

HAKING EXTRA

Issue 4/2012 (11) ISSN 1733-7186

FORENSICS IN THE CLOUD

A hand is shown pointing towards the viewer, with the index finger extended. The hand is superimposed over a glowing, golden circuit board. In the background, a blue and green globe of the Earth is visible. The overall scene is set against a dark, textured background with various electronic components and labels like 'C16', 'C17', 'C18', 'C19', 'C26', 'CB23', 'CB29', 'C37', 'C33', 'CB41', 'CB42', 'C40', 'CB20', 'C12', 'C3', 'C29' scattered throughout.

A FORENSIC LOOK INSIDE NAS
LINUX FORENSICS

ANDROID VS. BLACKBERRY FORENSIC TECHNIQUES

FACEBOOK ACCOUNT FORENSICS

DATA HIDING TECHNIQUES

PLUS

HPA AND DCO VS. FORENSIC
IMAGING TOOLS

Atola Insight

That's all you need for data recovery.

Atola Technology offers *Atola Insight* – the only data recovery device that covers the entire data recovery process: *in-depth* **HDD diagnostics**, **firmware recovery**, **HDD duplication**, and **file recovery**. It is like a whole data recovery Lab in one Tool.

This product is the best choice for seasoned professionals as well as start-up data recovery companies.

Emphasized features at a glance:

- Automatic in-depth diagnostic of all hard drive components
- Automatic firmware recovery and ATA password removal
- Very fast imaging of damaged drives
- Imaging by heads
- Case management
- Real time current monitor
- Firmware area backup system
- Serial port and power control
- Write protection switch



Visit atola.com for details

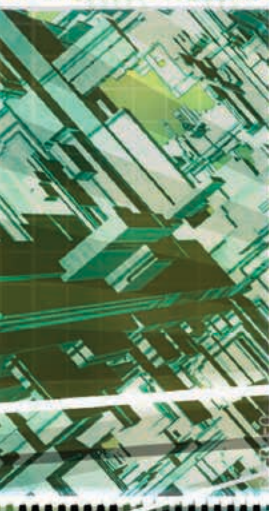


The Industry's First Commercial Pentesting Drop Box.

THE Pwn Plus:



Air Freshener?



Printer PSU?
...nope



FEATURES:

- ★ Covert tunneling
- ★ SSH access over 3G/GSM cell networks
- ★ NAC/802.1x bypass
- ★ and more!



PWNIE EXPRESS

@ pwnieexpress.com

Discover the glory of
Universal Plug & Pwn

t) @pwnieexpress **e)** info@pwnieexpress.com **p)** 802.227.2PWN

**Managing:**

Michał Wiśniewski
m.wisniewski@software.com.pl

Senior Consultant/Publisher:

Paweł Marciniak

Editor in Chief:

Grzegorz Tabaka
grzegorz.tabaka@hakin9.org

Art Director:

Marcin Ziółkowski

DTP:

Marcin Ziółkowski
www.gdstudio.pl

Production Director:

Andrzej Kuca
andrzej.kuca@hakin9.org

Marketing Director:

Grzegorz Tabaka
grzegorz.tabaka@hakin9.org

Proofreaders:

Dan Dieterle, Michael Munt,
Michał Wiśniewski

Top Betatesters:

Ruggero Rissone,
David von Vistauxx,
Dan Dieterle,
Johnette Moody,
Nick Baronian,
Dan Walsh,
Sanjay Bhalerao,
Jonathan Ringler,
Arnoud Tijssen,
Patrik Gange

Publisher: Hakin9 Media Sp. z o.o. SK
02-682 Warszawa, ul. Bokserska 1
www.hakin9.org/en

Whilst every effort has been made to ensure the high quality of the magazine, the editors make no warranty, express or implied, concerning the results of content usage. All trade marks presented in the magazine were used only for informative purposes. All rights to trade marks presented in the magazine are reserved by the companies which own them.

To create graphs and diagrams we used program by Mathematical formulas created by Design Science MathType™ **DISCLAIMER!**


The techniques described in our articles may only be used in private, local networks. The editors hold no responsibility for misuse of the presented techniques or consequent data loss.

DEAR HAKIN9 EXTRA READERS,

I AM, AGAIN, VERY GLAD TO INVITE YOU TO READ HAKIN9 EXTRA. THIS MONTH WE ARE GOING TO EXPATiate ON FORENSICS. THE NEW IDEA BEHIND HAKIN9 EXTRA WAS TO EXPAND IT AND GIVE YOU, DEAR READERS, MORE VALUABLE CONTENT. THE STORY BEHIND THIS MONTH'S TOPIC GOES BACK TO AN E-MAIL I RECEIVED FROM ONE OF OUR BETA TESTERS SAYING THAT SPANISH HACKING GURU PANCAKE HAS CREATED A USEFUL TOOL FOR REVERSE-ENGINEERING CALLED RA-DARE. THAT GAVE US SOME FOOD FOR THOUGHT AND WE DECIDED THAT WE SHOULD GIVE THIS TOPIC OF FORENSICS A GO. IN THIS MONTH'S HAKIN9 EXTRA: OUR LONG-TIME COLLABORATOR – NILESH KUMAR WILL PRESENT YOU LINUX FORENSICS. OUR COLLEAGUE FROM SEAGATE – DMITRY KISSELEV IS GOING TO TAKE A FORENSIC LOOK INSIDE NAS. FEDERICO FILACCHIONE WILL TAKE US FOR A TRIP INTO THE CLOUD FORENSICS. HPA AND DCO FEATURES WILL BE COVERED BY AMY COX. YURY CHERMERKIN, ONE OF OUR MOST EAGER COLLABORATORS, IS GOING TO JUXTAPOSE ANDROID AND BLACKBERRY FORENSIC TOOLS. UĐUR EKEN WILL SHARE WITH YOU THE KNOWLEDGE OF DATA HIDING AND ITS TECHNIQUES. TONY RODRIGUES WILL SHOW YOU HOW TO FIND A NEEDLE IN A HAYSTACK, OR SIMPLY A LOST FILE ON YOUR HDD. RUGGERO RISSONE TOOK A STROLL DOWN THE DEFT CON 2012 BOOTHS AND PREPARED A REPORT FROM THE AFOREMENTIONED CONFERENCE WITH THE IMPACT ON FORENSICS. DAVE SAUNDERS WILL GIVE US AN INSIGHT INTO "CORPORATE FACEBOOK ACCOUNT LOGIN FORENSICS AND US LAW". FINALLY, JERRY HATCHETT HAS A LIST OF PERFECT TIPS FOR YOUNG FORENSICATORS. IF YOU KNOW SOME INTERESTING TRENDS IN IT-SECURITY, YOU WANT TO BECOME A COLLABORATOR OR YOU HAVE SOME REMARKS CONCERNING HAKIN9 EXTRA – PLEASE DROP ME SOME WORDS VIA E-MAIL.

IT'S BIGGER,
IT'S BETTER,
IT'S HAKIN9 EXTRA!!!

MICHAŁ WIŚNIEWSKI
M.WISNIEWSKI@SOFTWARE.COM.PL



**Bad things can
happen to a laptop.
They don't have to
happen to the data.**

Seagate Recovery Services work on any disk drive to support forensic investigations

Seagate takes the dread out of data mishaps in forensic investigation scenarios. From file deletions to physical tampering causing hard disk damage - from any brand - we make it easy to get the files back for law enforcement agencies to crack criminal cases. For more information, please visit www.seagatedatarecovery.com.



8. An Overview on Cloud Forensics

By Federico Filacchione

There's not a single law. Preserving the chain of custody means that you've to comply with specific laws, regarding a specific country. But in a cloud perspective there's no single country. A huge network of data centers means a huge network of jurisdictions. So could be very complex to interact with some countries that don't have modern computer crime laws.

10. All Present and Accounted for?

By Amy Cox

Like a HPA it is not removed during a regular wipe or format. Though unlike the HPA it is created by the manufacturer and at the time of writing I am not aware of a way to create a DCO artificially after the drive is sold. That notwithstanding they can still be located and their contents copied to ensure they contain nothing of significance. Another difference between the two is that unlike the HPA which isn't hidden from the BIOS, this function even tells the BIOS that the disk is the smaller size.

16. A Forensic Look Inside NAS

By Dmitry Kisselev

Standard packages under a GNU general public license (e.g. apache, vsftpd and samba) are used by storage manufacture to package NAS functionality into the box. These packages' main purpose is to provide an administrative interface for the user's customized configuration as well as provide feature rich network data sharing capabilities. Often times, CIFS/SMB, NFS, FTP, AFS, WebDAV protocols are bundled with in NAS. It's important to note that this type of feature is the main differentiator for NAS devices. Every one of the protocols and administrative interfaces generate traces, logs and other useful information for forensic analysis and investigation.

22. Linux and Disk Forensics: A General Approach

By Nilesh Kumar

Complete description of tools and their uses are out of scope of this article, we'll be just using them for our forensics, as you may get a fair idea about them during our process. We shall be using BackTrack(BT) for our analysis. You could pretty much use any distro available as all have mostly common necessary tools. You could use any normal Linux flavors such as Fedora, RedHat, Ubuntu as well, but the advantage of using distros like BT is that they already have a fair collection of these tools, otherwise you may need to install them.

28. Comparison of Android and BlackBerry Forensic Techniques

By Yury Chemerkin

Logical methods manage with non-deleted data are accessible on the storage. The point is that previous case is about «simple» data type(format), while SQL db files as all-in-one file may keep deleted data in the database. While recovery of the deleted data requires special tools and techniques, it is possible to recover deleted data from a logical acquisition. Physical techniques as techniques aimed to gain deleted data without relying on the file system itself to access the data, so it is missed too. Let's gain the main logical acquisition differences between two kind platform throughout way to data store, developers API and tools, free and paid investigation tools, logs, backup some more and others tricks.

38. Data Hiding Techniques

By Uğur EKEN

In NTFS file system meta data category information is stored into Master File Table entries and their attributes. Each default entry and attribute contains descriptive information about the files and directories. As I previously mentioned this information includes file and directory locations, permissions and MAC(Modified, Accessed, Created) timestamps[Carrier, 2005]. In this category Alternative Data Streams are one of the common areas data hiding techniques can be implemented in NTFS file system.

46. A Needle in a Haystack

By Tony Rodrigues

Note, also, that we will use SHA1 as hash algorithm. We use `-a` option to enter the filename we will search. The Needle.pl will write to standard output (stdout) all hashes calculated. We can also use `-t` option and request hash calculation just for a specific size. Even though, the most usual usage will be just passing `-a` option, redirecting the output to a file.

52. DEFTCON 2012 Report

By Ruggero Rissone

Yes, one of the main problem in Digital Forensics is that available tools are often expensive commercial software and dedicated to specific aspects in the landscape of cybercrimes. Deft could be executed on every x86 architecture (future plans could be the support for SPARC architecture) and could be installed on a limited hardware; one typical application is a forensic duplicator realized with DEFT installed on a Netbook (a commercial one could cost more than 1000\$, i.e. Tableau TD1 Forensic Duplicator).

54. Corporate facebook account login forensics and u.s. law

Dave Saunders

The Computer Fraud and Abuse Act (CFAA) was instituted in 1986 to reduce hacking and unauthorized access of computers. Most recently, in United States v. Nosal, the 9th Circuit narrowly held the CFAA is limited to prohibit actions by individuals that exceeds authorized access and violations of restrictions on access of information, and not restrictions on its (the information's) use. The 9th Circuit most recent ruling is important in it confirms and criminalizes potential unauthorized access of private corporate information by a third party.

56. Forensic Tools Free and Paid

Jerry Hatchett

"Which software should I buy, EnCase or FTK?" As someone who's been practicing digital forensics for a long time, I can't count the number of times I've fielded that very question from an eager young forensicator. ("Forensicator" is an industry word; the dictionaries haven't caught up yet.) I remember asking it myself. I remember researching it to the nines myself. I remember my choice, and I remember how quickly I learned that the question of which forensic tool is "best" is a question that never gets answered. Want to know why?

AN OVERVIEW ON CLOUD FORENSICS BY FEDERICO FILACCHIONE

FEDERICO FILACCHIONE

In the cloud everything is different, even doing a classical, very formal and standard activity like computer forensics. It doesn't matter if you're a forensic professional or a cloud service provider, are you doing enough to be ready? Do you know what the new challenges are?

What you should know:

- General knowledge of what a cloud service is
- General knowledge of how computer forensics is done on traditional systems

What you will learn:

- What are the main issues on doing forensic on the cloud
- How to beat the new challenges that the cloud model poses
- What are the new opportunities of the cloud forensics

Introduction

Cloud computing is one of the greatest innovation of the recent times. It introduces a new way to implement complex architectures that just ten years ago would have a cost of hundreds (if not thousands) times more.

This is because virtualization of resources due to the high power of modern computers, and a new scalable and flexible organization of the services, oriented more in a Service business model, instead of a Software one.

This, together with a new way to use applications (via browser access and not via the old client-server model) has introduced what the NIST describes as:

A model for enabling convenient, on-demand network access to a shared pool of configurable resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

But there's a downside with all these new features. The downside is that all the traditional models that existed in the old environment, in Cloud must be re-thought.

One of them is obviously how to do forensics activity on systems *in the cloud* and how to interact correctly with cloud providers.

In this article we'll try to see a general overview on what are the main challenges a forensic professional has to win to do everything right, and what anyone who want to use a cloud service, or host one, has to keep in mind to avoid this kind of issues in the future.

What is different?

Why there are all these new challenges? Let's see what's different in the cloud.

The classic forensic model is based on:

- Collecting the evidence on site.
- Respect the law and preserve the chain of custody.
- Analyze the data to find evidence, and prove it.

Now, in the cloud, everything changes.

There is no site. Yes, you've read correctly. *In the cloud* there's no place you can go and seize a pc or server to be analyzed after. In the cloud you've a lot of datacenters holding in the

same time the data you're searching for. This is really a kind of nightmare, since we're just at the first step and everything seems so overwhelming. Since the cloud architecture relies on sharing data, load and services over a lot of datacenters, virtualized infrastructures, and services open to possibly anyone, there's no way to be sure that you've collected all the data you need.

There's not a single law. Preserving the chain of custody means that you've to comply with specific laws, regarding a specific country. But in a cloud perspective there's no single country. A huge network of datacenters means a huge network of jurisdictions. So could be very complex to interact with some countries that don't have modern computer crime laws.

There's not a single provider. Analyze the data means that what you've collected is consistent and can be used to prove a point. But if the cloud service you're analyzing relies on another infrastructure, could you be sure that you're not missing something more. And who can give you further assistance? This is a real and present issue, let's make an example. There's a very well-known service that allows you to sync files over different devices and computers. You just *drop the files in a box* and they're available everywhere. Well, this cloud service is hosted on an upper level, run by a very well-known *female horse rider* internet bookshop. The question now is: who is in charge? Who really run this service? The answer is that after dealing with multiple jurisdictions, the challenge here is understand what your data means, and what the two entities involved are using when running the service. And this would be very tough.

So is everything complicated? **No, not everything.**

Since the cloud model is a great opportunity for everyone, can be a great opportunity also for forensic professionals. But this implies that the cloud providers would be ready to confront their systems to the new challenges that the new model presents.

Many cloud providers don't know what to say when it's the time to talk forensics, because they don't know the issue. They don't care because the cloud model is too quick, too scalable, and too wide to comply with this kind of issues that came from the past.

But this is terribly wrong. It is wrong for the customer, because when something happens (and it will happen) you'll discover that you're in big trouble. And even worse for the cloud service provider since you are responsible for what happens inside your systems. In the good times, but a lot more in the bad times.

The cybercrime is not watching

Since now we've talked about regular services. But as you may guess the cloud is not only a very efficient way to do business for normal companies, it's also a great model for criminals too.

This generates an incredible threat for everyone who's hosting a cloud service. And the threat is dual.

The first huge issue is that someone will discover that your service is used by cybercriminals to do illegal thing. For example there are proofs that some online malware-scanning services are used by criminals to quote their new malware. Is like: See? Nobody recognize it, it's very good, buy it from me! This is a not so legal use of a perfectly legitimate tool, so maybe sometime the guys in blue will come to you to ask *what's happening?*

The other issue is that even criminals are deploying their cloud services. Nowadays there are a lot of malware toolkits available to buy, even for a few dollars. And those products are sold and supported (yes, they provide a helpdesk too) via infrastructures very close to the software-as-a-service model we all know. And when a system like this is blocked or seized, how can a forensic professional understand what's happening? How can he take data to prove the crime?

Those are real cases, and real challenges to deal with.

By the way, think about it: isn't a botnet a *cloud service* too?

Opportunities

We've say that before, and we'll say again: be ready. This is a huge issue, and that's not to be underestimated.

But this means also that many forensic professionals can really help to implement forensics tools and strategies in the new cloud services. Since this is an issue that every cloud provider will face sometime, this is an excellent opportunity to work on deploying a forensic strategy.

The tools are very important too, so why not think again every standard tool you use today, and translate them in a cloud configuration. This, with the help of a cloud provider could be very effective to adopt a working and very efficient strategy.

The cloud model is so versatile that could be possible, and someone is already working on this, to develop and deploy a Forensics-as-a-Cloud-Service platform, made specifically for the cloud.

And if you host a cloud service this could be an excellent opportunity too, since designing a forensic strategy and architecture for the cloud means implementing those tools and procedures in your cloud service, and that is what is need to be done now. That means a proactive data collection, based on forensic tools (even standard ones!), a detailed incident response procedure that includes forensics, training to all you staff to understand what are the risks (this is always true, anyway).

But this also includes understand the law of those countries where your datacenters are. And this is important not only to be prepared in case of incident, but also to avoid a possible prosecution if the law is too tight.

In the US, for example, many courts and judges are asking more and more for digital evidence. They're asking how the evidence was collected, and how it was preserved and analyzed.

And if it's your company on the court's bench, and if you're company is not ready to answer the judge's questions, you could face sanctions.

Again, don't expect your turn to be in trouble, prepare.

FEDERICO 'GLAMIS' FILACCHIONE

is born and living in Rome, Italy. He works in the IT industry for more than 10 years, most of them trying to spread security awareness and convince his colleagues and bosses that security is not a tool you can buy, but a new way to think about the same old stuff. He loves quoting Schneier's "You can't defend. You can't prevent. The only thing you can do is detect and respond", but still finds people who doesn't understand the meaning.

You can read his thoughts (in Italian) on glamisonsecurity.com, follow him @glamis on Twitter, or contact him at glamis@glamisonsecurity.com.

ALL PRESENT AND ACCOUNTED FOR?

AMY COX

Host Protected Areas (HPA) and Device Configuration Overlay's (DCO) are both 'hidden' areas on a hard drive. They are prime examples of where suspects can hide their fiendish files. Although not new tech they are still missed by several of the leading forensic imaging tools. So how can we, as practitioners, retrieve the data stored in them?

This article covers a little of the history relating to each of the areas and how they work. We will then learn how to locate their presence on a drive and how to create your own HPA. Following on from this I will share my findings regarding several forensic imaging tools and their ability to detect and recover these areas. And to finish up we will go over how the investigator can remove the HPA area in order to recover its contents.

When an operating system is loaded it will locate the size of each drive attached to it. It will do this using the ATA command 'IDENTIFY DEVICE'. This will report the addressable sectors of the drive to the operating system which will then set about using the space as it sees fit. However, hiding at the end of the drive may be data that is undetected by this command and thus not registered by the operating system.

These areas are referred to as Host Protected Area's (HPA) and Device Configuration Overlay's (DCO) and being over a decade old they are by no means new tech.

So, why are we still writing about them? Simple, because the same problems exist now that existed when they were introduced. If we do not look for them they will not be found.

Please note that Host can be swapped with Hidden and Protected with Partition but let's not get bogged down with the acronyms.

A Short History Lesson

In 1998 the AT Attachment (ATA) standard was updated to ATA-4, this update included the support for a HPA to be added to a drive. This area is located at the end of the drive and was originally designed to allow manufacturers to store recovery code.

This was seen as a positive move as it meant manufacturers could:

- Stop providing end-users with recovery disks that could be lost.
- Protect the recovery area from viruses and other contaminating nasty's.
- Protect the area from the most dangerous thing to any computer - the user.

Large brands including Dell, IBM/Lenovo and LG Electronics are examples of Companies that have distributed recovery software using a HPA.

A major feature of this area is that it is not accessible to the user and only the Basic Input Output System (BIOS) is aware of its presence, the operating system itself will not register it. Also important to note is that it is not wiped or removed during a standard drive wipe or format.

In 2009 a standard was released that sets out a firmware interface which contained the capability for the operating system to access the HPA, it was named 'Protected Area Run Time Interface Extension Services' or PARTIES. This was a set of diagnostic commands that allowed the manufacturer/technician to access the HPA in order to run the recovery processes if required.

PARTIES relies on there being a 'Boot Engineering Extension Record' (BEER) in place in the final sector of the drive. This contains a pointer to the user area of the disk and another to the PARTIES service area within the HPA.

Companies are now moving away from this mode of Recovery distribution in favour of Recovery partitions. This means that a modern disk containing a HPA should be treated with added suspicion.

Finally, and the part that is most relevant to a practitioner, is that a HPA can be created very easily using a single Linux

command. This process is described in the paragraphs following.

In 2002 the HPA was joined by the DCO which was introduced as part of the AT Attachment 6 (ATA-6) standard. Its main function was to allow the manufacturer to set the drive size to whatever they wanted to sell the drive as. For example if they had several 80Gb drives but wanted to sell them as 60Gb a DCO could be created to set all the drives to 60Gb.

This area can appear in addition to the HPA and is also located at the end of the drive. It is used by the manufacturer to manipulate the drives:

- Bad Sectors
- Cluster size
- Reported size
- Features- these can be disabled using the DCO

Like a HPA it is not removed during a regular wipe or format. Though unlike the HPA it is created by the manufacturer and at the time of writing I am not aware of a way to create a DCO artificially after the drive is sold. That notwithstanding they can still be located and their contents copied to ensure they contain nothing of significance. Another difference between the two is that unlike the HPA which isn't hidden from the BIOS, this function even tells the BIOS that the disk is the smaller size.

Putting all this together indicates that a practitioner should still be interested in the contents of these areas.

The simplest way to locate the areas would be to compare the Logical Block Addressing (LBA) value recorded on the label with the number of sectors reported by your chosen forensic imaging tool.

But....

- What if this value or the label has been observed or removed?
- Or the drive label lacks the value altogether?
- Or the HPA/DCO was set by the manufacturer and therefore the label was incorrect from the very beginning?

ATA Commands

Both of the areas are controlled by ATA commands. As previously mentioned the drive size is reported to the operating system using the ATA command 'IDENTIFY DEVICE'.

The ATA command to set up a HPA is 'SET MAX ADDRESS'.

This command sets the size of the accessible drive. This is the sector size reported when the command 'IDENTIFY DEVICE' is answered.

If this command is used to alter the size of the drive then the area after the last sector becomes a HPA.

The ATA command 'READ NATIVE MAX ADDRESS' will display the real size of the disk regardless of what the 'SET MAX ADDRESS' states it is. Therefore if there is a difference in the 2 sizes then a HPA may be present.

Consequently, as practitioners we need to send the commands 'SET MAX ADDRESS' and 'READ NATIVE MAX ADDRESS' to the drive to locate a possible HPA.

The DCO is also controlled using the ATA commands 'DEVICE CONFIGURATION IDENTIFY' and 'DEVICE CONFIGURATION SET'. The first command sets the size of the disk and the second enables or disables functions on the drive. Unfortunately, there is no ATA command to locate the real size of the disk.

So now we know what we need to do is there a tool to help us do it?

Of course - Linux offers a very powerful tool named 'hdparm' to help us carry out the task of locating these areas.

hdparm

This tool is included in several distributions of Linux. However, for this testing I have chosen the user friendly Ubuntu 11.10 (Oneiric Ocelot). It has been preconfigured not to automount devices. In Ubuntu 11.10 automount is disabled via the dconf configuration editor which can be downloaded via the Software Centre. The relevant menu to edit is:

- org – gnome – desktop – media-handling

The *man* page for 'hdparm' is extremely helpful and is available at <http://linux.die.net/man/8/hdparm>. It contains several health warnings that the practitioner should take into consideration as this tool can fry disks if misused.

One of the functions of 'hdparm' is to identify the sizes of devices attached to the system. It does this by sending the ATA commands 'SET MAX ADDRESS' and 'READ NATIVE MAX ADDRESS' to the drive and displaying the two values to stdout. The flag required to carry out this function is -N.

The command line for this is:

- hdparm -N /dev/sdX

Figure 1 shows the stdout display of a disk configured to have a HPA.

Please Note that the current device naming convention for Linux is sdX where X is a letter; this represents the physical drive. An example of this would be 'sda' equating to 'Disk 0' in Windows. If a number follows the three letters then the drive is partitioned and each partition is represented by a separate entry and consecutive number. An example of this would be 'sda1' equating to partitioned drive 'C:' in Windows.

Another tool that used to perform this task was 'disk_stat' which was part of The Sleuth Kit (TSK) by Brian Carrier. But according to their WIKI pages it was removed in 2010 and 'hdparm' was recommended as a substitute.

'hdparm' can also be used to set a HPA; which we will do now.

- Make yourself 'root'.
- Connect the drive to your system over an ATA connection (e-SATA or SATA are most likely) and make sure you know which drive you wish to add the HPA to.
- You can find out the drive name using either 'blockdev --report' or 'dmesg'. Make sure you pick a physical device and not a partition.

The command line for setting a HPA is:

- hdparm -Np[size] --yes-i-know-what-im-doing /dev/sdX
- The 'size' is not the size you want the HPA to be but the size you want the drive to appear as. Therefore pick the size, in sectors, you want the HPA to be and then subtract that from the size of the drive.

```
root@ubuntu:/home/ubuntu# hdparm -N /dev/sdg
/dev/sdg:
max sectors = 781410000/781422768, HPA is enabled
```

Figure 1. Stdout for the 'hdparm -N' command

- The '-N' flag as before relates to the max size in sectors on the drive.
- The 'p' flag means that the change to the drive is permanent. Without the 'p' the change is only temporary and will disappear when it is next powered on.
- The '-yes-i-know-what-im-doing' flag means it.
- **Health Warning:** You can corrupt the drive and possibly loose data already stored on the drive using this tool so consider yourself warned. If the answer is really 'no-i-dont-have-the-foggiest' then this tool is not for you.

The tool also helps with DCO's, handy isn't it?

The tool will list any device configuration settings that are present in the DCO settings for the drive. This is done using the command:

- `hdparm -dco-identify`

The drive used for testing only contained DCO settings from the manufacturer so although they were listed they contained nothing of interest to an investigation. However using 'hdparm' I was able to locate them.

So what's the big deal?

Now you know how to create a HPA lets consider how you would locate and recover the contents of such an area during the forensic imaging process. Well you would rely on your forensic imaging tool to get you a copy surely?

The simple answer is not necessarily.

Several of the popular forensic imaging tools relay on the operating system to be able to see the drive in the first place and as we have already established that isn't the case with a HPA/DCO.

In order to demonstrate this I have tested the following forensic imaging tools on a 400Gb Seagate hard drive that I have used 'hdparm' to add a HPA to.

- LinEn 7.0
- Guymager 0.6.3-1
- dc3dd 7.1.614
- EnCase Forensic 6.18
- FTK Imager 3.0.0.1442

For the Linux based tools I connected the drive to the test imaging station running Ubuntu 11.10 (Oneiric Ocelot) using an e-SATA external docking station. This imaging station is pre-configured not to automount devices.

For the Windows based tools I used a Fastbloc FE hardware write blocker by Guidance Software and the operating system is Windows 7 service pack 1.

Table 1 shows a basic breakdown of the results but read on for the detailed answer.

Table 1. Results from testing forensic imaging tools.

Tool	HPA located	HPA copied
LinEn 7.0 (BIOS)	No	No
Guymager 0.6.3-1	Yes	No
dc3dd 7.1.614	Yes	Yes
EnCase Forensic 6.18	No	No
FTK Imager 3.0.0.1442	No	No

LinEn 7.0

A Linux based boot CD developed by Guidance Software that is used to image drives. This is free to download from Guidance's customer support portal as long as you have valid login credentials.

This tool has 2 modes for acquisition, the first is BIOS mode; this is the version of the tool tested as part of this experiment.

- The user must first prepare their target drive to contain a folder to store the image files in.
- The CD in this mode has a Graphical User Interface (GUI) which lists the devices attached to the system.
- From this list the user can see the device location of the suspect's drive. The tool offers the user the option to 'Acquire' and when chosen the user is led through the imaging process which results in Evidence (E0) files being created.
- The user can input case data and select various options such as error granularity, hash algorithm and password into the tool as part of this process.

LinEn 7.0 in BIOS mode did not locate the HPA or give any indication that one was present.

The second mode available using this tool is Direct ATA mode, this second mode relies on the practitioner having Guidance Software's FastBloc Software Edition (SE) module. This module communicates with the drive at ATA level and therefore claims to be able to capture any hidden areas present; specifically a HPA or DCO.

I do not have access to this module so could not test this mode however previous versions of the mode has been criticised for not being able to recover the areas in the way advertised [Source: Disk Imaging Evaluation EnCase 6.8/Linen 6.1 <http://www.ep.liu.se/ea/cis/2009/001/cis09001.pdf>].

Therefore further testing of this mode would be required before any final comment could be made.

Guymager 0.6.3-1

This free Linux based imaging tool is available to download from SourceForge.net or from the Ubuntu distribution repositories. It is included on several forensic live CDs including CAINE, DEFT and CTImager.

- The tool comes pre-configured and has a GUI that displays drives attached to the system (see figure 2).
- It has the ability to image a drive to either DD, AFF or E0 file format. The window shown in figure 2 contains a column named 'Hidden Areas' which details if a HPA or DCO is present.
- When the practitioner right clicks their suspect's drive they can then select 'Acquire Drive' and a single page GUI appears with the model and serial number of the drive already filled in for ease of use.

Guymager registered that a HPA was present and displayed this to the user; however, it did not recover the HPA area as part of the imaging process.

dc3dd 7.1.614

This Linux based tool is also available to download from SourceForge.net or from the Ubuntu distribution repositories. This tool is a patch for the original dd command and has some extra useful features. It is purely command line however the *README*

All present and accounted for?

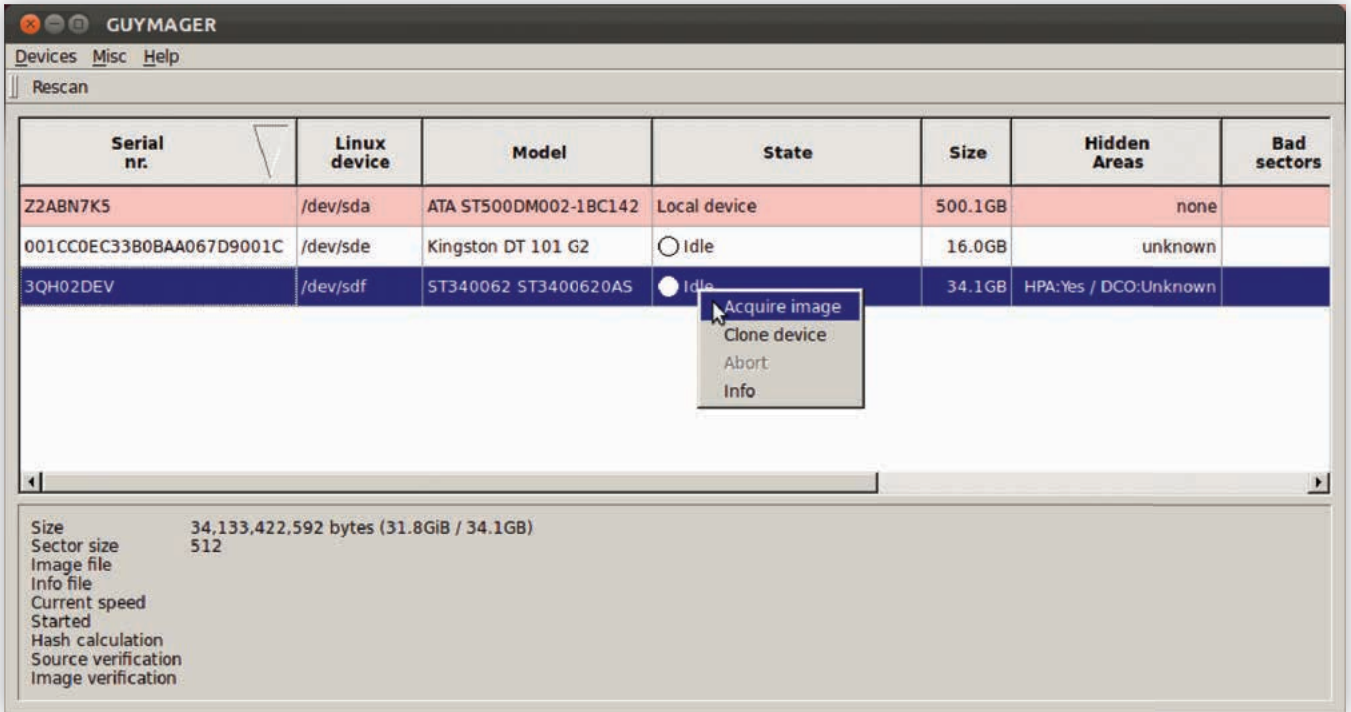


Figure 2. Guymager GUI interface

and `--help` files are extremely helpful. Be warned though the *info* and *man* pages simply point to each other and don't contain anything of value.

