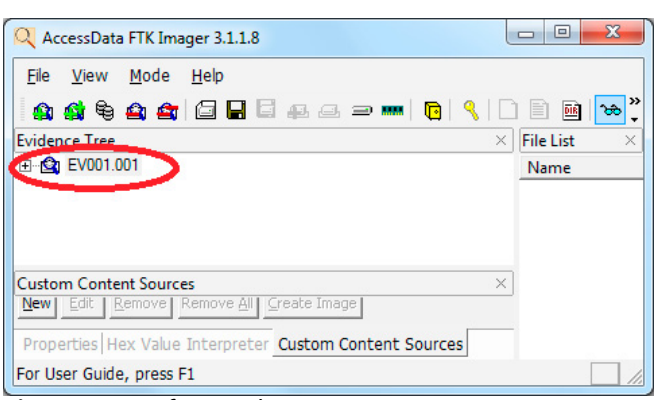


Figure 19. *Verify copied image in FTK Imager*

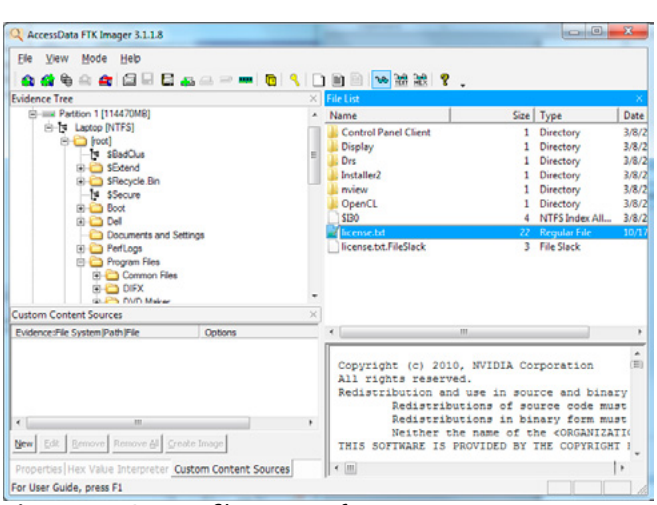


Figure 20. *Browse file system of image in FTK Imager*

There are quite a few others out there, mostly specialty software or boot CDs that provide GUI's to the dd* tools or are for specific platforms. These include Raptor, BackTrack, CAIN, Blacklight, Helix, and I'm sure many others.

Other tools an examiner should have in order to perform nearly any type of acquisition:

- portable USB CD/DVD player,
- bootable USB drive,
- collection of SATA, eSATA, SATA to eSATA, IDE, USB, FireWire, and Ethernet cables,
- Molex to Molex and Molex to SATA power adapters,
- universal power supply,
- adapters for SCSI, PATA, MicroSATA, MicroIDE, and ZIF connectors,
- universal IDE/SATA to USB bridge,
- anti-static mat,
- external SATA disk docking station,
- a variety of boot disks on both CD/DVD and USB,
- Linen EnCase Linux boot CD for network acquisition,
- network switch,
- antistatic evidence bags,
- printed chain of custody forms.

CONCLUSION

The process of digital image acquisition is not just obtaining a disk image. It is the collection of efforts that track the evidence from source to evidence locker to the courtroom. It is the foundation of the digital forensics field and proper documentation serves as its cornerstone. To create inscrutable digital images, the forensic examiner must be technically skilled, detail oriented, and able to improvise when the situation arises.

About the Author

Thomas Plunkett has worked in the information security and digital forensics field since 1995. He has acquired thousands of digital images, performed forensics and security incident response for clients ranging from Federal government agencies and large multi-national companies to local businesses and celebrities. He is a Certified Information Systems Security Professional (CISSP), EnCase Certified Examiner (EnCE), and holds a Master's Degree in Information Systems.

FTK IMAGER BASICS

by Marcelo Lau & Nichols Jasper

This article discusses a basic FTK Imager case study. In this case study a pen drive has been found with a suspect, but it appears to be empty. We will show how to image the pen drive's file system and how the FTK tool can help us to show traces of deleted artifacts in the evidence media.

What you will learn:

- How to create a disk copy from a pen drive recovered during an investigation
- How to use FTK Imager functions to mount the image and see deleted files from the file system.

What you should know:

- Basics of forensics principles and procedures to make a digital evidence preserve their admissibility.
- File system concepts, like allocations and deal location of files at NFTS.
- Installation and basic usage of Windows platform programs.

Mobile devices are used everywhere and IT professionals have at least one USB Storage device to backup data, and files to use in another desktop computer or server or use it like a cryptographic token.

In this scenario, we know many times this kind of media is susceptible to data loss due to small physical size. I have already lost two of my USB storage devices, fortunately backup practices saved me a few headaches. With ubiquitous computing and storage media, a forensics examiner would be able to use forensic artifacts like that, which could be essential to the investigation context.

When nonprofessional users are performing a logical format, they will never be sure if their data or secrets are discoverable. We will see that in the modern NTFS file system, it does not occur without a robust device's wiping method and we can recover an essential piece of the investigation puzzle to validate and prove a hypothesis that might solve the case.

CASE CONTEXT

You walk into the room. The suspect from a crime that involves intellectual property is in the other corner and he doesn't have a lot of time. He's afraid he might be caught. You notice he is quickly typing some commands in his Windows 8 station. When you look at the computer, you discover only an empty pen drive visible on Windows Explorer.

When you try to visualize the content, the following shows: Figure 1.

His manager says, "I know he had copied many files off of the server, but I have no server log that can prove it".

Upon analyzing the situation you ask questions on how did the suspect delete or destroy the data so quickly – perhaps Wipe? Hidden Files? Some Cloud Storage tool? Maybe he uses the CTRL+X command from his pen drive to do something?

You have a good idea of what happened, but how can we prove this in court? How to prove that what you saw is what really happened; if the computer that the files were copied to has no control applied to support your argument of a corporate fraud?

We need to present to the court judge a forensics report with convincing arguments, and evidence that organize the facts and reconstitute without any doubt what happened since defense lawyers will have many "IFs" to argue your report and to free or mitigate his client debts with the justice.

For educational purposes, we will present a way you can conduct the investigation, and we will assume that our suspect used CTRL-X command to clean his pen drive at the very least.

TECHNICALLY VIEW AND FORENSICS RESPONSE

Let's see what occurred from a technical outlook. The suspect was using his Windows 8 workstation, and you have the information that maybe he was unlawfully collecting information from the company, but you don't know how this occurred. When you and his boss entered his office, you noticed that he typed commands on his computer – these commands will show whether his actions were legal or if he was trying to hide something.

First, you lead the suspect away from his workstation and leave him with responsible collaborators. Examining the workstation, you saw that it was not connected to any network, no Ethernet cable or wireless device, so your attention now focus on the operating system and file system properties, in this case, NTFS.

After concluding, that network traffic or memory information is not the path to go with this investigation, you make a live image from the PC to save the running processes and other volatile information with a forensics system like CAINE and then turn off the PC and use your forensics disk duplicator to assure a copy of the hard drive and the pen drive is made.

In your forensics lab, you make another copy from the first duplication and start examining the second copy. You must ensure the hash integrity is maintained for future admissibility.

You believe that the suspect deleted the content of the mobile media or used the CTRL-X command (cut) to remove the content and save it on another device. But, how can we prove it?

If the person deleted the files, the NTFS file system run the following process: it marks the files as deleted within the MFT (*Master File Table*) entry, leaving your position on MFT ready to be reused by the operational system if some "write" operation on the device is needed. Therefore, the content will be there until this area of the storage device has been re-written, overlapping the last content. An interesting video from "WhereIsYourData blog" shows this process through the forensics point of view.

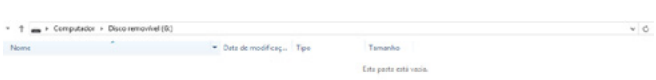


Figure 1. *The Pen Drive of Suspect*

Figure 2. *Access Data FTK Imager Download Page*

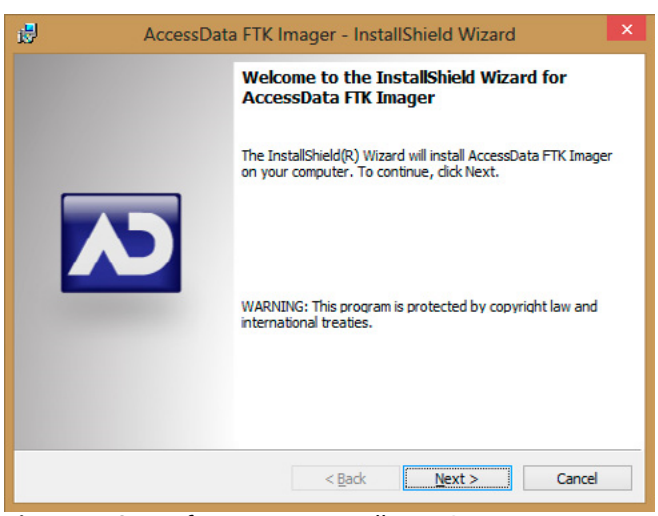


Figure 3. *Start of FTK Imager Installation Process*

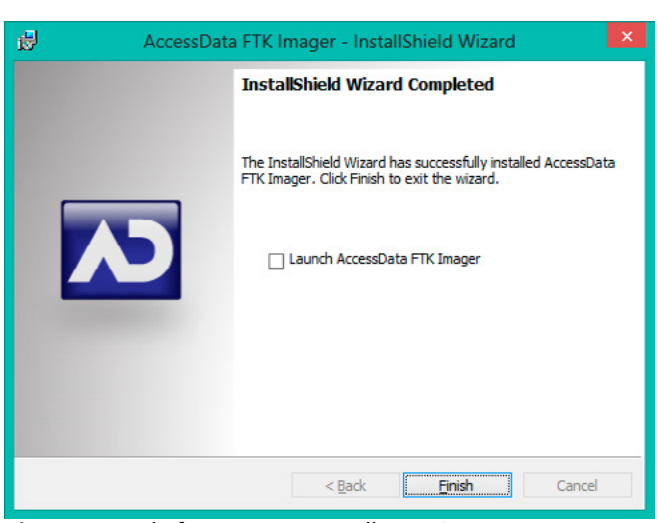


Figure 4. *End of FTK Imager installation Process*

If the content was not deleted, the CTRL-X Command is a combination of two operations from the Windows NFTS system:

First, there is a copy of the files to the memory and, when the insert place is selected and second on the moment that files were successfully moved there is the exclusion of the mark in the MFT Table, in other words, a real delete action in the file system.

Let's start the practical tasks with the forensic tool FTK Imager to show the process of cloning and analyzing.

DUPLICATING AND ANALYZING A THUMB DRIVE WITH FTK IMAGER

FTK Imager is forensic software created by Access Data that has features for creating forensic disk images, performing memory dumps and even resources for forensic analysis in the image created.

This article will focus on the forensics tool FTK Imager for two reasons: It's free, it's easily accessible and it's very intuitive to an enthusiast who knows a bit about computer forensics and wants to advance in this area.

Finally, let's start it. First of all, download the tool on the company's page: *http://www.accessdata.com/support/product-downloads* (Figure 2).

We will use FTK 3.1.2. After the short registration, get your package and start the installation process (Figure 3).

The installation process is very simple and requires almost no configuration; a NNF process will install the FTK tool on your Windows machine. In the final screen, don't check the box, because we can prepare the forensics station not to alter the original evidence (Figure 4).

In the sequence, use a hardware or software solution to block the "write" permission in the USB

