Proceedings of The 6th Australian Digital Forensics Conference

1st December 2008.

Edith Cowan University
Mount Lawley Campus
Perth Western Australia

Published By:
SECAU - Security Research Centre
Edith Cowan University
Mount Lawley Campus
Perth Western Australia

Edited By:
Associate Professor Craig Valli and Dr Andrew Woodward
SECAU Security Research Centre
Edith Cowan University
Perth, Western Australia

Copyright 2008. All Rights Reserved
ISBN 978-0-7298-0666-4 Proceedings of the 6th Digital Forensics Conference
ISBN 979-0-7298-0663-3 Proceedings of the 2008 SECAU Security Congress

CRICOS Institution Provider Code 00279B

Conference Foreword

Dear Delegate,

The 6th Australian Digital Forensics Conference has a number of high quality paper submissions from authors who represent a cross-section of international and Australian national perspectives in terms of digital forensics. These papers reflect the emerging themes of data acquisition, malware detection, and network vulnerability. All published papers were double blind peer-reviewed before acceptance into the conference for publication. There were a total of 38 papers submitted for review from which 21 were accepted and presented.

The 6th Australian Digital Forensics Conference is one of four conferences that form the 2008 SECAU Security Congress. The congress reflects an overarching commitment to bringing together a broad spectrum of security topics under a single assembly in order to promote and develop a cross disciplinary approach to a continuum of security research.

Conferences such as the 6th Australian Digital Forensics Conference take a great deal of co-ordination, time and effort in order to bring together the right people in a common forum in order to advance the wider security understanding and to progress the various research directions. To that end, I express my thanks to the conference committee for their hard work and dedication to the conference cause. In particular, I would like to commend the various reviewers, editors and proposal submitters for their devotion and perseverance in the face of countless other duties and engagements. In concert with this gratitude is another vote of thanks to the administrative staff within the School of Computer and Information Science, as well the SECAU Security Research Centre, for their patience, good-humour, and professional approach to ensuring the successful running of the conference

Sincerely,

David Cook
SECAU – Security Research Centre
2008 SECAU Congress Co-ordinator

Conference Committee

Professor William Hutchinson	Committee
Associate Professor Craig Valli (Editor and Chair)	Committee
Dr Andrew Woodward (Editor)	Committee
Dr Patricia Williams	Committee
Roslyn Dinkgreve	Committee
Peter Hannay	Committee
Patryk Szewcyzk	Committee
Marwan Al-Zarouni	Committee
Chris Bolan	Committee
Brett Turner	Committee
David Cook	Committee

Sponsors:
Best Paper Award – Presented by the Australia New Zealand Forensic Science Society
SECAU – Security Research Centre

Table of Contents

iPhone Forensics Methodology and Tools

Haitham AL-Hajri
Krishnun Sansurooah
School of Computer and Information Science
Edith Cowan University
iPhone@e4nzx.com
ksansuro@student.ecu.edu.au

Abstract

iPhone mobile devices are rapidly overtaking the new generation of mobile phones market, especially among the young generation. It is also gaining a lot of popularity among security specialists and fancy gadgets for collectors. The device is considered as a "special" mobile phone due to its ability to perform multi-operations if not multitasking. It can therefore be used as a entertainment media device, a camera, a GPS, Internet surfing via Wi-Fi technology, Internet Mobile Edge Services, personal organizer, and finally performing as a cell phone with all the usual services including sms, and so forth. However, the difference between the iPhone and the other conventional phones vendors is its ability to store and process huge volume of data which is supported by decent computing capabilities of the iPhone processor. As part of every technology, such a device can be used for legal and illegal activities. Therefore the potential risks from such "special" technology are not limited to the possibility of containing illegal materials, such as audios and visuals, including explicit materials, images, documents and the possibility of propagating malicious activities rapidly. Such modification can breach or tamper with the telecommunications network authorities and regulations. The goal of this paper is to focus on both the logical and the physical extraction of the iPhone generation one through the extraction of the iPhone flash drive NAND memory chip and also the logical extraction of data onto the second generation of iPhone using various techniques and methods at our disposal.

Keywords

Forensics methodologies, iPhone forensics, forensics tools, digital forensics evidence handling,

INTRODUCTION

iPhone mobile phone is becoming the most popular devices among the youth age. As iPod has changed the way of listening to music, as well the iPhone is revolutionizing the world of cell phone. iPhone is a result of combining the iPod capability with the functionality of cell phone, which has result in a powerful multimedia mobile phone. Adding the ability to store large volume of data has opened the gates for great potential misuse. Such a large storage device that can transfer stored data on the go can jeopardize serious risk of loss or theft if it contains sensitive cooperate data. In addition it can cause a threat to organization data when the iPhone can be synced to their system and can either download malicious tools or upload some sensitive executable files or documents. The iPhone device can be modified to be powerful hacking devices given its functionalities have been tailored to suit the malicious intention. Therefore the importance of having a sound forensic methodology or approach to acquire and analyze extracted data is essential. The aim of this paper is to have an overview of the tools scripts and forensics approaches that can be applied to the iPhone in case it has been found at a crime scene or been reported as an abuse of technology.

General Forensics Guidelines

Before arriving on any crime scene with any digital devices are involved be either an iPhone or any mobile devices, the following are general guidelines that could be put in practice prior and upon arrival.

Steps Taken Upon Arrival to the Scene

1. Obtaining appropriate approval and authorization letter.
2. Arriving to the scene.
3. Securing the scene.
4. Establish documentation and seizure form.
5. Locating the mobile device.
6. Photograph the crime scene and document everything
7. Use digital cameras or video recorders.
8. Record the names or ID of the examiners on the tape or the photos.
9. Cite the time, day and date, the venue, and also always get a witness to certify the collection of evidence.
10. Check the state of the mobile is it on or off. (iPhone has two buttons the home button, which turns the of sleeping mode and the power button which if it has been hold for more than 5 seconds will display a message if the user wants to turn the iPhone off).
11. Check if the device has password lock enabled, and if it is enabled, it should be obtained before disconnecting the phone.
12. Check the mobile connection. If it is connected to anything. In this case to any computer system that have iTunes (iTunes is a software used for syncing the iPhone to the computer).
13. If the mobile phone is connected to a computer, take photos of connection and the computer screen. Including the syncing process and documenting the software version used.
14. Tagging and bagging. In cases of ordinary mobile devices, it is recommended that the device to be placed in a Radio Frequencies Control bag. So device will need to be isolated from the network coverage and prevent it from receiving any calls or messages therefore it will not tamper with evidences on the devices.

A seizure form is attached in the Appendix 1 to have a better idea about how to fill in a seizure form when an iPhone is part of a crime scene.

FORENSIC METHODS

This section will demonstrate the forensic approaches and the tools required for each steps keeping in mind that these tools are subject to upgrades and might not matching set some phones with different firmware.

Logical Approach

The logical approach will illustrates the software tools that can be used for recovery of data. This section will list some tools and scripts associated with the forensics scene. The iPhone is a very useful tool, but one should be aware of some very important things due to its characteristics. It is considered as a fully-fledged computer, which is running a slimmed version of UNIX operating system and Apple's Leopard. Therefore, like most operating systems, deleting a file is simply just deleting the pointers that point to the physical location of the storage block hence allowing the initial injection of data on the storage disk initial to remain in the block. This method of data extraction will therefore explore all the possibilities while using a logical approach to extract all the data from the iPhone 3G. Obviously, it I very important that everything that has be extracted from the iPhone 3G follow the rules of evidence which are cited below:

(i) Evidence is admissible
(ii) Evidence is authentic
(iii) Evidence is complete
(iv) Evidence is reliable
(v) Evidence be understandable and believable

The free tool that was used is known as '***ibrickr 0.91***' which can be downloaded from www.ibrickr.com. Normally iPhone 3G will grant access to only "genuine" built-in mechanism signed by Apple to be installed. However, the ***ibrickr 0.91*** will use the loop-hole in the firmware and bypass, thus forcing an unsigned application to be installed even though it does not come from apple. Once that the software has been downloaded, one should run the application which is install itself in *C:\ProgramFiles\ibrickr\* once this has been performed, connect the iPhone 3G need to be connected to its dock connector and the other end to the USB port allowing the iPhone to maintain its charge while performing the recovery process. This will then allow a connection to be established between the iPhone and the toolkit.

Having already connected the iPhone 3G to the machine and detected by iTunes, we can start downloading the forensics recovery payload within the 'ibrickr'. The forensics recovery toolkit will comprise of OpenSSH, netcat, just to cite a few of them at this stage.

Once confirmed that the forensics toolkit has been properly installed and activated, the iPhone is ready to be jail broken and to start installing the payload, it could happen that the iPhone get jammed – i.e. the phone appear to be not functioning, then one should manually force it to go into recovery mode by holding the Power and the Home buttons until the iPhone hard powers itself off, powers itself back on, and finally will display the recovery screen. However, with the iPhone 3G with the new firmware version, it seems that the way that the iPhone communicates is different from the previous versions. With the 2.x version of the iPhone, the same approach is used but the way that it is delivered has considerably changed. Therefore, as mentioned previously the aim is still the same – i.e. booting form the iPhone boot ROM to accept unsigned applications.

A tool known as "Pwnage" is use to exploit this weakness which was designed and written by a group of hackers known as iPhone Dev Team. This tool is use to crack the iPhone by modifying the boot loader which in turn will normally is likely to destroy the file system on the iPhone 3G and before restoring the device firmware, we'll use another tool "Xpwn" which is an open source tool designed by David Wang to organize proprietary *img3* formats in which the RAM disk are stored. Once that you have already completed this section, we will need to unpack and repack the RAM disk through 'xpwntool' which is the actual RAM disk management tool. To be able to mount the RAM disk, we will need to unpack it first which will need to acquire the encryption key and the initialization vector.

Now that we have already extracted the RAM disk, the new file will contain HFS file systems that will be mounted as a read-write mode. Having completed the unpacking of the RAM, we will have to repack the decrypted RAM disk back to it correct format to overwrite the old one. This will be done through Xpwn tool downloaded earlier. Now that we have successfully created a custom firmware bundles we need to get the iPhone 3G into Device Failsafe Utility (DFU) mode. Obviously the forensic toolkit and all the changes can be undone after having created the image by using the iTunes 'restore' option to bring your iPhone 3G back to its normal (original firmware) state.

Wi-Fi & SSH Connection

Since a link has been created in between them, we can open a communication through the wireless network onto the iPhone 3G (make sure that the Wi-Fi is turn on). When the iPhone has already joined the wireless network, make a note of the IP address and confirm that the computer can ping the iPhone. Obviously if you don't have any access point (AP) available then you will have to create an ad-hoc network so that the iPhone can be allowed to connect to. Note that when creating the ad-hoc network some iPhone appears to have difficulties in connecting onto encrypted network which can be bypassed by recreating the network but this time without the password. Make sure that the iPhone is joining the wireless network and then select the option that the iPhone be connected through a static address on the network. Once the iPhone is visible and active on the network, we need to open a connection via SSH from the desktop by typing '*ssh –l root x.x.x.x*' where the "x" denotes the IP address of the iPhone which was recorded earlier and when prompted, enter the password which is normally "alpine" which was set initially when installing the forensics recovery toolkit and once that completed this would mean that you have been successful in logging into the iPhone and we should proceed with the recovery of the media partition.

With the recovery toolkit that was installed earlier on the iPhone 3G and the Wi-Fi connection with the desktop machine, we can commence to recover the media partition of the iPhone and for this we will need ***"dd"*** and ***"nc"***. The ***"dd"*** tool is a disk copy tool that allow you to copy the raw disk image where as the ***"nc"*** mostly known as the *"netcat"* tool is use to send and receive data across a network. Note that both command line tools need to be installed on both the iPhone and the desktop to be able to send and retrieve the data.