One of the tools extra features is the ability to identify the presence of a HPA and image its contents. This setting must be enacted during the tools configuration which is done using the following commands:

- `./configure --enable-hpadco`
- `make`
- `sudo make install`

Due to this downloading and configuring the tool yourself is recommended.

The command for imaging a drive is:

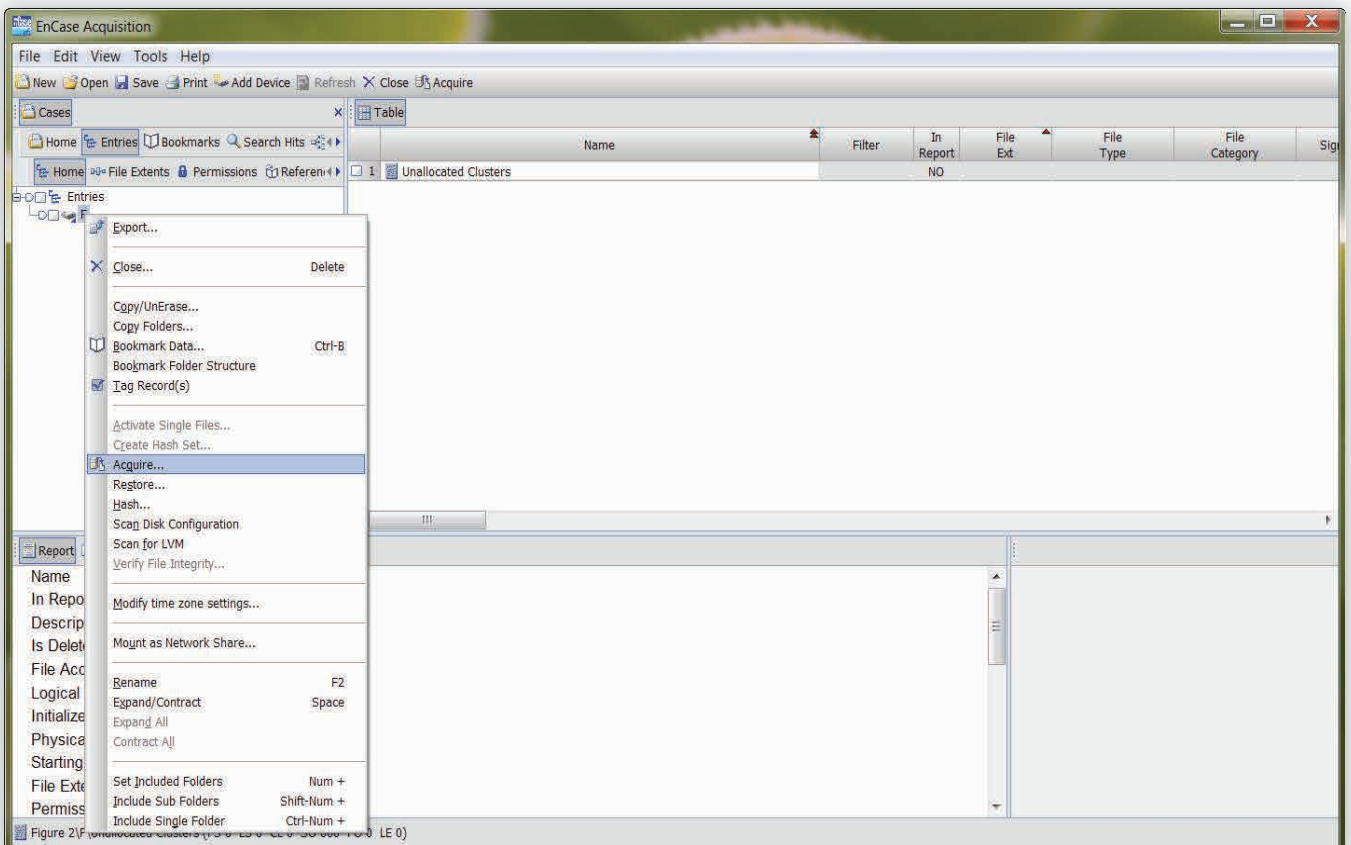


Figure 3. EnCase Forensic in Acquisition Mode.

- `sudo dc3dd if=/dev/sdX hash=md5 verb=on log=/media/log.txt hof=/media/output.dd`

The flags are broken down as follows:

- `if=` input file or the suspect drive
- `hash=` the hash algorithm you wish to use
- `verb=on` verbose logging on
- `log=` location of the log file, this will contain the verification details
- `hof=` the location of the outputted image file. The additional 'h' flag means that the file should be hashed to allow its verification against the input file.
- The resulting image file is a raw data dump format (dd).

This tool not only located the HPA but it successfully imaged it. The tool states that it will also image a DCO if present.

Following its successful imaging I ran the HPA identification command with `'hdparm'`. It indicated that the HPA had been removed from the drive. This is altering the drive and should be noted but fear not as none of the metadata will have altered as it was not mounted at the time.

EnCase Forensic 6.18

This licensed forensic tool is created by Guidance Software it is not only an imaging tool but is the standard investigation tool for many Companies around the world. Without the licence dongle EnCase Forensic reverts to 'Acquisition Mode'. It is available for download from Guidance's customer support portal as long as you have valid log in credentials.

- The imaging tool is enacted by adding the device to a newly created .case file and then right clicking a selecting 'Acquire disk'. This can be seen in figure 3.
- This action launches a GUI for the user to follow and once the input is complete Evidence files will be created in the chosen location.
- These can automatically replace the physical drive in the .case file if the user opts for this in the menu.

EnCase Forensic 6.18 in this mode did not locate the HPA or give any indication that one was present.

FTK Imager 3.0.0.1442

This free imaging tool is created by AccessData and can be downloaded from their website. Figure 4 show the GUI relating to this tool.

- The tool is able to image physical, logical or custom content image files.
- The resulting image files are either DD, ADD, SMART, Evidence or AFF.
- The GUI to create the images is a set of menus that allow the user to decide which options they require. It also allows the user to add case details to the image file.

FTK Imager did not locate the HPA or give any indication that one was present.

In conclusion, only one of the tools was able to successfully locate and image the contents of a HPA; this was dc3dd.

GuyMager was useful as it at least informed the user that there was a HPA present.

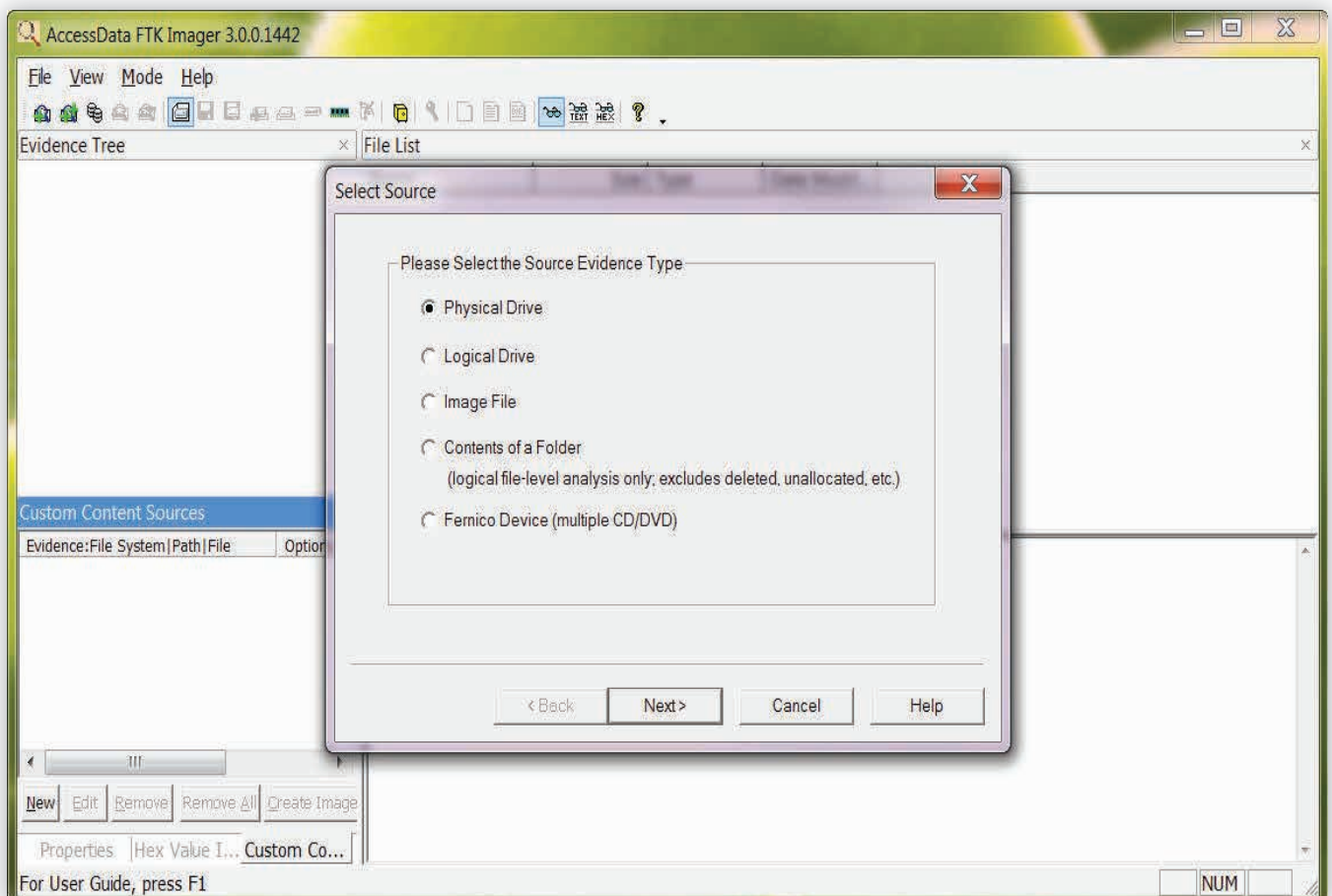


Figure 4. FTK Imager GUI

```

root@ubuntu:/home/ubuntu# hdparm -N /dev/sdg

/dev/sdg:
max sectors = 781410000/781422768, HPA is enabled
root@ubuntu:/home/ubuntu# hdparm -Np781422768 --yes-i-know-what-i-am-doing /dev/
sdg

/dev/sdg:
setting max visible sectors to 781422768 (permanent)
SET_MAX_ADDRESS failed: Input/output error
max sectors = 781422768/781422768, HPA is disabled
root@ubuntu:/home/ubuntu# hdparm -N /dev/sdg

/dev/sdg:
max sectors = 781422768/781422768, HPA is disabled

```

Figure 5. stdout for the hdparm commands to remove a HPA

The remaining forensic imaging tools were not able to deal with a HPA. This is due to the tools being reliant on the operating system being able to see the entire drive.

This highlights the fact that your choice in imaging tool will decide if you are able to recover the full drive or only the accessible parts. Your knowledge of the areas will also help you pick an appropriate tool.

Removing a HPA

It may be that you decide to manually remove the HPA from the drive in order to image it. This is possible and involves using 'hdparm' again.

Before you decide to take these steps though consider that this would be breaking the cardinal sin of computer based forensics – you would be altering the original device. In the UK we adhere to 4 main principles which are set out in the 'Association of Chief Police Officer's (ACPO) Good Practice Guide for Computer-Based Evidence' version 4. The first states that the practitioner should work on a copy of the devices contents and the original should remain unchanged.

As a qualified practitioner principle 2 does allow you to alter the original device as long as you can explain the whys and wherefores in a court of law. That's when the audit trail dealt with the third principle is key. If you decide to alter the original then your actions must be fully documented so that they may be subjected to review by the 'other side'.

If going down this route then the best practice would be to initially image the drive as it is and capture the entire live disk as it appears to the operating system. This way you have something, now if you damage the original removing the HPA when you won't lose the data currently stored in the user accessible part of the drive.

You must then calculate how many sectors should be on the drive. This is reported as part of the original 'hdparm -N /dev/sdX' command.

Then the same method for creating a HPA can be used to move the end of the user area of the drive to the 'real' end of the drive. This effectively moves the data from the HPA into the part of the drive the operating system can see.

The following command will do this:

- `hdparm -Np[total size of disk] --yes-i-know-what-im-doing /dev/sdX`

Figure 5 shows the output for this process.

Now if you remove the 'p' flag you may be able to then image the drive in Linux while the HPA is temporarily turned off and therefore not alter the original device. But, remember the re-boot into Windows would result in the temporary change being lost therefore to image using Windows you would have to set the 'p' flag.

Removing the HPA using the method above can be very dangerous and is not recommended; you may end up losing data that is essential to an investigation. It would be a safer and far more practical to make sure any device you suspect to have a HPA/DCO is imaged using a tool such as dc3dd. It is far more stable and you are less likely to lose the evidence you were recovering.

Another tool that should be mentioned here is the DOS based boot disk 'HDAT2' (www.hdat2.com). This tool states it is able to remove HPA's and DCO's; it also states it can recover the data from both areas. This has not been tested as part of this article however it may be an alternative to the Linux environment.

Conclusion

This article has dealt with HPA and DCO's. It has given the practitioner the tools required to create, locate and recover a HPA and to identify the presence of a DCO.

More importantly the goal of this article was to make the reader understand that if you forensically image drives without taking into consideration the presence of these hidden areas then you stand the chance of losing the HPA or DCO areas completely.

Always remember automated tools are there as an aid and can never replace the keen eye of an investigator.

AMY COX

graduated in 2008 with a first class honours degree in Digital Forensics from Teesside University, UK. She joined Greater Manchester Police's Hi-Tech Crime Unit early 2009 has a little over 3 years' experience with them as a computer forensic investigator. This role requires she stay up to date with her technical skills so she is at least one step ahead of the bad guys. She has recently completed a Post Graduate Certificate at Cranfield University, UK and is always looking for her next challenge.

A FORENSIC LOOK INSIDE NAS

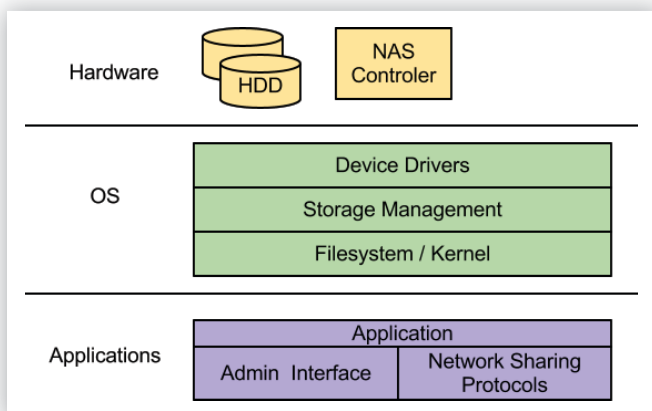
DMITRY L. KISSELEV

Network Attached Storage (NAS) devices are becoming de facto standards for external, computer independent storage for home use and small businesses. It's relatively easy to use, is cost effective and performs well. With a special purpose, NAS are computing devices that dedicate computing power to share and store digital files over the network. With the proliferation of Linux OS and affordable ARM-based-computers, the NAS market experienced substantial growth.

One small caveat is its complex design, which proves to be a challenge in a forensics analysis scenario. This article will explain at a high level how data is stored in NAS, a necessary step towards understanding how to efficiently work with computer evidence for forensics investigations.

How NAS systems are built

NAS systems are essentially composed of three layers: the Hardware layer, the Operating System (OS) layer and the Applications layer.



The NAS Hardware layer

Today's storage manufacturers create NAS appliances by utilizing standard SATA Hard Disk Drive (HDD) in a single or multi drive configuration. All of the drives are controlled by NAS controllers and are built around ARM or similar RISC based CPUs on a board with network interfaces. Most of the retailed NAS devices neither have any hardware based RAID controllers nor high performance HDD. The redundancy functionality and per-

formance of the system are passed on to the OS and application layer.

The NAS OS layer

The OS layer is responsible for managing storage, drivers and the overall network functionality used by the data sharing task of the Applications layer. Most of the time, a standard Linux distribution (like Debian) provides the OS functionality for NAS. In some cases, a Berkeley Software Distribution (BSD) flavored Unix or Microsoft Windows Server are used.

Storage management inside the NAS OS layer

The storage management functionality is a deciding factor for where and how digital files are stored in the physical device. A variety of storage management mechanism exists:

- A Logic Volume Management (LVM) and software RAID present necessary redundancy, space and performance. This type of configuration is a complex set up for forensic examiners because the end user data is not stored in a specific pattern, hence preventing direct access to the filesystem structures without first having to interpret it.
- A more sophisticated storage management technique is available, where the carving of disk space is utilized. In this set up, the disk space is partitioned to host the NAS operating system in a variety of auxiliary partitions and main data volume.

Useful Forensic Tools

During the forensic examination, partitions and filesystems are most likely targeted for analysis. Getting familiar with the disk structures of the filesystems ensures a successful metadata

check for filesystem creation times, deletion of inodes and traces of hidden or reformatted filesystems, or any abnormalities.

Forensic or data recovery tools that can help with this task are:

- WinHex- it provides ext3/4 filesystem parsing
- Filesystem debugger xfs_db - it is part of the xfs installation package

What you should know

- The main OS partition will carry a number of logs that are used to see what actions have taken place with the device, what user and access list was created to protect the files stored on the disk, as well as timestamps, all of which can be used in the investigation process.
- The swap partition carries pieces of log entries, filesystem records and other useful information, depending on the nature of the investigation.
- The main data volume displays access lists, file access/creation/modification timestamps as well as files.

The NAS Application layer

Standard packages under a GNU general public license (e.g. apache, vsftpd and samba) are used by storage manufacturer to package NAS functionality into the box. These packages' main purpose is to provide an administrative interface for the user's customized configuration as well as provide feature rich network data sharing capabilities. Often times, CIFS/SMB, NFS, FTP, AFS, WebDAV protocols are bundled with in NAS. It's important to note that this type of feature is the main differentiator for NAS devices. Every one of the protocols and administrative interfaces generate traces, logs and other useful information for forensic analysis and investigation.

A comprehensive analysis of NAS devices

As in any type of storage device, internal or external HDD, flash drives, NAS storage devices are meant to store all kinds of information for sharing among multiple users. When data is challenged, having an intimate knowledge on how to analyze data found on the NAS device could make or break the case.

Before the forensic analysis starts, access to the filesystem structures and filesystem is essential. However, NAS devices don't necessarily expose the filesystem's components over the network. Time and again, it is impossible to find information on the deleted files or access information/logs and hidden data. The only way to get that information is to analyze the drives where data is stored outside of the NAS enclosure. Knowledge of the principles associated with the NAS design and expertise in Unix filesystems will help with the analysis in any NAS devices set up.

NAS set up with one disk

This section will review a few single drive NAS devices made by WDC (My Book Live NAS) and LG (Network Storage N1A1). These devices can be purchased at most electronics stores or online. They are for general purpose, targeting the consumer market to enable file sharing such as documents, pictures, music and videos through standard network protocols like WebDAV, SMB, DLM, etc.

Most of the techniques used in the forensic field will apply to the filesystem in a single disk NAS because its architecture is simple. The storage control maps out the logical data space

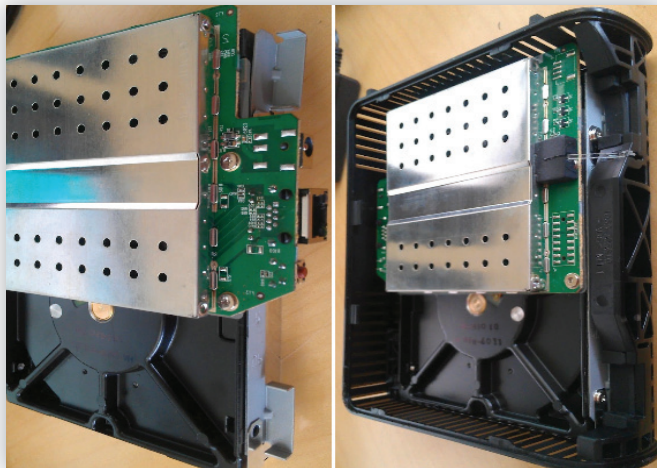
directly into the physical space. There are no layers of disk redundancy, and the volume management is rather typical.

Due to the self-contained nature of the drive, it is most likely used as a stand alone unit, through some manufacturers include a small partition space functionality which act as small external USB drive. No matter what the configuration is, one thing to consider is that the disk configuration could've been altered due to resizing and OS updates. Those actions will result in deleted files and hidden/lost filesystem structures, which may be important forensic evidence.

Steps to disassemble the one disk NAS enclosure

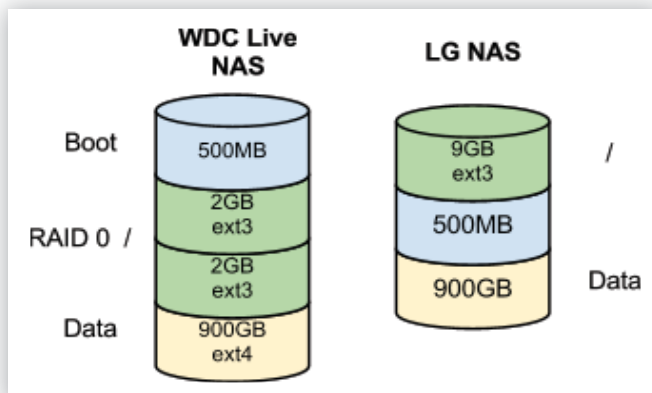
LG and WDC NAS, like with many single disk NAS, come with a disk drive that is firmly embedded in the enclosure. Extracting the drive from the enclosure in order to perform a forensic analysis requires ingenuity as Original Equipment Manufacturers (OEM) purposely intend for it to be a difficult task.

- To disassemble the enclosure, take a look at a drive attached to the NAS Printed Circuit Board (PCB). Most likely, this will be a SATA drive.
- Carefully disconnect the drive and prepare it for forensic imaging. This is an important step because a few forensically sound images will be created during the process. These images may not only serve as evidence, but may also help in case the image gets corrupted or damaged.
- Proceed with the analysis of the structures on disk



The disk Layout of the one disk NAS

Looking at the first two sectors of the drives, a different type of partition schema can be observed - WDC will have the GPT/GUID with 4 separate partitions and LG uses MBR the 3 partitions schema.



- Further analysis of the GUID partition type (A19D880F-05FC-4D3B-A006-743F0F84911E) on WDC Live NAS indicates that some type of RAID configuration is used on 2 GB partitions;

GUID Partition Table Entry #1		
400	Partition Type GUID	0F 88 9D A1 FC 05 3B 4D A0 06 74 3F 0F 84 91 1E
410	Unique Partition GUID	E5 F2 3B AF C3 46 66 44 91 FE 49 A9 34 C5 A2 87
420	Starting LBA	1032192
428	Ending LBA	5031935
430	Attribute Bits	00 00 00 00 00 00 00 00
438	Partition Name	primary
GUID Partition Table Entry #2		
480	Partition Type GUID	0F 88 9D A1 FC 05 3B 4D A0 06 74 3F 0F 84 91 1E
490	Unique Partition GUID	A2 BE 58 10 A6 A3 63 41 91 40 D8 38 C4 9A A8 2F
4A0	Starting LBA	5031936
4A8	Ending LBA	9031679
4B0	Attribute Bits	00 00 00 00 00 00 00 00
4B8	Partition Name	primary

- A parse partition at a latter stag will display in `/var/log/dmmsg` details of the RAID type configuration and how OS recognized these disks.

```
md: Autodetecting RAID arrays.
md: Scanned 2 and added 2 devices.
md: autorun ...
md: considering sda2 ...
md: adding sda2 ...
md: adding sdal ...
md: created md0
md: bind<sdal>
md: bind<sda2>
md: running: <sda2><sdal>
md0: WARNING: sda2 appears to be on the same physical
      disk as sdal.
True protection against single-disk failure might
      be compromised.
raid1: raid set md0 active with 2 out of 2 mirrors
md0: detected capacity change from 0 to 2047803392
md: ... autorun DONE.
md0: unknown partition table
kjournald starting. Commit interval 5 seconds
EXT3 FS on md0, internal journal
EXT3-fs: mounted filesystem with writeback data mode.
VFS: Mounted root (ext3 filesystem) on device 9:0.
```

- The same information can be obtained by analyzing the disk superblock/metadata at the end of the partition. That meta data is normally produced and served as a configuration for multiple device (md) driver. In this example, the md superblock is 128 sectors away from end of the partition and is clearly identified by its magic number `a92b4efc`.

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	
7A9EFFED	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
7A9F0000	A9	2B	4E	F5	00	00	00	00	00	00	00	5A	00	00	00	00	00	00	00	00	00	00	A3	9B	E7	83	4E	56	80	C4	00	00	01
7A9F0020	00	1E	83	C0	00	00	00	02	00	00	02	00	00	00	00	00	00	00	00	00	72	5F	AA	48	F7	D9	9C	B9	76	43	C0	BE	
7A9F0040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
7A9F0060	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
7A9F0080	4E	56	82	DD	00	00	00	01	00	00	02	00	00	02	00	00	00	00	00	00	00	00	00	00	00	CA	0F	CB	74	00	00	00	
7A9F00A0	00	00	02	9F	00	00	00	00	00	00	02	9F	FF	FF	FF	FF	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
7A9F00C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
7A9F00E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
7A9F0100	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
7A9F0120	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	

- The above disks are configured in a mirror RAID 1 configuration and are exact replicas of each other. Knowing this information can accelerate further analysis of filesystem on system and data partitions.

Conclusion: LG NAS has a slightly different layout that leverages older MBR partition to divide disk apart. In addition, LG decided not to use extra redundancy of RAID1 for main OS partition, yet leverages md driver to access partitions.

`/var/log/kern.log` has following data which shows layout md devices in LG NAS

```
kernel: [1.260000] md: Scanned 2 and added 2 devices.
kernel: [1.260000] md: autorun ...
kernel: [1.260000] md: considering sda3 ...
kernel: [1.270000] md: adding sda3 ...
kernel: [1.270000] md: sdal has different UUID to sda3
kernel: [1.280000] md: created md1
kernel: [1.280000] md: bind<sda3>
kernel: [1.280000] md: running: <sda3>
kernel: [1.280000] md: considering sdal ...
kernel: [1.290000] md: adding sdal ...
kernel: [1.290000] md: created md0
kernel: [1.290000] md: bind<sdal>
kernel: [1.300000] md: running: <sdal>
kernel: [1.300000] md: ... autorun DONE.
kernel: [1.300000] md0: unknown partition table
kernel: [3.650000] kjournald starting.
      Commit interval 5 seconds
kernel: [3.660000] EXT3 FS on md0, internal journal
kernel: [3.670000] VFS: Mounted root (ext3 filesystem)
      on device 9:0.
```

Both NAS devices samples here don't have any complex storage configuration to allow for a straight filesystem analysis without any additional steps of rebuilding RAID configuration.

The one disk NAS filesystems

The good news is that today, the majority of manufactures use comparable OS designs in a single disk NAS device. The filesystems are likely of linux origin - varying from ext(3/4) to xfs and in some rare occasions specialized derivatives of the above modified by by OEM.

WinHex recognizes ext3/4 filesystem on the WDC and LG NAS devices and allows for parse filesystem directly without any manipulation. In different types of configuration, the issue lays with the original disk layout because it is needed to identify the OS.

The one disk NAS OS

At the OS level, it is critical to examine the logs and configuration produced by the applications and different OS subsystems. On the disk logs, this depends on the type of services provided by NAS as well as the User Interface system used by the OEM. The analysis of the logs produced by the services will help identify the milestones during a NAS lifecycle, access and user lists, and systems data related to power on and shutdown sequences.

On the other hand, the analysis of data volume will help with identifying information related to the metadata of particular files, as well as anything that could relate to the deletion of the data from NAS. There is no data at the OS level. Instead, relevant information will have to be manually selected to accommodate for the level of forensic investigation.

For example, the following folders and structures are present at WDC and LG NAS:

Name	Type	Description
/var/log/dmesg*	Log(s)	Boot messages
/var/log/messages	Log	Boot and services startup messages
/var/log/samba*	Log and Directory	Contains information on the SMB/CIFS Windows client access and SMB server messages
/var/log/kern.log	Log	System Kernel messages
/var/log/vsftpd.log	Log	FTP Log with access and server messages
/var/log/apache2/access_log	Log	Access to Admin interface
/etc/fstab, /etc/mtab	Config	Mounted devices and correlating filesystem

This is nowhere a comprehensive list. Other logs may uncover additional data which will lead the understanding of systems behaviour during its lifetime.

NAS set up with two disks

Multiple drive NAS add additional complexity to the forensic analysis. In this set up, data is distributed over multiple drives to increase speed, capacity or redundancy. The two drive NAS samples analyzed in this section outline a simple RAID 0. RAID 0 distributes data between two disks to increase capacity and speed. It is not uncommon to see RAID 0 used to stripe data volume and RAID 1 used to protect system volume.

The de-striping of the drives can add complexity in accessing the filesystem. The below examples are two drive configuration NAS by Buffalo LinkStation Duo and DLink ShareCenter.



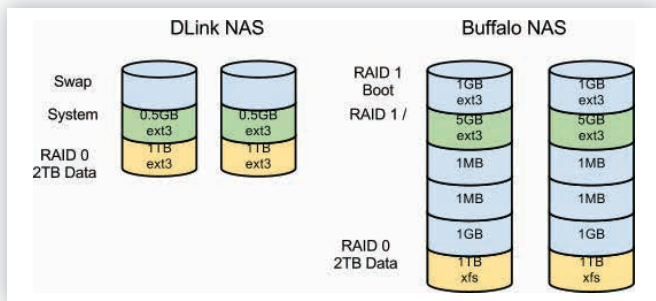
Disassembling the two disk NAS enclosure

Multiple drive devices allow easy access to drives through its special bays, although in some cases, accessing the drives seem to be as difficult was in the single drive NAS set up.

- Extract the drives from the NAS device
- Create forensics image of the disk before any analysis is started.

The two disk NAS layout

The DLink ShareCenter configuration puts all of the OS partitions into onboard flash memory, away from the HDD. It is still possible to retrieve content from it, but there is a level of complexity in reading that content directly from the flash memory. Using specialized memory chip readers and unsoldering the NAND flash chips off the NAS PCB will achieve just that.



The DLink uses a Master Boot Record (MBR) partition table with 3 partitions on each disk. Only Data Volume is set into the RAID 0 configuration and the rest of the volume is standalone.

The Buffalo LinkStation Duo NAS uses a GPT/GUID table partition scheme with 6 partitions, 2 of which are partitions that are in the RAID 1 configuration and the data partition, which is in the RAID0.

Both of the NAS devices employ a multiple disk (md) device to create soft RAID configurations and contain md superblock (old 0.9 and new 1.2 version) with the magic number a92b4efc either at the end (DLink) of the disk at the beginning (Buffalo).

```

000001000  c 4e 2b a9 01 00 00 00 00 00 00 00 00 00 21 07 5f 70 e5 a3 ee 70 3d c2 b1 3a 42 d8 e3 5e
000001020  55 4e 49 4e 53 50 45 43 54 2d 45 4d 42 35 41 3a 31 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000001040  46 0d 47 4f 00 00 00 00 01 00 00 00 00 00 00 00 e8 8f 98 00 00 00 00 00 00 00 02 00 00 00
000001060  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000001080  00 00 00 00 00 00 00 00 00 50 98 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000010a0  00 00 00 00 00 00 00 00 ee 33 76 f5 ae 03 09 ab e6 39 e7 af 6a ab b6 81 00 00 00 00 00 00 00 00
0000010c0  b9 f8 73 4f 00 00 00 00 8f 07 00 00 00 00 00 00 ff ff ff ff ff ff ff ff b4 68 32 af 80 01 00 00
0000010e0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000001100  00 00 01 00 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
    
```

In the above scenarios, RAID will need to be reconstructed before any filesystem data can be accessed for analysis. In addition, in the Buffalo NAS presence, the new version of md superblock will not allow filesystem parsers to recognize any NAS filesystem easily. By design, the new version of md superblock is located in front of the drive and is preventing the filesystem from being automatically seen by external tools.