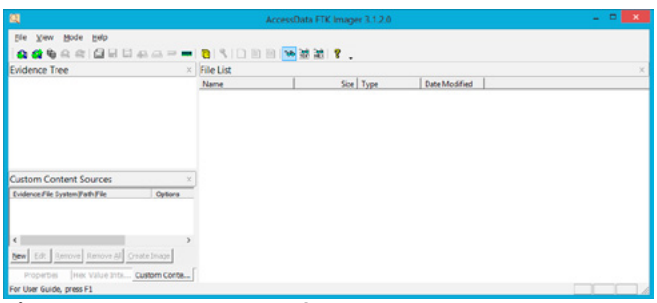


Figure 5. *FTK Imager Main Screen*

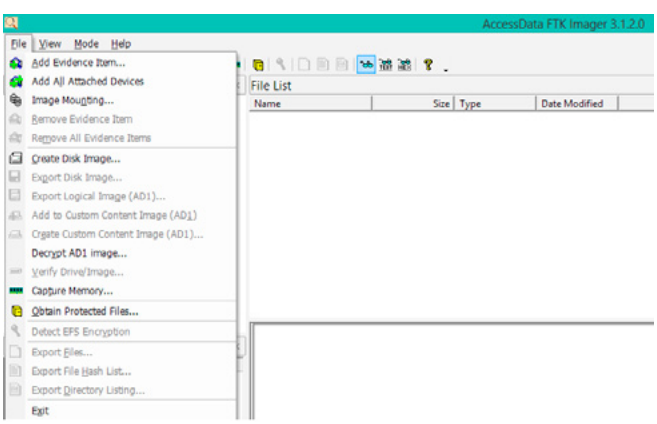


Figure 6. *FTK Imager Create Disk function*

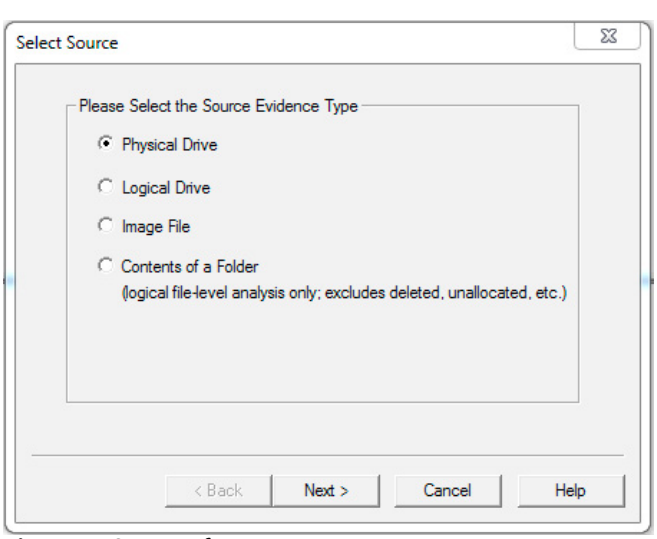


Figure 7. *Source of copy*

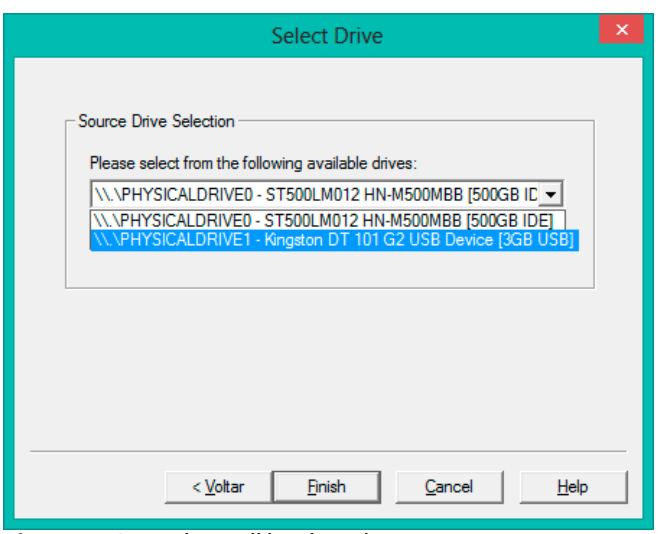


Figure 8. *Drive that will be cloned*

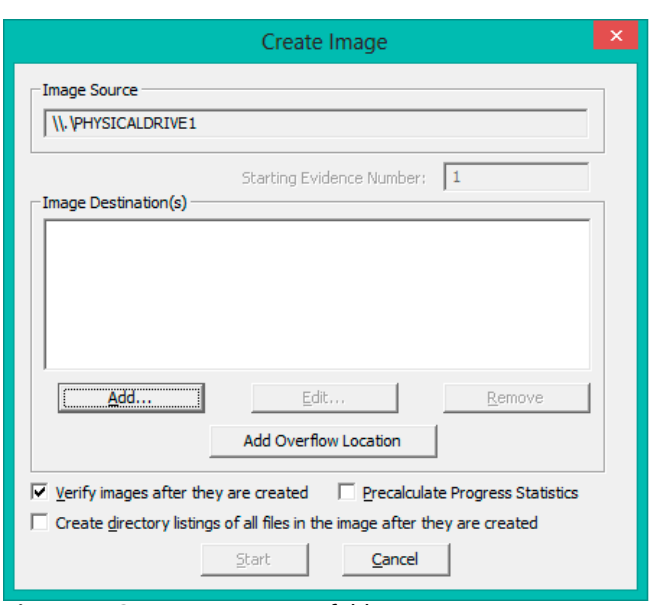


Figure 9. *Customizing output folder*

Ports of your Computer. The software solution changes the registry key below to assure that the operational system won't "write" on the device:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\
Control\StorageDevicePolicies\WriteProtect
```

Preferably, use another pen drive to test if the "write" protection is enabled. For some good references, use our "On the Web" section with valuable resources about Write Protectors.

Assuming the USB is already protected, let's plug in the pen drive collected from the suspect and start to produce the disk image. Start the FTK Imager with Administrator privileges (so it can find all the attached storage devices) and you will see your main interface: Figure 5.

In the menu File, use the option Create Disk Image: Figure 6. Then, select Physical Drive as copying source and click on Next (Figure 7). Choose the attached pen drive as Drive where the bit-a-bit copy will be made and click on Finish (Figure 8).

Click, Add… to customize your output folder (Figure 9). Check the image format like raw (dd) and it will get the bit-for-bit copy of the RAW data of

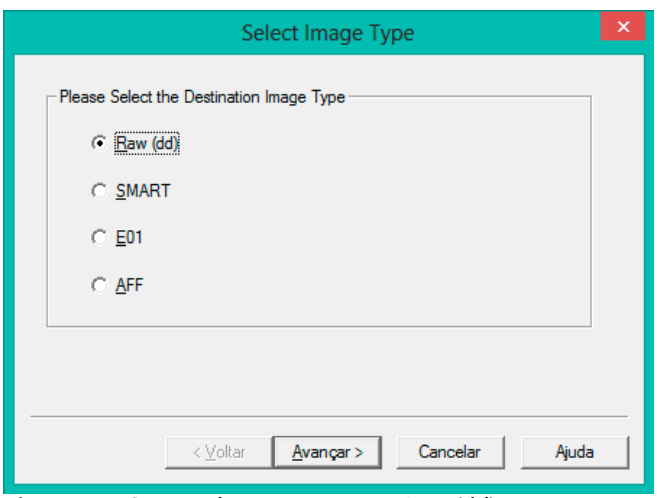


Figure 10. *Setting the Image Type as Raw (dd)*

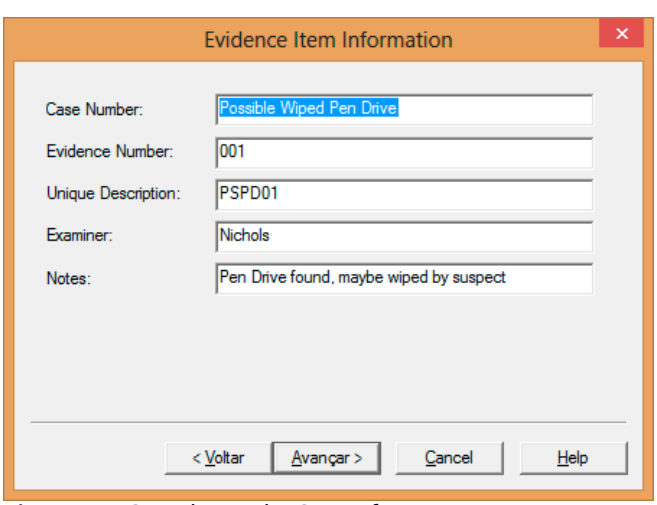


Figure 11. *Completing the Case Information*

the disk or volume picked. Click on Next (Figure 10). Fill the case information and you can register what is the link between the evidence and the case. Click on Next (Figure 11).

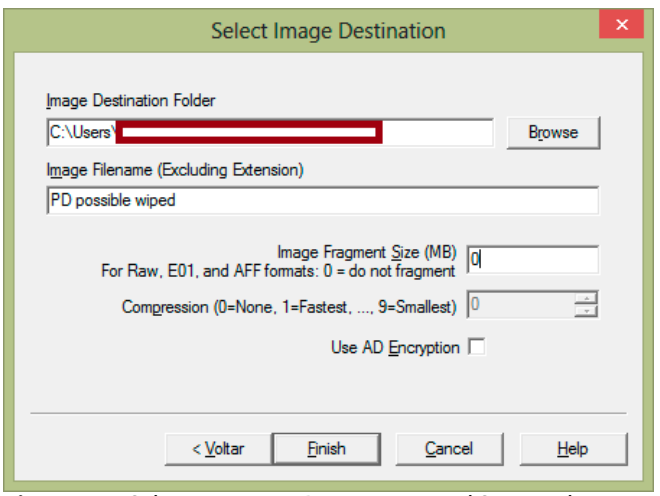


Figure 12. *Selecting Image Destination and Setting the Fragment Size*

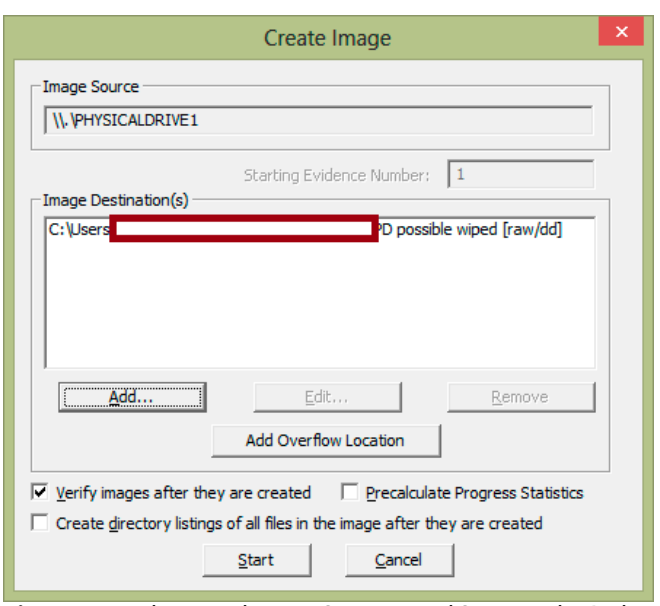


Figure 13. *Adjusting the Last Settings and Starting the Disk Cloning Process*

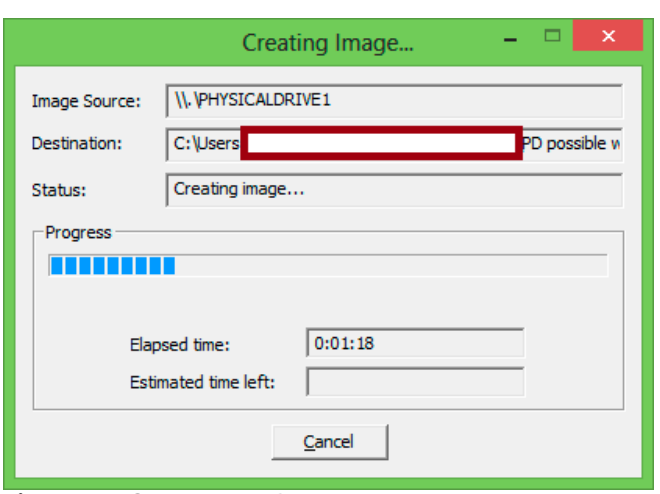


Figure 14. *Create Image Process*

Browse your file system and choose the folder where the pen drive image will be hosted. Additionally, put the image filename and *set the fragment size to 0 (zero)*, so only one image file will be generated instead of many files, according with the fragment size and the size of disk or volume involved in the process (Figure 12).

If all the options are ok, click on the Start Process and wait to the end of the disk duplication process (Figure 13).

The drive has been replicated bit-to-bit, preserving its integrity and showing the hash calculation in the end. If something goes wrong, the hash will mismatch and you should analyze what happened to correct the process error.

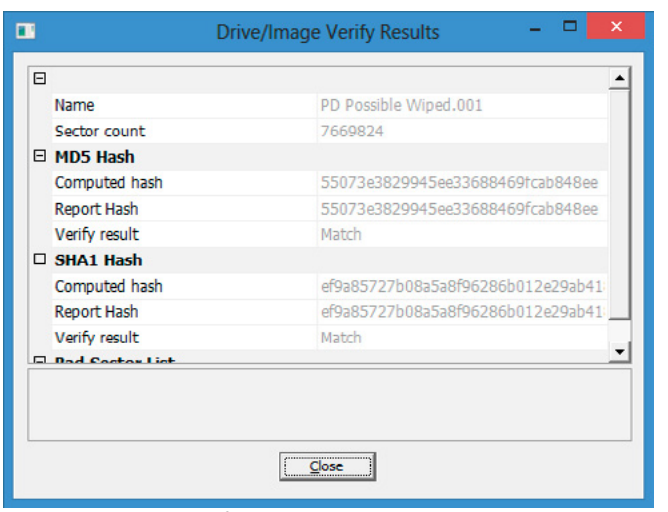


Figure 15. *Results of Cloning Process*

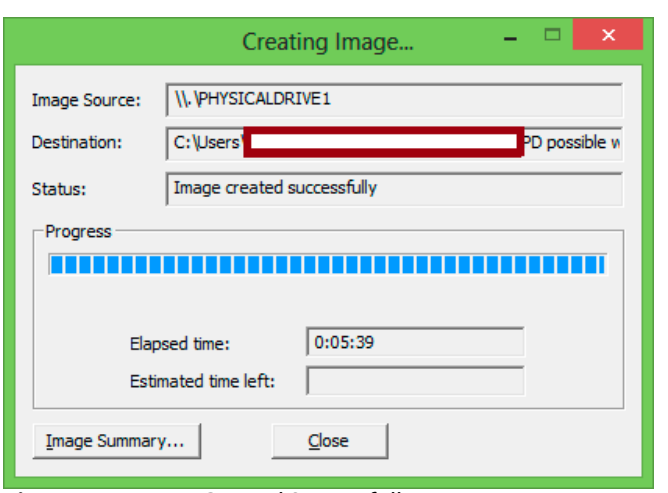


Figure 16. *Image Created Successfully*

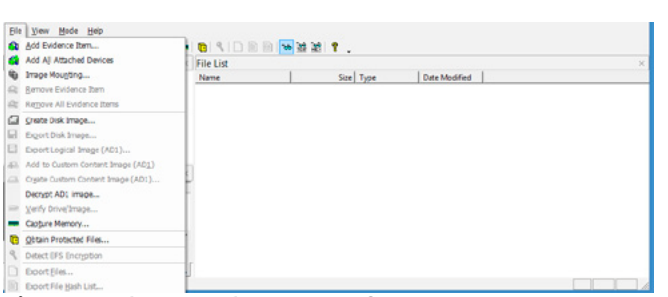


Figure 17. *Decrypt AD1 Image… Option*

After the image file creation, the verifying process will be shown (Figure 14-16).

Now we have the image file of the suspect's pen drive. Let's use the FTK Imager to analyze the image file and try to figure out and reconstitute the facts that could lead us to support our hypothesis or provide subsidies to a new line of investigation.

If you want to encrypt the image to send over hostile environments (like Internet) or host in a shared cloud host with more security you can use FTK Imager "Decrypt AD1 Image…" option to encrypt the .dd file as shown Figure 17.

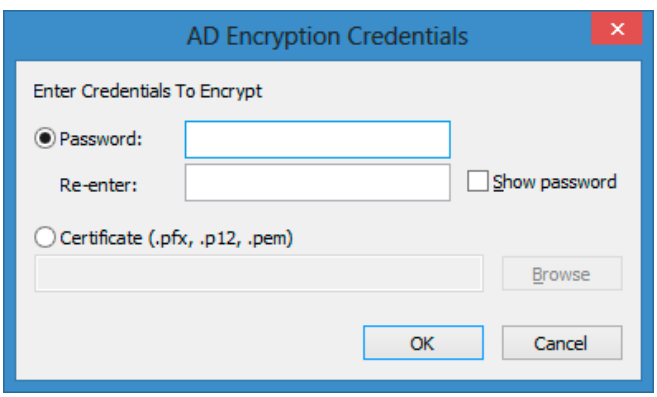


Figure 18. *AD Encryption Credentials Prompt*

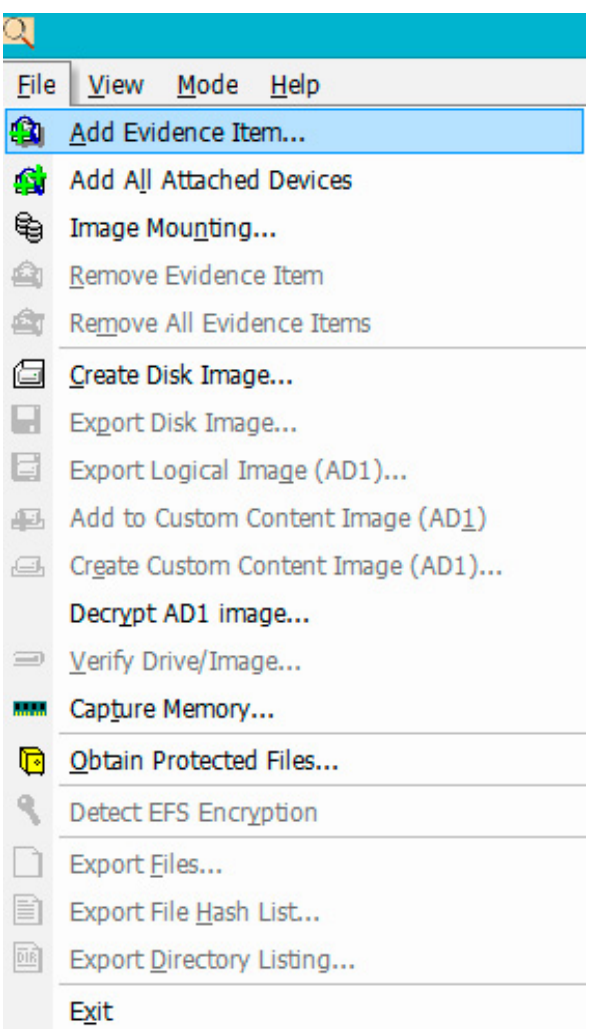


Figure 19. *FTK Imager Menu*

Clicking of previous option will appear a browser window to search your disk image file. Choose the directory were you saved the file and the output folder. After that you'll see the password prompt, asking for the user input or a certificate with a public key to encrypt the image so that only the owner of private key can decrypt the file: Figure 18.

Returning to the FTK Imager interface, let's use the *Add Evidence Item*… option (Figure 19).

If the image is encrypted, when you try to mount the image a password prompt will appear, and you need to enter the correct password to open the file (Figure 20).

The process of mounting the image on FTK is similar to the image creation. First, choose the source that will be mounted on FTK e.g. a Physical or Logical Drive, an Image File or simply a folder. Let's choose the *Image File* option, pointing to the *directory where the image file is* and observing the results in the following interface: Figure 21.

Zooming the FTK File List screen, we can see some files with an X mark. This is the sign the FTK Imager tool uses to show a file that has already

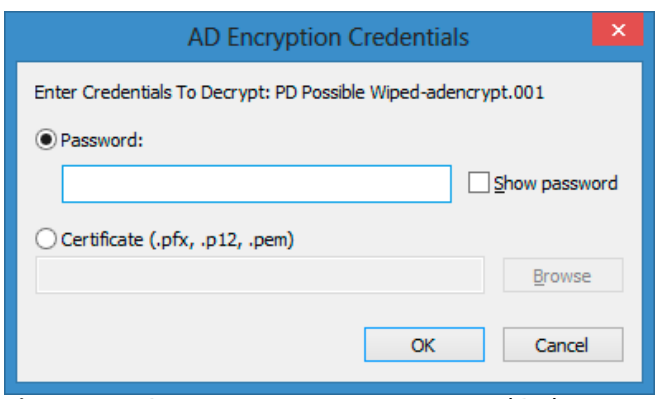


Figure 20. *AD Encryption to open an Encrypted Disk Image*

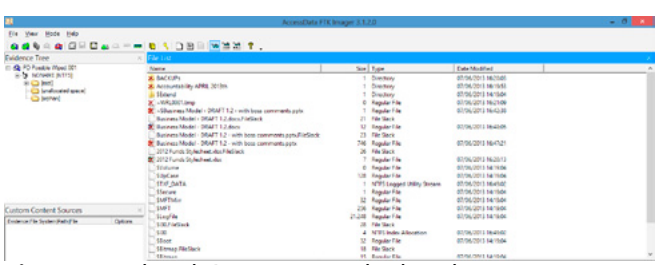


Figure 21. *Thumb Drive content displayed on FTK Imager*

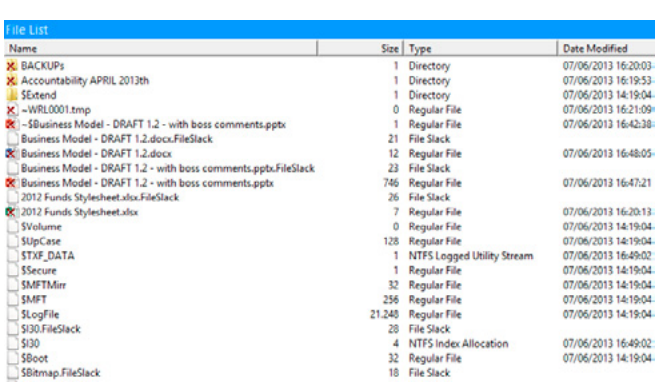


Figure 22. *Deleted Files are Marked with an X Sign*

been deleted from the File System, but it is accessible when the volume structure is deep analyzed (Figure 22).

We can see that many files were deleted from the pen drive. Based on the date of modification, we can see that the set of files were closely modified… maybe the author had just finished the edition of the files before the suspect made the unauthorized copy.

We can reconstruct exactly the content of the file if no overwriting was made. Simply right click on a deleted file and click on "Export Files", like showed below, and select a directory where the file will be saved (Figure 23 and Figure 24).

After the exportation, we can visualize the content of a PowerPoint presentation "*Business Model – DRAFT 1.2 – with boss comments*" and realize that the content is a critical document for a new business area the company will open to the market… maybe a competitor can find some valuable information here (Figure 25).

CONCLUSION

Finally, we understand the use of forensics tools are essential to a good computer forensics expert and the FTK Imager is a great and free tool that provides a forensics platform for investigations of digital evidences, unfortunately the tool lacks a mechanism of search and automation. It is a good resource that a beginner investigator could use to start his activities of collecting and analyzing digital evidence to present a detailed forensics report to his clients.

In many investigations, deleted files are a key point to show what's happened to the system or what activities are routinely performed on the computer. Knowing how to insert this kind of evidence in a case can be essential to show the client, judge or a court of law how the digital evidence can elucidate the facts and attribute responsibilities to persons involved.

Computer Forensics is a promising field and the knowledge required to be a good professional is very important. Combining many aspects of computer science, operational systems, computer networks, information security and digital forensics opens a rewarding professional career challenge for beginners.

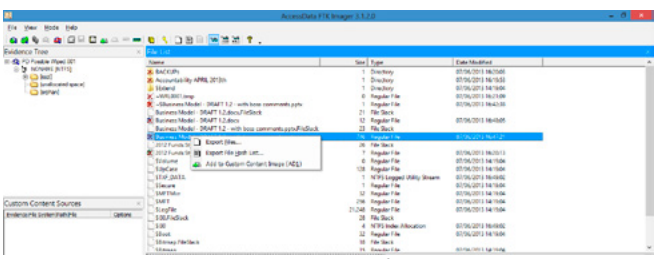


Figure 23. *Exporting the Content of a Deleted File*

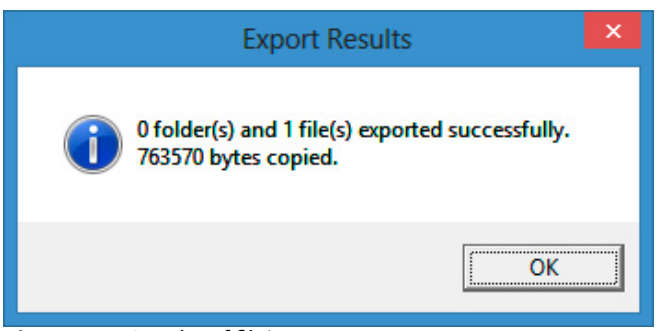


Figure 24. *Results of file's export*

Figure 25. *Presentation Recovered from the Pen Drive*

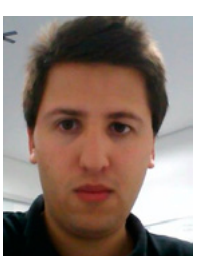
ON THE WEB

- *http://whereismydata.wordpress.com/2009/05/02/forensics-what-happens-when-files-are-deleted/* – Forensics: What happens when files are deleted?
- *http://www.forensicswiki.org/wiki/Write_Blockers* – Write Blockers.
- http://www.caine-live.net/ – CAINE Computer Forensics Linux Live Distro
- *http://www.forensicswiki.org/wiki/FTK_Imager* – Forensics Wiki of FTK Imager,
- *http://www.accessdata.com/support/product-downloads* – FTK Imager Download.

About the Author

Marcelo Lau is Engineer, post graduated In Administration, Communication and Art. Msc at University of São Paulo and experienced Information Security and Computer Forensics on several large banks in Brazil. Nowadays is owner and executive director at Data Security in Brazil. Well known professor at several universities in Brazil and other South America Countries, as Argentina, Bolivia, Colombia, Paraguay and Peru.

About the Author



Nichols Jasper is a security analyst with over five years of experience in consulting services, including collection and analysis of many cases of security incidents that demand a forensic report. The main subject of investigations is corporate fraud involving intellectual property events, and lawsuits that involves the use of electronic evidence.

DATA SECURITY
Computer Forensics Experts

Computer Forensics Services

We are prepared to attend the situation urgency supporting your needs and delivering our consulting solutions considering our worldwide cybercrime knowledge base by:

- Dispute support services
- Evidence Identify and Collection
- Evidence Analysis and Reporting
- Device analysis as: Computers, Smartphones, Tablets, Network, Printers, even Games Consoles...

Computer Forensics Training

Get in touch enjoying our cases applying methodologies and tools resolving a real forensic case in 40 hours. At last you will be submitted by a certification test (DSFE) proofing your skills.

R. Eça de Queiroz, 682 – Vila Mariana

São Paulo, S.P. 04011-033 - Brazil

Phone: +55 11 5011-7807

E-mail: datasecurity@datasecurity.com.br

facebook.com/data.secur.face

@datasecurity1

BASIC APPROACH TO INVESTIGATE A DIGITAL CRIME

by Ali Fazeli

If there is a computer on the premises of a crime scene, the chances are very good that there is valuable evidence on that computer.
Accidental/planned destruction of data, hardware failure or cyber attack can happen anytime and a computer forensics investigator may be called upon to respond, review and escalate the analysis to a formal investigation.

What you will learn:

- Basic computer forensic & investigation techniques
- Chain of custody & How to acquire the digital evidence
- Analysis and data extraction using proposed protocols
- Data recovery & Signature analysis

What you should know:

- A combination of education and work experience in computer security industry
- Good knowledge in operating systems & file structure
- Good knowledge in Networking

Computer forensics is a branch of forensic science pertaining to legal evidence found in computer and digital storage mediums. Kruse and Heiser defines, Computer Forensics is the preservation, identification, extraction, documentation and interpretation of computer media for evidentiary and/or root cause analysis. (Kruse and Heiser 2001)This article is designed to introduce you the fundamental of computer investigation concept.

INTRODUCTION