MD5 Hash

As with any piece of evidence for forensics use, it is very important to be able to verify the integrity of the information and one way to do so with the iPhone is to make an MD5 digest of the iPhone as it will ensure that the partition data has not been tampered during the transit and enforce data integrity.

This shows that the ssh connection to the iPhone and then goes up to the root level where it is then *umount* command unmounts the */private/var* partition by force. Given that there are other application that are using the iPhone's disk, it cannot be unmounted without force hence the option "*–f*" and then the partition is remounted with a read-only option (*ro*) hence backing it with the md5 command to create a digest of the raw device which depending on the capacity of the iPhone, this might be very time consuming to achieve. It is also noted that during this period of time, the iPhone has to be kept alive else it might go into a sleeping mode, thus causing the Wi-Fi connection to drop and hence freezing the whole process. Therefore a good trick to keep the iPhone alive is to run a ping session from the phone while waiting for the entire process to finish. Once this operation is finished, the md5 output will return the md5 Digest of the raw partition which just need to be copied and the stored in a safe location for further reference when you will need to compared the image.

However, the best and fastest way to recover the media partition is to send it over to the desktop PC without encryption. Unfortunately, using Wi-Fi will send encrypted data over the network due to the wireless WEP or WPA in place onto the wireless network that you are using. One way to achieve this is so setup up the command on both side (the iPhone and the desktop) to run the *"netcat"* command requesting in the first instance your desktop to listen to a network port while you the iPhone will be transmitting the data which is the disk image on that particular port. This means that on the desktop, request the *"netcat"* tool will to listen to a local port which when it receive the information, it is passed to the disk copy utility which will convert the data back into the image file. The following syntax used

```
$ nc –L -p 8888 | dd of=./rdisk0s2 bs=4096.
```

Once this is done, the desktop will be listening for incoming data sitting idle waiting to receive the data. Therefore, open an ssh connection to connect the iPhone using the following command

```
( $ ssh –l root x.x.x.x

# /bin/dd if=/dev/rdisk0s2 bs=4096  |  nc z.z.z.z 8888)
```

Then the raw partition will begin to transfer the data over the network and this will normally be time consuming depending of the size of the iPhone used. Note that while this is being carried out, only the iPhone disk storage will be transferred which will result in the capacity being less than what the advertised capacity is. When this is completed compared the md5 hash that was recorded previously from the iPhone to see that they both match hence no alteration has been made to the images. Make sure that when analyzing the iPhone that it is carried out only on a copy of the original disk as some tools have slightly altered the disk image during the examination and analysis. If the operation fails untimely, make sure that the iPhone is sitting on the dock and is charging else it will shut down the wireless connection when the battery enters the sleep mode.

As soon as the raw disk has been transferred and recovered from the iPhone, it can be injected through the numerous commercial forensics toolkits available to be read. However, the forensic investigator has to be very careful as the disk image is saved with an HFS/X image extension – i.e. the fifth generation of HFS which most of the commercial forensics tools such as Encase, FTK, Paraben Seizure Box does not recognize. The only way to be able to read this format with the other forensic toolkits is to change the identifier from HX to H+ denoting an HFS/+ file system. Note that prior to effect any change, documenting all the steps is compulsory so that you may have a trace of what has been done and where has it reached so far.

DATA CARVING

Data carving is the method of removing or extracting a collection of data from a larger data set. These carving techniques often occur during a digital investigation where unallocated file system space is analyzed to extract files. The files are "carved" from the unallocated space using file type-specific header and footer values. File system structures are not used during the process. (Dickerman, 2006).

Therefore to recover deleted files, a data carving tool is needed. The recovered raw partition will show as one huge partition to the desktop machine and contains both live and deleted data. (Zdziarski, 2008). The data carving tool will scan the disk image for traces of desired files, e.g. images and other files which are carved out for further analysis thus funneling the search further down. Such examples of data carving tools are Encase, CarvFs, Foremost, Scalpel. In this report we used Foremost as it is a free forensics toll which was developed by US Air Force of Special Investigations. Scalpel is a tool based on Foremost and handle the challenge that allow most forensic investigators to specify the headers and sometimes the footers which are optional – meaning allow the examiners to specify the beginning and the end of the desired data on the raw image.

Since Foremost and Scalpel uses both a *conf* extension, it is therefore very helpful for the examiners as it can be bundled together and allow the investigator to uncomment certain types of files that need to be carved due to the sample configuration loaded in the those tools.

The following are just examples where the investigator should be looking at – i.e. places of interest on the iPhone are:

- Photos taken by the iPhone camera;
- Photos that have been syncing with a desktop;
- Photos from web browsing;
- Google Maps tile;
- Contacts and their details;
- YouTube last visited to associate with any particular crime;
- Web browser of the visited pages;

Potential location of evidence within iPhone.

(i) **Voicemail messages** – the AMR codec considered as the standard speech codec used by the iPhone to store the voicemail messages which normally sit in nicely in a 65 K but for larger or longer messages extending the file size will allow you to get the entire message. Using the following syntax
amr y 65535 #! AMR

(ii) **Property List** – this is the list where all the XML configurations are held. This means that the examiner can retrace the different websites that the owner of the phone has visited or also what is the different Google Maps direction that he was looking at. Use the following syntax
plist y 4096 <plist </plist

(iii) **Dynamic dictionaries** – this is associated with keyboard caches used by the iPhone. Searching this allow you to see what the user has been entering, such as chat message, email messages, any password, username/ID, Short Messaging Services. Use the following syntax
dat y 16834 DynamicDictionary

(iv) **SQLite databases** – this allows the user to store address books, calendars, Google Maps tile graphics. These databases are normally 'live' on the iPhone, but deleted databases can allow the investigator to trace back what the user was looking at. Use the following syntax
sqlitedb y 5000000 SQLite\x20format

(v) **Electronic mails** – scanning of email headers is a very good way of recovering both live and deleted email. Syntax use is
email y 40960 From:

(vi) **Web pages** – scan all the websites visited and hence uses the following syntax
htm y 50000 <html </html

(vii) **Images** – all the formats that the iPhone allows such as gif, jpeg, png can be scanned by removing the prefix preceding the corresponding lines in the configuration file as shown through the following syntax
png y 40960 \x89PNG

(viii) **PGP blocks** – PGP-encrypted messages are not very useful without the key but often there are unencrypted messages within the same line. The syntax that accompanies this is
txt y 100000 -------BEGIN

(ix) **Other files** – specially*.pdf &.doc/.docx* can be stored locally onto the iPhone when emailing or when browsing the web. The following syntax are recommended;
pdf y 5000000 %PDF- %EOF
doc y 12500000 \xd0\xcf\x11\xe0\xa1\xb1

To carry this task it would normally take in between half an hour if using Scalpel and a few hours compared to using Foremost where the potential data will be recovered to a directory named foremost-output or scalpel-output. The other good things about this tools is that it also generate and audit file known as *audit.txt* within the output file confirming that information been recovered or what seems to be recovered data. Obviously it is at the investigator's discretion to determine whether the data is valid or not.

Validating Images

Recovery tools can sometimes make mistake when generating too much of data meaning that they extract a lot of data that may partially be corrupted or unneeded. Therefore, finding valid images can be time consuming when it has to go through thousands of files. Thus, the introduction of ImageMagick which contains a set of image processing benefits, one being the ability to display information about the images and this could give the examiner an advantage. The identify tool is great when bundled with the ImageMagick for filtering and analysing through the thousands of files generated after the data carving process.

String Dumps

Another way of addressing this issue is to make a string dump, meaning that the strings that were extracted from the raw disk image and be saved and exported and therefore analysed at a later stage. However, the output file will be huge, but it will allow loose text searches for a particular conversation or message or any other data to be found.

Timestamp

Having already gone through the process of data carving which is very practical when it comes to recover files or data that have been intentionally deleted, the disk image can also be mounted as a live partition to allow the forensic investigator to work/access live data on the iPhone. As we all know that when you are to present evidential data in a court of law you need to be precise and very direct. One of the crucial aspects is timestamping to determine when did that action occurred. In the iPhone many of the timestamps are represented in a UNIX timestamp format that needs to be converted to actual time and date. This can be done by either using a UNIX time stamp converter or using a command written in Perl script.

Disk Analysis

Once that the disk image has been transmitted from the iPhone – i.e. – in the HFS/X file system or if you have converted it to HFS+ this can be mounted on a Windows machine for further analysis through some modifications. Since Windows does not like and understand the HFS/X disk image by default, we need to download the HFSExplorer which is an application that will allow the extraction of the HFS+ volume and the Sun Java Virtual Machine and hence upload raw image files as the one that we have dumped from the iPhone earlier. Please make sure that the disk image is the saved as a*.dmg* extension.

IMAGE GEOTAGS

Geotagging is the process of adding geographical identification metadata to various media such as videos, websites, photographs or RSS feeds and is a form of geospatial metadata. This normally consists of latitude and longitude coordinates, though it can also comprise of altitude, bearing, and place names. This process can help users locate a wide variety of location-specific information. Geotagging enabled image search engine, information services, news, websites, or other resources. (Wikipedia, 2008). When it comes to the iPhone, these are images. This can be either enable or disable and in many cases people forget to disable it which when extracted will reveal the exact location of where this photo was taken and that location can help the examiner to arrest a criminal or a pedophile. For that to happen, the forensic examiners need to use ***Exifprobe*** which a camera utility that has the ability to extract image metadata. Therefore if the image is tagged, there will be a GPS latitude and longitude as shown below:

JPEG.APP1.Ifdo.Gps.LatitudeRef	*= 'N'*
JPEG.APP1.Ifdo.Gps.Latitude	*= 31,55.60,0*
JPEG.APP1.Ifdo.Gps.LongitudeRef	*= 'E'*
JPEG.APP1.Ifdo.Gps.LongitudeRef	*= 115,49.60,0*

Added to these extracted details, the timestamps can also be retrieved to indicate at what time was the shot taken as described below:

JPEG.APP1.Ifdo.Exif.DateTimeOriginal *= '2008:10:28 15:47:39'*

JPEG.APP1.Ifdo.Exif.DateTimeDigitized = ‘2008:10:28 15:47:39’

PHYSICAL APPROACH

The physical method will include the possibility of opening the device and to look into extracting data from the memory chip. Therefore the forensic examiner should be familiar with the device memory technology that has been used. In addition the type of memory that might have the evidential data.

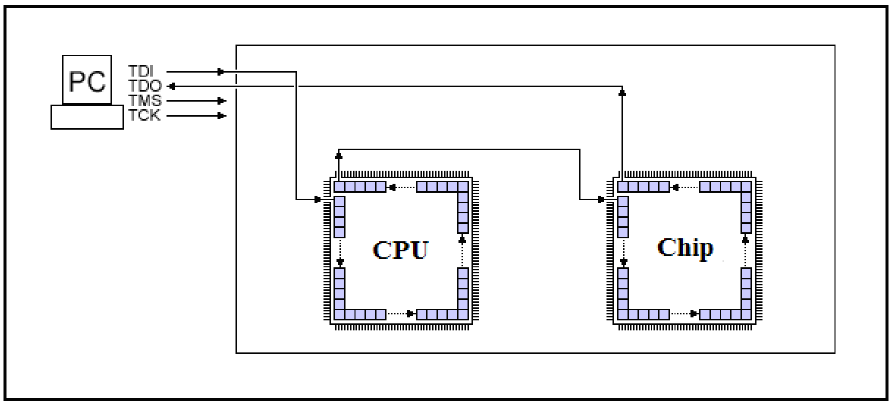
The iPhone has two main memory chips technology installed, NOR and NAND chip. The main differences between NAND and NOR chip is in the way of handling writing and reading of data. NOR has the ability of reading faster than writing which it is a good candidate to handle all the iPhone applications and process, for the reason that NOR has a function called execute in place (XIP) which allows programs that have been stored in NOR flash memory chip to be executed from the chip itself instead of loading the program to the RAM for it to run from there. By doing so NOR will help reduce the need of using RAM resources to execute programs allowing NOR to act as fast random access memory(RAM) with the luxury of having the programs in same location on the flash memory chip all the time. While NAND chip writing capability is faster than reading therefore it is a good candidate to be used for storing media and data. (Calligaro, 2005). The forensic examiner will look into the content of the devices which it's usually stored on the NAND chip. However it is important to know what sort of application could be stored or booted in the NOR chip, having the possibility of stored malicious code or encryption code could be resident in NOR.