Two disk NAS reconstruction options

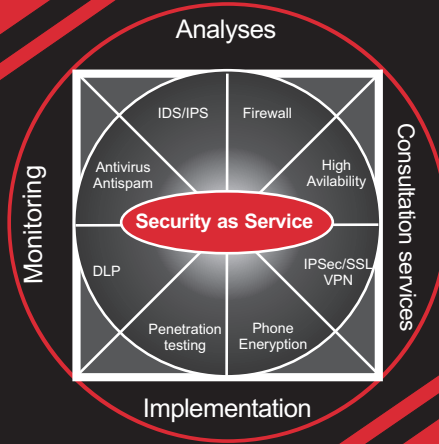
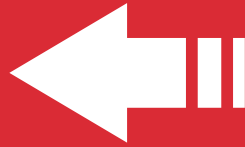
- When running Linux, connect the images of the NAS drives to the computer and use mdadm to reattach the md device to the system. Then, the necessary analysis can be performed applying SleuthKit, filesystem debuggers or hexeditor.
- In the Windows environment, several options are available through commercial software. These options facilitate the “virtual” attachment to a RAID array in order to see a contiguous space, or perform a “destripe” function to a separate drive by rebuilding its filesystem while effectively removing any RAID striping.

In normal cases, these utilities will have to be configured too, before they correctly characterize the targeted volume. Indicating the offset from the beginning of an image/partition to the

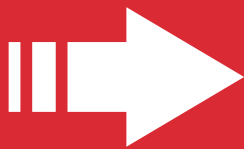
4safety

PROTECTING YOUR DATA

SECURE calling



www.4safety.cz



Trainings for administrators



CONTACT 4Safety, a.s.
ISO 9001



Krohova 2264/1
Praha 6, 160 00 CZ

Phone: +420-222 365 265
E-mail: info@4Safety.cz

LINUX AND DISK FORENSICS: A GENERAL APPROACH

NILESH KUMAR

In this article we'll discuss general and initial approach of performing disk forensics on Linux machine. This is a basic article about disk forensics and describes initial stages for an easy understanding of forensics and its approach. Forensics is a vast subject and it's all aspects can't be covered in a single article. I have tried to provide the major important steps followed in disk forensics using few tools. It will provide an insight towards moving forward while doing disk forensics.

What you will learn:

- Tools, methods and steps in performing a forensic.
- What you should know:

What you should know:

- Basic knowledge of Linux commands will be an advantage.

A digital forensic investigation generally consists of five major steps [Figure-1]:

- Identification
- Data Acquisition
- Data Recovery
- Analysis
- Reporting

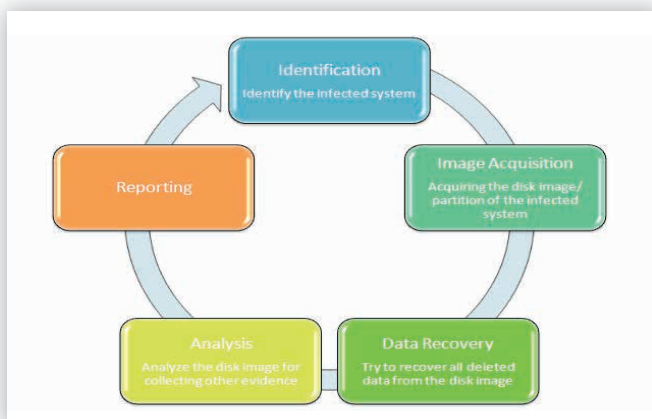


Figure 1. Forensics steps

A brief about various Linux tools available

There are multiple Linux tools used for imaging and analysis of a disks and drives. They also come as several distros containing all necessary tools to carry out Forensics, e.g. BackTrack, FIRE, Knoppix-STD, Linux LEO, penguinsleuth to name a few. All of them have an excellent collection of tools required for forensics. Some useful tools we require for a forensics are following, but not limited to:

- **Image-acquiring tools:**
dd, dd_rescue, dc3dd, Aimage etc
- **Data Recovery tools:**
foremost, magicrescue, safecopy etc
- **Forensics Analysis tools:**
bulk_extractor, missidentify, reglookup, readpst etc
- **Forensics suites:**
autopsy, sleuthkit, ptk

Complete description of tools and their uses are out of scope of this article, we'll be just using them for our forensics, as you may get a fair idea about them during our process. We shall be using BackTrack(BT) for our analysis. You could pretty much use any distro available as all have mostly common necessary tools. You could use any normal Linux flavors such as Fedora,

RedHat, Ubuntu as well, but the advantage of using distros like BT is that they already have a fair collection of these tools, otherwise you may need to install them.

To keep our work neat, clean and easily understandable, we may create a few directories in order to organize the data. We may need one directory for collecting our proofs and another directory to browse the suspected image of the disk. We shall redirect all the results from our analysis to the proof directory. Location of the directories is completely arbitrary but I prefer them on the root's home:

```
# mkdir /evidence
```

Now make a directory where we'll be doing our most of the analysis:

```
# mkdir /mnt/investigation
```

Here we are creating the folder *investigation* under *mnt* directory, where we shall mount all the external data for our investigation. You are completely free to create your own folder at any place; this is just for sake of better organization.

Acquire the Image

Identify the machine which needs to be investigated and take an image of the hard disk. You can capture the disk and connect to your forensics machine in order to take its image. The disk may be anything from a Hard disk to a Floppy. That way you'll have two copies of the suspected disk-one image as well as the physical disk itself. We'll be examining both images one by one. The tool *dd* can be used to take an image of the disk by using this command:

```
dd if=<media/partition on a media> of=<image_file>,
```

Example: *dd if=/dev/sdc of=image.dd*, here we are taking image of disk *sdc* and saving it as *image.dd*. You can give any name of the image and *.dd* is an extension just to denote that it's an image taken through *dd* tool.

Now for this article, we'll use sample test images already available on few open source sites such as <http://dfftt.sourceforge.net/>, <http://pyflag.sourceforge.net> or <http://linuxleo.com/> etc. They list excellent test images in every format to carry out test forensics. Download any disk images and unzip it in the *evidence* directory already created.

I shall be using one of the images already downloaded from similar sites at my PC. This was created using the same command *dd if=/dev/sdc of= pyflag_stdimage_0.1*, where we have taken image of disk *sdc* [Figure-2]:

```
root@bt:/evidence# ls -l
total 13120
-rw-r--r-- 1 root root 12288 2005-01-06 09:31 lost+found
-rw-r--r-- 1 root root 1474560 2012-02-21 02:31 disk_image1
-rw-r--r-- 1 root root 10485760 2012-02-21 20:27 pyflag_stdimage_0.1
-rw-r--r-- 1 root root 1474560 2012-02-21 02:31 testImage.dd
```

Figure 2. List of sample images

Once you download the above image copy it in blank floppy disk, we may require it later on:

```
# dd if= pyflag_stdimage_0.1 of=/dev/fd0
```

So, the image is copied into your floppy device(/dev/fd0). Now we have two copies, one in the */evidence* directory and one in physical floppy device.

Image Analysis

Now that an image has been captured, let's mount the contents to see how we can use tools, we'll mount it in our */investigation* directory:

```
# mount -o ro,noexec,loop pyflag_stdimage_0.1 /mnt/investigation
```

Here *ro* and *noexec* denotes that the file should be mounted as readonly and nonexecutable.

Now switch over to */mnt/investigation* directory, where you can browse through the file system of the disk image [Figure-3]:

```
total 4775
drwxr-xr-x 4 root root 1024 2005-01-06 09:54 .
drwxr-xr-x 29 root root 1474096 2012-02-20 23:02 disk_image1
drwxr-xr-x 3 root root 1471024 2005-01-06 09:43 Documents and Settings
-rw-r--r-- 1 root root 200736 2005-01-06 09:51 DonVittos_private_key.txt
-rwxr--r-- 1 root root 100427 2005-01-06 09:49 dscf1052.jpg
-rwxr--r-- 1 root root 1461565 2005-01-06 09:45 dscf1080.jpg
-rwxr--r-- 1 root root 1525183 2005-01-06 09:44 dscf1081.jpg
-rwxr--r-- 1 root root 1494120 2005-01-06 09:45 dscf1082.jpg
-rw-r--r-- 1 root root 10485712 2005-01-06 09:41 hello.txt
drwx----- 12 root root 112288 2005-01-06 09:31 lost+found
-rw-r--r-- 1 root root 258502 2005-01-06 09:43 rk_044.zip
-rw-r--r-- 1 root root 200881 2005-01-06 09:43 test.txt.gz
-rw-r--r-- 1 root root 203 2005-01-06 09:44 test.zip
```

Figure 3. File system in the disk image

Now you can redirect the above output to a simple file and place it into your *evidence* directory. This file can be used for analyzing the files, their various attributes [Figure-4].

```
#ls -Ralt > /evidence/ListOfFiles
```

```
root@bt:/evidence# cat ListOfFiles
.:
total 4775
522242 drwxr-xr-x 4 root root 4096 2012-02-21 22:51 .
2 drwxr-xr-x 4 root root 1024 2005-01-06 09:54 .
23 -rw-r--r-- 1 root root 736 2005-01-06 09:51 DonVittos_private_key.txt
22 -rwxr--r-- 1 root root 100427 2005-01-06 09:49 dscf1052.jpg
20 -rwxr--r-- 1 root root 1461565 2005-01-06 09:45 dscf1080.jpg
19 -rwxr--r-- 1 root root 1494120 2005-01-06 09:45 dscf1082.jpg
18 -rwxr--r-- 1 root root 1525183 2005-01-06 09:44 dscf1081.jpg
17 -rw-r--r-- 1 root root 203 2005-01-06 09:44 test.zip
16 -rw-r--r-- 1 root root 81 2005-01-06 09:43 test.txt.gz
15 -rw-r--r-- 1 root root 258502 2005-01-06 09:43 rk_044.zip
1281 drwxr-xr-x 3 root root 1024 2005-01-06 09:43 Documents and Settings
14 -rw-r--r-- 1 root root 12 2005-01-06 09:41 hello.txt
11 drwx----- 2 root root 12288 2005-01-06 09:31 lost+found

./Documents and Settings:
total 3
2 drwxr-xr-x 4 root root 1024 2005-01-06 09:54 .
1282 drwxr-xr-x 3 root root 1024 2005-01-06 09:54 Administrator
1281 drwxr-xr-x 3 root root 1024 2005-01-06 09:43 .

./Documents and Settings/Administrator:
total 1434
1282 drwxr-xr-x 3 root root 1024 2005-01-06 09:54 .
1285 -rwxr-xr-x 1 root root 147456 2005-01-06 09:53 outlook.pst
1283 drwxr-xr-x 2 root root 1024 2005-01-06 09:43 Local Settings
1281 drwxr-xr-x 3 root root 1024 2005-01-06 09:43 .
13 -rw-r--r-- 1 root root 1310720 2005-01-06 09:41 NTUSER.DAT

./Documents and Settings/Administrator/Local Settings:
total 35
```

Figure 4. Detailed listing of files

Using this list you can search for any specific file such as *.txt* [Figure-5]:

```
# grep txt ListOfFiles:
```

```
root@bt:/evidence# grep txt ListOfFiles
200736 2005-01-06 09:51 DonVittos_private_key.txt
200881 2005-01-06 09:43 test.txt.gz
12 2005-01-06 09:41 hello.txt
```

Figure 5. Using grep to search specific files

Another useful command might be for checking the file types. This might be useful in the scenarios where file extensions are modified. So if any *.txt* file is modified as *.jpg* the *grep* command won't be able to find that. Go to your investi-

gation folder and provide the following command and again redirect the results in your *evidence* directory:

```
# find. -type f -exec file {} \; > /evidence/TypeOfFile
```

Go to evidence directory and see the contents of TypeOfFile,

```
root@bt:~/evidence# cat TypeOfFile
./hello.txt: ASCII text
./Documents and Settings/Administrator/Local Settings/index.dat: Internet Explorer cache file version Ver 5.2
./Documents and Settings/Administrator/outlook.pst: Microsoft Outlook email folder (<=2002)
./Documents and Settings/Administrator/NTUSER.DAT: MS Windows registry file, NT/2000 or above
./rk_044.zip: Zip archive data, at least v2.0 to extract
./test.txt.gz: gzip compressed data, was 'test.txt', from Unix, last modified: Thu Nov 4 03:50:19 2004
./test.zip: Zip archive data, at least v2.0 to extract
./dscf1081.jpg: JPEG image data, EXIF standard 2.2
./dscf1082.jpg: JPEG image data, EXIF standard 2.2
./dscf1080.jpg: JPEG image data, EXIF standard 2.2
./dscf1052.jpg: JPEG image data, JFIF standard 1.01
```

you get the nice view of filetypes [Figure-6]:

Figure 6. Type of files

Now we may want to view the contents of the files:

```
# less hello.txt
# strings hello.txt
```

Or, if you want to see HexDump:

```
# xdd hello.txt
```

Searching strings can be also useful in the cases where you might want to look for notoriously used terms that may give you some idea about the incident and purpose, such as, ransom, virus, secrets etc. They may provide vital clue to in the investigation. So, we shall extract all the zipped files in order to search them for any particular string.

The following commands now can be used for searching a term [Figure-7]:

```
# grep -r -i secret ./
```

```
root@bt:~/mnt/investigation# grep -r -i secret ./
./test_extract/tmp/test.txt:This is a secret sentence, find it if you can!!!!
./test.txt:This is a secret test file.... Lets see if I can find it.
```

It will look for term *secret* in all the files of current directory

Figure 7. Looking for suspicious keywords through the image

```
# grep -r -i rootkit ./
root@bt:~/mnt/investigation# grep -i rootkit /
./rk_044_extract/rk_command.c: void process_rootkit(char *theCommand)
./rk_044_extract/rk_command.c: sprintf("rootkit: process_rootkit command %s, len %d", theCommand, strlen
theCommand);
./rk_044_extract/rk_command.c: char help[] = "Win2K Rootkit by the team rootkit.com\r\n \
./rk_044_extract/rk_command.c: // echo back the string, useful for rootkit patches that need
./rk_044_extract/rk_command.c: * skeleton functionality of rootkit is supplied
./rk_044_extract/rk_command.c: * Shutdown rootkit
./rk_044_extract/rk_command.c: // kill all traces of rootkit & shutdown permanently //
./rk_044_extract/rk_command.c: void process_rootkit(char *theCommand)
./rk_044_extract/rk_command.c: sprintf("rootkit: process_rootkit command %s, len %d", theCommand,
strlen(theCommand));
./rk_044_extract/rk_command.c: char help[] = "Win2K Rootkit by the team rootkit.com\r\n \
./rk_044_extract/rk_command.c: // echo back the string, useful for rootkit patches that need
./rk_044_extract/rk_command.c: void process_rootkit(char *theCommand);
./rk_044_extract/rk_defence.c: * If rootkit detects itself being monitored, it
./rk_044_extract/rk_defence.c: * - Stealth functions will attempt to hide rootkit
./rk_044_extract/rk_defence.c: DbgPrint("rootkit: FindTrackHandle() with handle %X\r\n", aHandle);
./rk_044_extract/rk_defence.c: DbgPrint("rootkit: found handle\r\n");
./rk_044_extract/rk_defence.c: DbgPrint("rootkit: AddNewTrackHandle()\r\n");
./rk_044_extract/rk_defence.c: DbgPrint("rootkit: GetRegValueMapping()\r\n");
./rk_044_extract/rk_defence.c: DbgPrint("rootkit: checking value map real %d t
0 trojan %d\r\n", rv->RealIndex,
./rk_044_extract/rk_defence.c: DbgPrint("rootkit: GetRegSubkeyMapping()\r\n");
./rk_044_extract/rk_defence.c: DbgPrint("rootkit: checking subkey map real %d
to trojan %d\r\n", rv->RealIndex,
./rk_044_extract/rk_defence.c: DbgPrint("rootkit: found the handle, cutting\r\n");
./rk_044_extract/rk_defence.c: DbgPrint("rootkit: detected invalid last ordering (a) \r\n");
./rk_044_extract/rk_defence.c: DbgPrint("rootkit: detected invalid list ordering (b) \r\n");
./rk_044_extract/rk_defence.c: DbgPrint("rootkit: internal error: attempt to free i
nvalid memory\r\n");
./rk_044_extract/rk_defence.c: DbgPrint("rootkit: CreateNewTrackHandle()\r\n");
./rk_044_extract/rk_defence.c: DbgPrint("rootkit: AddRegValuePair() %d %d\r\n", realIndex, trojanIndex);
./rk_044_extract/rk_defence.c: DbgPrint("rootkit: adding new regmap\r\n");
./rk_044_extract/rk_defence.c: DbgPrint("rootkit: AddRegMapPair() %d %d\r\n", realIndex, trojanIndex);
./rk_044_extract/rk_defence.c: DbgPrint("rootkit: adding new regmap\r\n");
```

Figure 8. Looking for suspicious keywords

It seems that a rootkit has also been injected to the system. But where is the file? If we search through the file, at the bottom we find path of rootkit binary file. Finding of NTROOT.sys suggest that the system was infected from process hiding trojans. So here' another search [Figure-9]:

```
./rk_044_extract/rk_kpatch.c: set the table to point to our trojan function. It is up to our trojan
./rk_044_extract/rk_kpatch.c: NewCXX is the rootkit trojan version of the function.
./rk_044_extract/rk_kpatch.h: 3. prototypes for our trojan calls
./rk_044_extract/rk_memory.h: prototypes for memory trojan calls
./rk_044_extract/rk_process.c: rootkit trojan function hooks. These are the meat and potatoes kids.
./rk_044_extract/rk_process.h: prototypes for our trojan calls
Binary file ./rk_044_extract/output/NTROOT.sys matches
./rk_044_extract/rk_utility.c: DbgPrint("rootkit: Exception occurred in ReadRegistry(). Unknown error. \n");
Binary file ./rk_044_extract/output/NTROOT.sys matches
Binary file ./Documents and Settings/Administrator/NTUSER.DAT matches
```

Figure 9. We confirmed the path of rootkit file

Now let's move to few more interesting folders. Let's see what lies inside them. Keep digging into them. The file *DonVittos_private_key.txt* contains a private DSA key, which might have been used to get access the machine [Figure-10]:

```
# less DonVittos_private_key.txt
```

```
-----BEGIN DSA PRIVATE KEY-----
Proc-Type: 4, ENCRYPTED
DEK-Info: DES-EDE3-CBC, 4E86848101D3CBF1
#-----
#-----BEGIN DSA PRIVATE KEY-----
DonVittos_private_key.txt (END)
```

Figure 10. View of a private key file

Let's drill into other folders such as *Document and Settings*. One file *index.dat* under *Document and Settings/ Administrator/ Local Settings* gives track about sites visited. Keep looking other folders, we found something more interesting, one outlook.pst file under *Document and Settings/ Administrator/*. This may give us more information. We will use a tool called *readpst* available in BackTrack. *readpst* is a command line tool which converts pst files into mbox format which in turn can be viewed and manipulated using any mail reading software.

```
# readpst -D outlook.pst
```

Option *D* includes deleted items in the output.

This creates several folders Inbox, Sent Items, MailBox, Deleted Items etc [Figure-11].

```
root@bt:~/mnt/investigation/Documents and Settings/Administrator# readpst -D outlook.pst
Opening PST file and indexes...visible directory tree elsewhere:
Processing Folder "Deleted Items"
Processing Folder "Inbox"
Processing Folder "Outbox" newdir
Processing Folder "Sent Items" containing the directory dir
Processing Folder "Calendar"
Processing Folder "Contacts"
Processing Folder "Journal" dir
Processing Folder "Notes" table dir
One can "Notes" => 0 items done, 1 items skipped, count subtree
Processing Folder "Tasks" dir
"Tasks" => 0 items done, 1 items skipped.
Processing Folder "Drafts" dir
Processing Folder "10 items done, 0 items skipped"
```

Figure 11. Output of pst file

Now all of these folders which are in mbox format can be easily viewed using any mail client. For this purpose, I shall use *KMail* to open the items. You just need to show the path of the folders and it will open them in your client interface. So if there are any attachments in the mail, you can easily open it and download for further evidence. So, while examining Inbox we can see the mails, one of the mails says [Figure-12]:

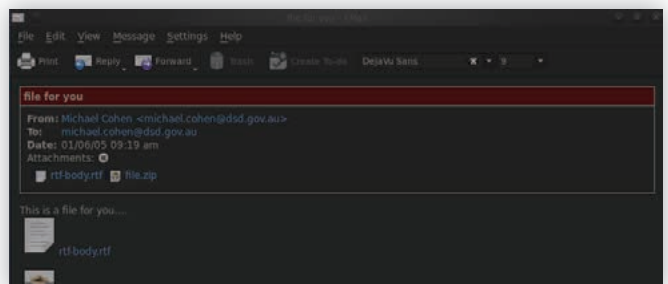


Figure 12: View of one received mail

Linux and Disk Forensics: A general approach

We can view any attachment now easily. Similarly let's examine Sent Items folder now [Figure-13].

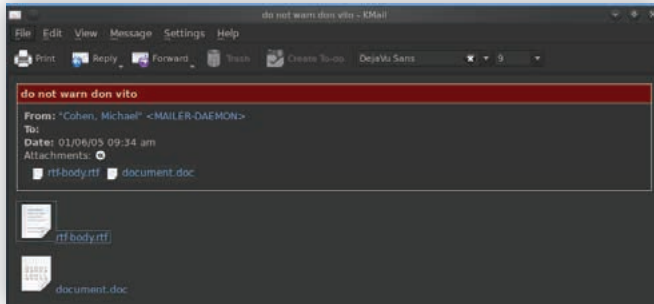


Figure 13. View of a mail in Sent Items

Above mail appears to be threatening mail. We can see the contents of the files as well. Let's open *document.doc* above attached above to see its contents [Figure-14].

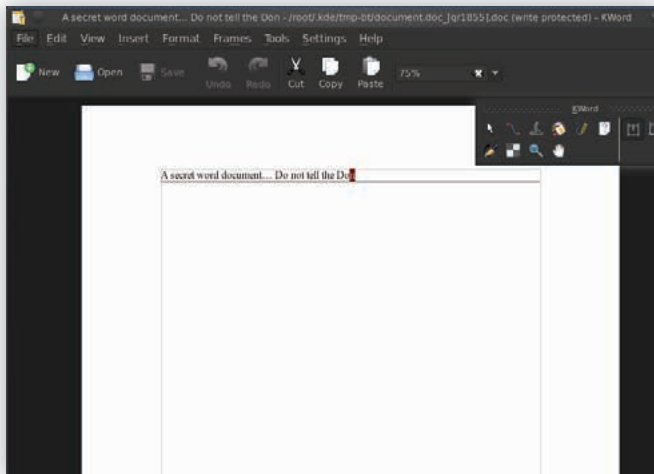


Figure 14. View of an attachment in the mail

Further we would like to redirect all the mail items to our *Evidence* directory in order to collect them for producing proof:

```
# readpst -D outlook.pst -o /evidence/MailsEvidence
```

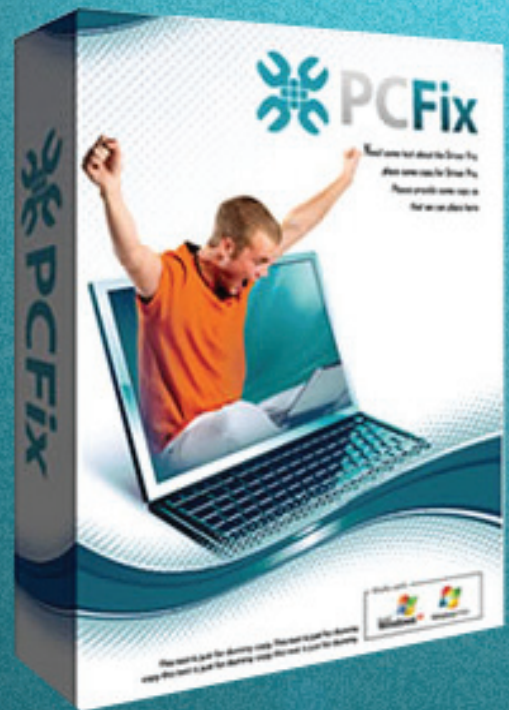
Data Recovery:

Now, I shall introduce one more tool which has a nice GUI, *Autopsy*. Autopsy analyzes the disk image and helps you browse the file contents and recover the data. Even it has capabilities for retrieving deleted files as well. So, once you are done with the image acquisition, we can use Autopsy to analyze the image [Figure-15].



Figure 15. Autopsy tool

PC Fix



Fix Windows Registry & Repair PC Errors!



Before you continue:

- ✓ Free scan your Computer now!
- ✓ Improve PC Stability and performances
- ✓ Clean you registry from Windows errors

Instant Scan

We'll open a new case, provide the case name, description and investigator names at second step, then add host third step and in fourth step you need to give path of the image stored at your machine. In few next steps, it will ask you to select the filesystem and partition etc. Once done, click on *Analyze* button, the following screen will appear [Figure-16].

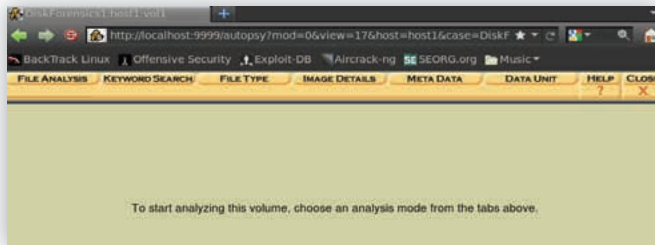


Figure 16. Autopsy functions

File Analysis lets you browse through the entire file system, *Keyword Search* can be used for searching specific terms in the file system, *File Type* lets you see the allocated and unallocated files, *Image Details* gives information about File System architecture, size and other metadata information, *Meta Data* gives information about Inodes, *Data Unit* shows contents of any fragment. Of our interest, is File Analysis and Keyword Search. Let's click File Analysis tab [Figure-17]:

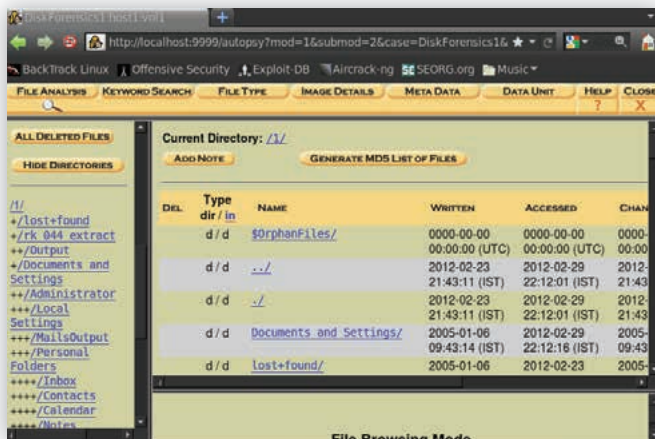


Figure 17. Browsing the file system

We can browse and view contents of the entire file system. The files in blue are existing files and if in red, they are deleted. We can see the contents of the files by clicking on them. Autopsy also gives you much information about dates of access, change, size, name etc of the files. Let's click on of the red files, which were deleted. It shows the contents of the deleted file, file type and various option for displaying it in ASCII, HEX or export the file [Figure-18].

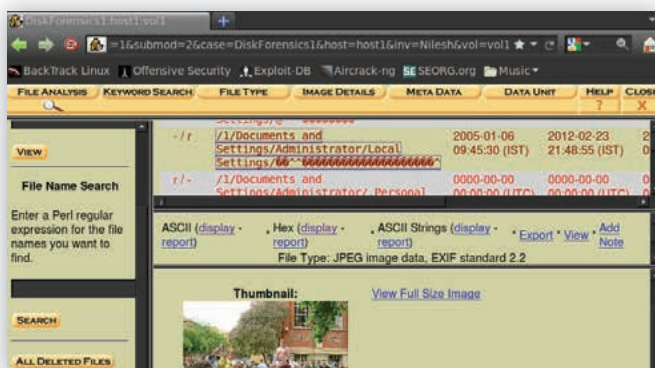


Figure 18. View of deleted files

Now we'll go to *Keyword Search* tab and try to search some important terms such as SSN, ransom, virus, Trojan, secret etc. Actually, it's suggested to create a list of terms or regular expressions, which can give vital clues about the file. We searched for the term *warn* here, it shows all of the hits it found in the entire filesystem. We can show the contents of the file after clicking on them [Figure-19].

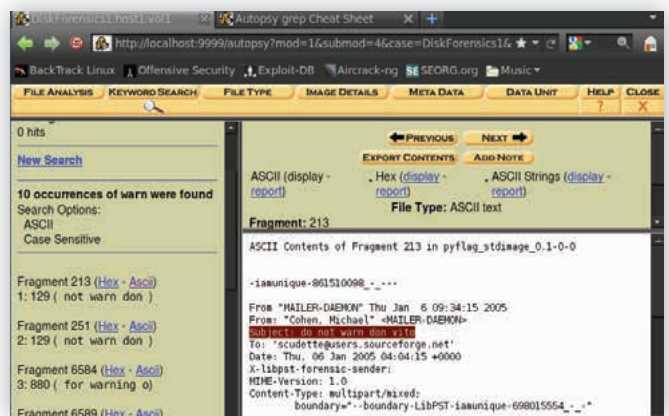


Figure 19. Keyword searches in Autopsy

Apart from these, there are a few more options which we can try and get more useful evidence as far as possible. It's not possible to cover all of the tools and their functionalities in single article; hence we may look at them in future articles.

References:

- <http://dftt.sourceforge.net/>
- <http://pyflag.sourceforge.net>
- <http://linuxleo.com/>
- <http://www.deflinux.net/> and more...

NILESH KUMAR

Nilesh is working as a Sr. Security Analyst with Honeywell Technology Solutions Lab, Bangalore, India. He is mainly focused on Application Security, Network Security and Wireless Security. Apart from that he shows interest in Reverse Engineering and Forensics (Blog: nileshkumar83.blogspot.com) http://www.mega-wallpaper.com/wallpapers/big/160_linux_pingwin_little_penguin.jpg

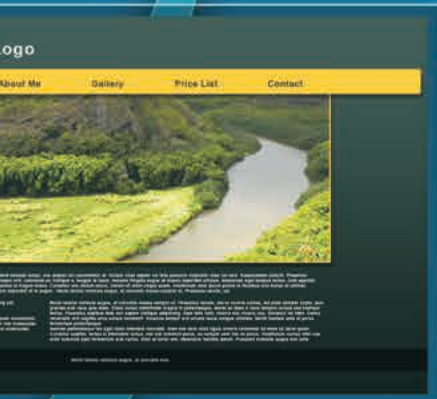




Web Audit Authority

The Internet service conducting fully-automatized web audits:

- Web Standards Audit
- Web Access Audit
- Web Usability Audit



www.webauthority.eu



Co-funding support provided by European Union from European Regional Development Fund

COMPARISON OF ANDROID AND BLACKBERRY FORENSIC TECHNIQUES

YURY CHERMERKIN

As digital data is omnipresent now, the digital forensics has quickly become a legal necessity. Mobile devices have quickly grown and extend their own features which simplifying makes them less unique. Developers API, SDK, NDK provide great opportunity to build live, DLP or spyware for data extracting.

What you will learn...

- How many differences between BlackBerry and Android forensics techniques

What you should know...

- Basic knowledge about Forensics (Classic and Live)
 - Basic knowledge about BlackBerry Forensics
 - Basic knowledge about Android Techniques
-

This mainly based on examine how many differences do exist between BlackBerry and Android OS. It's would interested to highlight whether one techniques provide more easy implementation, investigation and handling or not, what common differences examiners may encounter and what they should as concept be involved to forensic handling with these platforms. "Android Forensics: Investigation, Analysis and Mobile Security for Google Android" written by Andrew Hoog and my article "To Get Round To The Heart Of Fortress" published in Hakin9 Extra are the basis of my researching.

Mobile Forensics

Mobile device forensics is relating to recovery of digital evidence or data from a mobile device. The memory type, custom interface and proprietary nature of mobile devices require a different forensic process compared to other forensics. Nowadays mobile extraction techniques tend to be less unique especially throughout logical acquisition. This level manages with known data types for any user and this data set rarely differs among of iOS, Android or BlackBerry. This data set often contains the following items such as messages (SMS/MMS/Email/IM), social network data, contacts, calendar, phone logs, wallet and other

financial application data, media data (Audio/Photos/Videos) and other data even file structure, browser data (web history as a timeline and bookmarks), and shared folders.