In general, the goal of forensics analysis is to identify digital evidence for an investigation. Examples of investigations that use digital forensics include unauthorized use of corporate computer, child pornography, and a perfect crime that's suspected to use a computer. At the most basic level, computer forensics has three major phases, Acquisition, Analysis, and Presentation.

The Acquisition phase saves the state of digital evidence so that it can be later analyzed. As in the physical world, it is unknown which data will be used as digital evidence so the goal of this phase is to save all digital values. In the first step all the allocated and unallocated areas of a Master hard disk or any other storage device are copied and captured, and stored to another device. There are many tools that can be used in this phase to copy data from a suspected storage device to a trusted device. These tools must not modify the suspected device and copy all data.

The Analysis Phase takes the acquired data examines it to identify pieces of evidence. There are two major categories of evidence that

computer forensics professional are looking for, Exculpatory Evidence and Evidence of Tampering. Exculpatory Evidence means, evidence which support a given theory.

Exculpatory Evidence means, evidence which contradicts a given theory.

Evidence of tempering means, evidence which cannot be related to any theory, but shows that the system was tampered with to avoid identification.

Analysis phase includes:

- Transform the Voluminous amount of data collected during data collection phase into a more manageable size and form for analysis
- Employ data extraction techniques like Keyword searches, file timeline/Mapping, hidden data discovery, etc.
- Examine, analyze, and even reconstruct the data to answer critical investigation question.

Presentation phase is the conclusions and corresponding evidence from the investigation.

The purpose of the presentation is to present relevant findings to management, legal personnel or law enforcement.

ANALYSIS PHASE

The data analysis phase is the most complex and time consuming phase in the digital investigation process. Once the computer system has been seized, it is time to examine the evidence. There may be many elements of the computer structure which will give a clue of a crime, and it is very important not to target one piece of information. As I said before, the purpose of the data analysis phase is conforming or refuting an allegation of suspicion. Several models have been proposed for the analysis and examination phase.

First, two phase approach is examination and analysis. The examination phase is primarily characterized by search and extraction activities, whereas the analysis phase is primarily characterized by subsequent activities that generate useful information from the extracted data.

Secondly, a single data analysis phase, subdivided into “data preparation” and “data Analysis” Sub-Phase. Data Preparation Sub-Phase includes: file list creation, deleted data recovery, an allocated space recovery, statistical data collection, partition table and file system identification, file signature analysis, and known system file identification. The data analysis Sub-Phase includes: Email attachment extraction, installed application review, string searches, software analysis, network based evidence review, and specialized analyses.

Third is live system processing and data collection,’ and (2) ‘the analysis of secured data’. This approach is more useful from a network forensics perspective, thus live system processing and data collection includes copying system files, logical volume imaging, and obtaining system date/time information. Analysis of secured data includes logical analysis of the media structure, collecting operating system configuration information, file system mapping information collection and analysis, file signature analysis, identifying file content and type anomalies, evaluating program functionality, text string and key word searching, evaluating virtual memory, and evaluating ambient data.

CHAIN OF CUSTODY

There are several guidelines to consider when seizing evidence. First, Computer, PDA, or mobile phone should not be simply turned on. This may change some of the data on the device. Every time a digital device is powered on the access times of certain files are altered; these files contain crucial information to the investigation.

It is advisable to photograph the scene, certainly record and document, location, time & those present. In addition anything present in the scene may be used as evidence. Next step is to record all serial numbers, tag cables and record how and where they were placed as part of the configuration.

Before an investigator starts the examination process, a systems data drives should be mirrored and the original stored in protected manner. There are many different physical forms the original data may arrive in, and an investigator needs to be prepared to handle all of them. In addition, most data will arrive on hard disks, but data can also come in the form of removable media such as DVD’s, floppy disk, Flash memory, etc.

Time is a critical element in investigations, beware of different time zones and never change the clock setting on a seized computer. Don’t forget electronic documents by their nature are constantly changing, even opening a file to view will change the metadata, and may change the substance as well.

WRITE-PROTECTION

All media must be write protected in the first place, and write-protecting media prevents data from being added to the media. Write-protecting digital evidence guarantees that the evidence is not altered or erased when an investigator is working with it.

MIRROR COPY PROCESS

Next step is to make an exact copy of the disk. There is software available that can make a mirror image of a disk, bit-by-bit. Hard drive imaging must be carefully conducted to ensure no accidental overwrite of data occurs and the hard drives themselves are physically unharmed. The process must be recorded in detail to later prove the original image was not altered and the copies are true copies. (Appendix A). The methods are used in fo-

rensics system imaging, original system imaging, and system-to-system imaging.

Original system imaging uses the original computing platform to perform the imaging. A new blank drive is added to the system, and a special boot disk is used to run the imaging software and create the image. This method may be necessary if it is not possible to remove the original drive from the system. This method is most often used when the investigator has to travel to where the original system is located, instead of being able to take receipt of the original drives for imaging.

System to system: system to system imaging method uses two different computer system, typically the original system and a forensics imaging system. Both are booted from a special CD or floppy disk that load imaging software for transferring data between the computers using parallel, serial, Ethernet, or USB connection ports. This method is slower than the others, but may be necessary when trying to create an image from two incompatible hard drive formats, such as SCSI and IDE

In case of any problems, all analysis must be done on the copy rather than the original data.

The first copy made is referred to as the "master copy", and is not used for performing analysis, but rather for creating additional mirror copies on which analysis will be perform. In this manner, the original data only needs to be handled once to make a "master copy" and you need to keep it in safe storage or released from custody. This is important as an investigator can never boot the original drive because this will change information on the drive, including data of potential evidence. (Appendix A)

HASH VALUE

Once a master copy has been created, hash value processing has to be implemented to both the master and copy of the evidence. A few tools use Message digest version 5 (Md5), or Secure Hash Algorithm (SHA), so it can be used to confirm that the duplication process has been done properly.

These programs generate a unique value for both the data on the original hard drive and data on a master and copy of the hard drive, in order to further verify the results of the duplication process. These programs rely upon 'hashes' to confirm that the duplicating process has been done properly.

VIRUS CHECKING

Virus checking is the next step. If a virus is detected, an investigator has to record all information about the virus detected. They should be careful not to take steps to clean the media and eliminate the file, because it may change the evidence.

OS & FILE STRUCTURE

An intimate knowledge of Operating systems, file allocation and structure of storage drive are helpful to choose the right tools and find evidence. (In this paper I only focused on Windows systems).

Examine the physical level of partition is the first steps in this process. Norton Disk Edit, WinHex, or Hex Workshop can be used to examine the physical level of a partition. These tools allow investigators to view file headers and other critical parts of a file. Both involve analyzing the key hexadecimal codes that the operating system uses to identify and maintain the file system.

Examining the File allocation Table (FAT) is the next step. "The FAT database contains filenames, directory names, date and time stamps, the starting cluster number, and attribute (archive, hidden, system, and read-only) of files on a PC. PCs use the FAT to organize file on a disk so that the OS can find the files it needs. The FAT is typically written to the outermost track on a disk." (Bill Nelson 2003). Reason of this process is to find and clarify, Drive Slack, which is any space not used by an active file.

RECOVER DATA

The actual examination of the disk is next. After analyzing a disk, it is time to retrieve deleted files and e-mail for items that have been purposefully hidden or deleted. However, before an investigator starts to look for hidden or deleted data, it is advisable to analysis the user created data (available data) first that is available and accessible to investigator. This includes email, message, word-processing document, spreadsheets, database, electronic calendars, etc.

In addition free space and file slack have to be examined as well. Free space is the available storage space on the disk. This includes space where files may have resided at one time, but have since been deleted. These areas can provide key evidence to a case.

Below are a number of areas on the disk that might contain data and must be examined:

- Slack space: The space between the end of data and the end of a block of the file system which may contain fragmented or deleted data. For example, if a cluster is 512 bytes and a file only uses 312 bytes, the free or unallocated space is 200 bytes.
- Swap file: a hidden windows system file named Pagefile.sys that is used for virtual memory.