JTAG METHOD

JTAG was initially created as a testing solution for issues with circuit boards also known by logic boards. JTAG is a project of a group of European Electronic Companies called Joint Test Action Group (JTAG. The development of printed circuit board has become more popular and complex in both size and functionality (Oshana, 2002). The joint test action group worked on performing a boundary – scan testing in the hardware from the IC level. That specification result in establishing the IEEE 1149.1 standard detailing the specification in how to access the embedded chip for testing purposes, the technique was named after the group initials JTAG. JTAG method can be used to produce a forensically sound image of an embedded memory chip within small digital devices such as cell phone. JTAG has the capability of allowing a direct access to the memory in order to create a memory dump using the debugging mode or the extest mode. The two techniques is being explained in detail on the small scale digital forensics article (Breeuwsma, 2007).

JTAG also referred to as a boundary scan control uses serial protocol for scanning the boards. The five pins are defined as follows:

JTAG Testing Pins



Example of JTAG Testing Pins. Figure 1

TCK/CLOCK: this pin is detected for clock testing. It synchronizes the internal state of machine (device) operation.

TMS/MODE: this pin stands for Test Mode Select Input. It determines the state/mode of the device controlling the test logic by receiving the incoming data. (Davis, 2008)

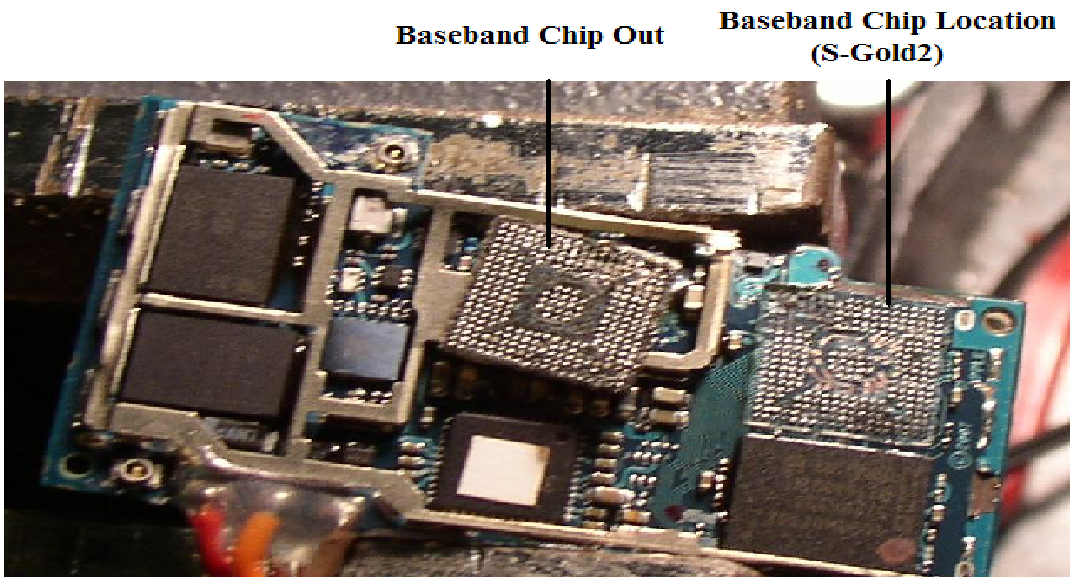
TDI/DATA INPUT: this pin stands for Test Data Input. It receives input of data depending on the state on the tap controller. (Davis, 2008)

TDO/DATA OUTPUT: this pin stands for Test Data Output, where the received signals on the data input will show in the data output depends on the TAP controller sometimes the TDO will be shifted a number of clock cycle depends on the internal length. (Davis, 2008)

Finally the fifth pin that may be used in some boards is the reset pin as follows:

TRST/RESET: this pin resets the JTAG test logic regardless of the state if TMS or TCLK

According to George Hotz also know by (geohot) JTAG pins can be found under the iPhone baseband chip known by (S-Gold2). Once the baseband is chip out of the logic board, its easy to connect the five JTAG pins TDI (Test Data In), TDO (Test Data Out), TCK (Test Clock), TMS (Test Mode Select) and TRST (Test ReSeT). but it requires a lot of soldering and modification to power up the baseband chip (S-Gold2) (Madigan, 2007).



IPHONE logic board with baseband chip detached (Madigan, 2007) figure 3

Such approach has its own advantages and disadvantage the following is a summary of main advantages and disadvantages of JTAG approach:

Physical Extraction Method (Memory Removal)

Physical extraction of the iPhone memory chip is not an easy task to operate on. As shown on figure(x1) the Samsung NAND chip is a 48 leg chip that seats on the edge of the iPhone logic board. it requires a lot of work around and de-soldering technique. However it is a method to be used in cases where other methods fail to obtain data. This method is ideal to be used on extreme cases where the mobile device is damaged severely, that have damaged the logic board; therefore it failed to communicate with its components, to the extent that it can't be turned on. in some cases the device will have destroyed connectors on board that won't allow cables to connect. This method is been used by criminals as an anti-forensics technique to avoid any possibility of extracting data from the seized device. (Keonwoo Kim, 2007).

Memory removal stages

The memory chip has to be extracted from the printed circuit board by de-soldering the chip from the board. However, there are three possible methods on extracting the embedded memory chip in general the following is the three stages in physical extraction approach:

A- De-Soldering of the memory chip.

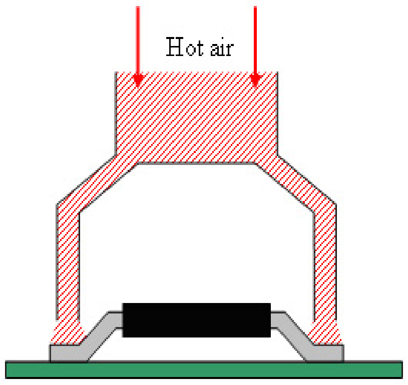
In this stage the chip will be de soldered from the board using special de-soldering equipment because the chips are usually packed in TSOP (Thin Small- Outline Package) or Micro BGA (Ball Grid Array) and this packages require handling with extra care to prevent it from getting damaged in the process of de –soldering. TSOP CHIP can be de-soldered using many ways but the best and most preferred method is using Hot Air. This method relate by applying hot air streams to the board that is hot enough to melt the soldering of the memory chip connection. Then using a vacuum to suck the chip out of the board figure 3 illustrates the method of extracting the chip. Extracting Micro BGA is a complex process using hot air and rework station, by using temperature profile to melt the connection but the issue here is that there are chances that the high temperature will burn the chip. Therefore sensitivity and care should be taken while handling this approach (Breeuwsma, 2007).

B – Preparing the chip for further process.

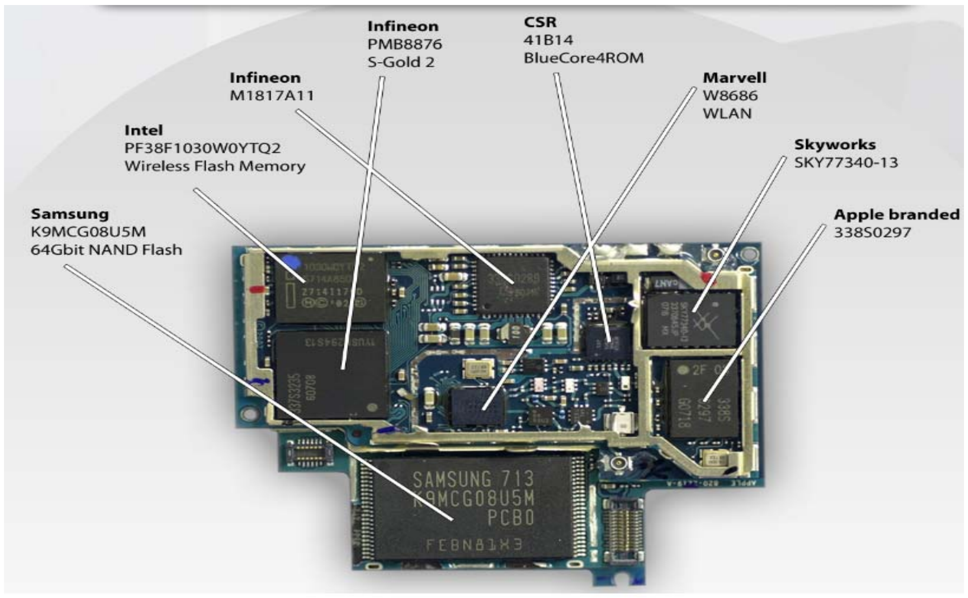
After the memory chip has been extracted from the printed circuit board, the chip needs to be cleaned from the leftover solder in the connection pin. Some pins might need to be aligned and straighten again in order to be plugged in the chip reader/programmer device for further investigation (Breeuwsma, 2007).

C – Flash memory reader/programmer

Few tools have been developed to read memory chip set, these tools are commercially available. One of the famous venders in this trend is the BPM Microsystems who will know by producing programmers/reader tools for different micro chips programs to suite a wide range of different type of memory chip set. Each chip requires a specific software driver to enable the device to communicate with the memory chip set. In case there is no any available driver for the specific chip, it might take longer to develop a suitable software solution in order to read the chip (Microsystems', 2008). There are other commercial tools, which can be used to read from the extracted memory chips and can be used for different memory chip set and they don't require any driver.



Extracting TSOP Using Hot Air Method Figure 3



The iPhone chip processors & internals figure (X1) (mydigitallife, 2008)

REFERENCES

CarvFs (n.d.). The Carve Path Zero-storage Library and file system. Retrieved October 19, 2008, from http://ocfa.sourceforge.net/libcarvpath/

Calligaro, M. (2005) RAM, ROM, NAND, NOR that's a lot of capital letters. Retrieved September 10, 2008 from http://blogs.msdn.com/windowsmobile/archive/2005/08/19/453784.aspx

Davis, L. (2008) JTAG Interface (JTAG Bus Description) Retrieved September 28, 2008, from http://www.interfacebus.com/Design_Connector_JTAG_Bus.html

Foremost (2008). Foremost – Latest version 1.5.4. Retrieved October 17, 2008 from http://foremost.sourceforge.net/

Keonwoo Kim, D. H., Kyoil Chung, and Jae-Cheol Ryou. (2007, December). *Data Acquisition from Cell Phone using Logical.*

Approach. Paper presented at the World Academy of Science, Engineering and Technology.

Madigan, J. (2007) iPhone JTAG Interface discovered. Retrieved September 17, 2008, from http://www.jasonmadigan.com/2007/08/02/iPhone-jtag-interface-discovered

Marcel Breeuwsma, M.D.J., Coert Klaver, Ronald van der Knijff and Mark Roeloffs. (2007) Forensic Data Recovery from Flash memory. Small Scale Digital Device Forensics Journal, Vol 1 (No.1).

Microsystems, B. (2008). Web site information related to services and products. Retrieved September 13, 2008 from http://www.bpmmicro.com/whoweare.htm

Mydigitallife. (2008). Apple iPhone Internal Hardware Components Close Look! My Digital Life. Retrieved August 19, 2008, from http://wwwmydigitallife.info/2007/07/03/ apple-iPhone-internal-hardware-components-close-look.