One of the main ongoing considerations for analysts is preventing the device from any changes, that's sometimes achievable, like making a network/cellular connection, because it may bring in new data, overwriting evidence, etc. Any interaction with the devices, whether you simply move it or even physically unplug the device, will modify them. If you instead decide to examine the device while it is running, all interactions change the device. To further complicate an investigation, it is possible that the computer is leveraging encryption and, while the device is running, that data may be accessible. However, if the device is powered off and you don't have the encryption keys, then you may permanently lose the ability to recover that data. Android devices are nearly impossible to forensically analyze without any kind of impacting, because unlike desktops, notebooks, and servers, Android storage cannot be easily removed that often leads to changes by changing state from turn off to powered or something else. There was a little data stored often on SIM-card when mobile phones were first introduced. It was possible to remove the SIM card to extract data.

Did you know?

Device Switched On

If the device is in the on state, you act immediately to get power to the mobile device. Now it will not lose the volatile information. Then you need to take the device to a secure location like a Faraday Cage or turn off the radio before beginning the examination

Device Switched Off

If the device is in the off state, you need to take the device to the shielded location before attempting to switch on or place the device in room that can block the signal well enough to prevent the data push. This case for Android not BlackBerry means the best chance to boot device into recovery mode to test for connectivity and root access and access to data without booting into normal operational mode (if only USB debugging is enabled or owner's device have rooted it).

Device in its Cradle

If device is in cradle, you have to remove any connection from the PC despite possibility that a sophisticated suspect might have a tripwire device and once it disconnected it could activate script to erase potential evidence.

Password Protected

The thing has to be known when it comes to password protection is the fact that the password itself is not stored on the device. The only thing stored on the device is a hash of the plain-text password. This storage is similar to the storage used by the majority of operating systems out there.

Wireless Connection

You must avoid any further communication activities, if possible. Eliminate any wireless activity by placing the device into a cage that can isolate the device.

External Memory Card

You must not initiate any contact before taking components off. This includes any devices that supported external media types of cards.

As many examiners likely know, it is important to isolate the device from the network as soon as possible. In the worst-case scenario, a remote wipe could be initiated on the device which would prevent the recovery of any data. While most remote wipes are done over the data network, some can be triggered over SMS, and hence ensure the device is fully isolated to prevent remote wipes. In other circumstances, additional messages on the device could be received or even removed by triggers outside your control. To prevent a connection mobile devices will often be transported and examined from within a Faraday cage. As it may be a bit expensive, there is a more powerful way named air-plane mode or some-kind techniques are almost look likes in the same manner on both devices. It brings disadvantage sometimes. Talking about Android you should press and hold the Power off button and select Airplane mode at first, and then press Menu (from the home screen), then Settings, then the Wireless option which is generally near the top. You also may turn off Mobile Networks from this screen. If you're going to disable wireless connection like Bluetooth or WiFi you have to walk out home screen to the settings that have upset because you're not sure whether you have enough time or not. On another hand, only touch or flip BlackBerry model bring the really fast way to turn on/off all

connections by clicking around tray and date'n'time place on your home screen. Both ways need you to have an access to devices for password locked case. Moreover, the device continues running with temporal data remains. SIM card removing doesn't bring the same result, because your device reboots or wipes out like BlackBerry.

There are several techniques is pertaining to mobile forensic:

- Physical acquisition technique is a bit-by-bit copy of an entire physical store. It has the advantage of allowing deleted files and data remnants to be examined. Physical extraction acquires information from the device by direct access to the flash memories. Generally this is harder to achieve because the device vendors needs to secure against arbitrary reading of memory so that a device may be locked to a certain operator.
- Logical acquisition technique is a bit-by-bit copy of logical storage objects (e.g., directories and files) that reside on a logical store (e.g., a file system partition). Logical acquisition has the advantage that system data structures are easier for a tool to extract and organize. This usually does not produce any deleted information, due to it normally being removed from the file system of the phone. However, in some cases the phone may keep a database file of information which does not overwrite the information but simply marks it as deleted and available for later overwriting.
- Manual acquisition technique as kind of utilizing of the user interface to investigate the content of the memory. Therefore the device is used as normal and pictures are taken from the screen. The disadvantage is that only data visible to the operating system can be recovered and that all data are only available in form of pictures.

The last acquisition has no difference among of BlackBerry or Android, so I miss this. Logical techniques often provide easy and fast data extracting and accessing that physical cause of time operating. Logical methods manage with non-deleted data are accessible on the storage. The point is that previous case is about "simple" data type(format), while SQL db files as all-in-one file may keep deleted data in the database. While recovery of the deleted data requires special tools and techniques, it is possible to recover deleted data from a logical acquisition. Physical techniques as techniques aimed to gain deleted data without relying on the file system itself to access the data, so it is missed too.

Let's gain the main logical acquisition differences between two kind platform throughout way to data store, developers API and tools, free and paid investigation tools, logs, backup some more and others tricks.

Forensic Investigation of the Android vs BlackBerry

A BlackBerry is a handheld mobile device engineered for email. All models now come with a built-in mobile phone, making the BlackBerry an obvious choice for users with the need to access their email from somewhere besides the comfort of a desk chair. The BlackBerry device is always on and participating in some form of wireless push technology. Because of this, the BlackBerry does not require some form of desktop synchronization like the other mobile device does. BlackBerry OS has numerous capabilities and features like over the air activation, ability to synchronize contracts and appointments with Microsoft Outlook, a password keeper program to store sensitive information and the ability to customize your BlackBerry display data.

An Android powers millions of phones, tablets, and other devices and brings the power of Google and the web into your hands. With an amazingly fast browser, cloud sync, multi-tasking, easy connect & share, and the latest Google apps (and thousands of other apps available on Google Play) your Android powered device is beyond smart. Android has a large community of developers writing applications (“apps”) that extend the functionality of the devices. While Android is designed primarily for smartphones and tablets, the open and customizable nature of the operating system allows it to be used on other electronics, including laptops and netbooks, smartbooks, ebook readers, and smart TVs (Google TV). Further, the OS has seen niche applications on wristwatches, headphones, car CD and DVD players, smart glasses, refrigerators, vehicle satnav systems, home automation systems, games consoles, mirrors, cameras, portable media players landlines, and treadmills.

Push-Technology

You see the changes provide goals a wide-spaced. Since the BlackBerry is all always on, push messaging, device information can be pushed to it at any time. Note that pushed information has the ability to overwrite any data that possibly was previously deleted. The BlackBerry device is not really “off” unless power is removed for an extended period. If the blackberry is powered back off then any items that were in the queue waiting to be pushed to the device could possibly be pushed before you could stop them. In android case, you have a bit more time to set state, you even may don’t touch it to not update email folder except inbox folder and malicious cases like BlackBerry Playbook not real BlackBerry device BIS or BES data plan. Android brings push feature only with enterprise connection, after ~5 seconds you press power button to display turn on or when you run applications, however even gmail application need a time or manually “update”-button pressing to retrieve new data from Internet.

Password Protection

BlackBerry devices come with password protection. The owner has the capability to protect all data on the phone with a password. He may also specify the amount of attempts for entering the password before wiping all data from the device. If you exceed your password attempts limit (defaults to 10, but you can set it as low as 3, Playbook may differ from 5 to 10), you will be prompted one last time to type the word BlackBerry. The device will then wipe. It will be reset to the factory out-of-the-box condition, and the password reset. You will lose everything in the device memory, with no possibility of recovery. It will not reformat the microSD card if it’s smartphone external storage, because that’s not part of the factory configuration, but if you have a BlackBerry Playbook you’ll get factory defaults at all. The phone will still be usable, and the operating system will be unchanged. So this technique cannot be used to roll back from an OS upgrade problem.

The ability to circumvent the pass code on an Android device is becoming more important as they are utilized frequently and, in most cases, do not allow data extraction as well as for BlackBerry. While there is no guaranteed method, there are a number of techniques which have worked in certain situations. There are three types of pass codes Android devices currently support. The first is a pattern lock as default on the initial Android devices when users are accessing the device should draw a pattern on the locked phone. The second type of pass code is the simple personal identification number (PIN) which

is commonly found on other mobile devices. The final type of pass code currently found on Android devices is a full, alphanumeric code. If you the screen of the device is active, strong consideration should be given to checking and potentially changing its settings. For devices that have pass codes, there is a short period of time (from less than a minute up to about 1 hour) where full access to the device is possible without re-entering the pass code. Sometimes possible to determine the pattern lock of a device by enhancing photographs of the device’s screen. The lesser the interaction a first responder has with the screen, the higher the success rate of this technique.

Password Extraction/Bypassing

So-called Smudge Attack for Android’s pattern lock

As Android devices used the pattern lock for pass code protection instead of a numeric or alphanumeric code, there’s a interesting option that a clean touch screen is primarily, but touch screen marked with fingerprint and fingerprint’s directed a good solution to bypass pattern lock.

Screen Lock Bypass App for Android

Security researcher Thomas Cannon recently developed a technique that allows a screen lock bypass by installing an app through the new web-based Android Market. This technique utilizes a new feature in the web-based Android Market that allows apps to be installed directly from the web site. As such, you must have access to the Android Market using the primary Gmail user name and password for the device, which may be accessible from the primary computer of the user. Alternatively, you could access the Android Market if you knew the user name and password and had sufficient authority. Changing the user’s Gmail password would not work in this instance.

The procedure is quite simple really. Android sends out a number of broadcast messages which an application can receive, such as SMS received. An application has to register its receiver to receive broadcast messages and this can be done at run time, or for some messages, at install time. When a relevant message comes in, it is sent to the application and if the application is not running it will be started automatically. Once launched it is just a matter of calling the `disableKeyguard()` method in `KeyguardManager`. This is a legitimate API to enable applications to disable the screen lock when, say, an incoming phone call is detected. After finishing the call the app ought to enable the screen lock again, but we just keep it disabled.

Use Gmail User/Pass for Android

On most Android phones, you can circumvent the pass code if you know the primary Gmail user name and password registered with the device. After a number of failed attempts (ten attempts on the G1), you will be presented with a screen that asks if you forgot your pass code. From there, you can enter the Gmail user name and password and you will then be prompted to reset the pass code. This technique does not require the phone to be online as it uses credential information cached on the phone. So, if you’ve already get somehow this credential data, it’s good. Others, if you do not have the current Gmail user name and password, but have sufficient authority (i.e., court order) to reset the password, you could attempt to compel Google to reset the account password. You would then have to connect the Android device to the network and gain access. This issue presents many challenges, including the need to place the de-

vice online, putting it at risk for remote wipe in addition to making changes to the device.

Password brute-force for BlackBerry

You can access encrypted information stored in password-protection backups if the original password is known or recovered with Elcomsoft Phone Password Breaker (<http://www.elcomsoft.com/eppb.html>). Elcomsoft Phone Password Breaker grants forensic access to protected information stored in BlackBerry devices by recovering the original plain-text password. The toolkit allows eligible customers acquiring bit-to-bit images of devices' file systems, extracting phone secrets (passcodes, passwords, and encryption keys) and decrypting the file system dump. Access to most information is provided in real-time. In addition to Elcomsoft Phone Password Breaker, the toolkit includes the ability to decrypt images of devices' file systems, as well as a free tool that can extract the encrypted file system out of the device in raw form. To unlock Apple backups even faster, the tool engages the company's patent-pending GPU acceleration technology.

Three key features are:

- Decrypt encrypted BlackBerry backups
- Recover original plain-text passwords
- GPU acceleration

Spyware for BlackBerry

As some kind of attack as was presented by Thomas Cannon and previously described, you had have installed spyware to extract password from device. Almost of all possible techniques to live extracting from BlackBerry was discussed several times in my articles, so I briefly remind it some tricks. First tricks exploits default feature to show password without asterisks that's a possible to screen-capture. If restricted API disable you've have a BIS device, it works. Second trick is about scaled preview for typed character through virtual keyboard. Third tricks provides you techniques to steal password during synchronization from BlackBerry Desktop Software as well as redrawing your own fake-window to catch typed password.

Classic Forensic

A typical forensic investigator performs the investigation by hand-reading mail and data files, checking for system activities through different log files, and verifying the consistency of the data through the time stamps associated with files on the file system. First, forensic software must be running on the local machine, and may have to be installed. Second, running such software locally risks damaging or contaminating data. Third, if the machine has been compromised, the investigation may produce suspect results - or worse, may alert the attacker.

Gathering Logs and dumps

BlackBerry

The main classic forensic procedure of evidence collection violates the forensic method by requiring the investigator to record logs kept and dump. Investigator can view some log on the device pressing hotkeys or throughout several applications from BlackBerry SDK Tools. Don't forget that the counter is always running, even when the radio is turned off, so to be sure to record these values as soon as possible to avoid log overwrites. BlackBerry hotkeys for quickly extracting log data was discussed with details in my articles "To Get Round To The Heart

Of Fortress". As I know Android didn't provide the same hotkeys. These log events depend on debug information added by developers, so it often may not exist.

Another way to collect the log information is using loader.exe from BB SDK tools or BBSAK. It extracts a full copy of BlackBerry event log to text file stored on your drive. Let's see some useful command of javaloader.

JAVA LOADER USAGE

Usage: *JavaLoader [-p<pin>] [-d0|-d1] [-w<password>] [-q] <command>*

-p<pin>	Specifies the handheld PIN (hex pin prefix '0x')
-w<password>	Connects using the specified password
<command>	is one of
dir [-d] [-s] [-l]	Lists modules on the handheld
-d	Display dependency information
-s	Display siblings
-l	Single column output
deviceinfo	Provides information on the handheld
save {<module> ... -g <group>}	Retrieves modules from the handheld
-g	Retrieves all modules in a specified group
info [-d] [-s] [-v] <.cod file>	Provides information on the specified modules
-d	Display dependency information
-s	Display sibling information
-v	Display verbose module information
eventlog	Retrieves the handheld event log
radio on off	Turns the handheld's radio on or off
siblinginfo <.cod file>	Provides sibling information on the specified modules
screenshot <.bmp file>	Retrieves the contents of the specified screen and saves as a BMP file.
logstacktraces	Dumps the stack traces for all threads to the event log

To extract event log from device

- Plug it to PC via USB cable
- Open command shell and type *javaloader.exe -wPASSW eventlog log.txt* where *PASSW* your password for device.

Command *dump* gives us all .cod modules stored on device in root subfolder *dump*. To get dump of BlackBerry device let's use a Loader from BlackBerry Device Manager.

LOADER USAGE

Usage: *loader.exe /<command>*

command	is one of:
eventlog	output filename
screenshot	output filename
screenshot active	output file
screenshot primary	output file
screenshot auxiliary	output file
deviceinfo	output filename
dir	output filename
radio	on off
dump	output filename

Dump extracting is the same the log previous. However, before you will be asking to enter a device's password. Note, dump beginning is required a device reboot. It can erase log to overwriting some information. Do not forget about encryption feature of BlackBerry Storage Protection based on Password & ECC. If it is on the dump result is empty obvious. Dumps and logs will provide you information about device like hardware id, pin, os version, others id, name-version-size-created date for .cod modules with their dependency as well as vendor info or description. Event log also can provide with date-time stamp and guids of applications.

Android

As some kind of data storage mechanism available to developers is the network to store and retrieve data on your own web-based services via packages named as java.net and android.net. These packages provide developers with the low-level API to interact with the network, web servers, etc. As an interesting example, such files (text log or xml) may store an actions with date and time stamps, error/warning/successful authenticate events, logins, some data as email addresses, access keys,

private keys or application id keys as well as SQL db files may store all upload, downloaded and transferred data via an application often without ciphering. They contain as much more data than BlackBerry at first glance, however, if developers didn't hear about it or didn't build them, they might get anything valuable.

The most know developer tool and command from Android SDK is adb pull command that provides copying to the files to desktop workstation for further analysis. Unless an Android device has root access or is running a custom ROM, the adb daemon running on the device that proxies the recursive copy only runs with shell permissions. As such, some of the more forensically relevant files are not accessible. However, there are still files which can be accessed. Successful accessing aims to extracting(copying) the entire "/data" partition to the local directory. If devices has not have root access, this technique may appear to be of little value. However, on nonrooted devices, an adb pull can still access useful files such as unencrypted apps, most of the tmpfs file systems that can include user data such as browser history, and system information found in "/proc," "/sys," and other readable directories (Table 3).

Table 3. ANDROID DEBUG BRIDGE (ADB) USAGE

command	description
-d	directs command to the only connected USB device; returns an error if more than one USB device is present.
-s <serial number>	directs command to the USB device or emulator with the given serial number. Overrides ANDROID_SERIAL environment variable.
devices	list all connected devices
connect <host>[:<port>]	connect to a device via TCP/IP Port 5555 is used by default if no port number is specified.
disconnect [<host>[:<port>]]	disconnect from a TCP/IP device. Port 5555 is used by default if no port number is specified. Using this ocmmand with no additional arguments will disconnect from all connected TCP/IP devices.
device commands:	
adb push <local> <remote>	copy file/dir to device
adb pull <remote> [<local>]	copy file/dir from device
adb sync [<directory>]	copy host->device only if changed (-l means list but don't copy) (see 'adb help all')
adb shell	run remote shell interactively
adb shell <command>	run remote shell command
adb logcat [<filter-spec>]	View device log
adb forward <local> <remote>	forward socket connections forward specs are one of: tcp:<port> localabstract:<unix domain socket name> localreserved:<unix domain socket name> localfilesystem:<unix domain socket name> dev:<character device name> jdwp:<process pid> (remote only)
adb jdwp	list PIDs of processes hosting a JDWP transport
adb install [-l] [-r] [-s] <file>	push this package file to the device and install it ('-l' means forward-lock the app) ('-r' means reinstall the app, keeping its data) ('-s' means install on SD card instead of internal storage)
adb uninstall [-k] <package>	remove this app package from the device ('-k' means keep the data and cache directories)
adb bugreport	return all information from the device that should be included in a bug report.
adb help	show this help message
adb version	show version num
DATAOPTS:	
(no option)	don't touch the data partition
-w	wipe the data partition
-d	flash the data partition

Data Extracting through the Backup

Android

Android did not provide a mechanism for users to backup their personal data. As a result, a large number of backup applications were developed and distributed on the Android Market. For users running custom ROMs, there was an even more powerful backup utility developed called nandroid. Many of the backup utilities have a “Save to SD Card” option (which users found extremely convenient) as well as several options to save to “the cloud.” Either way, users could take a backup of their devices, and if needed they could restore required data. This is not only a great way for users to protect themselves from data loss, but it can be a great source of information for forensic analysts.

Anyway, backup area is covered by following items:

- Application install files (if phone has root access, this includes APK Data and Market Links)
- Contacts
- Call log
- Browser bookmarks
- SMS (text messages)
- MMS (attachments in messages)
- System settings
- Home screens (including HTC Sense UI)
- Alarms
- Dictionary
- Calendars
- Music playlists
- Integrated third-party applications

Despite of that the backup API is now available the synchronization provide outlook linking.

Regardless of the backup app, forensic analysts should determine if one was installed and, if so, where the backup data is stored. The SD card should be examined as well as other devices such as a computer or laptop. The data saved in a backup is obviously of significant value in an examination.

BlackBerry

First, you need to download and install BlackBerry Desktop Manager. Use the following link to select and download the install file that fits your system or version. Once BB Desktop Manager installed, connect the device to PC. Then Click “Back up” button for a full backup of the device or use the advanced section for specific data. In the options, you can find a destination folder where your “.ipd” file will save. Note, that ipd-file can be encrypted with password less even than 4 characters. BlackBerry backups contain essential information stored in the device. User data such as email, SMS and MMS messages, Web browsing history and cache, call logs, pictures and photos, contacts, calendars, appointments, and other organizer information are stored in BlackBerry backups. Access to information stored in BlackBerry backups can be essential for investigations, and is in high demand by forensic customers. Note, that the backup file does not save your email attachments, moreover if email-message is more than to 8Mb data Base64 non-encoded per whole file (if attachments more than one then each file will encoded and summary size limits more faster), there will be only a message with notification about truncation. The most known tool to extracting data from .ipd files are MagicBerry IPD Reader, Amber BlackBerry Converter, Elcomsoft BlackBerry Backup Explorer, Paraben

Device Seizure. So, what you’ll be able to do with “Magic Berry IPD Parser”:

- Read ipd files
- Split ipd files
- Export MS Messages, Phone Calls Log, Memos, Tasks, Calendar, and Address Book to CSV
- Edit Service Books
- Merge two ipd files

Elcomsoft BlackBerry Backup Explorer allows forensic specialists investigating the content of BlackBerry devices by extracting, analyzing, printing or exporting the content of a BlackBerry backup produced with BlackBerry Desktop Software. Elcomsoft BlackBerry Backup Explorer supports BlackBerry backups made with PC and Mac versions of BlackBerry Desktop Software. You can export information from BlackBerry backups into a variety of readable formats (PDF, HTML, DOC, RTF,...). Also BlackBerry Backup Explorer can access encrypted information stored in password-protection backups if the original password is known or recovered with Elcomsoft Phone Password Breaker. Elcomsoft Phone Password Breaker grants forensic access to protected information stored in BlackBerry devices by recovering the original plain-text password. Elcomsoft BlackBerry Backup Explorer is totally the same with Amber BlackBerry Converter.

As an alternative to acquiring the BlackBerry through “BlackBerry IPD Reader”, Paraben’s Device Seizure is a simple and effective method to acquire the data. Device Seizure was designed from the ground up as a forensic grade tool that has been upheld in countless court cases.

- SMS History (Text Messages)
- Deleted SMS (Text Messages)
- Phonebook (both stored in the memory of the phone and on the SIM card)
- Call History
 - Received Calls
 - Dialed Numbers
 - Missed calls
 - Call Dates & Durations
- Scheduler
- Calendar
- To-Do List
- Filesystem (physical memory dumps)
 - System Files
 - Multimedia Files (Images, Videos, etc.)
 - Java Files
 - Deleted Data
- GPS Waypoints, Tracks, Routes, etc.
- RAM/ROM
- PDA Databases
- E-mail

There’s a briefly general draft to examine data with Paraben Device Seizure.

- Create a new case in Device Seizure with File | New.
- Give the case a name and fill in any desired information about the case on the next two screens. The third screen is a summary of the data entered. If all data is correct click Next and then Finish.
- You are now ready to acquire the phone. Go to Tools | Data Acquisition.

- You are prompted for the supported manufacturer. Select RIM Blackberry.
 - Leave supported models at the default selection of autodetect.
 - Connection type should be set to USB.
 - For data type selection select Logical Image (Databases).
 - Confirm your selections on the summary page and click Next to start the acquisition.
- ### BlackBerry Simulation
- This feature unfortunately unavailable for Android, so it will be discussed only for BlackBerry. BlackBerry Simulator built for simulating a backup copy of the physical device. This is helpful if the device is low on battery, needs to be turned off, or you do not want to alter the data on the physical device. Following steps are suitable for each BlackBerry device model.
- Select a simulator from the drop-down list on the BlackBerry website and download it. Then install it
 - Select and download BlackBerry Device Manager. Then install it.
 - Run BlackBerry Device Manager and BlackBerry Simulator
 - Select Simulate | USB Cable Connected.
 - Select File | Restore to simulate with physical data evidence on BlackBerry Simulator.
- *Process Management*
(both Android ' n BlackBerry)
 - *Memos and Tasks*
(seems only BlackBerry)
 - *Screen-shots*
(both Android ' n BlackBerry)
 - *Camera-shots*
(both Android ' n BlackBerry)
 - *Videocamera-shots*
(both Android ' n BlackBerry)
 - *Clipboard*
(both Android ' n BlackBerry)
 - *Location tracking*
(cell, wifi, gps, bluetooth)
(both Android ' n BlackBerry)
 - *SMS/MMS/Emails*
(both Android ' n BlackBerry)
 - *Pictures, Videos, Voice notes, and other file*
(both Android ' n BlackBerry)
 - *File and Folder structure*
(both Android ' n BlackBerry)
 - *IMs*
(both Android ' n BlackBerry)
 - *Passwords*
(very differ)

Also, you mount a SD-card "copy" to the BlackBerry Simulator. Now you may turn off blackberry wireless communication holding power on and then examine evidence with up state device-simulator.

Live (Spy) forensic

In some situations, it is not desirable to shut down, seize the digital device, and perform the forensic analysis at the lab. For example, if there is an indication that an encryption mechanism is used on the digital device that was discovered, then the investigator should not shutdown this digital device. Otherwise, after shutdown all the information (potential evidence) that was encrypted will be unintelligible. By performing Live Analysis, the investigators attempt to extract the encryption key from the running system. That's known as "Live Analysis" or "Non-Classic Forensic". The goal of any live forensics task should be to extract and preserve the volatile data on a system while, to the extent possible, otherwise preserving the state of the system. Additionally, this is often the first step of an incident response scenario where a handler is simply trying to determine if an event has occurred. The benefit of using this approach is you have a forensically sound data collection from which to proceed with a full forensic analysis if the initial analysis indicates one is required.

Potential Data as Evidence

Potential attack vector can be various, however, the most popular of them are

- *Address Book*
(both Android ' n BlackBerry)
- *Calendar Events*
(both Android ' n BlackBerry)
- *Call History*
(both Android ' n BlackBerry)
- *Browser history and bookmarks*
(both Android ' n BlackBerry)

Android's data set stores on internal storage as well as on external, but only internal storage keeps a strong folder structure because it's controlled by Android API. Typically internal place to store any kind of data is "/data/data/" where cache and databases stored in "PackageName" folder. Android data stored on internal and external storage as binary (or simply text) files as well as packed into xml or SQL-lite database formats. XML format allows including Boolean, integer, float or string data types provide developers to create, load, and save configuration values that power their application.

Internal files allow developers to store very complicated data types and saved them in several places on the internal storage that by default, can only be read by the application and even the device owner is prevented from viewing the files unless they have root access. While files stored on the internal device's storage have strict security and location parameters, files on the various external storage devices have far fewer constraints.

First, one important motivation (beyond cost) for using a removable SD card is that the data could be used on other devices, presumably upgraded Android devices. If a consumer purchased a new Android device, inserted their previous SD card containing all of his or her family pictures and videos and found they were unable to access them, they would be quite upset.

SQLite is one of the most popular database formats appearing in many mobile systems for many reasons such as high quality, open source, tend to be very compact, cross-platform file, and finally, cause of the Android SDK provides API to use SQLite databases in their applications. The SQLite files are generally stored on the internal storage under /data/data/<packageName>/databases without any restrictions on creating databases elsewhere.

All of them you can extract using the official BlackBerry API and Android API routines. Let us examine some of them to find out the common sense. What is in an up-to-date BlackBerry Address Book? A lot of contact's data, such

as several mobile or home phone number, faxes, emails, work and home addresses, web-pages or dates. Also we can add a IM data and social data. In our Address Book, we have much valuable information about friends; social network gives an up-to-date avatar, calendar (in spite of our calendar that filled our sleeping time at least), GPS location points, and SW names that provide several pieces of information. Due to victim's calendar info and GPS info (from photo exif or FaceBook likes), private data such as tracking info, habits, time marked a free, time when you're possible sleeping, time when you're at home/company can come to light. In additional, if you involve call history with gps records as two part of evidence you provide yourself with many opportunities to draw a social graph of accomplices. Extracting all possible fields from the object called PIM is goal for gathering more information about the attacked individual from their profile overall.

Classic Forensics techniques manage with BlackBerry backup file or with data stored on `"/data/data/com.android.providers.contacts"` for Android internal storage. This app stores the Call Logs for the device in the calls table. There are over 30 tables in `contacts2.db`, so further inspection may be required. The data table contains additional values about contacts and the `raw_contacts` contains additional data about some contacts extending by different accounts including Gmail, Exchange, Facebook, Twitter, and more. If pictures of the contacts are available, they are stored in the files directory and named `thumbnail_photo_[NNNNN].jpg`.

Facebook data stores on `"/data/data/com.facebook"` where `fb.db` contains nearly all of the information includes albums, `info_contacts`, notifications, chatconversations, `mailbox_messages`, photos, chatmessages, `search_results`, `default_user_images`, `mailbox_profiles`, `stream_photos`, events, `mailbox_threads`, friends and others. Gmail data is located on `"/data/data/com.google.android.gm"` which stores each configured Gmail account via separate SQLite database filled by the entire e-mail content. GMaps data located on `"/data/data/com.google.android.apps.maps"` stores amount of information about maps, tiles, searches, and more in the files directory often provide by `"search_history.db"` or actual spoken directions stored as map data on the SD card in `.wav` files; the time stamps on the file prefaced with a `"_speech"` simplify movement timeline.

Mentioned on the net password tips are revoked by the tendency in matter to complexify. How many web sites do you log in, Facebook, Myspace, LinkedIn, Twitter and any number of other social networking sites? Probably a dozen. Shopping sites? Yes, a several. Emails, IMs, etc. Every site requires you to create a password, strong password. Some kind people solve it with digit wallet. All password managers are describing, as is indispensable tool for the active internet and shopping user. In addition, it fully automates the process of entering passwords and other data into websites and saves the user going to the trouble of creating and remembering multiple passwords. It is still unsecured. Do not neglect a spyware that able to capture screens of your device. Ok, forget about that kind of malware. Examine a logical way to break into. You need to see it to type or need to copy into clipboard. Moreover, no one software producer can protect it, because need to put data into public text-box. In other words, end-point object is vulnerable. By the way, there's a `getClipboard()` method to retrieve the system's clipboard object though the BlackBerry API or Android API.

Next victim is message (sms, mms, email, further email). Email is one of the most common ways people communicate. From internal meeting requests, distribution of documents and general conversation one would be pressed to find an organization of any size that does not rely on email. Studies have shown that more email is generated every day than phone conversations and paper documents combined. Many users store their personal calendars, contacts and even synchronize their email clients with their mobile devices.

Less interesting part of evidence concludes browser history, browser bookmarks, memos, tasks, etc. Such kind of forensic has sense in case of violating company policy by visiting certain sites or time aspect (when the computer was connected to a site at the time when something happened) and reconstruct a detailed history of a computer's use by examining a handful of files that contain a web browser's past operation. One more part of it is "Favorites folder" that contains the URLs of web sites saved by the user, probably because they are of interest to the user and are frequently visited explicit storing of these links indicates intent.

As BlackBerry classic forensic extraction manage with backup again, Android provide a file-folder storage located `"/data/data/com.android.providers.telephony"` filled by the MMS attachments (images, video, or any other supported data), sms message as database table with all messages. A bit more information filepath `"/data/data/com.android.mms"` provides with cached data or data is outcoming.

Pictures, Videos, Voice notes, and other files. Let's start from its last object "other files". Voice notes, videos and pictures show us in general what interesting in particular our "victim". It may be enterprise presentation that he videocaptured or audiocaptured. This case is useful for us, because we don't need to intercept API events; all we need is listen file events of creating and deleting files.

Pictures are more inquisitive as camera-snapshots since it has exif-header. Metadata is, quite simply, data about data. Many digital camera manufacturers, such as Canon, Sony and Kodak implement the use of EXIF headers. This header is stored in an "application segment" of a JPEG file, or as privately defined tags in a TIFF file. This means that the resulting JPEG or TIFF is still in a standard format readable by applications that are ignorant of EXIF information. However, not only basic cameras have these headers, but both mobile devices provide you "Camera Make" as RIM/BlackBerry/Android/HTC data as well as "Camera Model" may often be device model. GPS tag often renames filename by placing into beginning city name. To get date and time stamps you don't need to examine EXIF, because it's enough to check file name again.

Android Media database located on `"/data/data/com.android.providers.media"` contains contain the volume ID as a file system volume ID. If an image was deleted, the thumbnail likely still exists. Also, even if the metadata record is deleted, it is likely recoverable due to the YAFFS2 file system. Also this place is scanned for audio files, albums, and etc by media scanner to find media data or thumbnails referred to the deleted pictures and videos. Also, YouTube preferences, including device key(s) and watched videos stores in `"/data/data/com.google.android.youtube/"`, cached data stores in `"/data/data/com.google.android.youtube/cache"`.

Instant messaging is a well-established means of fast and effective communication. IM forensic were to answer the two questions as identifying an author of an IM conversation based strictly on author behavior and classifying behavior characteristics

For example, BlackBerry stores all chats (from Google, Yahoo, Windows Live, BlackBerry Messenger, AIM(AOL)) in plain-text mode in .csv file. File paths are often easy to find too.

Conclusion

The BlackBerry devices as well as Android devices share the same evidentiary value as any other Personal Digital Assistant (mobile device). As the investigator may suspect of most file systems, a delete is by no means a total removal of data on the device. However, the RIM's always-on, wireless push technology adds a unique dimension to forensic examination. Android, instead tends to be more offline and wake up by user actions.