- Unallocated Space: When data is deleted, it is a reference to the data within file allocation table that is actually deleted. This means, the data may still exist on the storage medium, but the OS will not know how to access this data.
- Unused Partitions: Space that is allocated and formatted, but does not appear to contain any data.

- Hidden Partitions: Hidden space that might contain unallocated space that may also deliberately hide data.

To recover and examine data, many software tools have been developed to assist the computer forensics examiner. Below are just a few of the tools on the market:

- AcoDisk- A CD recovery tools
- Encase-Forensic software application that manages and enables viewing of all evidence
- Coroners Toolkit- a kit of UNIX and Linux data collection and analysis tools
- DtSearch- A Keyword indexer and search tool
- GetSlack- a tool that collects all available disk space on a particular driver, saves it to a location and makes it available for analysis with other tools
- Net Threat Analyzer- A Tool that identifiers past internet activity; examines windows swap file and reveals evidence of browsing activity.

DATA ANALYSIS SECOND PHASE

SIGNATURE ANALYSIS

Every file has a signature, which define what type of file it is so that a program can properly recognize and associate it. Some users are known to deliberately change the file extensions to hide data. When a file extension is changed the file signature does not change, but most programs won't recognize what the file is. To make sure there is no data that is being surreptitiously stored, it is important to perform signature analysis.

In this process, an investigator must have a file signature table, or use a tool which can help them to identified file signatures. Below, are four possible answers which may come out as a result:

- Bad Signature, means the extension, exists in the file signature table but header is incorrect, and the header is not in the file signature table.
- Match, means the header matches the extension
- Alias means the header is in the file signature table and the extension is incorrect. This is an indication that the extension of file has been changed.
- Unknown, means neither the header nor the file extensions are listed in the signature table.

HIDDEN DATA PLACEMENT

Components of the complex system that compose a computer offer many places for data to be hidden. There are many places that criminals can use to hide data. One of the most popular methods which have been used is Steganography. Several open source tools are available that make it possible for anybody to hide messages of many types in graphics and images. All one needs is the identical tool on the sending and receiving ends of transmissions and data can be sent undetected.

The first phase of work in this area is to find hidden data in images and graphic data streams. The goals expressed most often by researchers are to enhance and develop techniques in the following categories:

- Image quality standards: Detailed metrics regarding resolution, size, aspect ratio, color pallet, and other graphic attributes applies to images of differing types. The standard metrics could then be analyzed and compared to those narrow tolerances known or expected for that graphics type. Any flaw or potential compromise would be detected. (DFRWS 2001).
- Blind detection, This is a method that doesn't require an original graphic for comparison. Comparing copies of suspect digital imagery or graphic data representations with known uncompromised versions of the same is a common detection method. To be fully successful this would require a repository that contained all the images in existence. Instead, researchers are looking for ways to analyze the structural elements of the data to look for anomalies. (DFRWS 2001).
- Watermarking: The same concept is being studied and researched to decide if it can be reliably applied to the digital data we create and transmit daily. The digital watermark would have a highly detailed and defined structure that could be analyzed to assess if any compromise has occurred for that digital product. (DFRWS 2001).

OBTAINING MAGNETIC RESIDUE DATA

In some cases an investigator needs to recover some data that has been overwritten; because the hard disk or other storage devices are physical devices, and they consist of a stack of storage covered in magnetic material which stores the pattern of 1's and 0's that make up the data. When a track is overwritten with new data, traces of the old data remain underneath. This is due to the inability of the writing device to write in exactly the same location each time, and partially due to the variation in magnetic media sensitivity and field strength over time and among devices. Specialized equipment is needed to recover some of the overwritten layers through the use of magnetic force microscopy (MFM). MFM creates patterns of the magnetic data on the disk. Thus, any traces of old data will appear on the image of the patterns. The numbers of layers that can be read depends on the sensitivity of the instrument used to perform the MFM. Generally know that these machines can read the first two layers quite easily.

CHECKLIST

After all processes have been done, it time to double check all the processes, it is a good idea to use a checklist to go through all the processes one more time. (Appendix B)

CONCLUSION

This paper has examined many steps which have been taken during the analysis phase during an investigation. Any forensics professional, which is involved with an investigation case, should be familiar with basic technology involved in how to gather data from digital evidence, and how to ensure that the information will be presented as evidence to law enforcement.

Moreover, computer forensics is a new field that will continue to grow, especially as computer technology becomes more involved with our life.

Case Number: The number assigned by your organization when an investigation is initiated.

Investigation organization: The name of your organization.

Investigator: The name of the investigator assigned to the case. If many investigators are assigned, insert the lead investigator's name.

Date & Time: The date and time

Operation Case No: This is the each operation case number.

Operation: This Parameter specifies one of the two values.

- Copy indicates that the evidence under investigation analysis is to copy from the original evidence device directly to the destination device.
- Image specifies that the evidence under investigation analysis is to copy from the source device to an image file and then restore the image file to the destination device.

Src and Dst: The type of disk access interface for the source and destination drove is specified.

Rel Size: The relationship between the source and destination is specified.

Appendix A

Mirror Copy Form						
Investigation Case No				Date		
				Time		
Investigator				Investigation Organization		
Operation Case No	Operation	SRC	DST	Real Size	Partition	Operation Errors
Operation Tools						
Operation Record By						
Time				Signature		

Appendix B

Checklist for Electronic Media Analysis
Assign a unique number to each piece of media.
Write-Protect all media
Make a mirror copy. Restore each piece of media to a file with a name that corresponds to the number assigned to the media being restore
Verify that all files on the directory listing appear in the copy restored
Compare the copy evidence's Hash value with Original hash value
Viruses check all media. Record any viruses discovered and immediately notify the producing party.
Secure the source media
Analyze Operation system and file structure
Retrieve deleted file, e-mail, hidden file, hidden partition, swap file, and slack space
Check Files signatures
Check all Hidden data replacements possibility
Document and print all the evidence has been found
Prepare your report

Partition: Specification of a partition type indicates that the digital evidence is in a partition.

Operation: The parameter value no spec indicates a full disk operation.

Errors: A value of Error indicates that a corrupted image file should be used.

About the Author

Ali Fazeli has been involved in the information technology field since his younger days. He is a highly interactive and innovative trainer, whose work is mainly based on research and applied knowledge from extensive experiences in Computer Security and Data Recovery. He has extensive experience in digital forensic investigation, anti-forensic and cyber war. During his 10 year career, Ali has assisted and trained numerous information technology security professionals in Asia. He is accredited as a specialist in the successful investigation and prosecution of fraudsters and other criminals from the dark underworld of the internet. One of the few areas of Ali's expansive expertise lies in lecture deliveries. With a vast knowledge of various IT security sectors, Ali is competent in lecture deliveries on topics such as: Computer Forensics Investigation and Anti forensics techniques, Penetration Testing, Investigating IT Misuse, Wireless Network Hacking, Enterprise Network Security, Social Network Forensics.

INTRODUCTION TO NETWORK FORENSICS USING WIRESHARK

by Dauda Sule

Network forensics involves recording, monitoring, capturing and analysis of network traffic in a bid to uncover how incidents occurred (like a breach, attack, abuse or error). Network data is highly volatile and may be easily lost if not captured in real-time; for example, if malicious code is sent to an endpoint, the source or path of the code would be difficult to discover if the traffic data was not captured as it was coming in through the network. There are various tools that can be used to capture and analyze network traffic such as NetworkMiner, tcpdump, snort, windump and Wireshark. This article introduces the use of Wireshark for network analysis.

What you will learn:

- Definition of network forensics
- Basic understanding of network forensics
- Basic network analysis using Wireshark

What you should know:

- Basic understanding of computer networks
- How to operate computer applications and software
- Basic understanding of TCP/IP

Wildpackets (2013) defined network forensics as "the process of capturing, storing, and analyzing network events", data of which can be used to solve network breaches, improve network performance and identify rogue activity. The site further states that network forensics can be used to monitor users and devices, identify sources of data loss and points of security breaches, analyze business transactions and point out the origin of intermittent network issues. Activity monitoring can help identify abnormal traffic, like a change in the network use pattern of a particular endpoint which might signify something is wrong. Network logs from network control mechanisms like routers and firewalls usually provide a good source of digital evidence (Casey, 2004).