Oshana, R., (2002) Introduction to JTAG. Retrieved September 23, 2008, from http://www.embedded.com/columns/beginners corner/9900782

Siliconfarest,. (2005), Boundary-Scan Testing / JTAG Standard. Retrieved September 26, 2008, from http://www.siliconfareast.com/jtag.htm

Wikipedia (2008). Geotagging. Retrieve October 21, 2008, from http://en.wikipedia.org/wiki/Geotagging

Appendix 1

SEIZURE FORM

Case Number: (1-25042008) **Date**: (25/04/2008)

Time: 11:45 AM

Location: 145 central street

Suburb: Victoria park

Post Code: (6001) **State**: WA

Other Notes

Investigator Name	**Haitham AL-Hajri**
Investigator ID	**1000 60 20**
Investigator Name	**Krishnun Sansurooah**
Investigator ID	**1000 60 21**

IPhone Type	**[ ] First Generation (2G) [ ] Second Generation (3G)**
IPhone Module	**[ ] 4GB [ ] 8GB [ ] 16GB**
IMIE Serial Number **(located on bottom back of the iPhone)**	**(00002000204)**
password protected	**[ ] Yes [ ] No**
State of the Mobile	**[ ] ON [ ] OFF [ ] Syncing**
SIM CARD	**[ ] Yes [ ] No**
IS the iPhone attached/connected to	**[ ] Power Source** **[ ] Computer /Laptop**
Is the computer Running iTunes software	**[ ] Yes (Version Number : _ _ _ _ _ _ _ _ _ _ _ _)** **[ ] No**
Dose the Location support Wi-Fi facility	**[ ] Yes (Access Point name : _ _ _ _ _ _ _ _ _ _ _ _)** **[ ] No**

Notes:

Name: Haitham AL-Hajri **Signature: Haitham**

Witness: **Signatures: haithy**

Time: 13:25

An overview of Mobile Embedded Memory and Forensics Methodology

Haitham AL-Hajri
School of Computer and Information Science
Edith Cowan University
Memory@e4nzx.com

Abstract

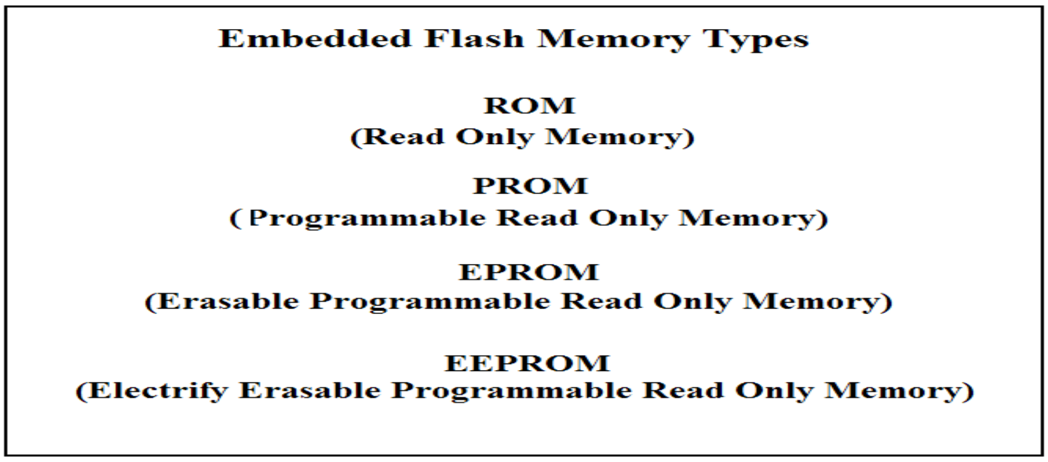
Embedded flash memories are developing very rapidly. On the last few years the revolution of using high capacity embedded flash memory has reached its peak. In the current time embedded flash memory is used in about every aspect of technology starting from small thumb drives to cell phones and cameras. The extraordinary huge capacity in such a small size, permit this flash memory chip to be a very popular aspect to be included in devices that contribute to every moment of our daily life. The aim of this paper is to focus on the types, methods and approaches that can be taken in a forensic investigation of a device that has one of this embedded flash memory technologies and how to acquire data from it. However, the sole focus of this paper will be to investigate the popular types of flash memory chips used on board of cell phones as primary data storage.

Keywords

Mobile Phone Forensics, small scale device forensics, ROM, PROM, EPROM, JTAG, Forensics, Memory extraction, NOR memory, NAND Memory

INTRODUCTION

Embedded flash memory fall under the family of solid state – non-volatile memory, that are being used in thumb drives (USB stick), cell phones, game console , Secure Digital cards (SD cards) and Multi Media Cards (MMC cards). This technology differs from the normal 3.5 and 2.5 hard disk by not containing any moving parts such as arms and cylinders. In addition the physical size of the embedded memory chips makes it a good candidate to be used in every device that interacts with our daily life. The benefits of embedded memory continue to higher life expectancy of the memory chip due to a reduction of mechanical failure even if it had been used in high vibrated or trembling environment (Hamalainen 2007).


Embedded Flash Memory Types

ROM
(Read Only Memory)

PROM
(Programmable Read Only Memory)

EPROM
(Erasable Programmable Read Only Memory)

EEPROM
(Electrify Erasable Programmable Read Only Memory)

Figure 1: Embedded Flash Memory Chips

Read Only Memory (ROM)

ROM chip is a **non alterable** and non-volatile memory that stores and returns information even if it has been powered off. For that reason ROM is being used in devices that require certain information in order to operate and function in digital devices such as computers and cell phones. The ROM chip will accumulate the loaders and the boot up information that is necessary for the device to operate. ROM chips are benign used widely in many devices for its ability of storing and maintaining critical information without the need of a power source that is connected all the time (Linda 2006). Figure two illustrates a closer look of the chip.

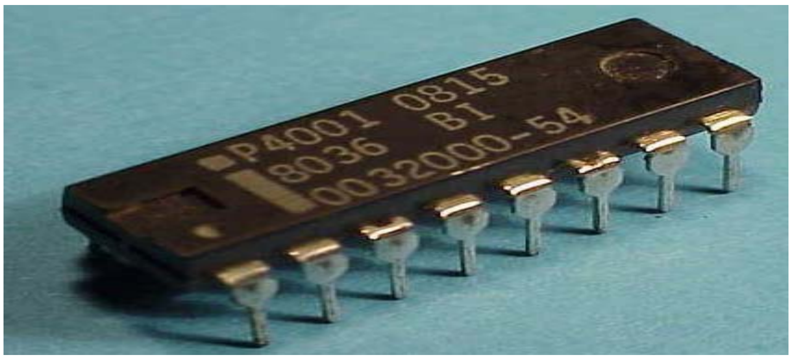


Figure 2: ROM chip

Programmable Read Only Memory (PROM)

PROM is a non-volatile memory chip that can be purchased blank and then the user can load it with his own program using a tool (device programmer) that will allow the user to program the chip. Once it is done, the data cannot be altered or modified. This chip is also known by the name of OTP (One Time Programmable) chip which is an advance technology of ROM. Figure three illustrates a closer look to the chip.

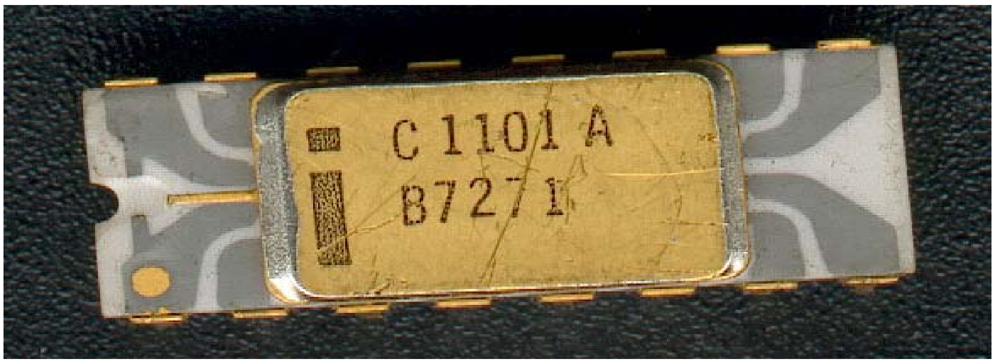


Figure 3: PROM Programmable Read Only Memory

Erasable Programmable Read Only Memory (EPROM)

EPROM chip is alterable and non-volatile as the name suggest, this EPROM chip has the ability to be reprogrammed with different set of data, by simply erasing the previous information using ultra violate light that is strong enough to discharge the electrons on the chip. The following scenario details how electron functions. Figure four illustrates a closer look to the chip.

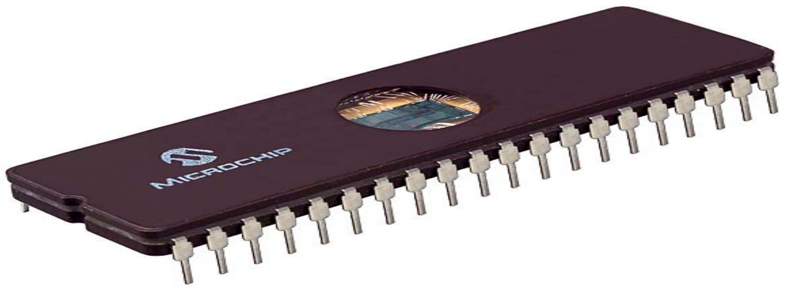


Figure 4: EPROM Erasable Programmable Read Only Memory

Basics of Electron State (Key Function of Electron Process)

The following is an explanation of electron charge movement and representation within the memory chip.

Turning to 1 meaning the erasing of the chip by adding electron charges into the chip

Turning 1 to 0 meaning the writing to the chip by getting rid of electron out of the chip

Turning here is the effect of pushing the electron to gate

We can turn 1 to 0 when the chip is programmed (adding data)

Turning 0 to 1 meaning we have to use the UV light erase the data

We think of a chip as a chamber full of electron charge; this is the empty (blank) state. To add data to the chip (writing to the memory), we have to get rid of the electron charge by pushing it out of the chip, to make room for the new data to be stored. This state is known by writing to the chip. Now to erase the new data we have to bring back the electron charge into the chip by pushing the new data away. In EPROM the only way to do so is by using ultra – violate light that will add electron charge to the chip. Once the chip is full, it will indicate that the chip has been erased (empty). Now the whole chip will be erased because EPROM chip don't support portion erasing of data. It will erase the whole chip at once. (Tyson 2000)

The EPROM chip is made with a window made of quartz therefore a proper ultra – violate light with a suitable frequency should be used. In order to ensure that light will by pass the quartz window and not affect the chip, it has to be in a certain frequency. The ideal frequency of 253.7MHz is perfect to ensure a proper erasing of the EPROM chip. Using a higher frequency will result in damaging the memory chip.

On the other hand, this chip is ideal for loading new information on it and it is also an ideal solution to be used in devices for testing purposes or devices that require frequent updates. So the chip can be updated and loaded with new information when it is required. The chip is built with a silicon window allowing the ability to erase the chip with a proper exposure of UV light beam. This will result in having the chip to return to a programmable state where the chip is then ready to be loaded and installed with new programs and scripts (Allen Kent,; Williams et al. 2002)

Electrify Erasable Programmable Read Only Memory (EEPROM)

EEPROM is an alterable and non - volatile memory chip that is a step ahead of EPROM. The main difference between the EEPROM and EPROM is in the way they handle erasing of data on the chip. EPROM requires ultra violet light beam in order to erase the chip. EEPROM uses an inbuilt electro charge mechanism which helps in erasing bit by bit. This method is best for eliminating the biggest draw back of EPROM but raises anther one which is slowness of operation for the reason that EEPROM erases bit by bit all the time(Mueller 2008) . Figure five illustrates a closer look to the chip.