As the BlackBerry is an always-on, information can be pushed to the device through its radio antenna at any time, potentially overwriting previously "deleted" data. Without warning, applications such as the email client, instant messaging, wireless calendar, and any number of third party applications may receive information that makes the forensic investigator's attempts to obtain an unaltered file system much more difficult. In order to preserve the unit, turn the radio off. You make release the same action for Android, however, you need to perform this quickly and the two best ways a Faraday Cage or Airplane mode. Airplane mode may be harmful because, the device still continues interacts with local data. Otherwise you may not be access to active devices to bypass password. As a native feature android device have a pattern lock bypassed via fingerprinting; blackberry, instead, doesn't provide this techniques, however third-party application may be easy found on market especially for Playbook as a tablet.

Classic forensics for Android a bit easy than BlackBerry, because for BlackBerry there's no way except having a BlackBerry Backup file. Moreover, this backup file may be emulated after you restore this on BlackBerry Simulator via USB Plugged option; SD card may be copied into folder and attached to simulator. Android, instead, doesn't have this feature, but you can extract all database files from plugged device more successful if it's rooted.

If the RIM is password protected, you have to get the password, because the password doesn't stor on the unit; rather an SHA-1 hash of the password stored and compared to a hash of what entered. The examiner only has the opportunity to guess 10 times before a file system wipe occurs to protect the data. This wipe will destroy all non-OS files. No software exists to circumvent the password protection. A direct-to-hardware solution will be required if the password is not available. Android devices present opportunity to (after unsuccessful attempts rich limit) unlock device via Google credentials that leads to strongly rule named "placing device online". It's a kind of risk to add some changes, but it's a better way than BlackBerry. All live techniques may be valuable when you've installed "spyware" but don't offer a successful end according to password. On other hand, live techniques offer you simplifying of investigation, because you don't need to analyze a SQL-Lite database and can extract data in any suitable format. Live techniques covers the same points data interested for researchers, so there's no valuable difference between BlackBerry and Android. Commercial tools as well as free provide enough covering data extracting via live techniques without needs for develop them. Thus, the RIM's currently unsurpassed portability is the examiner's greatest ally more than android if we're talking about password. If we need

to emulate data – BlackBerry provides more native user tools to prevent change becoming.

On the Net

- <http://www.amazon.com/Android-Forensics-Investigation-Analysis-Security/dp/1597496510> - Android Forensics: Investigation, Analysis and Mobile Security for Google Android. Andrew Hoog
- <http://hakin9.org/to-get-round-to-the-heart-of-fortress/> - To Get Round To The Heart Of Fortress. Hakin9 Extra. Yury Chemerkin

YURY CHERMERKIN

Graduated at Russian State University for the Humanities (<http://rggu.com/>) in 2010. At present postgraduate at RSUH. Information Security Researcher since 2009 and currently works as mobile and social infosecurity researcher in Moscow. Experienced in Reverse Engineering, Software Programming, Cyber & Mobile Security Researching, Documentation, Security Writing as regular contributing. Now researching Cloud Security and Social Privacy.

Contacts:

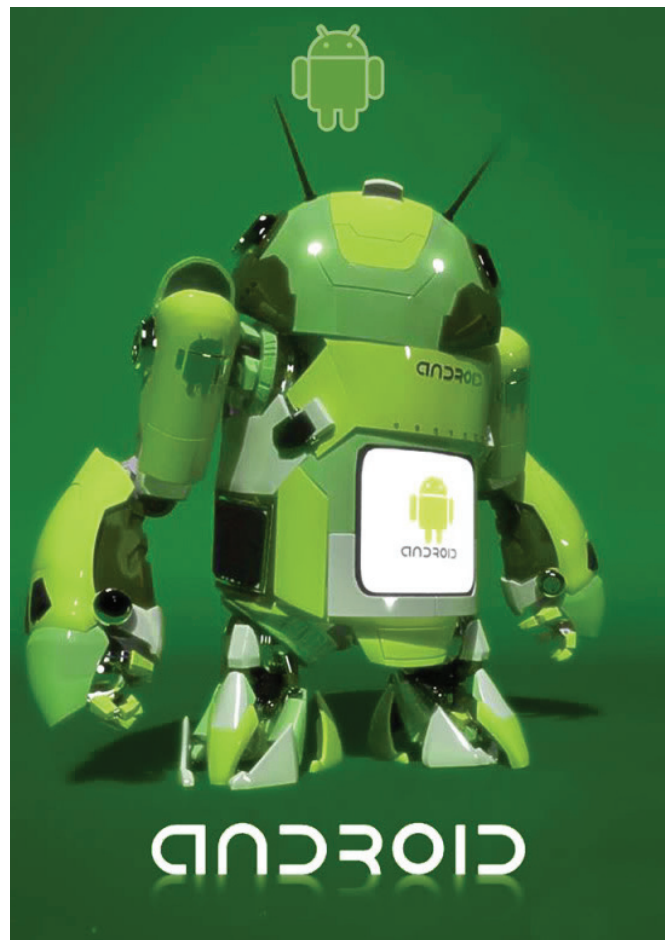
I have a lot of social contacts, that's way you're able to choose the most suitable way for you.

Regular blog: <http://security-through-obscurity.blogspot.com>

Regular Email: yury.chemerkin@gmail.com

Skype: [yury.chemerkin](https://www.skype.com/en/contacts/yury.chemerkin)

Other my contacts (blogs, IM, social networks) you'll find among http links and social icons before TimeLine section on Re.Vu: <http://re.vu/yury.chemerkin>
<http://4.bp.blogspot.com/-R09jvrMJW6I/TzARr9Ksx6I/AAAAAAAAACDo/9CRo9LDMjJ0/s1600/Android+robot+Wallpaper+2012+new+hq.jpg>





[GEEKED AT BIRTH.]

PWR: 110%

IM Geek PH: 877 IUAT

[IT'S IN YOUR PULSE.]

LEARN:

Advancing Computer Science
Artificial Life Programming
Digital Media
Digital Video
Enterprise Software Development
Game Art and Animation
Game Design
Game Programming
Human-Computer Interaction
Network Engineering

Network Security
Open Source Technologies
Robotics and Embedded Systems
Serious Game and Simulation
Strategic Technology Development
Technology Forensics
Technology Product Design
Technology Studies
Virtual Modeling and Design
Web and Social Media Technologies



You can talk the talk.
Can you walk the walk?

www.uat.edu > 877.UAT.GEEK

DATA HIDING TECHNIQUES

UGUR EKEN

Abstract

Data hiding can be classed as one of the important methods of anti forensic technique which can be implemented by many open source and commercially available tools by having access to hidden data in storage media, file system and in applications when it is possible and make it difficult for forensic examiners where time and costs are very crucial in order to get a conviction or prove an innocence.

History

The art of hiding information has been with us thousands of years and it goes back to ancient Greece. The Herodotus ancient Greek historian lived in the 5th century BC states that Histiaeus the tyrant of the Miletus wanted to send a message to his son-in-law Aristagoras to rebel against Persians. Histiaeus called one of his trusted slaves, shaved his head and tattooed the message on his head. When the slaves hair was long enough to cover the hidden message Histiaeus sent his slave to Aristagoras in order to deliver the secret message and consequently the message had been delivered successfully.

In 20th and 21st century the idea of hiding information and covert communication stayed the same but an advance in technology provided new tools and techniques that allowed us to hide large amounts of information in a digital form. However this great opportunity introduced great challenges in form of anti forensics for computer forensic examiners in order to hide many illegal and destructive data such as password loggers, key loggers, Trojans, Viruses, copyright materials, child pornography, intelligence concerning national security, etc.

In this article I will be explaining some of the major data hiding techniques, potential data hiding areas, and forensic examination techniques of exploiting data hiding implementations in a storage media and a file systems. In order to explain these techniques and give a general idea I will be using NTFS file system for its complexity and with it been one of the mostly used file system in today's computing.

Data Hiding Techniques

The storage devices are used by computers in order to store and retrieve users digital form of data. These devices are manufactured in different architectures and sizes that it can be divided into two categories such as primary(volatile memory) and secondary (non-volatile memory) storage. The volatile memory requires a constant power supply in order to keep digital data. The Random Access Memory is a great example for volatile memories and it is a great storage space for hiding malicious data such as Viruses, Trojans, and the Worms. Hiding such a data in a volatile memory has great strengths that it gives at-

tacker capability of storing and executing malicious code and be able to destroy this data immediately after when the power supply is switched off. This introduces certain challenges for forensic examiners that examination of volatile memory requires live system. There are many great open source and commercial tools to examine volatile memory in order to examine malicious data as well as encryption keys.

The non-volatile memory totally opposite to volatile memory does not require constant power supply. This type of memory also known as secondary storage is used for digital data to be stored in long term basis. The ROM (Read Only Memory), hard disk drives, magnetic tapes, and optical drives are great examples for this types of storage.

Since data retained is stored in a non-volatile memory even when it is not powered this creates a great data hiding ground for opportunists through large data storage capacity and the providing capability to access hidden data when it is needed. In next chapters I will be explaining potential data hiding areas in a secondary storage devices.

Data hiding techniques takes advantage of slack space and unallocated space created during the formatting process while logical data structures such as file system, partitions, system records, and files mapped into physical drive[Bergel 2007] and also takes advantage of vulnerabilities in system data structures. These are some major areas which I would like to discuss and each of these areas has their own strengths and weaknesses for hiding data. These areas are;

- Physical Layer
- File System Layer

The data hiding can also be implemented in the application layer such as steganography and it is a wide area of subject. In this article I will be mainly concentrating on physical and file system layers.[Knut Eckstein, M.J.2005]

Physical Layer

The data hiding techniques in physical layer takes advantage of limited accessibility of Operating Systems and architecture of the physical drives. The following areas are main areas where data can be hidden in this layer;

- Volume Slack
- File System Slack
- Host Protected Area/Device Configuration Overlay

The digital data stored in fixed equal size logical data units known as sectors and clusters. For example single sector can

be 512 bytes in size and the consecutive series of sectors forms the clusters. The volume slack occurs when the file system size doesn't exactly match with the volume size in one to one basis. When the file system is mapped into volume this left and unused space between the file system and the volume becomes potential area for data hiding. For example following "hdparm" output shows total amount of sectors in the volume.

```

ueken@fedora15:/home/ueken
File Edit Tabs Help
[ueken@fedora15 ~]$ su
Password:
[root@fedora15 ueken]# man hdparm
[root@fedora15 ueken]# hdparm -ig /dev/sdb

/dev/sdb:
SG_IO: bad/missing sense data, sb[]: 70 00 05 0
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
geometry      = 1018/124/62, sectors = 7831552,
HDIO_GET_IDENTITY failed: Invalid argument
[root@fedora15 ueken]#
    
```

Figure 1. hdparm output showing total amount of sectors in volume

As we can see in Figure 1. there are 7831552 sectors in the entire volume. Since we determined total amounts of sectors in the volume the next step is the find out total amount of sectors in file system for determining volume slack.

```

FILE SYSTEM INFORMATION
-----
File System Type: NTFS
Volume Serial Number: 9674E14874E12C25
OEM Name: NTFS
Volume Name: NTFS
Version: Windows XP

METADATA INFORMATION
-----
First Cluster of MFT: 262144
First Cluster of MFT Mirror: 2
Size of MFT Entries: 1024 bytes
Size of Index Records: 4096 bytes
Range: 0 - 256
Root Directory: 5

CONTENT INFORMATION
-----
Sector Size: 512
Cluster Size: 4096
Total Cluster Range: 0 - 978942
Total Range in Image: 0 - 983999
    
```

Figure 2. TSK fsstat output for file system details

The Figure 2 indicates total file system sector range is 978942 sectors. When we subtracted the total file system sector range from the total volume sector range we can see that 1 sector of volume slack is exist in this system. The forensic examiner can extract actual content of volume slack by running following hdparm, fsstat and Linux dd commands or by using commercial tools such as AccessData FTK or the EnCase.

- `hdparm -ig /dev/sdb`(This command displays total sectors in the volume)
- `fsstat ntfs.001 -f ntfs` (This command displays total sectors and clusters in file system)
- `dd if = ntfs.001 bs=512 count= (Number of sector in volume slack) skip=(Amount of sectors allocated to file system of= (Image destination folder)`

As soon as volume slack is imaged then the content of the volume slack can be examined through plain sight analysis by using hex editors, keyword searches and data carving.

The file system slack is another area in physical layer and it occurs at the end of the file system. For example two consecutive series of 512 bytes sectors forms an one cluster. The exact file system size is 5511 sectors and in this case system will dedicate 2756 clusters for the file system. In this case remaining one sector becomes a file system slack and makes it available space for data hiding.

The Host Protected Area and the Device Configuration Overlay data structures are other interesting places for data hiding in physical layer and these data structures are can be found at the end of the volume. The HPA also known as Hidden Protected Area is protected from the potential user, Operating System, and the application access for allowing manufacturer of the storage device to embed recovery system and significant system configuration backup data in this area [Carrier, 2005].

In Host Protected Area large amount of spaces can be created and these spaces can easily be overlooked by other users. However hidden data in HPA can be detected by many special forensic tools, and plain examination. During the examination of HPA forensic examiner must consider checking status of storage device if it is on HPA mode or not. The main reason for that is if the device on HPA mode there is a high possibility there that HPA is potentially containing hidden data.

The status of Host Protected Area can be determined by following command in Linux;

- `hdparm -N /dev/sda`

This command will display the status of HPA mode. If the HPA mode state is enabled than forensic examiner can determine amount of sector allocated to HPA by calculating the difference between maximum disk sector and maximum user sector displayed in hdparm output.

FILE SYSTEM LAYER

In order to store and retrieve data from storage media in an organised and efficient manner Operating Systems need some kind of mechanisms and these mechanisms are provided by the file systems. The FAT32, NTFS, ExFAT, Ext4, UFS, and HFS are some of well known file systems in today's computing. The each file system has their own unique data structures, and method of storing and retrieving digital data from the storage

devices. For this reason techniques of data hiding and examination of system differ in each file system and requires good knowledge about the functionality of file system and its data structures.

In file system layer data hiding will be implemented in file system data structures. As I mentioned earlier in each file system these data structures differ from each other. Therefore hiding data and conducting computer forensic examinations requires learning relevant file system data structures. In order to do this Brian Carrier developed a basic reference model. The reference model provides systematic approach to the learning and also examining these file systems by categorising into five major categories and each category refers to different data structures of the file system. This way potential places for data hiding can be discovered or examined systematically. These categories are;

- File System Category
- Content Category
- Metadata Category
- File Name Category
- Application Category

The file system category provides general information about the file system and its data structures. The file system category information includes location and size of the data structures, and data units information such as sectors, clusters or block sizes. These informations are usually located at the boot sector of file systems. The information gathered from this category particularly important that the map of the file system, potential areas for data hiding can be determined and certain data structures can be manipulated in order to conduct these operations.

The content category contains actual data or contents of the file or directories. The data in the

content category stored and organised into equal size of data units such as sectors and clusters. Hiding data in file slacks, creating additional clusters and creating fake bad clusters are main data hiding techniques that can be implemented in this category .

The meta data category includes descriptive information about the files and directories. It contains

information about file and directory locations, permissions and MAC(Modified, Accessed, Created) timestamps. In this category fake bad clusters can be signed, and more clusters can be allocated to files in order to hide data.

The next category of the basic reference model is the File Name Category. Brian Carrier [Carrier, 2005] points out that the file name category data structures are needed in order to link a file and directory names with the appropriate contents related to that file and directory by using meta-data structure.

The final category is the Application Category. The application category is not necessary for the file system in order to function properly. The application category data structures contains features such as encryption, compression and journalling capabilities in order to increase efficiency in file systems[Carrier, 2005].

The file system layer provides easy access for anti forensics and there are variety of spaces available in order to hide data. However these spaces are usually small, scattered around the file system and also some of the data structures in this layer are important for functionality of file system and can be protected by checksum values. In this case hiding data in these data structures may change certain values and may cause file

system to fail. Therefore one who is hiding data should be considering how much space is needed and what category or even what layer is most suitable, how long data going to reside hidden, reverse engineering of data structures without affecting functionality of system, hiding same data more then one places as a backup, encryption and the existing methods.

In this section I will be explaining and demonstrating how data hiding techniques can be implemented in file system layer by using NTFS file system and Brian Carriers basic reference model.

NTFS FILE SYSTEM

The NTFS file system was introduced in 1993 for Windows NT 3.1 and it is supported by many Operating Systems in today's computing. The NTFS file system has complex data structures compared to other file systems and provides many application level features such as journalling, encryption, compression and be able to support large volumes such as RAID drives. However the most distinction characteristic of NTFS compared to other file systems everything is a file and these files can be located anywhere in the volume except boot sector which is located at first sector of the volume layout. For this reason NTFS file system don't have exact general volume layout like FAT32 or even ExFAT file system. However figure 3 shows potential layout of NTFS file system [NTFS.com, 2011].

Table 1. Potential NTFS File System Layout

Boot Area	Master File Table	System Files	Data Area
-----------	-------------------	--------------	-----------

HIDING DATA IN NTFS FILE SYSTEM CATEGORY

The file system category in NTFS file system contains general information about the file system. The Master File Table is most significant data structure for functionality of NTFS file system that it contains general information about all files and directories. Each file and directory has 1024 bytes entry in Master File Table. Since Master File Table is a file itself entry 0 is dedicated to \$MFT file. Therefore by examining \$MFT file entire file system can be mapped.

The MFT entries in \$MFT file contains data structures called attributes and each attribute contains different information about the files an directories. These are some of the MFT entry attribute data structures.

Table 2. Master File Table entry attributes[Carrier, 2005]

Attribute Name	Type Identifier	Description
\$STD_INFO	16	Contains meta data about directory and files
\$FILE_NAME	48	Contains file name and parent directory information
\$DATA	128	Contains contents of the files
\$ATTRIBUTE_LIST	32	Contains location of other attributes
\$OBJECT_ID	64	Contains global object identifier
\$REPARSE_POINT	192	Used for files that are reparse points
\$INDEX_ROOT	144	Root of the index tree (resident entries)
\$INDEX_ALLOCATION	160	Non resident entries stored
\$BITMAP	176	Keeps record of allocation status of clusters

The attributes are divided into two categories as resident and non resident attributes. The resident attributes are only requires

small amount of storage space and they are located at the Master File Table data structure. If the attribute needs more space than it is located into separate location and it becomes non resident attribute. The pointer in Master File Table indicates where non resident attribute for the relevant entry is located in the volume. There are many other entry files in NTFS file system and each of these entries provides different information about file system data structures.

\$BOOT FILE & \$BOOT RECORD

The \$BOOT file occupies seventh entry of the Master File Table and \$DATA attribute of \$BOOT file is located at the first sector (sector 0) of the file system. This meta data file contains crucial information for hiding data and examining file system. These data includes sector and cluster sizes, location of data structures such as MFT and total amount of sectors in file system that this information will help in order to determine volume and file system slack and create map of the file system. The many open source and commercial tools will provide this information. This table and figure explains data structure of NTFS boot sector.

Table 3. NTFS Boot Sector Data Structure[Carrier, 2005]

The \$BOOT file occupies first sixteen sector of the file system. The half of the allocated sectors contain non-zero values in boot file and rest of the sectors contains 0's. But is it possible to hide any kind of data in this available eight sectors in \$BOOT file. In order to determine this I used following Operating Systems, tools, storage media and procedures.

- **Windows 7 Professional Operating System** (Updated at Tuesday 3 April 2012 16:41:21 BST (UTC/GMT London)) I used this Operating System in order to run "CHKDSK" command for checking relevant storage media for any errors, formatting storage device with NTFS file system, and cross referencing the results with FTKImager hex utility.
- **Windows XP Mode** (Windows XP Professional Service Pack 3 running on Windows XP Mode) This Operating System is used in order to run HxD Hex Editor and hide data in \$BOOT file.
- **HxD Hex Editor** (Version 1.7.7.0 April 3, 2009) This tool is used in order to edit contents of \$BOOT file in Windows XP
- **FTKImager 3.1.0** The end results confirmed by using hex viewer utility of FTKImager
- **Windows CHKDSK command** This command is used in order to determine integrity of the file system.
- **Lexar JD FireFly 4GB USB Storage Media** This media is used in order to find out outcome of data hiding procedures in NTFS file system \$BOOT file
- **NTFS File System** (Version 3.1 Non-Bootable) The NTFS file system version 3.1 used in order find out outcome of data hiding procedures in NTFS file system \$BOOT file

In order to determine results the first relevant storage device formatted with NTFS file system in Windows 7 environment with default values and this procedure is followed by checking the file system and storage device for errors by using Windows CHKDSK utility. When Windows CHKDSK utility run following results found and they are indicated in Figure 3.

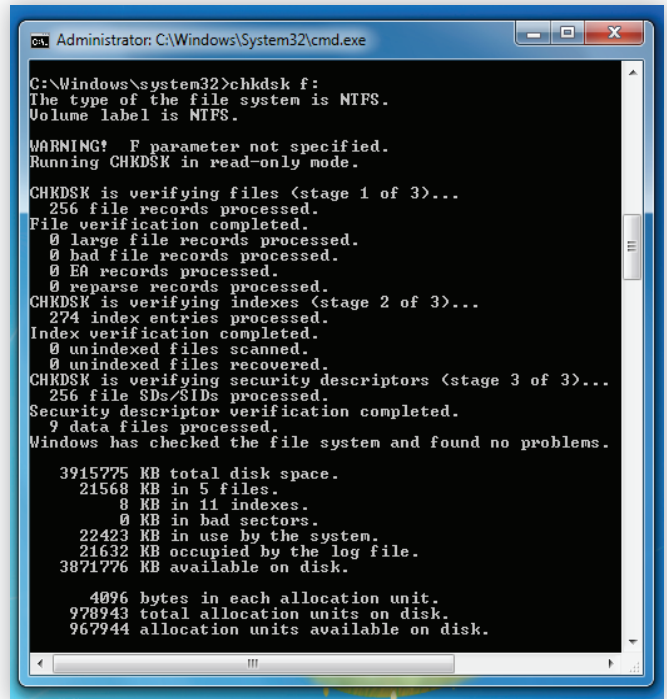


Figure 3. CHKDSK results in order to check integrity of system.

The results indicates that storage media and file system has no errors. This procedure followed by mounting relevant storage device to XP Operating System running on Windows XP Mode in order to hide data by using HxD Hex Editor. The data hiding procedure is implemented from sector 8 (started from offset 4144) up to sector sixteen. The Figure4. shows implementation of data hiding.

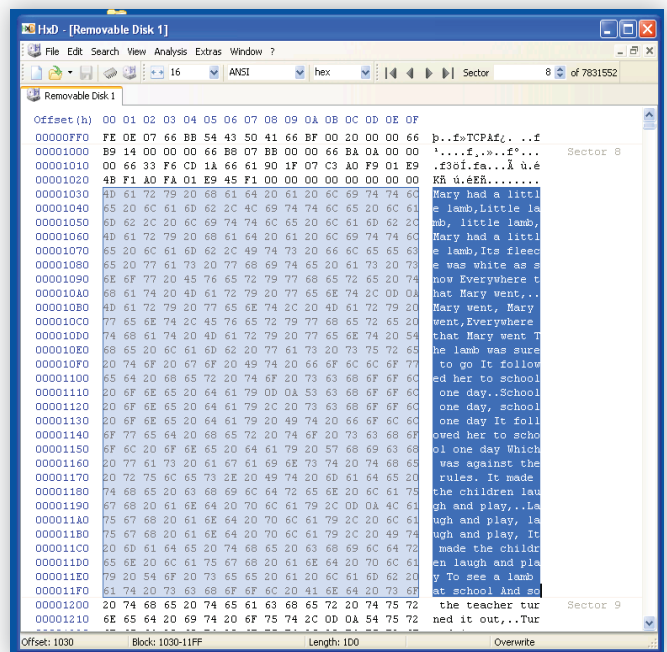


Figure 4. Displays hidden data in \$BOOT file

When data is saved in \$BOOT file storage device is mounted in Windows 7 Operating System and contents of \$BOOT file are checked by using FTKImager hex utility. The FTKImager indicated that data included in \$BOOT file is still exist and storage device mounted without any complications. The hidden data is displayed in FTKImager output in Figure 5.

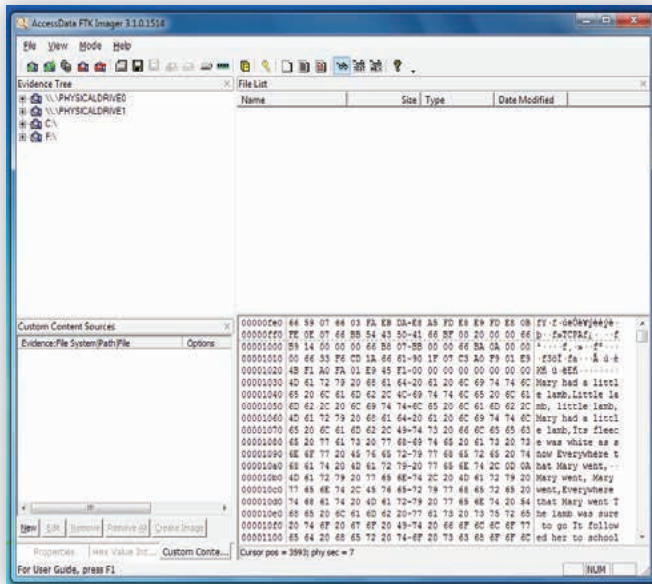


Figure 5. FTKImager displays hidden data in \$BOOT file.

Straight after determining the contents of the \$BOOT file by FTKImager CHKDSK utility run again for checking the integrity of the storage device and file system. The Figure 6 displays the results found from the process.

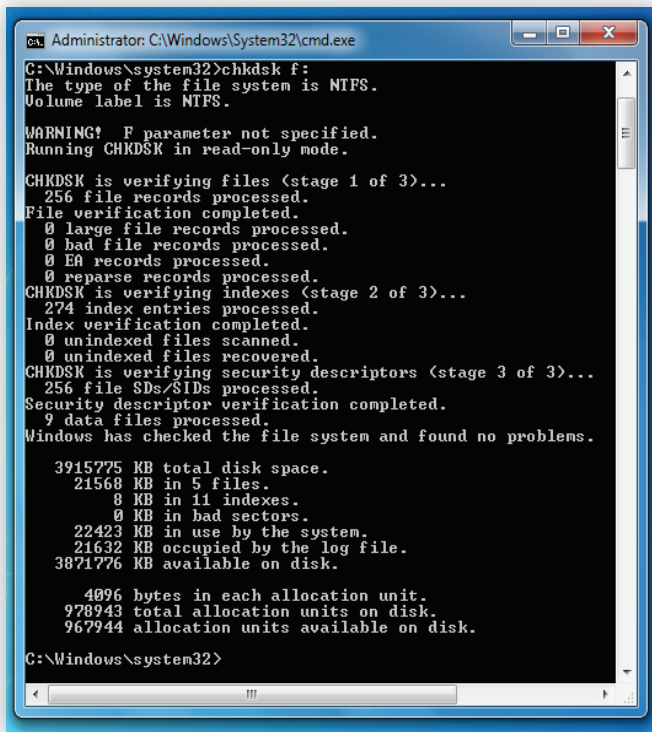


Figure 6. CHKDSK results for checking integrity of storage device and file system

The findings indicates that it is possible to hide data between 8th and 16th sector of the \$BOOT file where values are 0 and this procedure is not affecting functionality of the file system. However this procedure can cause file system failure in some of the other file systems such as ExFAT file system. The main reason for that is main and backup boot area in ExFAT is protected by checksum values which any changes in this data structure will fail the file system immediately. However findings indicates that this is not the case in NTFS file system.

Another interesting area is where the boot code is located in boot sector. The offset between bytes 84 and 509 in NTFS boot sector contains the boot code. However when file system is non bootable between these bytes BOOT MGR is missing error message is embedded and this area allows data hiding without causing file system failure. However size of the hidden data is only limited to 426 bytes.

These areas in \$BOOT file are well known areas by forensic examiners. The examiners will conduct a thorough examination on \$BOOT file for abnormalities by plain sight examination, comparing backup and original boot sector by calculating hash values, running data carving tools(Encase, FTK, Foremost, Scalpel) and keyword searches. The NTFS file system final sector contains a copy of boot sector for backup purposes and this sector is a potential data hiding environment for itself. Therefore forensic examiners should examine this sector for anti forensic implementations by comparing with the original boot sector in file system.

HIDING DATA IN NTFS CONTENT CATEGORY

The content category in NTFS file system contains actual contents of files and directories. These contents are stored in data units known as clusters which comes from a consecutive series of sectors. The first addressable cluster in NTFS file system starts from cluster 0 which it is first sector of the file system and where boot sector is located.

\$BADCLUS

The \$BADCLUS meta data file entry occupies 8th entry of the Master File Table. The \$BADCLUS file is responsible for keeping track of damaged clusters by assigning them its \$DATA attribute known as \$BAD. \$BAD data structure is a sparse file that it can grow entire size of the file system. Between operations when Operating System finds bad clusters it adds them to \$BAD attribute. However in today's technology most of the hard disk drives capability of finding these faulty data units(sectors) before the Operating System[Carrier, 2005].

The Figure 7. from SleuthKit/Autopsy forensic tool output shows \$DATA attribute of the \$BADCLUS file has one resident \$DATA attribute and one non resident \$DATA attribute and this attribute known as \$BAD. The non resident attributes indicates its actual size however it doesn't indicate any allocated bad cluster and their location. Therefore there is no faulty sectors in this storage media.

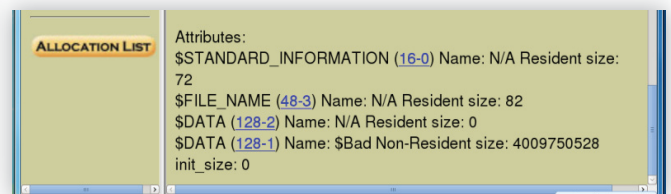


Figure 7. TSK/Autopsy output for displaying bad sectors in storage media

\$BAD data structure has capability of growing entire size of the file system. In this case one can create fake bad sectors/clusters and be able hide data in these data units. There are many available forensic tools has capability of extracting bad sectors/clusters from file system. Therefore if any data hiding implementations in bad sectors/clusters can easily be determined by forensic examiners. On the other hand data carving tools can extract hidden data through their file header information also known as magic numbers. The TSK istat command will display bad sectors in file system through command line

by executing following command which output is very similar to TSK/Autopsy output in Figure 7.

- `istat ntfs.001 -f ntfs 8`

ADDITIONAL CLUSTERS

Another technique of data hiding in NTFS file system is allocating additional clusters to existing file. By implementing this technique many additional clusters can be added manually to existing file and additional clusters can be located anywhere in the volume. This can provide great environment for data hiding.

In order to store content of files and directories NTFS file system uses attributes and as I mentioned previously attributes are divided into two different data structures. These are known as resident and non-resident attributes. The resident attributes can only store data up to 1024 bytes. If the file size bigger than 1024 bytes non-resident attribute will be dedicated to a file which this file can be GBs in size. The location of the non resident attribute of file is indicated in its header data structure.

The non-resident attribute contents are stored in series of clusters known as cluster runs. The same file can be stored in different non-resident attributes in different locations. This way different cluster runs can be allocated to same file. For example we have three clusters allocated to a file which is 10, 11 and 12. The cluster run starts from 10 and it has a length of 3 clusters. In order to keep track of these runs NTFS uses Virtual Cluster Number (VCN) and Logical Cluster Number (LCN). For example the file stored in clusters 10, 11 and 12 has 3 cluster runs. Therefore 0 to 2 is VCN and 10, 11, 12 is the LCN and this run list information is located at the attribute header [Carrier, 2005].

To create additional clusters to the relevant file for data hiding purposes the first cluster run list information must be examined from attribute header. When cluster run determined than attribute header cluster run LCN value will be modified to appropriate value which additional clusters to size of data to be hidden. When these values are modified we also have to modified VCN to appropriate values. For example if the original file size 3 clusters and starts from cluster 10 LCN will be 10, 11, 12 and VCN will be 0 to 2. If we add 2 more additional clusters we changing LCN to 10, 11, 12, 13, 14 and accordingly we have to change VCN value to 0 to 4.

After modifying the attribute header the size of the file must be modified from \$FILE_NAME attribute and allocation status of clusters has to be changed from unallocated clusters to allocated clusters in \$BITMAP file. The any of these procedures are skipped or not implemented appropriately file system will fail.

The \$BITMAP is a important data structure during creating additional clusters to relevant file. The \$BITMAP file entry occupies 6th entry of the Master File Table and it is responsible for keeping track of allocated and unallocated clusters in file system. The \$BITMAP achieve this by dedicating a bit for each cluster in its \$DATA attribute. If bit is 0 cluster is unallocated and if it is 1 cluster is allocated.