Wireshark is an open source network sniffer and protocol analyzer. A packet sniffer is a passive tool used to capture messages being transmitted to and from a system over a network. It is passive because it only monitors and records packets being sent and received on a system, not sending or receiving directly any itself nor interfering with the packets (Kurose and Ross, 2009). Rather what it captures are copies of packets moving within protocols and applications on the system. Wireshark can be used to troubleshoot network problems, examine security problems, debug protocol implementations, and to understand internals of network protocols (Lamping, Sharpe and Warnicke, 2013). Wireshark captures network traffic from both wired and wireless Ethernet networks; however, it does not capture traffic from mobile net-

work dongles on Windows (at least for now). A list of networks that Wireshark can and cannot capture is available here: *http://wiki.wireshark.org/CaptureSetup/NetworkMedia*.

It should be noted that network analysis tools like Wireshark can be used both positively and negatively; network administrators, network security personnel and investigators, and so on use them for troubleshooting, debugging, investigating intrusions and the like, but malicious persons can use it to monitor, spy on and gather reconnaissance data on potential victims. Wireshark is available for free download from the Wireshark website (*http://www.wireshark.org/download.html*).

In an investigation, into a network breach for example, a network sniffer can be used to analyze captured network traffic to discover the path that the intruder followed to get into an organization's network. The network sniffer is able to view the IP address that the intruder used to get in to the network, which can be a starting point for the investigation, even though that may not be a smoking gun. The entry of a malicious code like a network worm can be traced using a network sniffer; it can be used to trace how it got onto the network: could have been downloaded from an endpoint then spread, or could have originated from an endpoint not via download – that could imply infection from a storage device like a thumb drive. Leakage of sensitive data to a competitor could be traced or discovered with a network sniffer by discovering its movement from an IP address in the organization's network to an external IP address. The preceding are just a few basic examples of what network sniffers can be used to uncover whether through analysis of already captured and stored network traffic or live monitoring.

CAPTURING NETWORK TRAFFIC

Following is the use of Wireshark version 1.8.6 on a Windows system to capture and analyze traffic over a network (a wireless network). The wireless access point being a smart phone, and the endpoint a Windows-based Laptop.

Once installed, run Wireshark. The Graphic User Interface as shown in Figure 1 comes up.

The interface is quite user friendly with a variety of options like user guide, help, opening previously captured files, and so on as is visible from Figure 1. Actions can be carried out from the file menu bar, and for some actions shortcuts below the file menu bar and on the interface page (like the starting network capture). For example, clicking on the "Interface List" option under "capture" on the page can be used to view the available network interfaces on the computer whose traffic Wireshark can capture. Once clicked it shows the available network cards and the packets that are sent and captured on them, selecting an available network interface by clicking the checkbox to its immediate left activates the option to start network capture on it (note: there is only one network interface on the system used for this illustration, hence only one card is available in the option). Options for capturing packets can be edited by clicking the options button. Clicking on the details button pops up information about the network interface; like the vendor, the status of the network (connected or disconnected), the throughput and so on, as shown in Figure 2.

A live packet capture can be started by any of the following:

- selecting the required network interface and clicking the start button in the "Interface List" as described above;
- by selecting the desired interface on the main page then clicking "start" above it;
- clicking "capture" on the file menu bar then click "start";

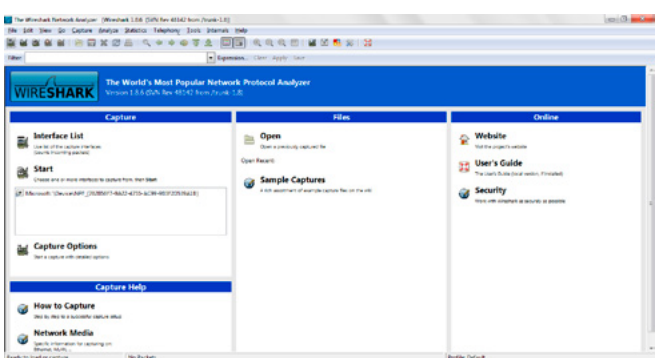


Figure 1. *Initial view when Wireshark is run*

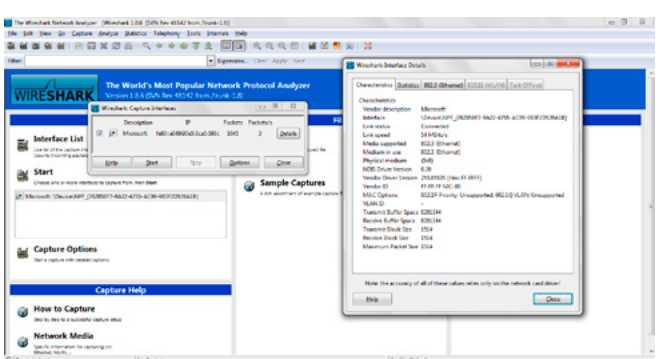


Figure 2. *Interface list showing details of network card*

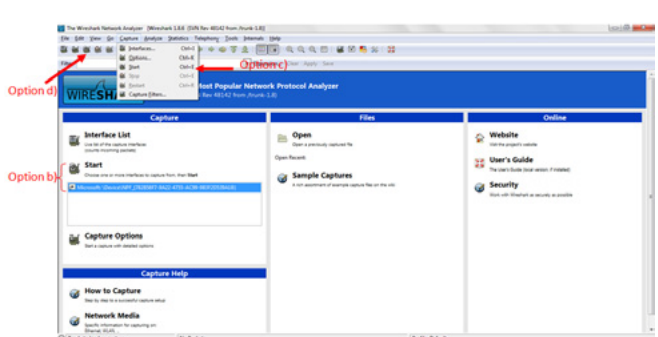


Figure 3. *Capture start options*

- clicking the start shortcut on the bar below the file menu bar;
- using [Ctrl + E] keyboard shortcut.

Locations of options b) to d) are depicted in Figure 3.

Once the packet capture is initiated from any of the above mentioned options, Wireshark starts capturing packets as depicted in Figure 4. The main subdivisions of the interface follow:

- The command menus: located at the topmost, these are made up of the file menu bar and capture menu bar. The file menu bar is a normal file bar, while the capture menu toolbar consists of capture shortcuts which can be gotten from the file menu.
- Packet filter toolbar: this is just below the capture menu bar. It is used to filter the type of packets information displayed in the packet list pane; for example based on protocol, this makes it possible to display only packet data of the selected protocol.

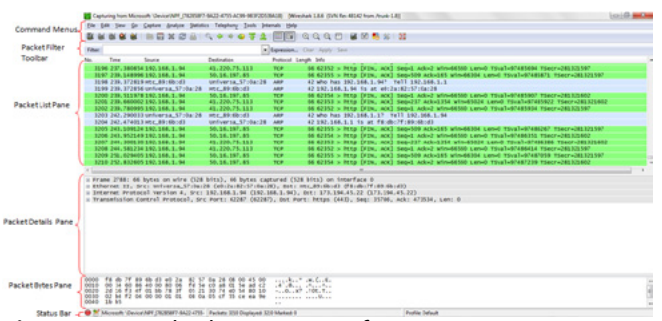


Figure 4. *Wireshark capture interface*

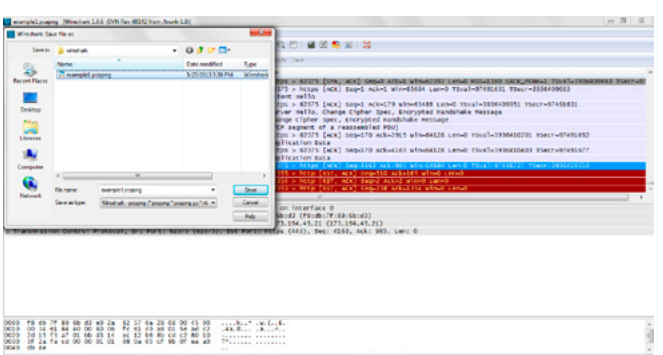


Figure 5. *Saving a packet capture*

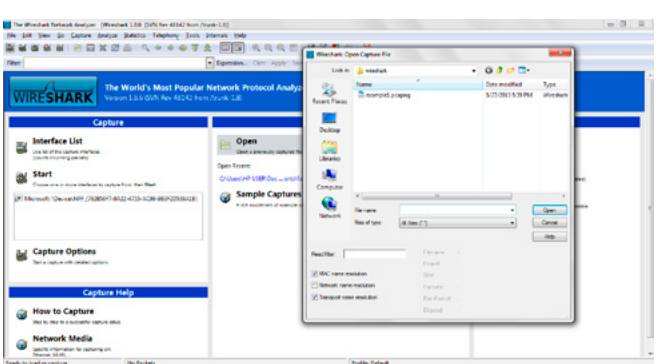


Figure 6. *Opening a stored packet capture*

- Packet list pane: this displays the summary of packets captured each in a row. It shows in each row the Wireshark assigned frame number for each packet, the time the packet was captured, the source and destination addresses, the type of protocol, the length and information pertaining to the protocol type.
- Packet details pane: this shows detailed information on any selected packet in the packet-listing window. Any packet selected by clicking on it in the packet-listing window will have displayed in the packet-header details window details of the Ethernet frame, the Internet Protocol, and other protocol details (like TCP, UDP) depending on the protocol of the selected packet. Each of these can be expanded to show further details.
- The packet bytes pane: this shows all the contents of the captured frame in ACSII and hexadecimal format.
- The status bar: shows some details regarding the state of Wireshark and the packets captured.

The packet capture can be stopped from the file menu bar by clicking capture then stop from the drop-down; the stop button on the capture menu bar (the fourth from the left); or hitting Ctrl + E on the keyboard again. The captured packets can then be analyzed immediately or saved till later. The packet capture is saved just as any normal file is saved (Save, Save As, the floppy disk icon shortcut), as shown in Figure 5. A saved packet capture, or captured network traffic stored in logs, can be retrieved and analyzed by opening the file from the directory in which it is stored. The opening is done like any normal document opening from the file menu, or folder icon shortcut, or the “Open” shortcut in the middle of the initial interface page, to retrieve the captured packet file from the location it is stored (Figure 6).

ANALYSIS OF CAPTURED PACKETS

The time a packet was captured is viewable under the time column in the packet list pane. The time display is set by default to the number of seconds from beginning of a capture, which can be adjusted as required using the view option from the file menu bar. From the view option, move the cursor to “Time Display Format”, which will give a drop down list of options, UTC date and time of the day format is chosen in Figure 7. This enables one to know the time (UTC) and date a specific packet was captured. (Note: the UTC date and time of the day format was chosen just for illustrative purposes, it’s not a requirement). If a packet is of particular interest (especially when analyzing an archived network log), knowing the time it was received/sent on a network can help identify who was responsi-

ble, for example if an endpoint is shared by employees working in shifts. The timing can be very useful in an investigation, the time packets were transferred over a network (whether local or UTC or otherwise) on the suspect endpoint is available on the captured network log, this can be used to verify/nullify a suspect's alibi, even more so if combined with CCTV footage or eye-witness accounts. A suspect in a workplace may try to make it look like an infraction took place at a time when he was off-duty, trying to exonerate himself from the infraction, but the logs can reveal the time such an infraction took place, which when combined with the time the suspect was on or off duty can reveal the truth of the matter.

The filter toolbar can be used to select packets based on type of field or protocol. For example, TCP, HTTP, DNS can be criteria for filtering, which will display packets with such criteria in the packet list pane. This is achieved by typing in the criteria in the filter toolbar and clicking on apply. Wireshark is case sensitive and requires that the characters for the filtering criteria be entered in lower case. Figure 8 shows the packet list pane showing filtered results for DNS. This allows the analyst to view and analyze DNS related packets.

To view details and analyze a network packet, the packet is clicked on in the packet list pane, making it highlighted. The time the packet was sent from one IP address to another can be seen under the "Time" column. The IP address from which it was sent, and the one which received it, are visible under the "Source" and "Destination" columns respectively. The protocol type is visible under "Protocol", length shows the packet size in bytes, and information gives a general description of the packet. In Figure 9, the packet selected has a Wireshark frame number of 916 in the capture; it was captured at 20:19 (8:19 PM) on 28th May 2013, sent from IP address 192.168.1.94 to IP address 192.168.1.1, is a ninety-one bytes long DNS packet, and was a standard query.

The selected packet can be further scrutinized in the packet details pane. For this particular packet, you can view details of the Frame, Ethernet II, Internet Protocol version, the transfer protocol (UDP in this case) and DNS; each of them is expandable for full details. For example, expanding the Frame gives further details pertaining to the frame, like the actual time the frame was captured, the frame number, the packet length (in bytes and bits), the protocols in the packet, and so on. The status bar indicates what each detail represents if the detail is clicked on. Figure 10 shows the expanded Frame details in the packet details frame. Ethernet II shows the source port and the destination represented as "Src" and "Dst" respectively. In this example, Ethernet II indicates as below:

```
Src: Universa_57:0a:28 (e0:2a:82:57:0a:28),
Dst: Htc_89:6b:d3 (fb8:d3:7f:89:6b:d3)
```

That means the source of the packet is `Universa_57:0a:28` (which is the endpoints network card), and the destination is `Htc_89:6b:d3` (the destination, in this case a smart phone wireless hotspot). The hexadecimal figures in brackets after both the source and destination represent their MAC (Medium Access Control) addresses in 48-bit – that is the network card's 48-bit MAC address is `e0:2a:82:57:0a:28`, while that of the wireless hotspot is fb8:d3:7f:89:6b:d3. (Note: the 48-bit MAC addresses are in hexadecimal format, the first six digits identify the vendor – called the Organizationally Uniqiue Identifier, OUI – the last six digits represent the MAC's serial number). Once Ethernet II is expanded, it is divided into the destination and the source which are also expandable. Selecting any component of the expanded destination or source highlights the

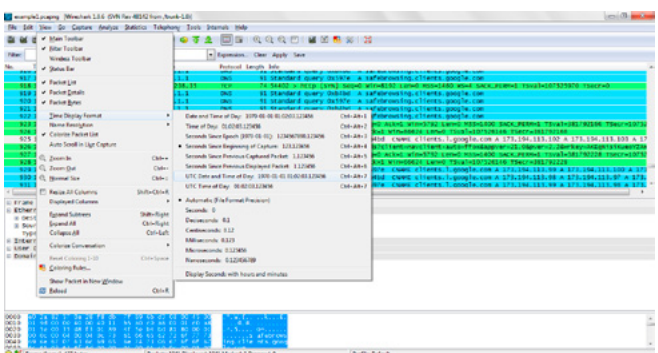


Figure 7. *Changing time display format*

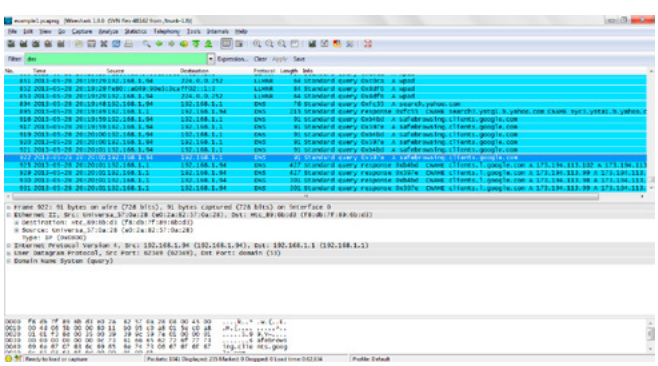


Figure 8. *Filtered DNS results*

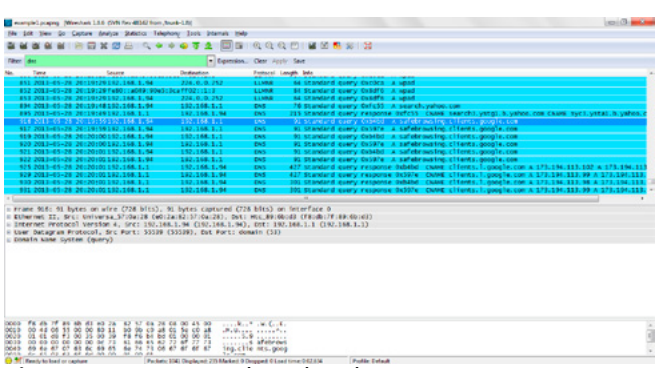


Figure 9. *Viewing a selected packet*

bytes representing such in the packet bytes pane as depicted in Figure 11.

In the Internet Protocol field under Ethernet II in the packet details pane, the version of Internet Protocol, source IP and destination IP address are visible. Upon expansion, it is further broken down into “Differentiated Services Field”, “Flags”, and “Header Checksum”, each giving further information and expandable. It can be observed from Figure 12 that the packet has version 4 of Internet Protocol and the header length is 20bytes. The source and destination IP addresses are also visible as was seen in the packet list pane: 192.168.1.94 and 192.168.1.1 respectively. The User Datagram Protocol field shows the source and destination port numbers, once expanded checksums can be viewed if available. Figure 13 shows the source port as 55539 and the destination port as domain or 53 (port 53 is the default port for Domain Name Server – DNS – protocol), and checksum unavailable.

Ports are used to direct different types of network traffic to specific programs that handle them (SYBEX Inc., 1998). Touch et al (2013) indicated that ports are assigned in different ways based on three ranges viz: system ports – 0 to 1023; user ports – 1024 to 49151; and dynamic and/or private ports – 49152 to 65535. Some common default ports are:

- Port 21 for FTP
- Port 23 for Telnet
- Port 25 for SMTP
- Port 53 for DNS
- Port 80 for HTTP, World Wide Web
- Port 110 for POP

Traffic direct to and/or from a particular port can be used to determine what type of traffic was transferred, for example, traffic on port 25 would most likely be e-mail related. Also, when looking for a particular type of traffic, for example Internet traffic, analysis could be narrowed down to port 80. It should be noted, however, that these ports can be changed; that might be used by an intruder as a way of masking his activities. It is also possible for an organization to use different port numbers than the default for protocols, probably for administrative reasons. Hence, one should have it at the back of one’s head that the port number might not have been used in default form when carrying out an investigation.

Casey (2004) mentioned a case in which a disgruntled staff of an organization configured his endpoint with the organization’s CEO’s IP address and used that to send offensive messages – giving the impression that such messages were sent by the CEO. Investigation of network data showed that the CEO’s address was temporarily set with a diffract MAC address from the CEO’s, the MAC address was discovered to be that of the disgruntled staff. Reviewing captured packets on Wireshark reveals both IP address and MAC addresses used to send and receive a packet, closer review can determine if the IP address used is the one that was allocated to a specific endpoint or not by com-

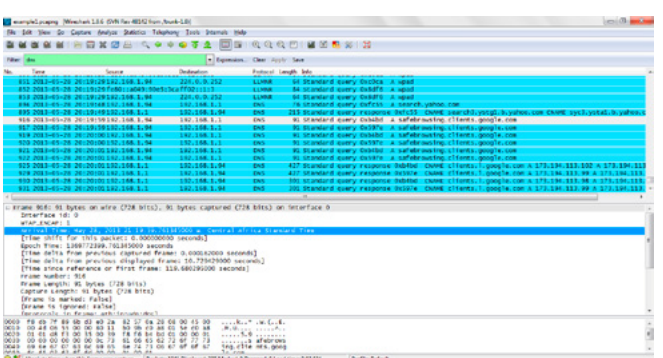


Figure 10. *Expanded frame details in the packet details frame*

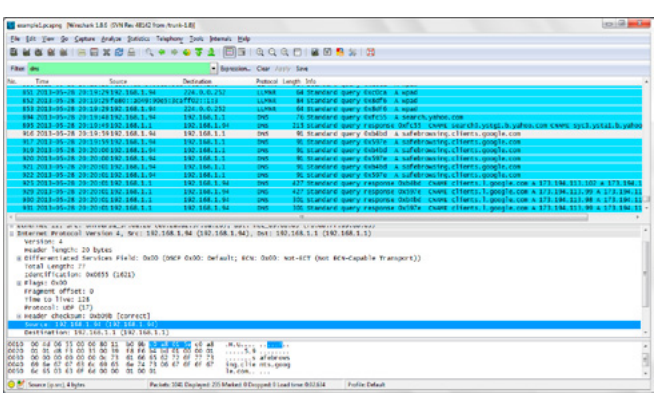


Figure 12. *Internet Protocol field expanded*

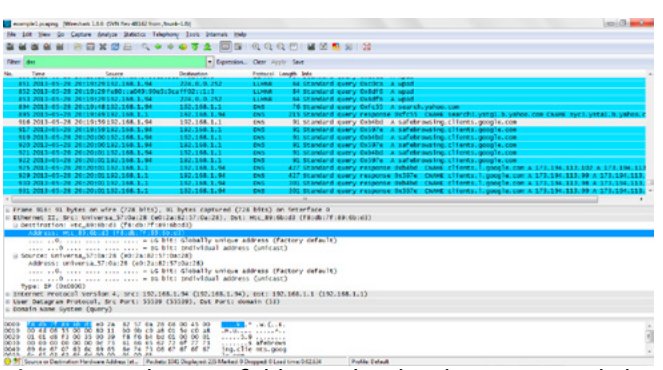


Figure 11. *Ethernet II field in packet details pane expanded*

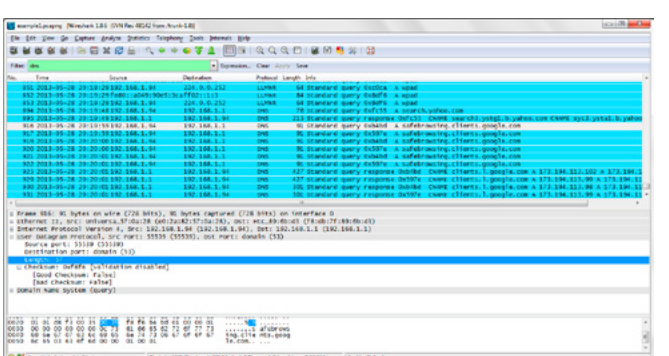


Figure 13. *User Datagram Protocol field expanded*

paring it with the MAC address. That can help to detect an IP spoofing attack.

Under the Domain Name System field, you have the flags and query which are both expandable. The field shows that the packet is a standard query to host address safebrowsing.clients.google.com, with transaction ID 0xb4bd (Figure 14); all which are visible in the information column of the packet list pane. It also shows that the query was responded to in frame number 930. A quick review of items in the packet details field of frame number 930 shows it is a response packet, source and destination from frame 916 are reversed and in the Domain Name Server field, it refers to the request being from frame 916. This confirms that frame 930 is the response to request in frame 916; hence in this case the source is 192.168.1.1, and destination 192.168.1.94.

ANALYZING HTTP PACKETS