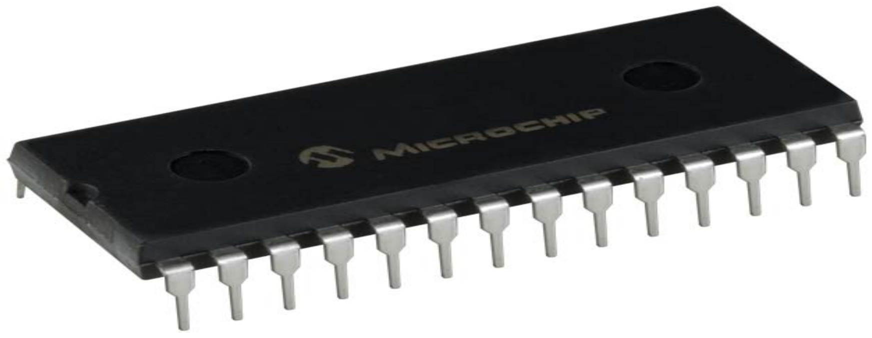


Figure 5: EEPROM Electrify Erasable Programmable Read Only Memory

FLASH MEMORY

Flash memory is the latest and considered as the improved version of EEPROM. Flash memory is a non-volatile memory storage chip, which has all the aspects of EEPROM except it has been improved in the field of erasing mechanism from byte by byte to sector by sector. The overall sector size falls between 256 to 16 KB. Other benefits of flash memory is the physical size and the price or production is cheap(Barr 2001).

Flash memory has two main parts NOR and NAND and they are both considered slower than RAM (Random Access Memory) but both share the advantages of ROM (Read Only Memory) which is the biggest advantage of storing data while the device is been powered off. Therefore, flash memory has become an ideal technology for saving data. The main differences between NAND and NOR technology is in the way of writing and reading data. NOR has the ability of reading faster than writing while in NAND writing is faster than reading.

On the other hand, these technologies could be ideal to use in different environments or a mixture of both NOR and NAND. An example of NOR use as a replacement of RAM, is the ability of restoring data even when the power has been turned off. The rise of using NOR memory as a replacement of RAM has become very demanding. In addition to NOR functionality, NOR has a function called execute in place (XIP) which allows programs that have been stored in NOR flash memory chip to be executed from the chip itself instead of loading the program to the RAM for it to run from there. By doing so NOR will help reduce the need of using RAM resources to execute programs allowing NOR to act as fast random access memory(RAM) with the luxury of having the programs in same location on the flash memory chip all the time. (Calligaro 2005).

Overview of Flash Memory Categories

The following will look in to the types of flash memory technologies from the prospective of their characteristic and use.

NOR TECHOLOGY

NOR is a flash memory type that has different specification than its twin, the NAND flash memory. Both of them fall under ROM ability but differ on its functionality. The following categorizes the two main features of the chip (Toshiba 2006)

Architecture

The internal circuit of NOR memory cell are connected in parallel allowing high reading times ideal for low density and fast process of application which has been offered by RAM but adding the privileges of storing and executing code from the same chip.(Toshiba 2006)

Density

NOR density is very low compared to its ability of reading and executing programs. It is considered as a trade off between the ability and writing data in high speed. For that reason the NOR chip is only available up to 256 MB as it has been stated on Toshiba electronics paper. But the wheel of development is still spinning and it might be available in higher memory capacity in short period of time. However, the point of having NOR chip as a small capacity device compared to NAND is because NOR is Ideal as a replacement of RAM. Also it allows the chip to have an extra storage which executes programs in place. In addition it has the ability to store codes or programs that are frequently modified or updated by the device. Therefore it doesn't require a large capacity of storage space. For the reason that NAND chip will be the primary storage space for all other types of data (Toshiba 2006) .

NAND TECHNOLOGY

NAND is the second part of flash memory technology. Where the advantage of NAND over NOR is on the technique used to stores data. NAND chip can write data in a faster rate than its ability of reading, where the NOR chip advantage is on the ability of performing faster reading rate than NAND chip. A closer look on NAND in the following attributes:

Architecture

The physical architecture of NAND memory chip is smaller in size, compared to the NOR chip. The chip itself is made of an array of eight memory transistors connected in a series. Implementing such architecture will achieve having a smaller size furthermore fast writing and erasing mechanism, allowing NAND chip to be an ideal piece of technology for storage devices. NAND chip is considered as a replacement of hard disk, due to the fact that it is smaller, faster and potentially live longer than an average hard disk. The small size makes the NAND chip ideal to be used in small scale technology that requires a storage device such as phones, PDAs, MP3,cameras, and small electronic devices that operates with memory (Toshiba 2006).

Density

The main advantage of NAND is in the high density that NOR lacks. The architecture of NAND made the capability of reading and erasing data in the chip faster.

Flash memory stores information using four main techniques

The following is illustration of the techniques used by the flash memory on writing data to the chip.

1-Single-Level Cell, **SLC**

Memory card stores one bit in each cell, leading to faster transfer speeds, lower power consumption and higher cell endurance. The only disadvantage of Single-Level Cell is the manufacturing cost per MB. Based on that, the SLC flash technology is used in high-performance memory cards.

2-Multi-Level Cell, **MLC**

Memory card stores three or more bits in each cell. By storing more bits per cell, a Multi-Level Cell memory card will achieve slower transfer speeds, higher power consumption and lower cell endurance than a Single-Level Cell memory card. The advantage of Multi-Level Cell memory card is the lower manufacturing costs. The MLC flash technology is used mostly in standard memory cards.

3-Multi-Bit Cell, **MBC**

Is a similar technology to the Multi-Level Cell but stores only two bits per cell.

4-Chip stacking technology

Chip stacking is used by many manufactures to double the memory card's capacity at considerable lower manufacturing costs. This is achieved by putting two chips together to form a single chip. For example, by stacking two 256 MB chips together they will form a single 512 MB chip. This technology is far less expensive alternative to the single-die chips or even called monolithic chips.

Methods of Retrieving Data from Flash Drives

This section of the paper will examine the forensics tools and methods of retrieving data from memory chips.

Flasher tools

Flasher tools or flasher box is a tool that could be used to modify or upgrade mobile device, flasher tools initially originate from(Knijff 2007):-

- Service centers of mobile phone devices: to allow technicians to upgrade, repair or troubleshoot the device.

Hackers and crackers: intend to modify their device to carry own extra functionality that may not be supported by the manufacturers of the device.

Consequently flashers are used more towards upgrading and modifying the device functionality. In addition it could be used for fast recovery of data and users can retrieve their data from their devices in case the device has failed or got damaged. Flashers are Ideal solution for the users who are not concerned about getting their data on a forensically sound and approved procedure; all they care is to get their data back in any way. Therefore flashers are not the perfect candidate to be used in a proper forensic analysis case. Flasher tools will allow some degree of data acquisition from the devices that has an inbuilt flash memory. Some drawbacks and limitations of the use of flasher tools have alerted that some flasher tools software's will only perform a portion of the device data acquisition. The reason behind it is the nature of the flasher tool software operation. Some flasher software's will not allow a full discovery of the device content, it will only recover whatever the software is designed to do , meaning it will not acquire a full image of what's on the device flash memory. The potential use of flasher in forensically sound manner is considered low , due to some of the following limitations of the flasher (Knijff 2007) :

1- The use of flasher tool on the device may change on the stored content
2- Flasher tools will not acquire everything on device; it will only show what the tool is designed to show.
3- Some flasher tools have a fatal function, which might result in a complete loss of data.
4- The only way to use flasher tool is on the device itself which is not a forensically sound procedure

Joint Test Action Group (JTAG)

JTAG was created by a group of European Electronic Companies called Joint Test Action Group (JTAG) for the purpose of solving the testing problems of printed circuit boards. The development of printed circuit board has become more popular and complex in both size and functionality (Oshana 2002). Testing schema has become very difficult, especially when it deals with mass production of the circuit board. This has actually become an issue. The joint test action group worked on performing a boundary – scan testing in the hardware from the IC level. That specification result in establishing the IEEE 1149.1 standard detailing the specification in how to access the embedded chip for testing purposes, the technique was named after the group initials JTAG.

JTAG also referred to as a boundary scan control uses serial protocol for scanning the boards. JTAG has what its been called TAP (Test Access Port) which has been defined by IEEE 2449.1 as a serial interface port. This utilizes four or five pins on the circuit board to carry on the testing and debagging of the board functionality(SILICONFAREST 2005) . The five pins are illustrated on figure six.

JTAG Testing Pins

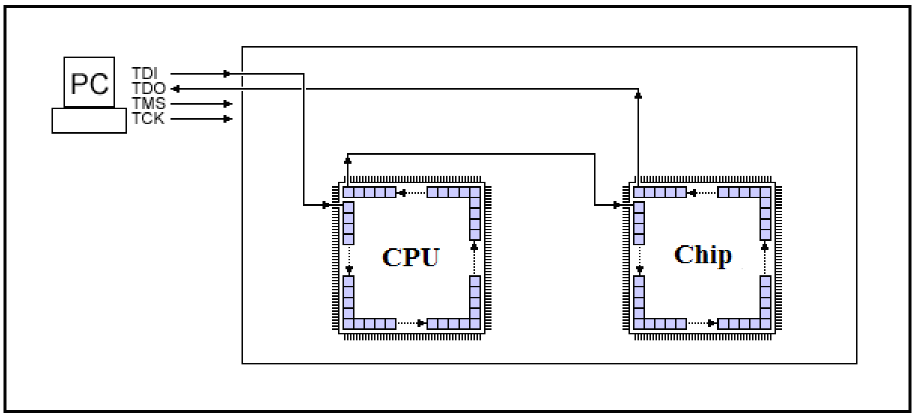


Figure 6: Example of JTAG Testing Pins.

The five pins are:

TCK/CLOCK: this pin is detected for clock testing. It synchronizes the internal state of machine (device) operation.

TMS/MODE: this pin stands for Test Mode Select Input. It determines the state/mode of the device controlling the test logic by receiving the incoming data.(Davis 2008)

TDI/DATA INPUT: this pin stands for Test Data Input. It receives input of data depending on the state on the tap controller.(Davis 2008)

TDO/DATA OUTPUT: this pin stands for Test Data Output, where the received signals on the data input will show in the data output depends on the TAP controller sometimes the TDO will be shifted a number of clock cycle depends on the internal length.(Davis 2008)

Finally the fifth pin that may be used in some boards is the reset pin as follows:

TRST/RESET: this pin resets the JTAG test logic regardless of the state if TMS or TCLK

As it has been understood previously, JTAG method was initially made for testing purposes but a wider range of using JTAG is yet to be discovered. JTAG method could be used to produce a forensically sound image of an embedded memory chip within small digital devices such as cell phone. JTAG has the capability of allowing a direct access to the memory in order to create a memory dump using the debugging mode or the extest mode. The two techniques is being explained in detail on the small scale digital forensics article(Breeuwsma 2007). The paper has details about the fact that it is possible to produce an image using JTAG connection on the processor. Since most of the devices are not JTAG enabled, it is ideal to connect through the processor to allow access to the memory which can be able to create a dump of what resides in the memory chip for further analysis.

The main advantage of using JTAG is:

1- A complete forensic image can be created
2- Tampering with evidence is very minimum
3- Doesn't need de - soldering of the chip

There are few disadvantages of using JTAG:

1- JTAG access pins can be hard to find
2- Not all Devices are JTAG enabled
3- It can get very slow

Physical Extraction

Physical extraction of embedded memory chips is not an easy task to carry on, but it is a method to be used in case other methods fail. This method is ideal to be used on extreme cases where the mobile device is damaged severely, that is, it can't be turned on or be connected using cables due to destruction of the connectors. This method can be used by criminals as an anti-forensics technique to avoid any possibility of extracting data (Kim 2007).

The memory chip has to be extracted from the printed circuit board by de-soldering the chip from the board. However, the process differs depending on the type of the PCB. For this reason some of the boards are thick and some are slim, therefore the de-Soldering process will be different and the chip will need to be prepared for further processing. The chip will be placed on a chip reader/programmer to read the content of the memory chip. To sum the physical extraction process in the following stages:

1- De-Soldering of the memory chip.