The Figure 8 indicates some of the allocated and unallocated clusters in file system. First 8 bytes in bitmap table indicates 0xFFFFFFFF0FF0FFFF values. When these values converted into bits we can see that clusters number 32, 33, 34, 35, 44, 45, 46, and 47 are unallocated and rest of the values are allocated.

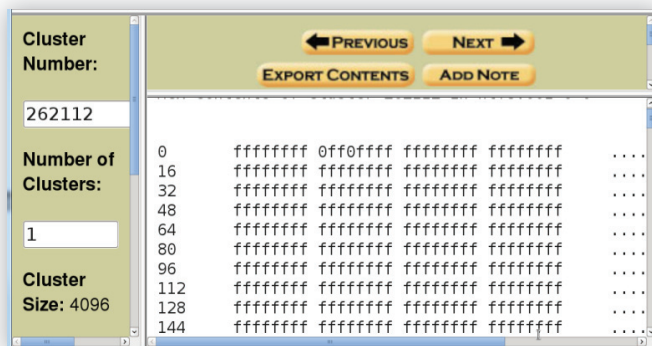


Figure 8. Allocated and unallocated clusters in \$BITMAP file.

When unallocated additional cluster positions are determined these bits can be changed from 0 to 1. Therefore system recognise these clusters are allocated and data can be hidden in these clusters.

However hiding data in additional clusters has its own disadvantages. If the original file grow in size it will be overwritten on hidden file unless the original file has permanent fixed size.

The forensic examination of hidden files in additional cluster can be time consuming process and it requires plain sight analysis, keyword search analysis, and data carving. The plain sight analysis can be accomplished by examining attribute header information, and comparing original size of the file against allocated clusters in file system.

FILE SLACK

The file slacks are created when file is stored into data unit which actual size of the file is smaller than the allocated data unit size. For example each sector 512 bytes in size and four consecutive series of sectors forms a cluster. The file going to be stored is 1750 bytes in size. In this case automatically one cluster will be allocated for relevant file which is 2048 bytes in size. Therefore 298 bytes of space becomes available since no data can be allocated in this space and it becomes ideal space for data hiding. In file system more than one file in fact most of the files can have even small amount of slack space. Some of these areas maybe padded by Operating Systems. This totally depends on functionality of the Operating System. For example in NTFS file system [Carrier, 2005] if one cluster comes from 8 sectors and first full four sectors and half of fifth sector used in order to store data remaining half sector in fifth sector will be padded with data by Operating System. The remaining three sectors will either wiped by 0's or not touched. By considering this one can hide data in more than one slack space in the volume even can implement certain algorithms to make it difficult to trace. However there are certain points to be considered. The stability of hidden data in file slack is totally depends on original file. If original file is removed the hidden data has a chance of overwritten by future file.

Cluster 9

Table 4. File Slack

Sector 47	Sector 48	Sector 49	Sector 50	Sector 51	Sector 52	Sector 53	Sector 54
-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------

Sectors with data

File Slack

The forensic examiner can use variety of forensic tools such as EnCase, and AccessData FTK in order to extract contents of file

slack exist in file system. On the other keyword searches, data carving and plain sight analysis can also be implemented. The extraction of slack space can be done by using combination of istat, icat, and dd TSK and Linux commands.

NTFS META DATA CATEGORY

In NTFS file system meta data category information is stored into Master File Table entries and their attributes. Each default entry and attribute contains descriptive information about the files and directories. As I previously mentioned this information includes file and directory locations, permissions and MAC(Modified, Accessed, Created) timestamps[Carrier, 2005]. In this category Alternative Data Streams are one of the common areas data hiding techniques can be implemented in NTFS file system.

ALTERNATE DATA STREAMS

The Alternate Data Streams also known as ADS are great data hiding areas in NTFS file system. When data is hidden in these data structures such as malicious programs most of the anti-virus applications may struggle to find these hidden data. So what is Alternate Data Stream?

In NTFS file system when data length over certain size it becomes non resident and data will be stored in an external cluster. In this case file will have two \$DATA attributes which one of them resident and the other one is non resident. This non resident \$DATA attribute is classed as Alternate Data Stream. This data structures give users capability of adding additional files to original existing file by using command line utility in Windows Operating System. When the new additional files are injected to original file the functionality of file system is not affected, additional files can not be seen in Windows Explorer and directory listing and size of the hidden data is unlimited. Lets have a look how Alternate Data Stream can be created and data will be hidden these \$DATA attribute data structures. In order to achieve this first we create text documents with some contents as it shown in Figure 9.

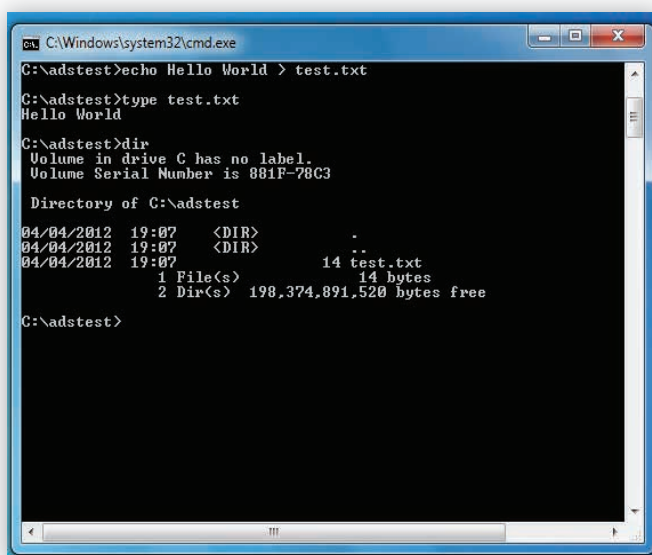


Figure 9. Creating text document with contents for hiding data in ADS

As soon as text file is created we are going to add additional data to original file and hide this file by following commands in Figure 10.

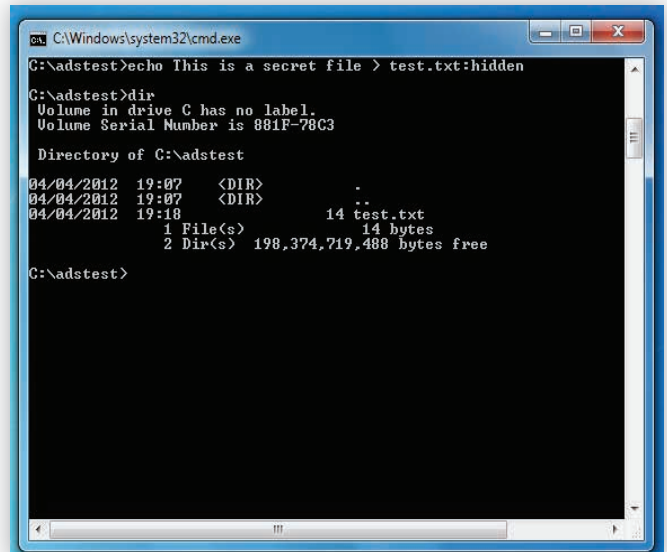


Figure 10. Additional data is added to original file

The file name "hidden" added to file call "test.txt" by commands in Figure 10. As you can see when the directory is listed by using "dir" command from command line we only be able to see the first original file we created which is "test.txt". Now lets have a look contents of test.txt file to see if any changes are made to its original contents during the process.

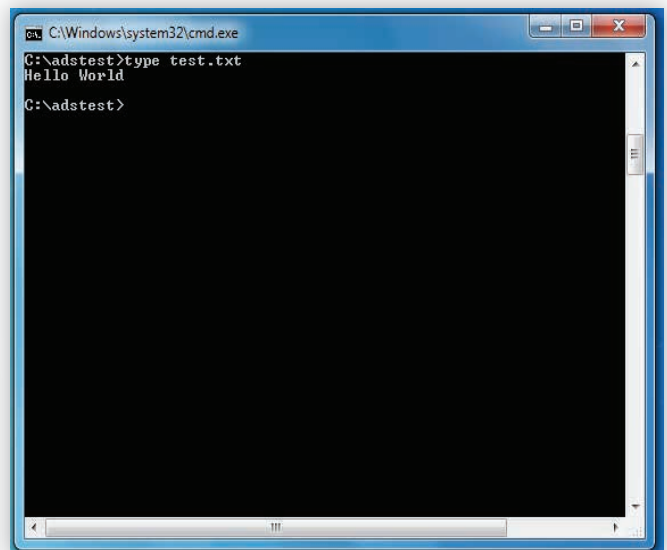


Figure 11. Checking original contents of test.txt file

The Figure 11 indicates that there are no changes made to contents of test.txt file and it remained the same when we added the contents of hidden.txt file. Now lets have a look to data we hidden into test.txt file \$DATA attributes.

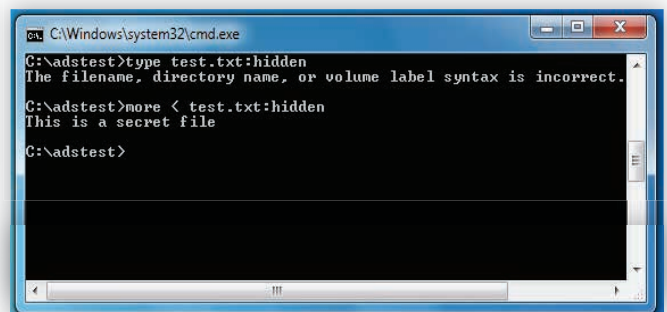


Figure 12. Contents of hidden.txt

The contents of hidden.txt added to test.txt file without any complication and it is very easy procedure to follow. As you noticed from Figure 12 we used “more” command on read mode to access contents of ADS. The main reason for that is “type” command is not supported by ADS.

Another interesting area is directories in NTFS file system which they can also have a \$DATA attribute. This means that relevant directory can store contents of the file, file list and the sub directories. Since directories can contain \$DATA attribute the additional \$DATA attributes can also be assigned to a relevant directory which these attributes are also known as Alternate Data Streams.

The forensic analysis of Alternate Data Streams can be done through examining each ADS on the file system. The main reason for that is ADS provides great ground such as simplicity which low sector level operations aren't required and size of the hidden data is unlimited. For this reason there is a high percentage of possibility hidden data can be found Alternate Data Streams. The forensic analysis of ADS can be done by using forensic tools such as EnCase, AccessData FTK, Streams, and Lads.

CONCLUSION

The file systems can have complex data structures as well as functionalities which NTFS file system is a great example for this. In order to hide data or examine file systems and storage devices these data structures and their functionalities must be understood appropriately. The main reason for that is each file system has their unique characteristics and data structures. For example some data structures are protected by checksum values in some file systems such as ExFAT file system and any parameter alterations may cause system failure.

The successfully hiding data requires finding new and unknown techniques, and taking advantage of new data structures. As soon as these techniques are known there is no point using that method because its not secret any more. Using complex algorithms and implementing these algorithms on stable data structures in systems also crucial for data hiding. If data structures where data is hidden change their size, or functionalities that may cause lost of data. For this reason person who hiding the data might consider backup of hidden data inside same system. In most circumstances encryption and changing file headers are also applied to hidden data in order to introduce certain difficulties for forensic examiners. Again all

these techniques directly related to functionality of file system, storage media and their data structures. When these factors are considered successful data hiding procedure becomes very complicated and again if techniques are known there is no point using that method because its not secret any more.

As I mentioned at beginning of this article these techniques we discussed well known techniques in today's computing. Without a doubt many data hiding techniques and tools are also introduced everyday because these techniques are only limited to imagination of a person. This brings great challenges for forensic examiners such as time consumption and cost where time is crucial in order to get conviction and cost is very important where limited budget resides. The forensic examiners can overcome these challenges by studying new relevant technologies and sharing this information within the forensic community.

References

- Berghel, H (2007) *Hiding data, forensics, and anti-forensics*. Commun. ACM 50, 4
- Carrier, B. (2005), *File System Forensic Analysis*, Pearson Education, Indiana, p. 173 - 215, 221-256, 261-305, 317- 369
- Knut Eckstein, M.J.(2005): *Data hiding in journaling file systems*. In: *Digital Forensic, Research Workshop*.
- NTFS.com. (13/04/2011) *NTFS - New Technology File System designed for Windows 7, Vista, XP, 2008, 2003, 2000, NT*. [WWW Document]. URL <http://www.ntfs.com>
- Wee Kai Cheong, *Analysis of hidden data in NTFS file system* [WWW Document] URL <http://www.forensicfocus.com/downloads/ntfs-hidden-data-analysis.pdf>

ABOUT US (WWW.ERFORENSICS.COM)

ER Forensics and Data Recovery is specialised on recovering lost and corrupted data from variety of storage devices such as hard disk drives, RAID, SD family, SSD, USB drives and many other storage devices as well as conducting digital forensic examinations. ER Forensics and Data Recovery provide these services without hidden costs, and forensically sound manner by considering relevant laws and legislations with specialised forensic examiners.

UĞUR EKEN (ueken@erforensics.com)

is founder of ER Forensics and Data recovery with his business partner Joseph Richards in Teesside United Kingdom.



NEEDLE IN A HAYSTACK

TONY RODRIGUES

It hasn't been long since file count in a hard disk was around a few thousand files. Nowadays, it's not uncommon to find more than a million files in a system. Finding a file in those HDs may be like finding a needle in a haystack. Even worse, if the file has been deleted. Let's see some techniques in order to make our lives, the forensics practitioners, a little easier.

Sometime ago, I was involved in a Digital Investigation in which I should check for the existence of a specific file in all company's hard drives. Even more, I should check whether it was currently in the hard drive or it had been deleted. The mentioned file could also have been altered, with some small parts changed. File wiping wasn't a concern, but the supposed file copy/access may have happened more than a year before the start of the investigation. In this article, you will learn a technique to deal with this sort of search and some ways to get the whole thing done faster.

Requirements

It's important to join all the conditions we have for this search. They will lead us to the most effective solution. Let's consider, initially, only one hard drive to check.

- We have to check the file existence in a live Windows system;
- The file may have never touched this hard disk;
- The file may have been copied to this hard drive recently or a long time ago;
- The file may have been to this hard drive, but was deleted;
- The file may have been deleted recently or a long time ago;
- The file may not still be exactly the same (small changes);
- File wiping is highly improbable

Maybe the first idea is to search for file name and check content. In our case, we can't guarantee that name will remain unchanged. Search by its keywords could also be inconclusive, as someone may have changed the file just in the place we were about to search it. Use the file hash for searching cannot help us, also, because file may be a little different and even may exist

no more. So, we have to take a really effective approach from the simplest search scenario (the file is unaltered and still exists in the hard drive) to the most challenging one (the file was altered and deleted a long time ago). In fact, the most challenging would be a wiped file, but, again, we are not considering this. Obviously, the best approach must address the most challenging scenario and, in order to achieve it, we have to review some NTFS concepts before.

NTFS and file Allocation

A file system has many functions, but the most important is, of course, to place the files in the disk in a way we can recover and access them. The media is organized and divided in small pieces that receive data. Two important concepts are sectors and clusters (or blocks). A sector, usually with 512 bytes, is the smallest physical storage unit on the disk. A cluster can consist of one or more consecutive sectors and is the smallest addressable unit of the disk. Each cluster has a number as a logical id, and file system is responsible to map each file to a set of clusters, contiguous or not.

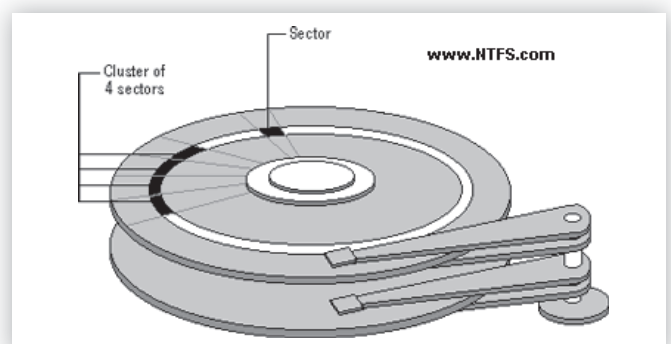


Figure 1. Sector and Cluster

We can find it!

Let's take some considerations to the most challenging situation. When a file is deleted, its clusters are free to be used, but they retain their content. So, until all clusters of a deleted file are completely overwritten, we can find at least a sector that has belonged to it. We may find that:

- The deleted file clusters were completely overwritten. For instance, its clusters were overwritten by contents of a bigger file, and this data isn't in the end of this bigger file.
- The deleted file clusters were overwritten, but not completely, with some clusters (contiguous or not) still remaining in the disk. For instance, the deleted file left 3 contiguous clusters and more 2 non-contiguous, but just the first 3 clusters were overwritten by a smaller new file.
- The deleted file clusters were completely overwritten, but the last cluster received less bytes (sectors) than the deleted file. For instance, the overwriting file was one or two sectors smaller and after the overwriting operation we still can find data from the deleted file.

The main point in letter a, b and c is that only the first one doesn't leave data from the file being searched. All the others will leave at least one sector, and this could be more than enough to say that the file being searched truly existed sometime in the hard disk.

The preparation

It's important to take the approach in two phases. First, we must calculate the file hash, but breaking it in small pieces with the same size of a sector. 512 bytes is the most common sector size, but we cannot guarantee we will find this size in all disks we will search for, so, we must consider calculating hashes dividing the file reference (the one searched) in pieces with size in a range from 512 bytes to 64 Kbytes. All hashes calculated will be used for searching the file. Let's see the first program, used for preparation, called Needle.pl.

```

1  #!/usr/bin/perl -w
2  # Calculates SHA-1 piecewise hashes. Sizes from 512b, 1k, 2k, 4k,
3  # 8k, 16k, 32k, 64k or specific
4  # usage: Needle.pl -a file [-t specific-size] [-h]
5  #
6  # -a :file
7  # -t :specific-size
8  # -h :help
9  #
10 #
11 #
12 use Getopt::Std;
13 use Time::Local;
14 use POSIX qw(log10);
15 use Digest::SHA1 qw(sha1 sha1_hex sha1_base64);
16
17 my $ver="0.1";
18
19 #options
20 %args = ( );
21 getopts("a:t:h", \%args);
22
23 #print help
24 if ($args{h}) {
25     &cabecalho;
26     print <<DETALHE ;
27     usage: Needle.pl -a file [-t specific-size] [-h]
28
29     -a :file
30     -t :specific-size
31     -h :help
32
33     Ex: Needle.pl -a secret.doc > contenthash.ini
34         Needle.pl -a sales.xls -t 65535 >> contenthash.ini
35
36     DETALHE
37     exit;
38 }

```

Listing 1. First block of lines of Needle.pl

In the Listing 1 we have the first block of code for Needle.pl. Not too much to say, just checking options. Note, also, that we will use SHA1 as hash algorithm. We use `-a` option to enter the filename we will search. The Needle.pl will write to standard output (`stdout`) all hashes calculated. We can also use `-t`

option and request hash calculation just for a specific size. Even though, the most usual usage will be just passing `-a` option, redirecting the output to a file. For example:

```
C:\>Needle.pl -a secretfile.doc > content_hash.needle
```

Listing 2 shows part of a typical Needle.pl output. This output can be matched to the output of SHA1DEEP or SHA1SUM, both well known utilities:

```
C:\>SHA1DEEP -p 512 secretfile.doc
```

```

[2048]
6b4f6104e15546821beb2fed667498fd90655a61
5940f62ab861598ac6b1d897bb30406570fe8bf2
2a84490eac2e09e94dac47a6d66c822875ce0569
35c7263a50305d1f8560fa1a2b98178052cc4a77
19ecbe2230dcb26f17ea2904940e79306992ef67
dde2849e977d014896773a387d7c773fffa05320
4f47b10b9c89122ac6eca6ece753a816f8248ecc
0bcc0237439c521fd6a96f9580e6abc439066296
4ce3add9f23f32ed4297b9e439bc888e2b013121
aebc46fe4adclb7662ea076d321ee2b4ffa87aee
b861c031e20f4da81f34cbcc73bd424066c340e1
605db3fdbaff4ba13729371ad0c4fba3889378e
420b509e63fb23326cc14857eee02b88e6a95944
605db3fdbaff4ba13729371ad0c4fba3889378e
4768ac85357276f36bb336fb497962559d4f0b5e
ec6af5cf65f23f58497ab98f2b2a40fa666fd4a2

[4096]
8dcc1e6b75f5358f8e770ed544ffaa2c6c41a26b
3c157d22680e71bfe9897b44216195ea5c3bab31
43d8b95e340624e6faeedba2b886b82200b6f56
c41fd40a452063b25925a563d205fc3e39eba3e4
e81040b680c3c5e4f538d367e31aa70dc88dfbd5
a86fec1e17074d1f214852e16f4fdffed06b94eb
1c295935f7432d612b8f0549598e781f9b7d2ceb
386eb4f59673522d825c8441d2260b4fa9c7e75e

[8192]
7ea60e111225b35c7ea0e9b6788590eba399cd74
de6263ce911bdcbd1fa17711de7110a575049360

```

Listing 2. Excerpt from Needle.pl output (a needle-file)

```

49 # From 512 to 64k
50 my $ini = 0;
51 my $fim = 7;
52
53 # check -t option
54 if ($args{t}) {
55     $ini = log_base(2, $args{t}/512);
56     $fim=$ini;
57 }
58
59 #open file as binary
60 open($loFile, "<", $args{a});
61 binmode $loFile;
62
63 #loop for different piece sizes
64 for (my $i=$ini; $i<=$fim; $i++) {
65
66     #size of piece
67     my $piecewise=(2**$i)*512;
68
69     #byte 0
70     seek $loFile, 0, 0;
71
72     my $sBlocoInteiro = "";
73
74     print "\n[$piecewise]\n";
75     while (read($loFile, $sBlocoInteiro, $piecewise) != 0) {
76
77         #calc hash after read data from file
78         my $digest = sha1_hex($sBlocoInteiro);
79         print "$digest\n";
80     }
81 }
82
83 #print the last section: hash for file
84 print "\n[arquivo]\n";
85
86 $old_rs = $/;
87 undef $/;
88 seek $loFile, 0, 0;
89 $sBlocoInteiro = <$loFile>;
90 my $digest = sha1_hex($sBlocoInteiro);
91
92 print "$digest\n";
93 $/ = $old_rs;
94
95 close($loFile);
96

```

Listing 3. Most important block of code from Needle.pl

In Listing 3 we have the processing code of Needle.pl. In short, it is a loop in the range of different sector sizes. Data is read and SHA1 hash is calculated from pieces of the file. Finally, from line 83 to 93, it is calculating SHA1 for the file. It's important to notice that if the file is smaller than the sector size being calculated, the resulting hash for this hash and next ones will be the same as the file hash. From now on, I will refer to the *Needle.pl* output file as *needle-file*.

The Search

For the next step we have more concerning in our search:

- We have to do the search in many workstations;
- This search operation is heavy in terms of resource-consumption;
- The file may have existed compressed in the hard disk (or had other kind of foreseen but heavy transformation);

From number 1) and 2) we will prepare the Search program with logging capabilities. We cannot wait in a console for the output, hã?!? Also, we must do the search off time, combining the activation of the search after a remote wake-up of workstations (Wake-up-On-LAN would work nice here).

Number 3) requires a previous analysis of what may happen to the file. We're not looking for small changes or data update, but considerable changes to file data, like file compressing or picture resizing. We have to apply each of these *file variations* to Needle.pl, saving the output for these needle-files. Of course, our search program will need to deal with more than one needle-file. Let's see the first block of SearchHaystack.pl, the searching program:

```

18 use Getopt::Std;
19 use Time::Local;
20 use Sys::Hostname;
21 use Socket;
22 use Time::HiRes qw(gettimeofday);
23 use XML::LibXML;
24 use Win32::API;
25 use Win32API::File 0.08 qw(:ALL);
26 use Digest::SHA1 qw(sha1 sha1_hex sha1_base64);
27 use Win32::OLE('in');
28 use constant whenFlagReturnImmediately => 0x10;
29 use constant whenFlagForwardOnly => 0x20;
30
31 # Paths for SleuthKit - CONFIGURE HERE BEFORE USE
32 my $PATH_TSK = "\\\\";
33
34 my $MY_COMPUTER = "localhost";
35 my $SER = "0.1";
36
37 #options
38 %args = ();
39 getopts("hd:f:ps:s:", \%args);
40
41 #show help
42 if ($args{h}) {
43     &showcall;
44     print <<DETAIL;
45     usage: SearchHaystack.pl -f needle_file1,needle_file2,needle_file_n -p path-result [-s sector-size] [-d volume] [-h]
46
47     -d :volume to search
48     -h :help
49     -f :comma-separated Needle filenames
50     -s :sector size
51     -p :output path
52
53     Ex: SearchHaystack.pl -f x.ini,arq.ini,x2.ini -d c -p d:\
54     SearchHaystack.pl -f arq.ini -p i:\Dir_resultado\ -s 512
55
56     The output file sha1_{IP}_{HostName}_{drive}.txt will be written to the output path
57
58     DETAIL:
59     exit;
60 }

```

Listing 4. First block of code for SearchHaystack.pl

Some highlights for these initial section of code:

- The program uses WMI to get Windows logical disks;
- The program depends on The Sleuth Kit (TSK) in some operations. It's path must be configured before usage in the line 32;
- Needle-files are passed to the program by *-f* option. The list is comma-separated;
- The option *-d* can be used to search in only one disk. All logical disks will be searched by default;
- Output file will be written to the path in the *-p* option;
- The sector size of the disk been searched will be calculated by default. Although, we can pass it through *-s* option.

A good usage example would be:

- `SearchHaystack.pl -f file.needle,file_zip.needle,file_arj.needle -p \\MyServer\SearchResults\`

```

88 @mAllDrives = ();
89
90 if ($args{d}) {
91     #just one volume
92     push(@mAllDrives, $args{d});
93 }
94
95 else {
96     #get all volumes
97     my $objWMIService = Win32::OLE->GetObject("winmgmts:\\\\$MY_COMPUTER\root\cimv2" or
98         die "WMI connection failed.\n");
99
100     my $colItems = $objWMIService->ExecQuery("SELECT * FROM Win32_LogicalDisk", "WQL",
101         whenFlagReturnImmediately | whenFlagForwardOnly);
102
103     foreach my $objItem (in $colItems) {
104         push(@mAllDrives, substr($objItem->{Caption},0,1));
105     }
106 }
107
108 #get computer name
109 my $sHostname = hostname();
110
111 #get computer IP address
112 my $sEndIP = gethostbyname($sHostname);
113 my $sRealIP = inet_ntoa($sEndIP);
114
115 $mParser = new XML::LibXML;
116
117

```

Listing 5. Warm up code

The code in Listing 5 is kind of a *warm up* code, preparing structures for the search. It collects the logical disks in the running workstation via WMI, or just use the option passed in the *-d*. Also collects the host name and IP Address. This information will compose the name of output file. Have you noticed the XML initialization code in line 116? Yeah, this is for the output file.

```

120 $mThisDrive = $_;
121
122 #start time
123 my $sTimeIni = gettimeofday();
124
125 #output filename
126 my $sNoneArg = $args{p} . "sha1_" . $sRealIP . "_" . $sHostname . "_" . $mThisDrive . ".txt";
127
128 if (c= $sNoneArg) {
129     #output file already exists. Load it.
130     $mXMLDoc = $mParser->parse_file($sNoneArg);
131     ($mXMLPai) = $mXMLDoc->findnodes("WasiHere");
132     ($mXMLHag) = $mXMLDoc->findnodes("WasiHere/Haguina");
133 }
134 else {
135     #start time running in this volume
136     $mXMLDoc = $mParser -> parse_string("<WasiHere><\WasiHere>");
137     ($mXMLPai) = $mXMLDoc->findnodes("WasiHere");
138     $mXMLHag = $mXMLDoc->createElement("Haguina");
139     $mXMLHag->setAttribute("IP", $sRealIP);
140     $mXMLHag->setAttribute("Host", $sHostname);
141     $mXMLHag->setAttribute("Disc", $mThisDrive);
142     $mXMLPai->appendChild($mXMLHag);
143 }
144
145 #get sector size
146 if (defined($args{s})) {
147     #informed via -s option
148     $mSizeSector = $args{s};
149 }
150
151

```

Listing 6. Runs for each drive

The main loop of *SearchHaystack.pl* is through all drives of the workstation. Actually, just one drive if *-d* option was used. Highlights:

- The output filename has workstation IP address, hostname and drive letter on the name;
- The program is prepared for running more than once in the same workstation. It loads previous results accordingly;
- From line 151 to 159 we get sector size for the current disk. Listing 7 shows function *getSectorSize*.

```

317 #get volume sector size
318 sub getSectorSize {
319     my $pPartition = shift;
320
321     #get info from the file system (uses TSK)
322     my $comstr = $PATH_TSK . "fsstat.exe -t -i raw \"$pPartition\"";
323     my $resp = `"$comstr"`;
324
325     my $expr = "";
326
327     if ($resp =~ /ntfs/) {
328         $expr = "Total Sector Range";
329     }
330     elsif ($resp =~ /fat/) {
331         $expr = "Total Range in Image";
332     }
333     else {
334         return 0;
335     }
336
337     #get sector size
338     $comstr = $PATH_TSK . "fsstat.exe -i raw \"$pPartition\"";
339     $resp = `"$comstr"`;

```

Listing 7. getSectorSize function

This function calls `fsstat.exe`, a utility from The Sleuth Kit, in order to get disk sector size. Notice lines 327-335, where it checks for file system. Only NTFS and FAT are accepted.

```

170 #Load Needle files only if this volume has a different sector size than the previous
171 if ($mLastSize != $mSizeSector) {
172     #Locate section and load the hashes
173     $mHashPieces = ();
174     $mHashPieces = loadHashPieces($args(f), $mSizeSector);
175     $mLastSize = $mSizeSector;
176 }
177
178 #Write sector size to the output XML
179 $mXML.Hq->setAttribute('TamanoSector', $mSizeSector);
180
181 #Execute Search
182 #FindPiecesByHash("\\\\\\\\\\\\$mThisDrive", $mSizeSector, $mHashPieces, $mXML.Doc, $mXML.Pai, $mNomeHq);
183
184 #Finish time
185 my $mTimeFin = gettimeofday();
186
187 #Log this execution
188 my $mXML.No = $mXML.Doc->createElement("Exec");
189 $mXML.No->setAttribute('Inicio', $mTimeIni);
190 $mXML.No->setAttribute('Fim', $mTimeFin);
191 $mXML.No->setAttribute('Tempo', &com($mTimeFin-$mTimeIni));
192 $mXML.Pai->appendChild($mXML.No);
193
194 #Save output file
195 $mXML.Doc->ToFile($mNomeHq);
196

```

Listing 8. Last code in the drives loop

After the code in Listing 6, the program loads the hashes related to the sector size of the current disk. This is done only if the sector size has changed from the previous disk. Check Listing 9 for the `loadHashPieces` function code.

The search is finally executed by the function `findPiecesByHash` (see Listing 10).

Finishing the program, it saves the time elapsed for this search and writes the XML file to disk. The loop continues to the next disk.

```

222 #Load Needle File
223 sub loadHashPieces {
224     my $pIniCollection = shift;
225     my $pSectorSize = shift;
226     my $mHashPieces = ();
227     my $mConteudoArquivo;
228
229     #Get all Needle files (they are comma-separated in the option)
230     my @mArquivosIni = split(/,/, $pIniCollection);
231
232     #Load each file
233     foreach (@mArquivosIni) {
234         my $mThisIni = $_;
235
236         #Get the whole file to a string
237         my $mFareq = $_;
238         $mFareq =~ s/\n//g;
239         open(RRQUIVO, "< $mThisIni");
240         $mConteudoArquivo = <RRQUIVO>;
241         close(RRQUIVO);
242         $mFareq = $mFareq;
243
244         #Get the hashes related to the sector size
245         $mConteudoArquivo =~ s/(\{ $pSectorSize \} \n (\{ a-f \} { 40 } \n) *) /g;
246         $mConteudoArquivo = $mFareq;
247
248         my @mLinhas = split(/\n/, $mConteudoArquivo);
249
250         #From array to perl-hash
251         for (my $i=1; $i < @mLinhas; $i++) {
252             $mHashPieces[$i] = "Arquivo $mThisIni, parte $i";
253         }
254     }
255     return $mHashPieces;
256 }
257

```

Listing 9. `loadHashPieces` function code

The `loadHashPieces` function takes all needle-files and reads each of them. Regular Expressions are used to filter out unnecessary hashes. The returned structure (a Perl Hash) indexes each piece hash to a description *File xxx, part nth*.

```

260 #Search pieces
261 sub findPiecesByHash {
262     my $pPartition = shift;
263     my $pTamSec = shift;
264     my $pPieces = shift;
265     my $pDoc = shift;
266     my $pPai = shift;
267     my $pErr = shift;
268
269     $pErr .= ".err";
270
271     #Open Volume
272     my $loFile = Win32API::File->new("< $pPartition");
273     binmode $loFile;
274
275     #byte 0
276     seek $loFile, 0, 0;
277
278     my $mBlocoInteiro;
279     my $mK=0;
280
281     while (read($loFile, $mBlocoInteiro, ($pTamSec*$mBLOCOLS)) != 0) {
282         #read data and after, calc hash piece
283
284         for (my $i=0; $i < $mBLOCOLS; $i++) {
285             my $mTamParcial = ($i*$pTamSec);
286
287             #check if has finished
288             last if ($mTamParcial > length($mBlocoInteiro));
289
290             my $mBloco = substr($mBlocoInteiro, $mTamParcial, $pTamSec);
291
292             #calc hash
293             my $mdigest = sha1_hex($mBloco);
294
295             #check existence
296

```

Listing 10. `findPiecesByHash` function - the heart of the search

In many aspects, the `findPiecesByHash` function, in Listing 10, is similar to the main loop of `Needle.pl`. This function is responsible for open the volume as raw data and read it, breaking the data pieces the same size of the disk sector size. If the hash of the piece is found in the Perl-hash structure (loaded in `loadHashPieces` function), a XML element is created to mark this finding, which piece of file was found and in which disk sector.