Start a Wireshark packet capture and then launch a browser or refresh an already open web page. In this example, an already open Google home page was refreshed. You can stop the packet capture once the web page has loaded. Filter out HTTP packets by entering "http" (in lower case and without quotation marks) into the Filter toolbar and clicking on apply. The first packet after filtering in the packet list as can be observed in Figure 16 shows the packet was captured 14:09 UTC on 5th June, 2013 with frame number 33, the source IP address being 192.168.1.94 and the destination IP address 173.194.41.215. The protocol is of course HTTP, the packet having a length of 571 bytes. The information column describes the packet as GET / HTTP/1.1, meaning it is a request to retrieve HTTP data. The source IP address is known to be the endpoint's IP address, while the destination IP address is for a web site. The destination web site can be figured out in the packet details pane, and using an IP address translator (IP address translators are available online and can be gotten using a search engine). 173.194.41.215 is an IP address for google.com, which is gotten using an IP address translator, and will be seen in the packet details pane.

A quick look at the packet details pane shows the frame number is 33 and that the packet contains 571 bytes of data; Internet Protocol was Version 4; and TCP source port was 60829 and destination port 80 (which is the default HTTP port). An expansion of the Hypertext transfer Protocol field reveals the language for the packet is US English (en-US); the packet is compatible with Mozilla 5.0, Microsoft Internet Explorer 10.0; and the website *www.google.com.ng*. The next packet in the example is frame number 67. It is quite similar to the previous frame number 33, save it is a GET image request. Hence, its description in the packet list pane is GET /images/srpr/logo4w.png HTTP/1.1 in the Hypertext Transfer Protocol field, details show that it accepts images in PNG.

In the packet that follows (frame number 68), in the Line-based text data under the Hypertext

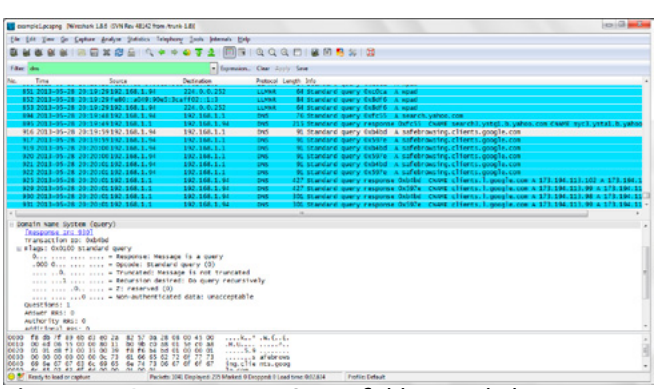


Figure 14. *Domain Name Server field expanded*

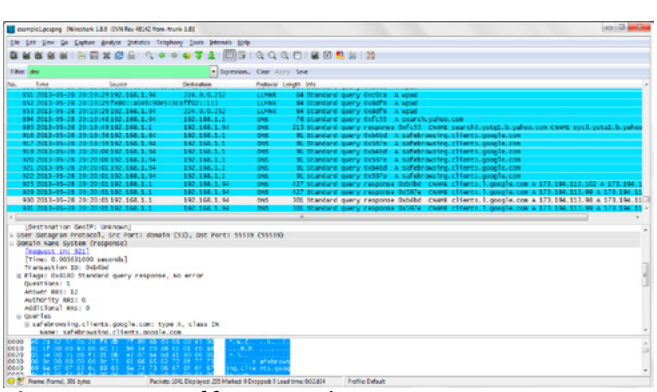


Figure 15. *View of frame number 930*

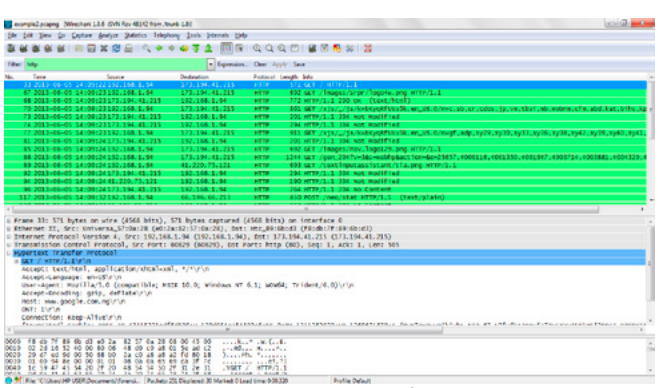


Figure 16. *Packet capture with HTTP filtered*

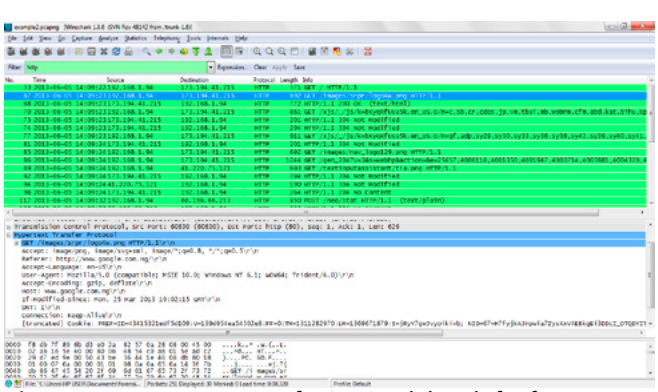


Figure 17. *Hypertext Transfer Protocol details for frame number 67*

Transfer Protocol field, the HTML script for the packet is displayed as shown in figure 18. The script can be used to reconstruct the web page.

Knowing the HTTP data can help identify which websites were visited and what was downloaded; this can help in tracing the source of a problem like malware or a slow network. Identifying pornographic websites or free download/torrent sites on the network can show the problem resulted from visiting such websites, which can further be traced to an endpoint. In the event a user accesses an unauthorized website using a browser in private browsing mode (in a bid to cover his/her tracks), Wireshark can be used to analyze network logs to identify the breech – the unauthorized website's HTTP data will be on the network logs, and the IP address of the endpoint that accessed such a site can be revealed. That is possible because the use of private browsing mode only removes traces from the browser, but cannot affect the network log. The packet sniffer can be used to reveal what sites were visited by a suspect which could be smoking gun evidence; it could, for example, provide breakthrough evidence in a child pornography case where it is discovered that a suspect visited site related to child pornography.

ANALYZING ETHERNET AND ARP DATA

The Address Resolution Protocol (ARP) is used to get the MAC address of a specific IP address. For example, when an endpoint is sending a packet to a destination host, it only has the destination IP address; hence as it sends the packet, it asks which host has the IP address, and the response form the destination states the MAC address. *Dynamic Host Configuration Protocol* (DHCP), on the other hand, uses MAC addresses to assign IP addresses to endpoints that are authorized on a network. Where DHCP is used, an unauthorized MAC address will not get an IP address assigned to it automatically (Casey, 2004). DHCP logs can be used to retrieve a MAC address that was assigned a specific IP address within a particular time frame, this can determine which endpoint was used to carry out a specific action based on packets captured.

In order to analyze Ethernet and ARP data, IP protocols view may be disabled. This is done by clicking Analyze in the File command bar, then clicking on Enabled Protocols and unchecking IP Version 4 and 6 (shown in Figure 19) which results in the look of the interface changing.

We can refer back to packet with frame number 67. In this interface, source and destination addresses are not stated in IP (as IP protocols have been disabled for the view); rather they are indicated by name and hexadecimal – that is source: `Universa_57:0a:28`, and destination: `Htc_89:6b:d3` – in the packet list pane. Within the packet details pane, under the Frame field the date and time of packet arrival is stated based on the time zone of the endpoint (June 5 2013, 15:09 West/Central African Time). The frame number and length are also available under the Frame field. Figure 20 depicts the Frame filed.

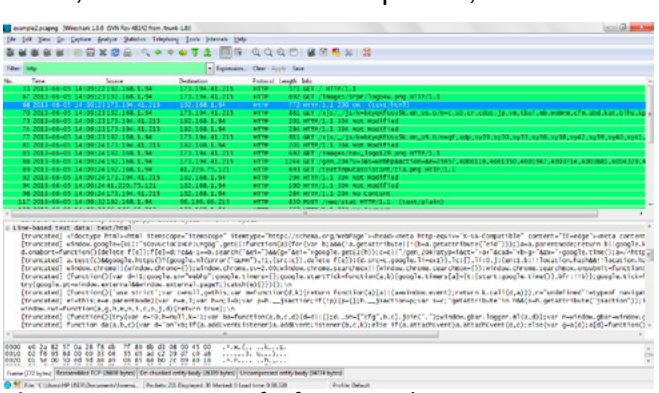


Figure 18. *HTML script for frame number 68*

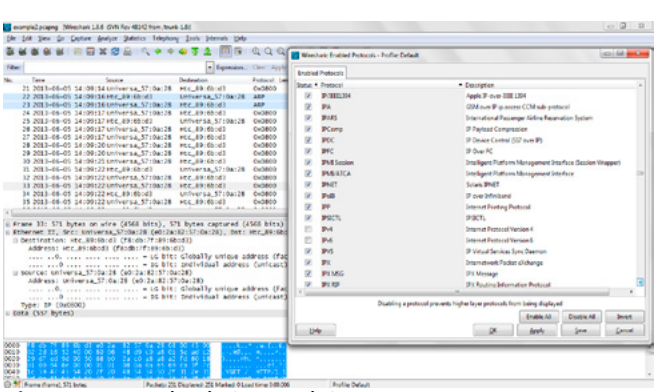


Figure 19. *Uncheck IPv4 and IPv6*

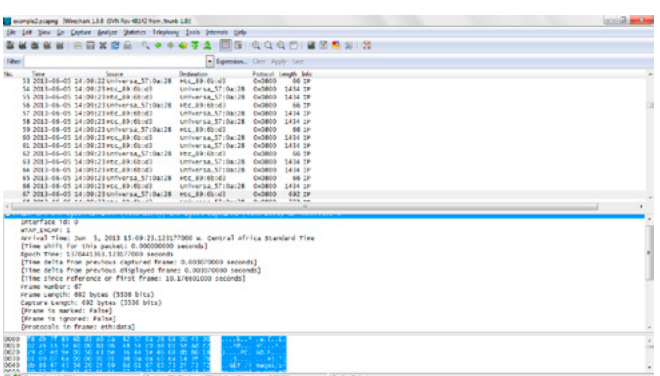


Figure 20. *Captured packet interface with IPv4 and IPv6 disabled and packet frame 67 selected*

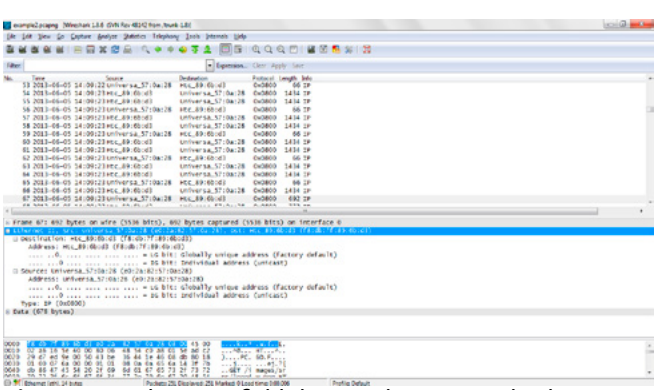


Figure 21. *Ethernet II field showing bytes in packet bytes pane*

Listing 1. *Print out of packet frame 7*