In this stage the chip will be de soldered from the board using special de-soldering equipment because the chips are usually packed in TSOP (Thin Small- Outline Package) or Micro BGA (Ball Grid Array) and this packages require handling with extra care to prevent it from getting damaged in the process of de –soldering. TSOP CHIP can be de-soldered using many ways but the best and most preferred method is using Hot Air. This method relate by applying hot air streams to the board that is hot enough to melt the soldering of the memory chip connection. Then using a vacuum to suck the chip out of the board figure seven illustrates the method of extracting the chip. Extracting Micro BGA is a complex process using hot air and rework station, by using temperature profile to melt the connection but the issue here is that there are chances that the high temperature will burn the chip. Therefore sensitivity and care should be taken while handling this approach (Breeuwsma 2007).

2- Preparing the chip for further process.

After the memory chip has been extracted from the printed circuit board, the chip needs to be cleaned from the leftover solder in the connection pin. Some pins might need to be aligned and straighten again in order to be plugged in the chip reader/programmer device for further investigation(Breeuwsma 2007).

3-Flash memory reader/programmer

There are tools commercially available to be used to read the flash memory chip, tools such as the 1600 BP that has been developed by BPM Microsystems. They produce programmers/reader tools for micro chips. The disadvantage of this tool is that it requires a software driver of the chip in order to read it. If there is no driver available for the chip it might take long time to develop a software in order to read the chip. More information regarding the device is available on their website(Microsystems 2008). There are other tools that can be used to read from the memory chip, some are universal and they don't require any driver. It functions by applying different type of protocols to read from the chip. Once it finds the correct response it optimizes that protocol and that saves the hassle of finding drivers for the memory chip to be read (Breeuwsma 2007).

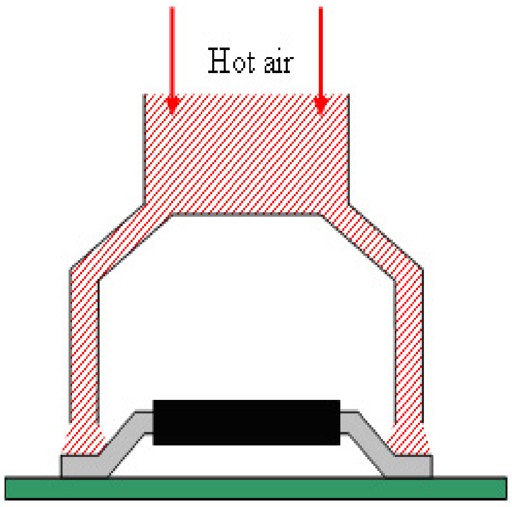


Figure 7: Extracting TSOP Using Hot Air Method

Advantages and disadvantages of using physical extraction method

The following will list some of the advantages and disadvantages of physical extraction method

Advantages

- Data can be retrieved from dead or damaged devices
- It is possible to generate a complete forensic image of the memory chip
- Writing or modifying data on the chip is very low

Disadvantages

- De-soldering the memory chip from circuit board might be difficult
- Expensive method to be used due to the complexity of the tools
- Possibility of damaging the memory chip in the process of extracting the chip

CONCLUSION

This paper has examined various types of embedded memory chips available on the market today. The development wheel of technology is still revolving, and soon there will be more advanced memory chips available. The technology is going towards stack die chips, where the some chip can be mounted on top of each other in a stack. The flash memory technology is all about speed. The requirement of the modern world is high capacity drives with high speed read and write. Some memory technologies are better in term of reading such as the NOR chips, rather other are faster on writing such as NAND chip. Therefore a new breed of combined memory chips technology, been utilized, to operate with the very some device. With the development in term of storage, other issue rises to the surface. The importance's of data retrieval development to be able to retrieve data from such drives is essential. Data recovery tools and methods should develop along with the development of the storage medium Forensics approaches should develop to meet the requirement of the modern world.

REFERENCES

Allen Kent, J. G., R. . Williams, et al. (2002). EPROM. Encyclopedia of microcomputers, CRC Press. **6**.

Barr, M. (2001). "Embedded Systems Memory Types." Retrieved 8 april, 2008, from http://www.netrino.com/Embedded-Systems/How-To/Memory-Types-RAM-ROM-Flash.

Calligaro, M. (2005). "RAM,ROM, NAND, NOR--that's a lot of capital letters." Retrieved 10 april, 2008, from http://blogs.msdn.com/windowsmobile/archive/2005/07/14/438991.aspx.

Davis, L. (2008). "JTAG Interface (JTAG Bus Description)

" Retrieved 28 april, 2008, from http://www.interfacebus.com/Design_Connector_JTAG_Bus.html.

imo D. Hämäläinen, A. D. P., Jarmo Takala, Stamatis Vassiliadis. (2007). "Embedded Computer Systems: Architectures, Modeling, and Simulation." Retrieved 2 April 2008, from http://books.google.com.au/books?id=1dCbChJPZj4C&pg=PA314&dq=what+is+Embedded+Flash+Memory&sig=rWO_zX4VJnn22XQreW5EeFQzJag#PPA314,M1.

Keonwoo Kim, D. H., Kyoil Chung, and Jae-Cheol Ryou (2007). Data Acquisition from Cell Phone using Logical

Approach. WORLD ACADEMY OF SCIENCE, ENGINEERING AND TECHNOLOGY, www.waset.org.

Knijff, R. v. d. (2007). "10 Good Reasons Why You Should Shift Focus to Small Scale Digital Device Forensics." Retrieved 10 april, 2008, from www.dfrws.org/2007/proceedings/vanderknijff_pres.pdf

Linda Null, J. L. (2006). The Essentials of Computer Organization and Architecture Jones & Bartlett

chapter 6 page 282.

Marcel Breeuwsma, M. d. J., Coert Klaver, Ronald van der Knijff and Mark Roeloffs (2007). "Forensic Data Recovery from Flash Memory." SMALL SCALE DIGITAL DEVICE FORENSICS JOURNAL, **VOL. 1**(NO. 1).

Microsystems', B. (2008). "web site inofrmation related to services and products " Retrieved 26 april, 2008, from http://www.bpmmicro.com/whoweare.htm.

Mueller, S. (2008). "Upgrading and Repairing PCs - Google Book Search." Retrieved 9 may 2008, from http://books.google.com.au/books?id=eV1_LjW3pTkC&pg=RA1-PA366&dq=EEPROM&sig=CuVPs5BnuCBWadLt7IyLf5tROi8

Oshana, R. (2002). "Introduction to JTAG." Retrieved 21 april, 2008, from http://www.embedded.com/columns/beginerscorner/9900782.

SILICONFAREST. (2005). "Boundary-Scan Testing / JTAG Standard." Retrieved 28 april, 2008, from http://www.siliconfareast.com/jtag.htm.

Toshiba. (2006). "NAND vs. NOR Flash Memory Technology Overview." Retrieved 10 april, 2008, from www.toshiba.com/taec/components/Generic/Memory_Resources/NANDvsNOR.pdf -.

Tyson, J. (29 August 2000). "How ROM Works." Retrieved 07 April, 2008, from http://computer.howstuffworks.com/rom.htm

Digital forensics and the legal system: A dilemma of our times

James Tetteh Ami-Narh
Edith Cowan University
taminarh@student.ecu.edu.au

Patricia A H Williams
Edith Cowan University,
trish.williams@ecu.edu.au

Abstract

Computers have become an important part of our lives and are becoming fundamental to activities in the home and workplace. Individuals use computer technology to send emails, access banking information, pay taxes, purchase products, surf the internet and so on. Business also use computers and the Internet to perform accounting tasks, manage customer information, store trade secrets, and develop new products and services. State, Federal and Local government agencies use the computer and Internet to create and access information. Similarly, digital systems have become the mainstay of criminal activity. Legal proceedings have always been influenced by tradition and court decisions. These legal traditions and decisions have necessitated the development of complex sets of rules that are used to assess forensic evidence in legal matters. Information and communication technology has impacted enterprise investigation and associated legal matters by requiring electronic evidence to be considered. However, not all evidence presented by digital forensic investigators in legal proceedings has been admissible. The digital forensics investigator must adopt procedures that adhere to the standards of admissibility for evidence in a court of law; proper content inspection of a computer system, proper analysis documentation and professional court representation to ensure a successful outcome. This paper presents an overview of issues in the discipline of digital forensics and explores some areas in the legal system where digital forensics evidence is most likely to be questioned. These include case jurisdiction, search and seizure, spoliation of evidence and issues of "good faith", evidence preservation, investigation and analysis.

Keywords

Digital forensics, legal issues, evidence, forensic process.

INTRODUCTION

The impact of information technology on the world provides limitless benefits for individuals, business, commerce and industry. Unfortunately, as technology develops so does the vulnerability of systems to failure, to unauthorised access and to attack. In the past few years law enforcement agencies have seen an increase in computer related crime including fraud, hacking, equipment misuse, cyber stalking, embezzlement, forgery, harassment, discrimination, sabotage, copyright infringement, security violations, illegal spreading of pornographic materials, theft and virus attacks. The 2005 FBI computer crime survey revealed that 75.1% of the 1762 organisations incurred a financial loss because of computer security incidents. Indeed, the total estimated loses of 313 respondents of the CSI/FBI 2006 survey amounted to $52,494,290 (Gordon et al., 2006). Computer and law enforcement professions have been challenged by the dynamic and evolving nature of computer crime to develop expertise to combat these crimes through the use the collection and analysis digital evidence (Vacca, 2005, p.4).

This paper presents an overview of legal issues in computer forensics. Further, it explores areas within the legal system where digital forensic evidence is most likely to be questioned, which includes case jurisdiction, search and seizure, evidence preservation, investigation and questions relating to analysis.

ISSUES IN EVIDENCE

Biros and Weiser (2006) define digital forensics as “scientific knowledge and methods applied to the identification, collection, preservation, examination, and analysis of information stored or transmitted in binary form in a manner acceptable for application in legal matters”. Digital forensic investigation requires defined procedures that comply with industry practice, organisational practice and appropriate laws, whether as part of a criminal investigation or as part of a more general security incident response. The technique and tools used by forensic investigators may vary, however the process generally includes planning, acquisition, preservation, analysis and reporting as shown in Table 1. Presenting digital evidence is a unique legal challenge facing computer forensic professionals (Kenneally, 2002). Evidence in legal cases is admitted or not admitted based on the relative weight of its probative and prejudicial value (Johnson, 2005, p.150). Given that the legal system is based on precedents, forensic investigators must introduce cohesion and consistency in the expanding field of extracting and examining evidence.

Table 1: Forensic investigation processes (Hershensohn, 2005; Ryder, 2002; Yeager 2006)

Process	***Description***
Identification	Incident is recognised as needing investigation. Triggered by the detection of irregularities in a system, information about a crime and so on.
Search and seizure	Obtain search warrant, prepare tools and techniques. Adopt strategy that maximises the collection of untainted evidence and minimises impact on victim.
Preservation	Involves taking steps to stop or prevent any activity that can damage digital information being collected. Consists of operations such as stopping ongoing deletion processes, preventing people from using computers during collection, using the safest way to collect information.
Examination	Systematic search of evidence about the incident being investigated. Examination of computer media, such as floppy disks, hard disk drives, backup tapes, CD-ROM's and any other media used to store data. Data objects may include timestamps, log files, data files containing specific phrases etc.
Analysis	Evidence analysis is required to identify the perpetrator of crime, claim damages and defend copyrights. Involves determining significance, reconstructing data fragments of data and drawing some conclusions based on the evidence collected. May require the use of tools, and test may also be done more than once to support the crime theory. Technical knowledge required to do undertake an effective analysis

	process.
Reporting	Translating, summarising and providing some conclusions on the analysis of the evidence. Presentation should be in a layperson's language.