The Search Result

The result file is a XML. Listing 11 shows an example. The element *Maquina* identifies the workstation, its IP address, name, disk letter and its sector size. There's one element *Achei* for each piece of the file found in the disk. It also has which piece this one refers, and where is it in the disk (sector number). In the end of the file, a *Exec* element has how long it took for the search. More than one *Exec* element means the program was run more than once.

```

1 <?xml version="1.0"?>
2 <Maquina>
3   <Maquina IP="10.1.1.2" Nome="forcomp" Disc="I" TamanoSector="512" />
4   <Achei Hash="983da59f8fc342828157c5b943f0badfedc5d" ParteArquivo="1" Bloco="20401"/>
5   <Achei Hash="810533803d1f7e9e8fc7a3cfd4731878e8a67" ParteArquivo="2" Bloco="20402"/>
6   <Achei Hash="ce3a9d2e23050ca4a3dad3e2ec0b405678dd99b7" ParteArquivo="3" Bloco="20403"/>
7   <Achei Hash="27f3584bcfd292b177f0ecaf3a356ef91e0998" ParteArquivo="4" Bloco="20404"/>
8   <Achei Hash="46ce8f31bf5a5c1ea2f4da96880b7a208919428" ParteArquivo="5" Bloco="20405"/>
9   <Achei Hash="9ad49421f8991a8c7c06cdf5932a1f89ed1282hd" ParteArquivo="6" Bloco="20406"/>
10  <Achei Hash="40cf8664c33f594b51ae67f43a08285429fd" ParteArquivo="7" Bloco="20407"/>
11  <Achei Hash="66d92e5d0007b3f56d0a2ef8bf4e1ad53a80" ParteArquivo="8" Bloco="20408"/>
12  <Achei Hash="e27f26517210b27ac3c4086ede0ba6472792b6f" ParteArquivo="9" Bloco="20409"/>
13  <Achei Hash="1df9558c043d9a8c15516ae4958c8a3ba8794352" ParteArquivo="10" Bloco="20410"/>
14  <Achei Hash="c132d17bc336271d187c8cd4a85f6630d2acc8" ParteArquivo="11" Bloco="20411"/>
15  <Achei Hash="cb584593d4fe91d9461fff2658fa79da522e41d3" ParteArquivo="12" Bloco="20412"/>
16  <Achei Hash="6106ad8bb9b260c6d2130cb316270107bc39d071" ParteArquivo="13" Bloco="20413"/>
17  <Achei Hash="ac177163537ac356405b40d14283ceab01931f" ParteArquivo="14" Bloco="20414"/>
18  <Achei Hash="8b110e237252748f18aed57779f5265a5c54870b" ParteArquivo="15" Bloco="20415"/>
19  <Achei Hash="7af248ac0d97585954a7224fb339a97065ea" ParteArquivo="16" Bloco="20416"/>
20  <Achei Hash="4020ae3c0d206993b5a80990eac3880bee1c7a0d8" ParteArquivo="17" Bloco="20417"/>
21  <Achei Hash="717f27b5d90798018ddc5a5ef364057ecf362" ParteArquivo="18" Bloco="20418"/>
22  <Achei Hash="7871fd6cae3893b0bfc50f4d71302781d108def" ParteArquivo="19" Bloco="20419"/>
23  <Achei Hash="029b153b368c5c0c88378998c0b0a54629074" ParteArquivo="20" Bloco="20420"/>
24  <Achei Hash="189514b183c813614bfe583a0b14020538b" ParteArquivo="21" Bloco="20421"/>
25  <Exec Inicio="1303791697.3125" Fim="1303791910.84375" Tempo="0m3n33s"/>
26 </Maquina>
27

```

Listing 11. `SearchHaystack.pl` output file

Limitations

Unfortunately, this technique is not 100% efficient. It cannot achieve its objective in some circumstances:

- **Wiping**
If the file has touched the drive but the user wiped it after, we won't have any directly reminiscent data; so, no data, no findings.
- **Cryptography**
If a ciphered file was saved in a disk, we cannot find it using its clear text version as input for `Needle.pl`. We will have the same difficulties with password compressed files. Obviously, it's ok if we have the ciphered/compressed version, or at least we know the software and password used to cipher/compress.
- **Disk Usage**
If the user continuously write and delete files to the disk, it will be less probable to find reminiscent data. Eventually, all sectors used by the searched file will be completely overwritten. Frequent disk defragmentation may also remove reminiscent file data.
- **The Last Sector problem**
It's hard to find files that are multiple of hard disk sector size. So, most of time, the last hash calculated for each sector size in the needle-file won't match anything. This happens because this hash is calculated considering the last bytes of the file while the hash of the sector will always consider a fixed size (the sector size). Similarly, file smaller than a sector will never have the same hash (they never match). This doesn't affect significantly our search be-

cause most of hard drives use 512 bytes for sector and documents/pictures are usually bigger than that.

False Positives

False positives may occur when the file we are searching for has a long sequence of 00 or FF bytes (probably happens to others). That's because those byte sequences are common in hard disks. So, it's recommended to refine the results, filtering out these sequences.

- **Collisions**
It's highly improbable, but hash collision may occur. Single pieces found must be refined and checked.
- **Improvements**
You will notice that error feedback isn't implemented, yet. It's important to write something to the output XML, avoiding wrong conclusions. I'll leave this for the 0.2 version;
- Hash for common byte sequences (00, FF, etc) may be filtered out or receive a special mark in the output.
- The Sleuth Kit dependency can be removed. As the search is focused on NTFS and FAT, it's not hard to get the sector size directly. Brian Carrier's book *File System Forensics Analysis* is great reference for this (he is also the TSK developer). Of course, you can also dive into TSK code.
- I've found some articles claiming MD5 is faster than SHA1. I haven't tested, but if we could save some time in each operation, the whole search could be one hour faster. Definitely, it deserves a test.



Prioritizing Suspects

Hash an entire disk, sector by sector, isn't as fast as we would like. This can last for hours in a single disk, even more with large HDs commonly found today. Can we select which workstations should be searched first? Yes.

There are some details that we can check in advance and score the workstation. Higher scores must be searched first. This checking is fast and doesn't require to be done off time. Some suggestions:

- Browse through directory tree of each drive, parsing \$I30 NTFS Index. If you find:
 - Same searched file name: add one point to the score;
 - Same name and Creation time greater than Creation-time of searched file: add two points;
 Just remembering, MAC times in \$I30 are copied from Filename attributes and it's possible to get file names and MAC times from files that have been deleted. Although all of this just work for NTFS, it's possible to do similar procedures in FAT (would work just for non-deleted files).
- If you're searching a graphic file, Thumbnail file retains its name and Last Modified timestamp even after file is deleted (or wiped). Parse then and add one point to the score if you find the same searched file name. You may also check visually the thumbnail picture.
- If the searched file is a document and workstations have Windows Search Indexer enabled, we can access Windows.EDB files and search inside for uncommon strings the searched document has. It may work even for wiped files!

Conclusion

Search for a specific file in many workstations can make a Digital Investigator have a hard time, but it's not an impossible mission. We can come up with the goods breaking the file in small – hard disk sector size - pieces and calculating their hashes. By studying the search scenario in advance, we can save much time prioritizing workstations that came up with some ease-to-extract artifacts related to the file being searched.

TONY RODRIGUES

is a Brazilian Computer Forensics Researcher with more than 20 years of IT experience, and 8 years in Information Security Management positions. He holds CISSP, CFCP, Security+, ACFCP and MCSD professional certifications and has led numerous digital investigations, incident responses and DFIR researches. Tony is Chief Forensics Researcher and Founder of OctaneLabs, a Brazilian Open Source Computer Forensics and Incident Response Research Team. Tony is also member of Comissão de Direito Eletrônico e Crimes de Alta Tecnologia da OAB-SP (High Tech Crime Investigation Commission of OAB-SP) and has presented in many international conferences held in Brazil (YSTS, H2HC, OWASP, CNASI, ValeSecConf, WebSecForum). Tony developed the first Rio de Janeiro's Computer Forensics training course and has trained DFIR practitioners for large companies and govern agencies, including Federal Attorneys and Police. Tony writes technical articles for his blog forcomp.blogspot.com and for SANS Computer Forensics Blog, http://3.bp.blogspot.com/-qYfGiDaHNzg/TcmbACnb2a1/AAAAAAAAABRo/8bNwPCIsIHs/s1600/computer_programming.jpg

Do You Want to Become a Cyber Security Expert? OR ADVANCE YOUR IT SECURITY CAREER?

- 🕒 Cyber Security has one of the largest market shares in IT
- 🕒 Government & Compliance Regulations are more and more enforced
- 🕒 Gartner Group predicts unprecedented growth and need in Cyber Security
- 🕒 Skilled Cyber Security Experts are in ever more demand

THE CYBER 51 EXPERT COACHING FORUM

- 🕒 Individual 1-on-1 Mentoring on Ethical Hacking, Penetration Testing and IT Security
- 🕒 Networking with other community members and moderators
- 🕒 Access to a wealth of tools and information not found on public domain
- 🕒 Permanent Job & Contract offers, Webinars and much more!

YOUR BENEFITS

- 🕒 Become an Ethical Hacker / Penetration Tester with 1-on-1 mentoring
- 🕒 Learn at your own pace at a fraction of the cost of regular courses

CYBER 51 COACHING FORUM

CYBER SECURITY FORUM



CONTENT:

1. General Topics
2. Service Assessment
3. Ethical Hacking
4. Cyber Threats
5. Mitigating Cyber Threats
6. Penetration Testing

CYBER 51 INSTRUCTORS



OUR CERTIFICATION LEVELS:

- Certified Ethical Hacker (C|EH)
- Forensic Investigator (C|HFI)
- Certified Security Analyst (ECSA)
- Licensed Penetration Tester (C|LPT)
- Network Security Admin (ENSA)
- ISC Consortium (CISSP)

FEATURES



ADDITIONAL FEATURES:

- 1-on-1 Coaching
- Trainers with Years of Experience
- Wealth of Tools
- Webinars
- Networking with other members
- Contract & Perm. Job Opportunities

WHY CYBER 51?

- 🕒 Learn whenever you want to
- 🕒 Dedicated 1-on-1 Coaching
- 🕒 Information you will not find on public boards
- 🕒 All Mentors work as Senior Security Consultants
- 🕒 Frequent updates
- 🕒 Great Value for money



CONTACT US TODAY

CYBER 51 LIMITED, 176 THE FAIRWAY, SOUTH RUISLIP, HA4 0SH, MIDDLESEX, UNITED KINGDOM

EMAIL: INFO@CYBER51.CO.UK

WEB: WWW.CYBER51.CO.UK

DEFT CON 2012

– WHERE DEFT DEVELOPERS MEET USERS

RUGGERO RISSONE



First of all, for the few guys that don't already know what is DEFT, I'm talking about a Linux distro (based on Lubuntu) customized to be the most comprehensive "solution toolbox" available for Digital Forensics activities. Moreover it's completely free (under CC License 3.0), and with a huge user manual (290 pages) available in Italian (this distro is 100% Made in Italy) and English (actually for version 6 only: work in progress for the version 7); future translations (expected in 2013) will be also in Spanish and Chinese. Further information to the DEFT Project are available on <http://www.deftlinux.net/>.

Coming back to the DEFT CON 2012, the conference was held in Turin in March 30th, 2012 and was a big success: 16 speakers and more than 200 participants, mainly Military and Police forces agents, but also IT Auditors, pentesters, private investigators and lawyers.

During the opening Stefano *youngSTER* Fratepietro, Founder and Project Leader since 2005, introduced the new release (v 7.1): It now cover also Mobile Forensics and Cyber Intelligence. Today there is no other freely distributed system that allows you to perform Intelligence tasks.

Other tools available in the distro are used for : Disk Forensics, Network Forensics, Live Forensics and Incident Response.

The approach of the *deft* team it's quite *simple* : finding the best and/or promising tools for digital forensics and apply them to real cases (most of all the developers are LEA officers or Digital Forensic Specialists for attorney and lawyers), adding functionalities and/or removing bugs, developing new tools not already present in the *free market*.

Yes, one of the main problem in Digital Forensics is that available tools are often expensive commercial software and dedicated to specific aspects in the landscape of cybercrimes.

Deft could be executed on every x86 architecture (future plans could be the support for SPARC architecture) and could be installed on a limited hardware; one typical application is a forensic duplicator realized with DEFT installed on a Netbook (a commercial one could cost more than 1000\$, i.e. Tableau TD1 Forensic Duplicator).

During the conference, several speakers presented some of the new (and old) features of DEFT Linux, each of them would need a dedicated article for cover the entire subject.

In the following lines, just an overview about the topics discussed.

- **Mobile Forensics.** Introduced with version 7, the tools available on DEFT don't support all the devices available on the market, but you have enough "firepower" to analyze Blackberry, iPhone (1,2 and 3) and Android smartphones. In particular, with SQLite Database Browser you could analyze most of all the information registered by mobile applications. Some problems in Mobile forensics could

come from the different filesystem chosen by vendors, in particular in mounting/manipulating YAFFS2 images.

- **Cyber Intelligence** is implemented in several ways in DEFT Linux : using **OSINT** (Open Source Intelligence) **framework**, grabbing informations from several sources with Maltego and/or with a customized Chrome browser with several plugins and resource (passive approach); using Proactive Resources (active approach).
- **HB4Most**, used as a GUI for Scalpel and Foremost, two of the most used tools for file carving, not so easy to manage from the command line, until now.
- **Timeline and Supertimeline.** Well, it's important to have a complete view about all what is occurred on the target. Log2timeline is a single tool that parse various log files and artifacts found on suspect systems and produces a timeline that can be analysed by forensic investigators/analysts. When you are going to analyze a timeline you have to keep in mind what are the differences between different OS and platforms, in particular how it is represented the system time : some applications and systems shows the date counting the msec from 1/1/1970, Apple starts from 2000.
- **Emule Forensics.** Yes, it's possible to perform forensic investigation into Emule and discover who is the first sharer and who are their primary and secondary contacts.
- **NFAT (Network Forensic Analysis Tool).** XPlico is great tool that allows data extraction from traffic captures (pcap files). It supports extraction of mail, VoIP streams, photos, texts and videos contained in MMS messages. Ideal for a quick analysis during a MiTM.
- **Record ALL.** In every forensic analysis it's strictly important that all the actions done are recorded in order to guarantee that nothing has changed the acquired data. A good approach is the usage of Screencast (using also the mike for describing what is going to happens) plus a Network Dump plus all the application logs coming from the tools (also the shell dump, history + outputs).
- **The cloud.** Resources in the cloud, available at very low costs on the market, are used for malware spread and botnet development too. We are in front of a new paradigm: Crimeware-as-a-service. The wide usage of cloud computing resourcing for conducting cyber-attacks is a problem for forensic analysts, in a scenario where cloud computing becomes an excellent anti-forensic tool if the ISPs don't support to police forces investigation.

It was a pleasure meeting almost all the DEFT Development Team member, even if the time available for Q&A sessions was too short.

Anyway, if you are interested in digital forensics, follow the DEFT linux Forum (<http://www.deftlinux.net/forum/index.php>) and the several WIP projects coming from them.



Get the best real-world
Android education anywhere!

Attend

AnDevCon III

The Android Developer Conference

May 14-17, 2012
San Francisco Bay Area

AnDevCon is the biggest,
most info-packed, most practical
Android conference in the world!

- Choose from over 65 Classes and Workshops!
- Learn from the top Android experts—including speakers straight from Google!

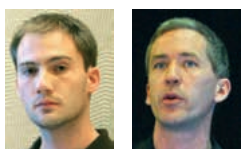
“AnDevCon had a good mix of presentations — some explored the newer cutting-edge technologies, and others offered a deep dive into existing ones.”

—Priyanka Kharat, Software Engineer, Intel

“AnDevCon is great for networking, learning tips and tricks, and for brainstorming innovative, new ways to create apps.”

—Joshua Turner, Software Engineer, Primary Solutions

Google keynote!



Romain Guy
and Chet Haase

Register Early
and SAVE!



AnDevCon™ is a trademark of BZ Media LLC. Android™ is a trademark of Google Inc. Google's Android Robot is used under terms of the Creative Commons 3.0 Attribution License.

Follow us: twitter.com/AnDevCon

A BZ Media Event

Register NOW at www.AnDevCon.com

CORPORATE FACEBOOK ACCOUNT LOGIN FORENSICS AND U.S. LAW

DAVE SAUNDERS

The Computer Fraud and Abuse Act (CFAA) was instituted in 1986 to reduce hacking and unauthorized access of computers. Most recently, in *United States v. Nosal*, the 9th Circuit narrowly held the CFAA is limited to prohibit actions by individuals that *exceeds authorized access* and violations of restrictions on access of information, and not restrictions on its (the information's) use. The 9th Circuit most recent ruling is important in it confirms and criminalizes potential unauthorized access of private corporate information by a third party.

Unfortunately, the Court has not been as judicious in application and interpretation of statutes focused on cybercrime. Generally, lack of developed cybercrime law and coherent application of existing law has made it difficult to prosecute unauthorized access of online information and has resulted in massive cross-jurisdictional inconsistencies.

Additionally, although the basic framework is in place for prosecution and criminalization of cybercrimes the process of prosecution – meeting the burden on the rules of evidence and procedural requirements of discovery can serve as a roadblock in criminal prosecution of cybercrimes.

Following judicial trend, Facebook has made slow progress in their efforts to offer better security and privacy for users, but it remains the responsibility of account holders to make sure they are utilizing all of the security features available.

Below are some tips and tools that maybe used to better protect your Facebook corporate account and manage potential incidents of breach by unauthorized third parties on your Facebook account.

One important feature allows members to monitor their profile for any unauthorized access to their Facebook account. The login alert security feature* on Facebook requires the user, such as the authorized administrator of a corporate facebook page, to go to the "Account" drop-down menu in the upper right hand corner of your Facebook profile page and choose the *Account Settings* option.

Select the "Account Security" option from the security menu:

Here, the authorized administrator can choose to browse Facebook over a secure HTTPS connection (United States only), as well as opt to have Facebook notify you by email if your account is accessed from a different machine.

The page also shows a list of devices that have recently accessed your account, and an approximation of the origin of the login based on the IP address.

The problem with this approach is that IP addresses can be spoofed or proxies can be used to disguise locations. Facebook acknowledges: *In certain cases, it might be possible for the attacker to see or redirect the response to his own ma-*

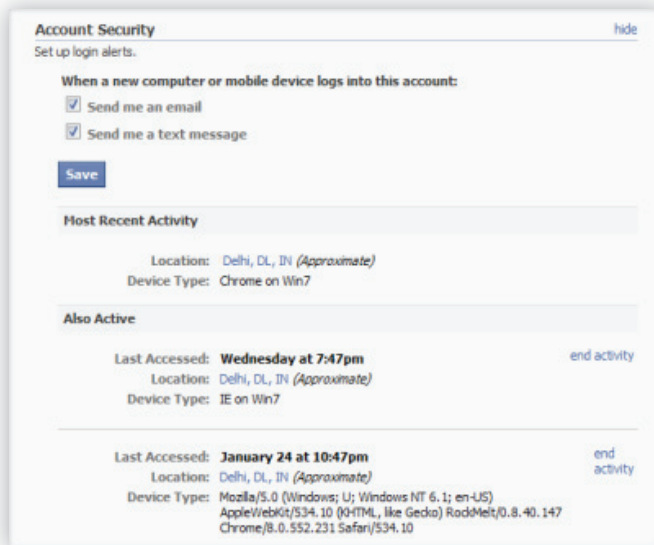


Figure 1. Account Security

chine. The most usual case is when the attacker is spoofing an address on the same LAN or WAN.

Nevertheless, login alert security will show you if your account has been accessed by someone other than yourself. If this is the case, you can choose to logout the unauthorized device.

But let's say you want to go further, and identify who is logging into your account. There are two options available to you, the user: (1) Trace the Facebook User by engaging in a chat with them or social engineering together with a phishing URL and (2) The Phishing Chat Trap.

Chat Trace

Using *netstat* command in windows. If you want to know the IP address of a specific person on Facebook or Orkut or any chat service: Just invite or ping him or her for a chat and while chat is ON open the *Command Prompt* on your PC : (*Start >Run>cmd*)

Note: before trying this make sure you close all the other tabs in your browser. Facebook is the only website that should be open. Also, if possible delete all the history and cache from your browser.

When command prompt opens type the following: command *netstat* – an and hit Enter.

And you will get all established connections IP addresses there. Take a screen shot of all of the suspicious IP's and related information.

The Next Step is to Trace that user using his or her IP address.

To do so we will be using an IP tracer service. Go to the following URL address http://www.ip-adress.com/ip_tracer paste the IP address in the box that says *lookup this ip or website* and it will show you the location of the user.

The website will show you all the information about that user along with her ISP and a location in the map. Inside the map, click on *click for big ip address location* which will allow you to zoom in. and try to recognize the area.

The Phishing Chat Trap

Alternatively, we can obtain the Facebook user's IP address by social engineering together with a phishing URL using the myipstest.com service.

First, go to <http://www.myipstest.com> and on the page "Get Someones IP" you will see:

- *Link for person* – the link that you need to give your unfortunate unauthorized user.
- *Redirect URL* (optional) – the specified URL that your unauthorized user friend will be redirected to after clicking the above link.
- *Link for you* – the link that you can check if your unauthorized user has clicked your link.

Second, fill in the *Redirect URL* (whatever you want, e.g. LNK.IN or TinyURL).

Third, copy the URL from *Link for person* and send it to your unauthorized user via Facebook message or chat.

Finally, follow the URL from *Link for you* and you will receive your unauthorized user's IP after he or she clicks on your link. Since this trick requires the other person's cooperation, you need to become or already be a friend with the unauthorized user person in Facebook in order to increase the chance of success.

Enter in the unauthorized user's IP address and screen capture the information.

It is important to note that, isolating the breach and locating the origin of the unauthorized access has become increasingly important because the 9th Circuit has held that costs associated with the tracking and discovery of the identity of the person who stole proprietary information from a company does constitute 'loss' for the purpose of calculation [and satisfying the damages requirement] under the CFAA.

Finally, it is possible to seek support from Facebook in managing and prosecution unauthorized access to corporate profiles. Facebook can and does provide a Facebook Account Report to law enforcement that generally includes single photograph, plus formal details such as: the image's caption, when the image was uploaded, by whom, and who was tagged. Other information released includes Wall posts, messages, contacts, and past activity on the site.

* Digital Inspiration posted an article titled *Check if Someone Else is Using Your Facebook Account* which highlights the login alert security feature on Facebook. <http://www.labnol.org/internet/check-your-facebook-account/18577/>

DAVE SAUNDERS

is the co-founder of Cloud Forensics Network. Dave speaks at various cloud computing and business recovery conferences around the world about cloud forensics tools and is a member of Cloud Security Alliance. Adrian Chua is the co-founder of Cloud Forensics Network and is a licensed attorney in California and Washington DC.

CLOUD FORENSICS NETWORK

Inc. is digital forensics software as a service which is monitoring social network data for advance e-discovery. Our forensic response tracks, traces and collects evidence of unauthorized access, breach, and exfiltration while proactively notifying of negative events i.e. defamation. The e-discovery process is based on a legal foundation designed to minimize resources and response time. We also maintain an inventory of cloud forensic tools and best practices to assist corporate clients in building capability.

FORENSIC TOOLS FREE AND PAID

JERRY HATCHETT

“Which software should I buy, EnCase or FTK?” As someone who’s been practicing digital forensics for a long time, I can’t count the number of times I’ve fielded that very question from an eager young forensicator. (“Forensicator” is an industry word; the dictionaries haven’t caught up yet.) I remember asking it myself. I remember researching it to the nines myself. I remember my choice, and I remember how quickly I learned that the question of which forensic tool is “best” is a question that never gets answered. Want to know why?

Here’s plain reality: No single tool can come close to meeting the needs of a typical digital forensic examination. Today’s devices are complex and growing moreso. Data volumes climb without pause. The operating systems of mobile devices can vary significantly, not just from manufacturer to manufacturer or carrier to carrier, but within devices of the same make and model. Keeping up is a challenge and to meet it you will need a lot of tools in your box. Some are software and some are hardware. Some of these tools are free. Some of them are commercial but priced within reach of the smallest practice. Others require a serious financial commitment.

Before we work through the process of building a toolbox, let me issue a wee disclaimer. Everyone has personal preferences, including me. No matter how hard I try to be objective, in the end I may choose one tool over another equally capable tool for no reason other than I like it more. Maybe you would choose differently, and that’s okay. The point is to put together a collection of tools that gets the job done, and if I leave out someone’s favorite tool ever in the history of the universe, it’s not a personal sleight, just a different choice. Also note that the case scenario below is not designed to be a pristine example of how to work a digital forensic case. The purpose is to cover a lot of common forensic tasks, and the tools that can help you do that, in a few thousand words. To make that happen, be advised that I will do something herein that you would never want to do in real life: Design a case workflow around a series of tools instead of around the case facts. For each task and tool, we will cast a discerning eye toward 1) forensic soundness of results, 2) efficiency, and 3) cost-effectiveness. (Clients often—not always, but often—care a lot about that last one.) Let’s roll.

As a workflow-oriented examiner, I think the most logical way to build a toolbox is to move through a case and look at ways to accomplish the most common steps along the way. Here’s the scenario: On October 12, 2011, without warning Jonathan Doesome left his six-year job as an engineer at High

Tech Enterprises (HTE), and went to work for a competitor named Research Forest (RF) on October 18. Less than a week after Doesome arrived at RF, that company issued a press release about a revolutionary new product called simply “X.” HTE quickly filed suit, alleging that Doesome had stolen the designs for “X” from them when he left and subsequently gave them to his new employer.

The plaintiff (HTE) hires us to find out and testify about what happened. (Congrats to us!) We are provided raw forensic images by the forensic firm hired by the defendant (RF), of Doesome’s RF computer and his home computer. HTE sequestered Doesome’s only device, a laptop computer, when he left, and we need to forensically image its hard drive. Our process must yield a bit-for-bit copy of the original and it may not alter the original in any way. We consider the following options for this, the collection phase.

PURE SOFTWARE: Protect the integrity of the original evidence by software and create the forensic image on a target computer without the aid of any hardware bridge between the two. As with any forensic process or tool, it is imperative that you thoroughly test software write-blockers before accepting their claims to not alter original evidence.

TOOLS TO CONSIDER

FastBloc Software Edition – Marketed as an EnCase module by Guidance Software, this Windows-based software write blocker will allow you to acquire directly into EnCase without the need of a hardware write-blocker.

SAFE Block – Sold by ForensicSoft, this software will let you write-block any disk of any type that you connect to your computer, except the boot drive. The obvious advantage here is that you can safely connect hard drives in a way that lets them run at their best speed. For example, SATA3 drives can be connected to a SATA3 controller and still be write-blocked. Very handy utility, but pricey at \$399 per workstation. Even worse, the support by

the company is abysmal. We purchased and installed a SAFE Block license, then decided to move it to another machine. We've been asking and waiting for that help for literally months.

Thumbscrew – Free applet that flips the appropriate registry bit needed to make your entire USB system read-only. Thumbscrew is my favorite, but there are several options to choose from out there, all free.

SOFTWARE AND HARDWARE COMBINATION: Hardware write-blocker to protect the integrity of the original evidence, acts as bridge between original device and target computer.

TOOLS TO CONSIDER: This niche of the field is filled with choices and I readily admit there are many brands and models of hardware write blockers that I have not had an opportunity to use. With that understood, I favor the products from the manufacturers listed below.

Tableau – If this space has a big primate, it's Tableau. They build a wide range of write blockers to accommodate virtually any device you're likely to encounter, and I have had excellent service from many different models over the years.

Wiebetech – These guys also make a good selection of write blocking devices, and they have steadily grown their presence and visibility in the industry over the past few years.

CoolGear – I'm a firm believer in finding tools that do the job I need at the most reasonable cost available, so there's no way I could write an article about forensic tools without giving a shout-out for these guys. They don't have a big selection of forensically-capable items yet—hopefully that will change—but at \$49, their SATA/PATA-to-USB 3.0 adapter with the write-block switch is a bona fide steal for forensicators. You can find the unit at Amazon and assorted other online retailers.

IMAGING SOFTWARE: Whether you go with software or hardware write-blocking, once you have a safe datastream established, you'll need software to read those bits and create a forensic image from them.

TOOLS TO CONSIDER

EnCase Acquisition – This version of EnCase will run without the need for a license on a dongle, and allow you to capture physical or logical devices as E01 images.

FTK Imager – Free to download from AccessData, Imager (as it's commonly referred to in the industry) should be a staple in every collection of digital forensic software. It will let you not only image physical and logical devices, but also open forensic images. Once you open an image, you can navigate its file structure, select and export files, output lists of files complete with hash values, and more.

Tableau Imager – Yes, Tableau, I know you call this software "TIM." But it's such a kicking tool that it deserves some attention and I felt it had a better chance of getting it if I call it what everybody but you calls it. So there. Whatever it's called, the latest freebie to show up on the scene from a major forensic name is definitely worth your consideration. It has a snazzy interface and snappy speeds, and most importantly, little Tableau utilities have a history of working long and well.

dd – Built into every *nix computer is the ability to mount and image any attached device, in read-only fashion, using the "dd" command-line application. Its power and configurability have made it a long time favorite throughout the industry. In addition to its use from a command prompt, numerous hardware acquisition devices use a variant of it internally to create RAW images if you choose that format. Oh, like most things in the open source world, dd is free.

PURE HARDWARE: Standalone forensic imaging device, no other hardware or software needed. All the devices discussed

below are SATA-to-SATA. Connection of SCSI, SAS, PATA (the cool way to refer to "IDE" now), etc., will usually require optional adapters or modules.

TOOLS TO CONSIDER

Tableau TD-1 and TD-2 – These two models are identical except for the TD-2's ability to write to two target drives simultaneously. They are fast, they have proven reliable for us, and they allow you to create images in either RAW or compressed E01 format.

Voom Technologies HardCopy 3P – As a user of HardCopy models going back years, I am a fan of the products and the company. The units are fast, durable, reliable, and the 3-button control system is a thing of beauty. If you are content or required to acquire all your images in RAW format, the HardCopy should be on your short list of choices. If you prefer to acquire to E01 format, however, the current model does not offer that option. We can keep our fingers crossed that the next model will.

ImageMASter Solo series from Intelligent Computer Solutions – An early vendor of hardware imaging solutions, these guys have continually developed new products to attempt to stay up with the times. Their latest products offer great sounding features and claim great speeds. I wanted to be able to talk about them from a more informed position, but they declined to provide me with a review unit, so I can only say that I am intrigued by their features and design, but would hate to invest that kind of money or recommend anyone else do so, without the ability to verify the claims in the real world.

Logicube Talon Enhanced – Logicube has been in this space for a long time, long enough that they were one of the very first vendors of hardware-based forensic imaging devices. They are fine folks and provided us with a review unit of the Talon Enhanced imaging device. The device was among the fastest we've seen, and its design of enclosing the target hard drive inside itself is interesting. A big negative was the sound; it's loud. A bigger problem is price. Feature for feature and spec for spec, Logicube products are two to three times the price of those from other manufacturers.

[SIDEBAR] Don't forget the little things! Any field kit should include a variety of cables, adapters, and assorted technowizardry to handle both the expected and unexpected parts of the job.

Labels: A variety of portable labelmakers are available at affordable prices. No forensic field kit should be without one. Whether it's for a criminal or civil case, we collect evidence, and it must be treated as such. Items should be clearly and immediately labeled. Every time. Without fail. Can you imagine a homicide detective just grabbing evidence from the scene and throwing it all in a box because he just knew he could remember where everything came from? Of course not. Don't be guilty of the digital equivalent.