```
No.   Time              Source      Destination Protocol Length Info
67 2013-06-05 14:09:23.123177000 Universa_57:0a:28 Htc_89:6b:d3 0x0800 692       IP
Frame 67: 692 bytes on wire (5536 bits), 692 bytes captured (5536 bits) on interface 0
Ethernet II, Src: Universa_57:0a:28 (e0:2a:82:57:0a:28), Dst: Htc_89:6b:d3 (f8:db:7f:89:6b:d3)
Destination: Htc_89:6b:d3 (f8:db:7f:89:6b:d3)
Address: Htc_89:6b:d3 (f8:db:7f:89:6b:d3)
.... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
.... ...0 .... .... .... .... = IG bit: Individual address (unicast)
Source: Universa_57:0a:28 (e0:2a:82:57:0a:28)
Address: Universa_57:0a:28 (e0:2a:82:57:0a:28)
.... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
.... ...0 .... .... .... .... = IG bit: Individual address (unicast)
Type: IP (0x0800)
Data (678 bytes)
0000 45 00 02 a6 16 5e 40 00 80 06 48 54 c0 a8 01 5e E....^@...HT...^
0010 ad c2 29 d7 ed 9e 00 50 43 be 36 44 1e 46 08 db ..)....PC.6D.F..
0020 80 18 01 00 07 6a 00 00 01 01 08 0a 0a 65 6a 14 .....j.......ej.
0030 3f 7b db 86 47 45 54 20 2f 69 6d 61 67 65 73 2f ?{..GET /images/
0040 73 72 70 72 2f 6c 6f 67 6f 34 77 2e 70 6e 67 20 srpr/logo4w.png
0050 48 54 54 50 2f 31 2e 31 0d 0a 41 63 63 65 70 74 HTTP/1.1..Accept
0060 3a 20 69 6d 61 67 65 2f 70 6e 67 2c 20 69 6d 61 : image/png, ima
0070 67 65 2f 73 76 67 2b 78 6d 6c 2c 20 69 6d 61 67 ge/svg+xml, imag
0080 65 2f 2a 3b 71 3d 30 2e 38 2c 20 2a 2f 2a 3b 71 e/*;q=0.8, */*;q
0090 3d 30 2e 35 0d 0a 52 65 66 65 72 65 72 3a 20 68 =0.5..Referer: h
00a0 74 74 70 3a 2f 2f 77 77 77 2e 67 6f 6f 67 6c 65 ttp://www.google
00b0 2e 63 6f 6d 2e 6e 67 2f 0d 0a 41 63 63 65 70 74 .com.ng/..Accept
00c0 2d 4c 61 6e 67 75 61 67 65 3a 20 65 6e 2d 55 53 -Language: en-US
00d0 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20 4d 6f ..User-Agent: Mo
00e0 7a 69 6c 6c 61 2f 35 2e 30 20 28 63 6f 6d 70 61 zilla/5.0 (compa
00f0 74 69 62 6c 65 3b 20 4d 53 49 45 20 31 30 2e 30 tible; MSIE 10.0
0100 3b 20 57 69 6e 64 6f 77 73 20 4e 54 20 36 2e 31 ; Windows NT 6.1
0110 3b 20 57 4f 57 36 34 3b 20 54 72 69 64 65 6e 74 ; WOW64; Trident
0120 2f 36 2e 30 29 0d 0a 41 63 63 65 70 74 2d 45 6e /6.0)..Accept-En
0130 63 6f 64 69 6e 67 3a 20 67 7a 69 70 2c 20 64 65 coding: gzip, de
0140 66 6c 61 74 65 0d 0a 48 6f 73 74 3a 20 77 77 77 flate..Host: www
0150 2e 67 6f 6f 67 6c 65 2e 63 6f 6d 2e 6e 67 0d 0a .google.com.ng..
0160 49 66 2d 4d 6f 64 69 66 69 65 64 2d 53 69 6e 63 If-Modified-Sinc
0170 65 3a 20 4d 6f 6e 2c 20 32 35 20 4d 61 72 20 32 e: Mon, 25 Mar 2
0180 30 31 33 20 31 39 3a 30 32 3a 31 35 20 47 4d 54 013 19:02:15 GMT
0190 0d 0a 44 4e 54 3a 20 31 0d 0a 43 6f 6e 6e 65 63 ..DNT: 1..Connec
01a0 74 69 6f 6e 3a 20 4b 65 65 70 2d 41 6c 69 76 65 tion: Keep-Alive
01b0 0d 0a 43 6f 6f 6b 69 65 3a 20 50 52 45 46 3d 49 ..Cookie: PREF=I
01c0 44 3d 34 33 34 31 35 33 32 31 65 64 66 35 64 31 D=43415321edf5d1
01d0 30 39 3a 55 3d 31 33 39 64 39 35 34 65 61 35 34 09:U=139d954ea54
01e0 35 30 32 65 38 3a 46 46 3d 30 3a 54 4d 3d 31 33 502e8:FF=0:TM=13
01f0 31 31 32 38 32 39 37 30 3a 4c 4d 3d 31 33 36 39 11282970:LM=1369
0200 36 37 31 38 37 39 3a 53 3d 6a 52 79 59 37 67 77 671879:S=jRyY7gw
0210 4a 76 79 6f 69 6b 69 76 62 3b 20 4e 49 44 3d 36 Jvyoikivb; NID=6
0220 37 3d 48 37 66 79 6a 6b 41 4a 6e 70 77 69 61 37 7=H7fyjkAJnpwia7
0230 5a 79 73 58 78 56 34 45 42 6b 67 45 69 33 44 44 ZysXxV4EBkgEi3DD
0240 63 49 5f 4f 54 51 44 59 49 54 69 65 51 48 79 34 cI_OTQDYITieQHy4
0250 4d 7a 53 71 43 57 35 57 47 74 68 67 58 71 6e 53 MzSqCW5WGthgXqnS
0260 38 69 6b 65 41 64 70 70 7a 33 53 77 47 39 34 43 8ikeAdppz3SwG94C
0270 73 6b 6d 51 66 47 6c 47 68 35 76 78 4a 79 53 58 skmQfGlGh5vxJySX
0280 34 63 6f 55 70 72 57 45 70 6d 2d 51 61 35 37 2d 4coUprWEpm-Qa57-
0290 35 54 69 64 39 74 73 74 64 78 48 41 48 59 4d 4d 5Tid9tstdxHAHYMM
02a0 70 78 0d 0a 0d 0a px....
```

In the Ethernet II field, the destination and source addresses are stated with the full 48-bit Ethernet addresses in brackets. The hexadecimal figures for both are visible in the packet bytes pane when the field is selected (Figure 21), and individually when either destination or source is selected.

Details of the packet frame 67 can be printed as shown in Listing 1, summarizing the captured packet. Clicking File and then print brings the dialog box in Figure 22; checking "Selected packet only" radio button ensures only the packet is printed.

Packets involving the *Address Resolution Protocol* (ARP) can be filtered out using the Filter bar. In the example, two such packets are found – ARP packets are normally a request and a reply as is observed in the example. Figure 23 shows the ARP request packet frame number 22, under the ARP field the packet type is in brackets as "request". The packet list pane shows that the source of the packet is the wireless access point with MAC address `Htc_89:6b:d3`, and the destination `Universa_57:0a:28;` and has a length of 42 bytes. The "Info" column has a question "who has 192.168.1.94? Tell 192.168.1.1". That is the ARP trying to resolve the end point's address 192.168.1.94 for the wireless access point 192.168.1.1. In the packet details pane, the target MAC address is stated as `00:00:00_00:00:00` as the address is not yet resolved; hence stated as unknown.

Frame 23 which is the ARP reply packet shows the response to the request in frame 24. The packet list pane shows the source and destination as the reverse of frame 24 as this packet is a response from the end point to the wireless access point; the length is the same 42 bytes. The "Info" column answers the question posed in the previous packet stating: `192.168.1.94 is at e0:2a:82:57:0a:28` – that is 192.168.1.94 belongs to the MAC address of the endpoint (in hexadecimal notation). Hence; under the Address Resolution Protocol field in the packet details pane, the source MAC address of the previous request is now identified and stated as the sender (`Univers_57:0a:28`), along with IP address details. A print out of the packet can also be done as was done for frame number 67 to yield a similar output.

ARP data can identify the endpoints and network interfaces that interacted over a network; this can be used to identify an unauthorized connection within the network. For example, a war-driver on a wireless network can be identified from ARP traffic analysis.

Figure 25 shows a DHCP (DHCPv6) packet filtered out. The Internet Protocol used is version 6, hence the IP addresses are shown in hexadecimal format (Internet Protocol field). The UDP field shows the source port to be 546, which is the DHCPv6 client port; and the destination port 547, the DHCPv6 server port. Under the DHCPv6 field, we can see the client identifier.

SUMMARY

Network forensics is a very important field in the information age. It can be used to monitor users and devices and to track network breaches, troubleshoot and improve network security and performance. It can also be used to track and indict of-

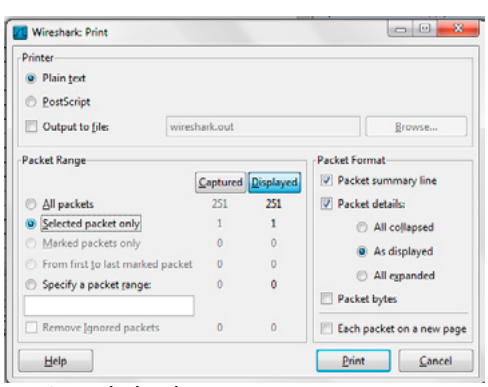


Figure 22. *Print dialog box*

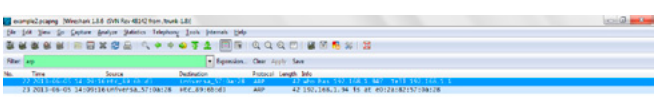
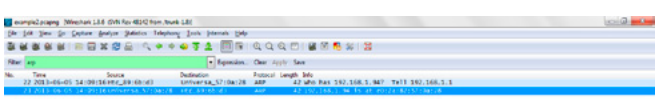


Figure 23. *ARP request packet*



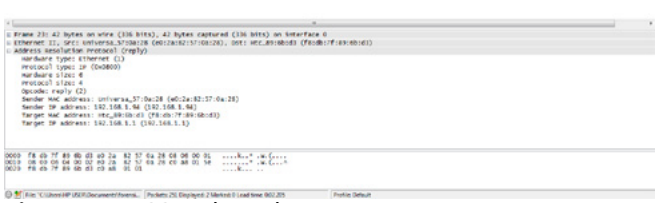
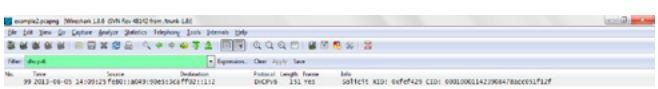


Figure 24. *ARP reply packet*



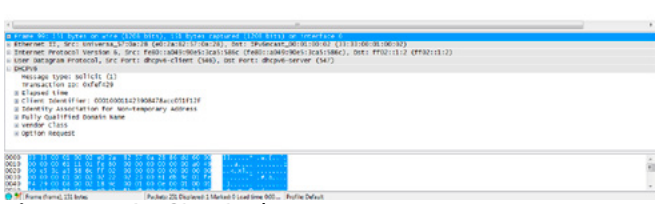


Figure 25. *DHCPv6 Packet*

REFERENCES

- Casey, E. (2004) Digital Evidence and Computer Crime: Forensic Science, Computers and the Internet. 2nd ed. Elsevier Academic press.
- Kurose, J.F. and Ross, K.W.. (2009) Wireshark Lab: Getting Started [Online]. Available from: *http://wps.aw.com/wps/media/objects/7134/7305312/WireShark_Labs/Wireshark_INTRO_Sept_15_2009.pdf* (Downloaded: 16 March 2010).
- Kurose, J.F. and Ross, K.W.. (2009) Wireshark Lab: HTTP [Online]. Available from: *http://wps.aw.com/wps/media/objects/7134/7305312/WireShark_Labs/Wireshark_HTTP_Sept_15_2009.pdf* (Downloaded: 16 March 2010).
- Kurose, J.F. and Ross, K.W.. (2009) Wireshark Lab: Ethernet and ARP [Online]. Available from: *http://wps.aw.com/wps/media/objects/7134/7305312/WireShark_Labs/Wireshark_Ethernet_ARP_Sept_15_2009.pdf* (Downloaded: 13 April 2010).
- Lamping, U., Sharpe, R. and Warnicke, E. (2013) Wireshark User's Guide for Wireshark 1.11 [Online]. Available from: *http://www.wireshark.org/download/docs/user-guide-a4.pdf* (Downloaded: 17 May 2013).
- SYBEX Inc. (1998) Using Port Numbers and Protocols [Online]. Available from: *http://msdn.microsoft.com/en-us/library/aa227632(v=vs.60).aspx* (Accessed: 17 June 2013).
- Touch, J. et al (2013) Service Name and Transport Port Number Registry [Online]. Available from: *http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xml* (Accessed: 17 June 2013).
- Wildpackets (2013) Four Ways Network Forensics Can Help You [Online]. Available from: *http://blog.wildpackets.com/2013/06/06/four-ways-network-forensics-can-help-you.html?goback=.gde_80784_member_247550610* (Accessed: 6 June 2013).

fenders. Wireshark is one of the tools for network forensics which is freely available.

Wireshark has a wide range of uses and interfaces which were not covered in this article, the article merely introduced the basics. Wireshark among other things can also be used to monitor and analyze mobile traffic and VoIP traffic. Packet sniffers come in very handy when analysis of network based evidence is required.

About the Author

Dauda Sule, CISA. He is currently the Marketing Manager of Audit Associates Limited which is a consultancy firm that specializes in designing and organizing training programs pertaining to auditing, fraud detection and prevention, information security and assurance, and anti-money laundering. He is a CISA and has a M.Sc. in Computer Security from the University of Liverpool. Dauda also has a first degree black belt in Taekwondo. He has previous experience of over five years in the Nigerian Banking industry, and also did some time in Gtech Computers (a computer and allied services company) as a systems security and assurance supervisor.

F.S.S.C.

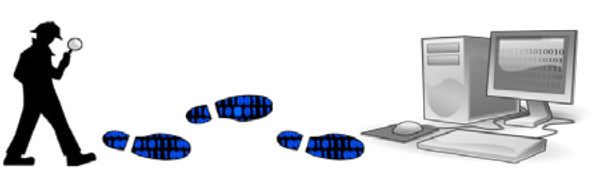
Forensic Security Solutions Co.

A Computer Forensics and Network Security Consulting Co.

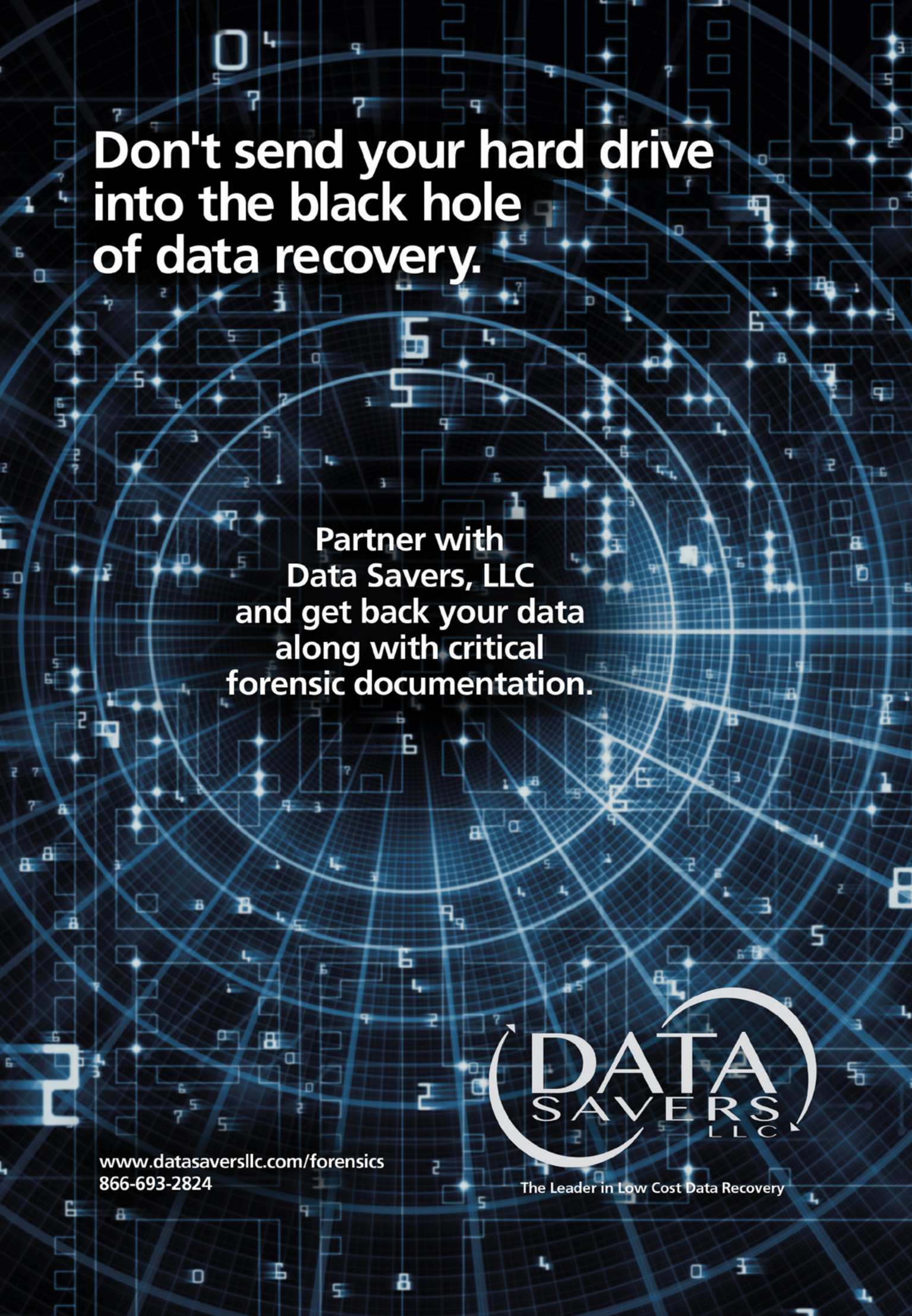
- Forensic Imaging & Preservation of Digital Data
- Forensic Analysis & Investigations
- E-Discovery Collections
- Targeted & Multi-User Collections
- Risk & Threat Analysis
- Vulnerability Assessment
- Penetration Testing
- Forensic Wiping of Digital Data Sources (Hard Drives, Thumb Drives, etc.)

Forensic Security Solutions Company is geared toward providing their customers with extraordinary project management and client interfacing that can be utilized for any size matter. Feel free to check us out at www.ForensicSSC.com

F.S.S.C.



Tel: (908) 917-1482 Email: Contact@ForensicSSC.com
www.ForensicSSC.com


Don't send your hard drive into the black hole of data recovery.
Partner with Data Savers, LLC and get back your data along with critical forensic documentation.
DATA SAVERS LLC
www.datasaversllc.com/forensics
866-693-2824
The Leader in Low Cost Data Recovery