JURISDICTION OF CASE

Portability and connectivity of computer systems lead to questions about jurisdiction (Allen, 2005). The legal system in one jurisdiction differs from another in the location of data and information (Wilson, 2008). Digital forensic evidence must meet the formal evidentiary requirements of the courts if it is to be admissible in a court of law in a particular jurisdiction. An act that may constitute a computer crime that is actionable in one country may be acceptable in another (Mohay et al., p.17). For instance in a landmark case, an Australian businessman was awarded the right to sue for defamation in Australia by the Australian High Court over an article published in the United States and posted on the Internet (OUT-LAW.COM News, 2002). The appeal court case Braintech v. Kostiuk is about the jurisdictional right of the Supreme Court of Canada to adjudicate a case involving an alleged wrongful doing by a resident of British Columbia through the use of the internet. The Court held that merely presenting information via the Internet which is accessible to users in foreign jurisdictions does not provide sufficient grounds to allow a court in another country to assert jurisdiction. Therefore, the Texas Court had no right to assert its jurisdiction over a British Columbian resident (Zorzi, 2000).

SEARCH AND SEIZURE OF DIGITAL EVIDENCE

Digital evidence seizure is closely linked to the issue of privacy. Article 12 of the UN Declaration of Human Rights endorses the right of privacy of everyone (CRL, 2006). Therefore, people have the right to be secure in their persons, houses, papers, and effects, against unreasonable searches and seizures. In the court case of United States v. Triumph Capital Group, the government sought and obtained a search warrant to search and seize a laptop computer in a public corporation case to avoid an infringement of privacy and spoliation of evidence (Kroll Ontrack Inc, 2004). This seizure is often a target for protestation in court.

Search warrant

Search and seizure of digital evidence is the first process that is most commonly disputed in court cases. During this initial process of forensic investigation, the use of an improper methodology or unlawful search and seizure can negatively affect the admissibility of the evidence (Rizvi & Misra, 2005). The forensic investigator must therefore ensure that the privacy of a culprit is not infringed in any search. The legal procedure for searching and seizing computers with a warrant largely mirrors the legal framework for other forensic investigations. Notwithstanding the similarities of the framework, digital forensic searches may adopt a non-traditional method which differs from other searches (USDoJ, 2002). In the case *State v. Cook*, the defendant contented that the search warrant used by the police search and seizure of personal property was stale and therefore sought to suppress the use of the evidence. The suppression of motion was overruled by the trial court stating that the warrant for the search and seizure was still valid (Wolff, 2002).

Limits of the warrant

The forensic investigator must not only identify and articulate a probable cause necessary to obtain a search warrant but also recognize the limits of warrants for the search and seizure. In a criminal case of child pornography, Yahoo! technicians retrieved all information from the defendants e-mail account after obtaining a search warrant. A lower court ruled that the seizure of the e-mails by Yahoo! was unlawful because of the absence of police presence during the search of the defendant's e-mail account. However a higher court reversed the lower court's decision. The higher court ruled that the search of the defendant's e-mails without a police officer present was reasonable under the US Fourth Amendment and therefore the defendant's privacy rights were not violated (Kroll Ontrack Inc, 2004).

A further example is Wisconsin v. Schroeder where more than one warrant had to be sought during the forensic investigation process. A search warrant for evidence of online harassment was issued and given to the detective to search and seize the defendant's computer and related items. During the initial search the computer lab examiner found some pornographic images of children. The search process was halted and a second warrant sought to provide authority to search for evidence of child pornographic pictures (Patzakis & Limongelli, 2003). Also in People v. Carratu, the defendant filed a motion to suppress the discovered evidence seized, claiming that the search warrants and supporting affidavits only limited to textual evidence relating to his illegal cable box operation and thus the forensic examiner violated the defendant's Fourth Amendment rights upon inspection of non-textual files with folder names clearly relating to other illegal activity. The court held that the search of the other files violated the Fourth Amendment because the warrant authorized only search for evidence pertaining to the device, and therefore did not authorize a search of image files containing evidence of other criminal activity (Kroll Ontrack Inc, 2004).

PRESERVATION

Evidence preservation is another important aspect of forensic investigation. The dynamic nature of electronically stored information and the routine operation of computers technology can modify or delete information. Therefore, it is critical to address evidence preservation early in an investigation (Alan & McCort, 2007). The forensic investigator preserves forensic evidence to ensure that that evidence is not destroyed or damaged.

Digital evidence can be very fragile, and inherently has several challenges unlike evidence encountered during traditional investigations (Kornblum, 2002):

- Memory resident programs can be lost when the system is shutdown
- Digital evidence can be manipulated during the collection, analysis and presentation of the evidence.
- Digital evidence can be altered without traces
- Digital evidence stored computer systems can be accessed several times
- It is sometimes difficult to attribute a computer activity to an individual, because the digital evidence is circumstantial

The evidence should be protected from virus infection, mechanical and electromechanical influences. The forensic investigator must be able to demonstrate that the evidence was not altered in any way before or during its collection or subsequently. In Weiller v. New York Life Ins. Co., the defendant in a class action lawsuit in New York was ordered to preserve documents. The defendant also asserted that preservation of the computer information would cost a considerable amount of money. However, the New York court determined that the federal preservation orders were not sufficient protection for the plaintiff (Kroll Ontrack Inc, 2006).

Similarly, in a multi-district litigation by Vioxx Products Liability Litigation, the court ordered all parties to preserve evidence relevant to the litigation including hardcopy and softcopy information (Westlaw, 2006). In another civil case, Kucala Enterprises, Ltd. v. Auto Wax Co., Inc., the district court dismissed the plaintiff's case because the plaintiff did not preserve evidence but rather destroyed evidence using Evidence Eliminator. The plaintiff was also ordered to pay the defendant's attorney fees and cost incurred with regard to the sanction (Patzakis, 2008).

SPOLIATION OF EVIDENCE

Individuals and organisations commonly destroy documents during the ordinary course of business activities (Walker & McCurdy, 2002). However, the obligation to preserve electronic data and documents requires good faith and reasonable efforts to retain information which can be relevant pending dispute or perceived litigation. Where relevant information is destroyed, the court may impose sanctions from monetary penalties to default judgement (Rothstein, Hedges & Wiggins, 2007).

One example of this is in the case of Associates International, Inc. v. American Fundware, Inc. The plaintiff complained the defendant deliberately destroyed its computer code after legal action was severed to the defendant. The court held that since the defendant was aware of the dispute it had the obligation to preserve the computer codes. Default judgement was granted in favour of the plaintiff (Walker & McCurdy, 2002). In another example, Mosaid Tech. Inc. v. Samsung Electronics Co., the defendant was sanctioned by the court because it failed to preserve discoverable evidence that was potential for the dispute (Venzie, 2007). Similarly, in the Applied Telematics, Inc. v. Sprint Communication dispute, the defendant failed to preserve the backup tapes of a computer system. The plaintiff argued that the defendant knew the about the potential importance of the information for the lawsuit. The court found out that the defendant did not deliberately destroy the evidence, and court therefore awarded plaintiff monetary sanctions for the destruction of evidence (Schauwecker, 2006).

EXAMINATION

The investigation stage of the forensic process requires a significant amount of planning and consideration to carefully preserve the original. The forensic investigator must ensure evidential integrity of the investigation by imaging an exact copy of all media (servers, floppy disks, hard disk drives, backup tapes, CD-ROM's etc.) using appropriate and usually proprietary imaging software. The collection of electronic data can be a complex task in a digital forensic investigation exercise due to the wide variety of electronic storage locations and the vast amount of data available. Mirror image can preserve the evidentiary value of the information recovered (Benson, 2004).
In the case Gates Rubber Co. v. Bando Chemical Industry, the court awarded sanctions when electronic evidence was destroyed by the defendant's employees. The defendant's expert was criticized by the court for not making an image copy of the drive at issue for production (Kroll Ontrack Inc, 2006). Also in State v. Cook, the defendant appealed against the court ruling for the receipt and possession of child pornography. The defendant claimed that part of the evidence admitted by the court were not were mirror image generated from his hard drive. The court discussed the mirror imaging process, the authenticity of the data taken from the image, and the likelihood for tampering; the appellate court upheld the decision of the trial court. The evidence was properly admitted (Kroll Ontrack Inc, 2006).

Whilst it is possible to fabricate electronic evidence, forensic processes can reveal intentional misrepresentation. In a property dispute, Premier Homes and Land Corp. v. Cheswell, Inc., the plaintiff claim that the defendant was not complying with the terms of a lease. To support its claim the plaintiff offered an email sent from a stockholder of the defendant from plaintiff's president. The defendant alleged the email was fabricated and filed a motion for the preservation and production certain electronic evidence. The court allowed the motion, stating that it was necessary to determine the origin of the disputed e-mail, ordered the defendant's experts to create mirror images of the plaintiff's computer hard drives, backup tapes, and other data storage devices. The plaintiff later admitted that he had fabricated the e-mail by pasting most of a heading from an earlier, legitimate message and altering the subject matter line. The court granted defendant's motion was granted and was awarded attorney and expert fees due to plaintiff's fabrication of email used as evidence (Kroll Ontrack Inc, 2006).

In another case, Commonwealth v. Ellis, the defendants filed a motion to suppress the computerized data obtained pursuant to the search warrants, claiming that the evidence was improperly seized and searched by a computer expert. The motion was denied. To broaden the case, the defendants alleged an infringement of his privacy due to the fact that the forensic expert used a file-by-file search instead of a keyword. The court noted that the expert began the computer investigation with a keyword search but could not properly execute the search because of the file system and structure. The court stated the fraud investigator's method of searching the data file-by-file instead of by a keyword search was logical under the circumstances. Based on the expert's representation, the court denied the defendant's motion, with a few exceptions, determining that the computer search was constitutional and reasonable (Kroll Ontrack Inc, 2006).

A final example of this issue is in the criminal case of United States v. Jackson, where the defendant filed a motion to exclude evidence of chat room conversations with the argument that portions of the transcript were omitted. The under cover police used a cut-and paste approach to capture the evidence. A computer forensics expert confirmed that the cut-and-paste method used by the police officer created several errors and that several portions of the defendant's conversations were omitted. The court decided that the cut-and-paste document was not authentic under the Federal Rules of Evidence and therefore not admissible evidence for trial (Dean, 2007).

EVIDENCE ANALYSIS

The admissibility of findings in a court of law are determined by the rules of evidence, which demand that the accuracy of the methods used to collect the evidence is known and that the evidence is not tampered with in the process of its analysis. (Huebner & Hensksen, 2008). Digital forensics involves essentially taking an autopsy of the forensic evidence using specialized software and techniques to analyse exactly what actions were executed on the computer and what data was stored (Thomas, 2004). Improper analysis of evidence can adversely affect its admissibility in court. The forensic investigator should be able to defend forensic finds in court.

Galaxy Computer Servs., Inc. v. Baker, is an example of a court case where experience of computer forensic expert was challenged. In this legal contention, the defendant argued that the testimony of the computer forensic expert of the plaintiff should be excluded because he was unqualified and had used incorrect procedures. The court rejected the defendant's motion stating that the computer forensic expert good educational background, skill, knowledge and experience (FindLaw, 2005). In Peach v. Bird, the defendant was first acquitted from the charge of possessing child pornography. The analysis of evidence taken from the defendant's computer could not link the defendant to the child pornographic websites. The plaintiff appealed and based on the use Encase evidence analysis and the testimony of an expert the appellant court overturn the dismissal and ordered a case retrial (Patzakis & Limongelli, 2003).

V Cable Inc. v. Budnick is also another example where the police seized computers from a defendant and asked an independent software company to analyse the evidence. The defendant argued that the evidence had been corrupted and therefore inadmissible. However, the court decided that the analysed evidence was trustworthy and admissible (Howell & Cogar, 2003).