Disassembly organization – How many times have you removed a hard drive from a laptop and before the job was over, you dropped one of those tiny screws and spent an hour looking for it? Here's what a \$3 fix looks like.

Once you have valid forensic images of all the known involved devices in hand, it's time to move to the next phase of the case. There is no industry-standard jargon I'm aware of to label the various elements of a digital forensic case, but terms I've heard bandied about for the next step are pre-processing, initialization, and processing. For the sake of this article, we will use the term pre-processing. It is here that we will get the evidentiary images loaded into the core forensic platform. We will ask that software tool to get some basic housekeeping out of the way, things like interpreting the forensic images so we can work with the original

device's file structure, calculating hash values and discovering file signatures for those files, expanding any compressed files or email containers, recovering deleted files that can be readily reconstructed, and more. The exact steps typically performed during the pre-processing phase will vary by platform, but whichever platform we choose, we want to come out of the pre-processing phase with our evidence loaded and ready for us to begin analysis.

Remember the question that started the article, the one about EnCase or FTK? This is where we will once and forevermore answer that question with finality. Just kidding! We will, however, expose the false premise of that question right now: EnCase (sold by Guidance Software) and FTK (the Forensic ToolKit, sold by AccessData), are two of the choices of core forensic software packages available to today's digital forensic examiner. Nothing more.

[SIDEBAR] BUSTING A MYTH: Marketing claims notwithstanding, there is no such thing as a "court approved" forensic software package. When on the witness stand, what matters is your credibility, the soundness and admissibility of your work, not the make of your tools. (Use common sense: You of course want to choose tools whose soundness is readily provable.) Consider an analogy: Imagine a trial in which a mechanic (technician may be more accurate in today's automotive environs) is accused of having botched an auto repair job. The car subsequently failed and caused a death. Do you think his cross examiner is more likely to focus on the fact that the mechanic was a) known to drink while working, or b) used a Craftsman ratchet instead of a Snap-On when he bolted the key part in place? Sound silly? That's the point.

Which one is most popular? Informal and unscientific surveys/conversations I've floated in the community consistently convince me that EnCase is the most popular of the core digital forensic examination platforms. Is this because it's the best, you ask? In my ever so humble opinion, no. I believe EnCase created and has maintained its status as 800-pound gorilla because it got visible to market earlier than its competitors and built a large user base among the law enforcement community. Making a move to another primary platform involves training, reduced workflow while the acclimation is underway, added expense, and all the other discomfort that comes with any significant change. In other words, in the opinion of this author, EnCase has remained on top largely due to familiarity and inertia.

Taking second place among examiners I've talked with is FTK. Its approach to processing and analysis is markedly different from EnCase, and it is a much easier tool to wrap your brain around out of the box, especially if you're new to the digital forensic arena. Its indexing feature, powered by the tried-and-true dtSearch search engine, inspires a great deal of loyalty among its users. As we'll discuss in more detail, AccessData has had some major gaffes over the past few years, so the company should be extraordinarily thankful for anything that kept their customers from jumping ship long ago.

Arriving in third place with an attitude, X-Ways Forensic is arguably the most powerful of the "Big Three," despite the fact that it sells for about half the price of its triad brethren. It also enjoys something I've never seen any evidence of among the user base of either EnCase or FTK: Unbridled enthusiasm and loyalty. Nay. That's not strong enough to describe it. X-Ways users are often fanatical in their devotion to the product. They don't just like it. They love it. Is the love justified? Fear not, we will explore that question. Finally, while my oh-so-imprecise information gatherings don't reveal big numbers for other tools, others do

exist both paid and free. A leader of the open source pack, the SIFT Workstation is in its own right a very interesting player in the space, and there are other examination tools out there that have managed to stay alive for years and warrant some time in any serious discussion about available tools.

With all that said, let's get down to some specifics, some pros, some cons, some objective observations, and I'll even throw in a few opinionated tidbits for flavor.

EnCase

Aside from the aforementioned familiarity factor, one of EnCase's biggest strengths is its EnScript feature. While the program comes out of the box with a spate of these task-driven applets, its real power lies in the large selection of third party scripts available. Some come with a price tag, but the vast majority of them are shared with the community at no charge by their creators. (To be certain, some of the paid scripts are well worth what they cost.) EnScripts are available for a wide variety of specialized processing and/or analysis chores, from extraction of USB history and other system artifacts, to file carving, to exporting results, to many other needs you may come across within the EnCase environment. If you're the codeslinging type, you can of course create your own.

On the wrong side of the balance scale, EnCase has its share of quirks, weirdness, and features that work in less than dependable ways, if at all. There's the White Screen of Wait, which becomes the White Screen of Death if you irritate it. Poor use of multi-core processors. You can of course report these issues, along with requests for new features, to Guidance Software. You might even get a speedy response telling you how much your patience or great idea is appreciated and how it has been entered as Issue XYZ, or Feature Request 16893246972. That may be the end of the story, but it does have some entertainment value.

All in all, EnCase is a good product that provides a broad and very granular feature set that will allow you to conduct a meticulous and fairly comprehensive digital forensic examination of most computers, be they PC, Mac, or Linux. It is a product used daily in our lab and will continue in that role for the foreseeable future. Finally, being somewhat infamous for my sense of candor, I must point out that all EnCase comments thus far refer to version 6. Its successor, creatively named version 7, has been on the market for almost a year as of this writing, but in the experience of me and many other examiners, it simply does not work and stands as one of the biggest debacles yet seen in the world of digital forensic software.

FTK

Without a doubt, FTK's claim to lasting fame is its robust dtSearch-based search engine. While many examiners use EnCase in an a la carte fashion, running specific processes as needed, FTK takes a get-the-waiting-out-of-the-way-up-front approach that can be a real pleasure to use once the initial processing is complete. This process can be configured to your preferences, of course, but a typical FTK pre-processing phase involves calculation of hashes for all files in the case, expansion of PST and other email containers, carving if you so desire, and creation of an index of words present across the body of evidence. This pre-processing can be lengthy if you're dealing with a lot of evidence and/or anemic hardware, but it can be significantly sped along these days by using the distributed processing capabilities built into the product since version 2. Getting the distributed processing, as well as the central database that stores all the particulars of an FTK case, properly tweaked can take some time but once it's done, the system works quite well.

With pre-processing done, searches are fast and you can add new search terms or change old ones to your heart's content. This makes FTK well suited for cases in which the investigation evolves depending on what you find and when you find it. It also has a user-friendly interface that can be learned with comparative ease compared to other products. The downside is a lack of granularity compared to its competition. Its scheme of defining filters has improved greatly as the product evolved, but is still cumbersome and time consuming. If you lack the resources to get product-specific training at the outset, FTK is a platform worth much consideration. Finally, in fairness to Guidance Software's epic failure with version 7, I must point out that AccessData pulled its own share of major gaffes when it made the move from its old staple, version 1, to version 2 and beyond, releasing build after build that simply did not work.

X-Ways Forensic

Don't expect a lot of glitz when you fire up X-Ways, because its interface is decidedly geared toward functionality over flash. The installation process is not as automated and slick as are many other applications in 2012, and it will take you a bit of time to carefully read the manual and properly configure its myriad settings. The construction and interface of this product are reminiscent of a race car, all the fluff stripped away for maximum performance. The result is an application that is measured in a tiny number of megabytes, a refreshing departure from today's ever-growing behemoth apps. And the philosophy works. X-Ways is hands-down the fastest application among the three competitors. It loads evidence faster, hashes it faster, carves it faster, and searches it faster, often dramatically so.

The bare-bones interface can be challenging and while the importance of proper training in our field can never be overstated, this is especially true with a product like X-Ways. Many of its most powerful features lie within its configuration panels and check-boxes, and they are not always obvious.

Open Source – Core Forensic Software

SIFT Workstation

Acronyming it to the max, the SANS Investigative Forensic Toolkit (SIFT) is provided by a team of forensic smart guys headed up by wizard Rob Lee at tech training organization SANS. Provided as a public service, the tool in its primary format is as a ready-to-run (Linux-based) virtual appliance that can be run using free virtual machine (VM) player software from VMware, or from their paid tools like VMware Workstation. It bundles together other open source forensic tools that have been around a long time, including open source stalwart SleuthKit.

In a couple of words? SIFT rocks. We will talk in more detail about some of its capabilities as our case progresses, but as an overview, there's not much in the way of digital forensic examination that you can't do with SIFT. You can image, pre-process, and analyze to your technoheart's content. No cost plus broad capability equals a very interesting opportunity for those who want to sample our world without a big fiscal commitment. And even if you have a well stocked toolbox full of commercial tools, SIFT offers some power that I haven't seen available anywhere else. At any price. If you do decide to work with SIFT, however, be prepared. It's not for the technically faint of heart. It's Linux and a lot of its power is accessible only via a command prompt. It can be a real challenge to use many of its tools, but if you're a geek who takes satisfaction in such things, you can reap yourself a boatload of gratification with SIFT.

OUR CHOICE: If it is within your budget, my recommendation is to use all of the above. Each of the tools discussed above is superior to the others in certain case situations. For example, FTK's indexed search approach is perfect for cases that have evolving search terms, since new search terms can be run and hits harvested almost instantly. Conversely, for live searches, X-Ways tears through data at a far faster rate than the others. Its search output is likewise clean, snappy, and easy to review and categorize.

Another critical feature is the ability to search unallocated space for email fragments that are encoded in Outlook Compressible Encryption (OCE). EnCase handles the search without issue, but its results are extremely difficult to export for use outside EnCase. X-Ways does a great job across the board with OCE, both finding the fragments and allowing you to get them out of the application in a readable form. FTK doesn't handle OCE fragments at all, a major shortcoming in an industry that so often deals with Outlook-based email.

The bottom line is an ideal forensic workflow should be designed not around tools, but instead your toolbox should be built around the evidence before you and the nature of what you're mandated to discover, interpret, and analyze. Also remember that key findings need to be cross-validated, making yet another strong case for a multi-tool approach to digital forensics.

Let us now accept that our case has been pre-processed, with the following steps complete:

- All forensic images have been properly interpreted and presented to us as valid file structures.
- Compressed files and containers have been expanded.
- All files have had MD5 hashes calculated.
- All files have had a signature analysis process run that checks to see if a file is what it claims to be.

With these steps out of the way, we can dive into the fun stuff. Let's figure out if Doesome is a data thief or a wrongfully accused man. A quick look at the user-created content on his old HTE computer shows several thousand documents, spreadsheets, and emails that span the period of his employment at HTE. A quick hash comparison determines that none of these files exists on his new RF computer, at least not in a forensically identical condition. A glance back at our pre-processing steps shows that we did not index the evidence, so any keyword searches we conduct will need to be live searches that comb the evidence looking for what we describe. This can take a while, so before we launch the search process, it would be a good idea to export any artifacts on which we intend to conduct external analysis. (Exporting files while a search is in process is easy enough in EnCase or FTK, but problematic in X-Ways.)

We select the following files for external analysis and export them from our core software:

Doesome's NTUSER.DAT registry files from both his old (HTE) and new (RF) computers.

SYSTEM registry files from both computers.

SOFTWARE registry files from both computers.

UsrClass.dat registry files from both computers.

All event logs (.evt and .evtx) from both computers.

All link files (.lnk) from both computers.

Setupapi logs from both computers.

Reminder: Do not use this as a template for a real-world case. This scenario is designed for exposure to certain tools and is not complete.

Windows registry files can be fertile ground to explore for clues to user behavior. And while each of the Windows-based core forensic tools we've talked about (EnCase, FTK, X-Ways) has registry analysis functionality built in, many prefer to do this analysis in external tools. Count me among that group. While it's possible to navigate through the structure of registry hives inside the core tools, I have found external tools to provide a far faster and more efficient route through Registry Land.

FEATURED TOOLS

RegRipper – This freeware tool, authored by well known and much respected digital forensic author Harlan Carvey, excels at pulling a plethora of useful information from the registry, including Most Recently Used (MRU) lists that document recent file activity and a long list of other configuration and activity artifacts to comb through.

USBDeviceForensics – Another no-cost solution, this tool pulls off some slickness by pulling information about previously connected USB storage devices like thumbdrives, flash cards, etc. It is possible to look this information up manually in the registry hives and achieve the same results, eventually, but this utility turns it into a one-minute job as opposed to the significant time it can take to find and correlate the information from numerous Windows system files into a unified chain of events. It is in fact the multi-source capability of this tool that sets it apart from others available. Where other USB-history utilities rely on information only from the SYSTEM registry file, USBDeviceForensics goes further and extracts information from both the NTUSER and SOFTWARE registry files, as well as SETUPAPI logs. The result is an impressive output of useful information about USB storage devices that have been plugged into a computer. You can download this marvel from Woanware at www.woanware.co.uk.

RegExtract – Slipping in another favorite, our friends at Woanware also created this tool. It can extract key information from a single registry hive, or all registry hives in a folder. It can run single, multiple, or all of a variety of plugins at the same time. Its output is clean, thorough, and useful. And once again, you just can't beat the price!

Using the tools above, we were able to determine that a thumbdrive with the volume name "Xstuff" was connected to Doesome's HTE computer about two hours before quitting time on his last day to work there. Three days later, that same thumbdrive was connected to his new RF computer. Quite interesting, wouldn't you say? With a device connection established between computers old and new, we will next look for any information we can glean about the potential transfer of files on that thumbdrive. For this task, a look at link files is a logical next step. Link files, or shortcuts, can be invaluable in cases involving alleged theft of data, because they often provide information about files that no longer exist.

FEATURED TOOL

LinkAlyzer – Although this is a paid tool that will set you back a bit over a hundred bucks, it has become my favorite link file analysis tool by far. In addition to its ability to ingest folders of link files (like those we exported above), LinkAlyzer will directly access an E01 forensic image and carve it for link files. This methodology can yield a gold mine of information about deleted link files, which in turn can provide valuable information about other deleted files. Crafted by Sanderson Forensics, LinkAlyzer generates a color-coded table of detailed output parsed from the link files it processes. Learn more about this tool at www.sandersonforensics.com.

Returning to our case, link file analysis has added another clue into our universe of known facts by establishing that a docu-

ment named "XMASTER.DOCX" was created on the "Xstuff" thumbdrive about fifteen minutes before Doesome clocked out of HTE for the last time. Three days later, and two minutes after the thumbdrive was connected to his new computer at RF, a file named "XMASTER.DOCX" was created on that computer's C-drive. Specifically, the file was created in a folder called "Xstuff" in the root of the C-drive. A compelling idea about what Doesome did is beginning to form, but it's not complete. Stealing someone's data is an awful thing to do but reality is that in civil litigation in the United States, this type of thing often spawns a "no harm, no foul" state of mind with the judge and/or jury if it cannot be established that the receiving party did something with the data that caused harm to its rightful owner. So our next question becomes, "If Doesome took the 'X' data and gave it to his new employer, RF, did they do anything with it that could be construed as harmful to HTE?" Let's explore that issue now.

There are often times when one source of information does not yield a clear picture of what took place. For example, even though link file analysis revealed the creation of "XMASTER.DOCX" on Doesome's RF computer, it was not able to provide any proof as to what happened with the file after it was created on that computer. We know the file doesn't exist any more on the RF computer, which brings our reasonable conclusion thus far to the likelihood that Doesome copied the file from his HTE computer and to his "Xstuff" thumbdrive, on his last day of employment there. Evidence then suggests that three days later he copied the file from the thumbdrive to a folder named "Xstuff" on the C-drive of his new computer at RF. What came next?

Windows systems now store data about folders in artifacts called "ShellBags." Extracted from a registry hive, these artifacts can provide a wealth of information about folders, including dates of creation, modification, and access, and even the size and layout of a folder's window when displayed on screen.

FEATURED TOOL

Our favorite tool for finding and extracting this information is a tiny command-line application called "sbg.exe." It can be downloaded free of charge from www.tzworks.net and it pulls data from NTUSER.DAT and UsrClass.dat registry hives to create a record of file folders as they existed in the past on the computer.

In this instance, our ShellBag analysis revealed that thirty-two minutes after the "Xstuff" folder was created on Doesome's RF computer, that folder was opened. One minute later, a new folder named "RF-Xstuff" came into existence. Our mental picture of what happened in this case is sharpening.

Armed with our expanding set of facts, HTE's attorney starts putting pressure on Doesome through his attorney. After a little back and forth and posturing, Doesome admits that he took the "XMASTER.DOCX" file with him when he left, but insists he never did anything with it. RF's announcement of their own "X" product just after he came on board was coincidence, nothing more. He claims that a look at RF's Facebook page will make clear that they already had the product in the works weeks before Doesome joined the team. His mention of Internet history was well timed, since it was already scheduled to be our next avenue of inquiry.

As is the case with many areas of inquiry, the core tools do have Internet history analysis built in. To a degree. Kind of. Sort of. EnCase's in-tool handling of Internet history is probably the most robust of the core tools, but it is still limited compared to more specialized tools. It also carries an ugly piece of overhead with it: Once the Internet history search has been run in EnCase (a process necessary to view the history), from that point

forward the case file will open very slowly as EnCase works to resolve and reconstruct the Internet history for viewing inside the application.

FEATURED TOOLS

NetAnalysis – Another product out of the UK makes our list. This product has been around more than ten years, somewhat of an eternity in this field, and it has an established and loyal base of users. In addition to doing a great job of parsing index.dat and other browser history files, an included module named HistEx does a fine job of carving deleted Internet history elements from unallocated space. The output is tabular, the interface snappy and intuitive. You can learn more about NetAnalysis at www.digital-detective.co.uk.

CacheBack – Made a bit famous in the Casey Anthony trial, this product takes a markedly different approach to Internet history analysis than its grid-i-fied competitors. While its results are presented in a partially tabular presentation, its big selling point is its ability to rebuild web pages and present them in a browser view that attempts to mimick the original user experience. There are times when such a presentation could have major impact with a judge or jury, and CacheBack is deserving of a look. Check it out at www.cacheback.ca.

Internet Evidence Finder – Previous versions of this application (usually referred to as “I-E-F” within the industry) were good tools. The latest and greatest version of IEF, version 5, is an outstanding tool. A single-user license will set you back between \$500 and \$1,000. It’s not a trivial number but in this case, the tool is worth every penny of it. IEF recovers browser history for the big three (Internet Explorer, Google Chrome, and Mozilla Firefox) and more, but it goes much further, digging through both allocated and unallocated space on a hard drive (or forensic image thereof) and carving out historical artifacts for a long list of Internet sites and applications. This list includes hard-to-get gems like Gmails (intact and fragments), Hotmail, Facebook interactions, cloaked browsing sessions like PrivacIE, and much more. The process is straightforward and streamlined. Its results are quickly and easily reviewable, and items of interest can be bookmarked for isolated presentation or extraction later. If IEF was a movie, it would get two enthusiastic thumbs up from this author. The tool resides at www.jadsoftware.com.

Thanks to our Internet history analysis, we have a little more informative picture of Doesome’s claim that RF’s Facebook page would prove exculpatory. What we found was that RF was indeed touting an upcoming new product on Facebook the week before Doesome joined the team. This did not, however, alleviate our concerns about his actions. It led us to scour his old computer from HTE for any evidence that might establish when his relationship with RF had begun. We hit the jackpot when we recovered a series of Gmail messages between Doesome and RF. Those messages had started nearly six months before his departure. Even worse, many of them had attachments that were loaded with proprietary material belonging to our client. HTE had been pouring money into the development of the “X” product for years, and the closer its release drew, the more quickly Doesome had fed critical design data to RF.

The HTE attorney presented the opposing side with a preview of our findings in hope of reaching a favorable settlement, but Doesome and RF appear dug in for a fight despite the bleakness of their position. We have been asked to wrap up our findings in a report that tells the story of what happened in a way that is compelling in word and visuals. To pull that off, we will use a variety of tools that may not come across as “forensic” in nature, yet play a huge daily role in our work.

FEATURED TOOLS

Microsoft Office – Once we step out of the meant-just-for-forensics arena, the Office suite easily tops our list of tools. Lots of forensic analysis tools output their data as spreadsheets, and while there are several different free Office-compatible suites out there that include a spreadsheet app, Excel is the king of this realm, period. Word and PowerPoint likewise rule their respective fiefdoms, making Office a no-brainer for our shop. If you can’t afford to pony up for Office quite yet, notable alternatives are OpenOffice and LibreOffice, both available for download free of charge.

SnagIt – Although I practiced for several years without this fantastic screenshot capture-edit-annotate application, I look back and wonder how. Grab the free trial from www.techsmith.com and you’ll see what I mean.

MacroExpress – I rarely encounter anyone else who has even heard of this one, but I’m on a lifetime mission to change that. Available for a pittance of about \$50 at www.macros.com, this tool is amazing in its power and ability to automate any routine tasks you encounter in Windows. If you can do it manually in Windows, you can probably automate it with MacroExpress. Need to watch for an error dialog, click OK, move the cursor to a certain row in a table, right-click, and export? MacroExpress is happy to handle this for you, whether you need to repeat it ten times or ten thousand times.

VMWare – Haven’t dipped your toes in the waters of virtualization yet? Shame on you!

Directory Opus – There are times when Explorer just doesn’t have the features necessary for the file operations before you. After evaluating (and even purchasing) several alternatives to Explorer, I discovered Opus and the search was over. Great interface, tons of built-in features like bulk renaming and de-duplication built in, and completely configurable to your needs and tastes. Multiple panes of files, vertically or horizontally oriented, with trees or without, if you can dream up your ideal file browsing and management scenario, you can probably implement it with this powerhouse. You can find it at XXX for \$\$\$.

TreeSize lets you quickly get an overview of volumes of data, and generates nice reports suitable for inclusion in deliverables.

Adobe Acrobat is pricey compared to the multitude of other products designed to generate PDFs, but it’s also in my experience the fastest, the most reliable, and by far the most likely to generate a document that looks like what you fed it.

Evernote – This cloud-based note repository is free for individuals and costs about five bucks a month to upgrade to premium features, most notably the ability to allow others to have read/write access to your shared notes. It handles text notes, graphics, and file attachments with ease.

Boxer – There are things Notepad can’t do. Boxer can.

Alpha Five – When you need to run more sophisticated operations on tabular data than Excel is intended for, you can do it in Microsoft Access, or you could use this much more user-friendly tool.

LogParser – This freebie will take you a little while to learn and a long time to master, but once you’re there, you’ll be able to parse, search, and extract records from virtually any Windows-based log file. Extremely powerful. Extremely granular. You can get the tool at XXX and I also recommend a quick visit to Amazon for XXX.

JERRY HATCHETT

practices digital forensics with a passion from his home base in Houston, Texas, USA, and writes related articles and papers from time to time. You can find him at jhatchett@forensicbit.com.

ATOLA TECHNOLOGY

WE KNOW HOW TO MAKE THE WORLD OF FORENSIC PROFESSIONAL

ATOLA FORENSICS

Forensics is one of the areas that foresees fundamental accuracy, exhaustive clarity and extreme precision. Atola Technology provides the best solutions for forensic through its products: Atola Forensic Imager and Bandura.

Why Atola Forensic Imager? Atola Technology is a specifically designed and unique product for forensics and e-discovery.

Atola Forensic Imager possesses the following key features for data capture:

- Imaging damaged drives:
 - Calculating checksum on the fly (MD5, SHA1, 224, 256, 384, 512);
- Wiping/erasing:
 - DoD 5220.22-M method;
 - NIST 800-88 method;
 - Security Erase;
 - Fill with pattern;
 - Zero fill;
- Case manager;
- Password removal.

It makes this tool perfect as an all-in-one solution for forensic specialists – that easily works with damaged or unstable hard disk drives. Atola Forensic Imager combines a fast imager with strong data recovery capabilities for creating forensic images. Big potential. Great opportunities.

Customize every step. You have the ability to adjust some of the parameters to make the imaging process most effective for you. It is able to work with damaged or unstable hard

drives in the field that cannot be imaged by regular forensic products. This means technicians can image more hard drives in the field without needing to take them back to their labs. Atola Forensic Imager provides a complete image and data for evidence by accessing more sectors. GList auto reallocation can be disabled to prevent the hard drive from contaminating evidence by remapping sectors.

Professional imaging meets any expectations.

Atola Forensic Imager presents the following imaging features:

- Calculate checksum on the fly (MD5, SHA1, 224, 256, 384, 512);
- Disable GList auto reallocation;
- HEX pattern to fill skipped sectors (00 by default);
- Image file size (chunk size);
- Timeout length (after failed sector read attempt);
- Number of sectors to skip after failed read attempt;
- Number of imaging passes;
- Apply Read-Long command on last pass;
- Reduce HDD operating speed to PIO mode;
- Copy direction (forward/linear or reverse);
- Specify read/write heads to transfer from;
- Abort duplication after X amount of HDD power cycles.



Figure 1. Atola Forensic Imager

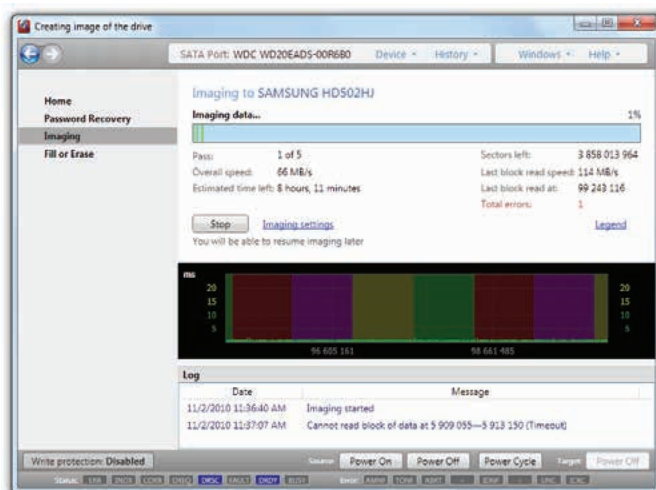


Figure 2. Imaging in Progress

Ability to quick erase. One of the most requested features for forensic professionals – is wiping the destination hard drive to prepare for a case. Our products allows for the guarantee of the accuracy of the data and checking the destination drive for any errors by writing to every sector. You can overwrite the sectors with any specified HEX patterns. It can also execute the Security Erase function, perform a Zero-Fill, NIST 800-88, and do a DoD 5220.22-M compliant wiping.

Effective case management. Our product offers the ability to keep the history of every one of your steps of the process in one place. The case management system works automatically, so all the information is recorded including characteristics, such as date, time and hash value. You can reach it at any time you want very quickly and easily.

Password? No problem! Atola Forensic Imager provides a powerful Automatic Password Removal function. You can remove any password from a locked hard disk drive very easily.

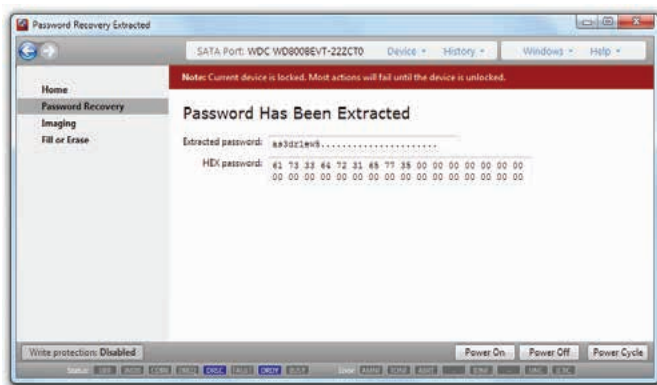


Figure 3. Password Recovery

Technical details. Main product features:

- 2 HDD ports;
- Native ATA/IDE and SATA I/II/III interface support;
- Supports HDD to host data transfer via ethernet (85 MB/sec max transfer speed);
- Supports direct HDD to HDD data transfer (110 MB/sec max transfer speed).

The way you work. Atola Forensic Imager is definitely the highest quality hardware that provides extreme speed imaging even for damaged or unstable hard disk drives. Its user-friendly interface makes every one of your actions clear and intuitive. Also, Atola Forensic includes automatic in-depth disk diagnostics, a built-in oscilloscope for current monitoring, and HDD power control buttons. Professional background. Fantastic results.

ATOLA BANDURA



Figure 4. Bandura

The easiest way to image even damaged drives. Atola Bandura is the only stand-alone tool that combines diagnostic and duplication features. You need only a 100-240V AC power socket. Bandura provides high-level speed disk imaging, automatic checkup, comparison, and secure data erasing. The tool easily deals with damaged disks, including bad sectors, and is supremely efficient. Just two touches is all you need to start any task. Its advanced touch screen makes your experience with the tool simple and natural. Smart technology. Brilliant in practice.

Where to use:

- Recovering data from damaged hard drives;
- Secondary duplication tool to free up *more expensive tools* for other tasks;
- Test disk drives for failures;
- Secure data destruction (disk wipe);
- Cloning only occupied sectors (quick cloning);
- 1:1 write-protected data acquisition;
- Disk comparison (locate sectors that are different on two drives).

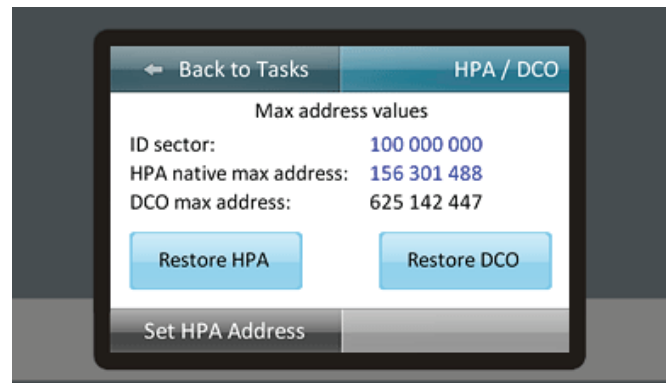


Figure 5. HPA/DCO Setup

Key features:

- Properly handles bad sectors and even severely damaged hard drives;
- Super-fast on non-defect drives: maximum duplication speed is 16 GB/min (265 MB/s);
- Multi-pass imaging (gets most data as quickly as possible on the first pass, then read the rest);
- Ability to stop/resume duplication sessions anytime;
- Disk diagnosis – PCB, head stack, firmware, SMART, media, file system checks;
- Checksum calculation: MD5, SHA1, SHA224, SHA256, SHA384, SHA512;
- Bad sector repair function;
- HPA and DCO max address management;
- Secure disk erasing with custom patterns: maximum speed is 17 GB/min (280 MB/s);
- Easy to use. Features 3.3-inch full color touch screen display.

Atola Technology applies only the latest innovative technologies in designing, and makes all efforts to create market-oriented, highly professional products for data recovery. We deeply value each client and strive to create a great experience in the market and offer the best solutions for everyday use.

Atola Technology. Inspired to be the best in data recovery.

CODENAME: SAMURAI SKILLS COURSE



<< Penetration Test Training Samurai Skills >>

- You will learn Real World Hacking Techniques for Targeting , Attacking , Penetrating your target
- Real Live Targets (Websites , Networks , Servers) and some vmware images
- Course Instructors are Real Ethical Hackers With more than 7
- years Experience in Penetration Testing
- ONE Year Support in Forums and Tickets
- Every Month New Videos (Course Updated Regularly)
- Suitable Course Price for ONE Year Support
- Take Our course at your own pace (any time , any where)
- Our Course is Totally Different from Other Courses (new Techniques)

We have Real World Hacking/Penetration Testing Lab with Over 20 Real Target

Protect your important data before it is too late!



No-one likes to consider the worst case scenario, but are you prepared for a loss of all computer assets?

Computers are easily replaced, but your critical data and files are something money can't buy.

zebNet offers powerful, easy-to-use and leading backup solutions for all major web browsers and email clients which are designed to protect you as much as possible.

With a backup solution from zebNet you will always be protected from the worst case scenario at an affordable price, starting at just \$9.99

Visit www.zebnet.us to be protected!

Highlighted features at a glance:

- Fast and reliable backup and recovery
- Self-restoring backup files
- Backup reserve copies
- Backup to any FTP server
- Scheduled backups on a regular basis
- Data migration between different computers
- Support for portable editions of your web browser/email client
- Create a portable edition of your web browser/email client
- And many more



Exclusive limited-time offer for you as a Hakin9 subscriber:

Get a **50% DISCOUNT** off any zebNet backup solution you wish by simply entering the discount code "**Hakin9**" in our store at www.zebnet.us

zebNet backup solutions are available for Microsoft Outlook, Windows Live Mail, Microsoft Internet Explorer, Mozilla Firefox, Mozilla Thunderbird, Mozilla Seamonkey, Google Chrome, Opera, Apple Safari, Postbox and IncrediMail.

For any questions you may have, please get in touch with us directly at info@zebnet.us or visit www.zebnet.us