CONFRONTING THE DILEMMA

The cases discussed highlight the dilemma that the forensic investigator faces in undertaking computer forensic investigation. Digital evidence has shifted paradigms in collecting, preserving, analysing and presenting forensic evidence. The forensic investigator must ensure that all the processes and procedures in conducting forensic investigation are within the required legal framework. A proposed representation of this is given in Figure 1. The computer forensic investigator has to conform with the rules and regulations of legal systems if the evidence uncovered is to be acceptable to the courts. Forensic investigators must have sound knowledge of legal issues involved in computer forensics investigations. These include the privacy protection rights of employees and other individuals; knowledge about what constitutes a legal search of a stand-alone computer as opposed to a network; laws about obtaining evidence and securing it so that the chain of custody is not compromised; and electronic communications that can be legally intercepted or examined (Wegman, 2004).

Information technology and the boundless environment of the Internet enable cyber-crime activities to span from personal desktops to fibre networks that circle the globe (Radcliff, 2008). Criminals in one country can perpetrate a crime against another person in another country using network servers located in a third country (Kessler, 2005). Digital forensic investigators need to know the legal issues that cross conventional geographic jurisdictions and that are fundamental to pursue digital evidence and wrongdoers.

The courts will uphold the fundamental rights of people against unreasonable search and seizure (Kenneally, 2002). The forensic investigator must ensure that authority for search and seizure of forensic evidence has obtained prior to the investigation. A search warrant must be clear about the searching of network and file servers, and backup media. Also, it must be clearly stated if hardware, software, and peripherals of crime scene can be removed to another location to conduct the search (Whitehead, 2005). In addition, any search warrant obtained should specifically identify the places to be searched coherently as possible, and also limitations imposed within the search warrant on what could and could not be searched (Baggilli, n.d.).

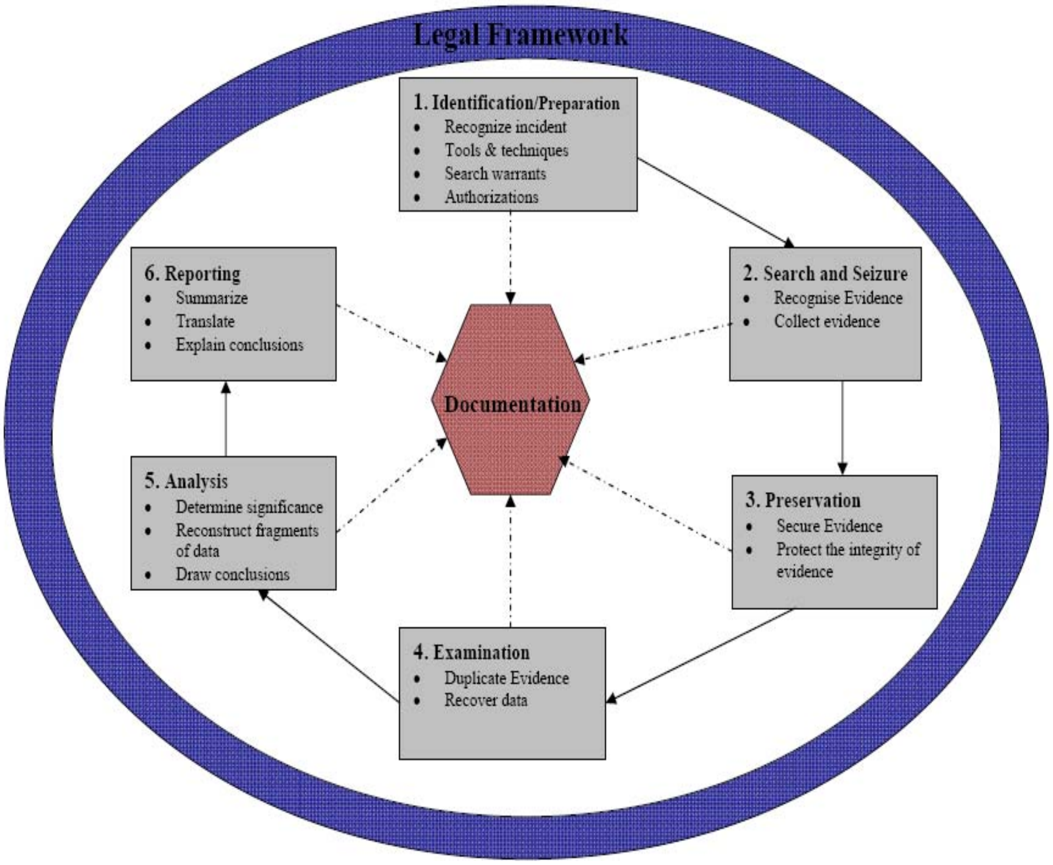


Figure 1: A process Model for seizure and handling of forensic evidence

In order to avoid pitfalls of spoliation of evidence, the forensic investigator should: ensure protection of evidence during investigation, prevent any introduction of computer virus, and properly manage relevant evidence and a continuing chain of custody (Leeds & Marra, 2000).

Forensic investigator can ensure preservation of evidence in accordance to legal requirements by taking the following actions (Thomas, 2004; Hershensohn, 2005):

- Avoid magnetic sources, humidity, excessive heat, or extreme cold, shock , etc
- Evidence removed should be documented and sealed
- Document the entire evidence transportation process
- Handle evidence and electronic equipment properly to avoid damages
- Maintain a chain-of-custody.

After documentation and custody procedures, the forensic investigator must use tools and techniques acceptable to the legal system to examine and analyse the evidence. In addition to having good communication and presentation skills, the forensic investigator should be able to defend the tools and methods used in the forensic process in court.

CONCLUSION

The widespread use of computers and the Internet in homes, businesses, and government facilities has revolutionized access and storage of information. The digital revolution has created the need for new laws, computer forensic investigators, forensic methods, forensic tools and techniques. Computer forensic investigation has become a dominant resource for attorneys and prosecutors in both criminal and civil proceedings. The computer forensic investigator faces the dilemma of conducting forensic investigation and presenting forensic evidence that would be admissible in court. The forensic investigator is expected to be competent in the use of a variety of forensic tools and ensure that every forensic investigation process is conducted within the acceptable legal framework of the court system. The model proposed gives a representation of the forensic processes which should be adhered to. Further development of this model could be undertaken to highlight particular areas of concern for the presentation of evidence in court. The areas identified as the most likely to be questioned in a legal case were case jurisdiction, search and seizure, spoliation of evidence, preservation of evidence, examination and analysis of evidence. In addition, common pitfalls and suggested rectifications for errors in the process could be investigated. There is a growing body of precedents in which objection to digital evidence has been challenged and this confirms the importance of forensic procedures and adherence to them. Models such as that initiated in this paper should be the cornerstone for development of an area which will no doubt become more important in the future.

REFERENCES

2005 FBI Computer Crime Survey. Retrieved September 5, 2008, from http://www.digitalriver.com/v2.0-img/operations/naievigi/site/media/pdf/FBIccs2005.pdf.

Alan, S. M., & McCort, M. (2007). *Maximize the effectiveness of your computer forensic expert and electronic data evidence*. Retrieved October 24, 2008, from http://technology.findlaw.com/articles/01195/010900.html.

Allen, W. (2005). Computer forensics. *Security & Privacy, IEEE, 3*(4), 59-62.

Angela Brungs, & Rodger Jamieson. (2005, Spring). Identification of legal issues for computer forensics. *Information Systems Management*. Retrieved from 26, 2008

Baggili, I. (n.d). *Search and Seizure from a Digital Perspective: A reflection on Kerr's Harvard Law*. Retrieved November 17, 2008, from http://www.forensicfocus.com/search-and-seizure-digital-perspective.

Berghel, H. (2007). Credit card forensics. *Commun. ACM, 50*(12), 11-14.

Carroll, M. D. (2006). Information security: examining and managing the insider threat. In Proceedings of the *3rd Annual Conference on Information Security Curriculum Development* (pp. 156-158). Kennesaw, Georgia: ACM.

Casey, E., & Stellatos, G. J. (2008). The impact of full disk encryption on digital forensics. *SIGOPS Oper. Syst. Rev., 42*(3), 93-98.

CCLSR. (2004). *Corporate law judgments*. Retrieved October 22, 2008, from http://cclsr.law.unimelb.edu.au/judgments/states/federal/2004/may/2004fca562.htm.

Dean, B. (2007). *Knoxville's EDiscovery newsletter*. Retrieved September 13, 2008, from http://www.forensicdiscoveries.com/previousnewsletters/September_EDiscovery_Newsletter.pdf.

Dixon, P. (2005). An overview of computer forensics. *Potentials, IEEE, 24*(5), 7-10.

Eggendorfer, T. (2008). Methods to identify spammers. In Proceedings of the *1st International Conference on Forensic Applications and Techniques in Telecommunications, Information, and Multimedia Workshop* (pp. 1-7). Adelaide, Australia: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

FindLaw. (2005). *Defense expert's notes, Depo Bar Trial Testimony in Merger Dispute*. Retrieved October 24, 2008, from http://news.findlaw.com/andrews/m/ese/20050622/20050622galaxy.html.

Gordon, L. A., Loeb, M. P., Lucyshyn, W., & Richardson, R. (2006) *CSI/FBI Computer Crime and Security Survey*. Retrieved September 5, 2008, from http://i.cmpnet.com/gocsi/db_area/pdfs/fbi/FBI2006.pdf.

Group, T. C. D. E. S. F. W. (2006). Standardizing digital evidence storage. *Commun. ACM, 49*(2), 67-68.

Guha, S., Tang, K., & Francis, P. (2008). NOYB: privacy in online social networks. In Proceedings of the *First Workshop on Online Social Networks* (pp. 49-54). Seattle, WA, USA: ACM.

Herath, A., Herath, S., Samarasinghe, P., & Herath, J. (2005). Computer forensics, information security and law: a case study. In *Systematic Approaches to Digital Forensic Engineering, 2005. First International Workshop* (pp. 135-141).

Hershensohn, J. (2005). *I.T. Forensics: the collection and presentation of digital evidence.* Retrieved October 20, 2008, from http://icsa.cs.up.ac.za/issa/2005/Proceedings/Full/076_Article.pdf.

Howell, R. T., & Cogar, R. N. (2003). *Record retention and destruction: current best practices*. Retrieved October 25, 2008, from http://www.abanet.org/buslaw/newsletter/0021/materials/recordretention.pdf.

Huebner, E., & Henskens, F. (2008). The role of operating systems in computer forensics. *SIGOPS Oper. Syst. Rev., 42*(3), 1-3.

Kenneally, E. (2002). *Computer forensics - beyond the buzzword*. Retrieved November 14, 2008, from http://www.usenix.org/publications/login/2002-08/pdfs/kenneally.pdf.

Kerr, O. S. (2005). *Search warrants in an era of digital evidence*. Retrieved September 8, 2008, from http://www.olemiss.edu/depts/ncjrl/pdf/02-KERR.pdf.

Kessler, G. C. (2005). *The role of computer forensics in law enforcement*. Retrieved November 12, 2008, from http://www.garykessler.net/library/role_of_computer_forensics.html.

Kornblum, J. (2002). *Preservation of Fragile Digital Evidence by First Responders.* Retrieved October 23, 2008, from http://helix.e-fense.com/Docs/Jesse_Kornblum.pdf.

Kroll Ontrack Inc. (2004). *Cyber Crime & Computer Forensics News*. Retrieved September 3, 2008, from http://www.krollontrack.com/newsletters/Cybercrime/oct04.html.

Kroll Ontrack Inc. (2006). *Cyber Crime & Computer Forensics News*. Retrieved September 3, 2008, from http://www.krollontrack.com/newsletters/cybercrime/dec07.html.

Leeds, G. S., & Marra, P. A. (2000). *Discovering and preserving electronic evidence: How to avoid spoliation pitfalls in the computer age.* Retrieved November 16, 2008, from http://www.spsk.com/Articles/artdscov.cfm.

Lingxi Peng, Zhengde Li, Jinquan Zeng, Jian Zhang, Caiming Liu, & ChunLin Liang. (2007). A Computer Forensics Model Based On Danger Theory. *In Intelligent Information Technology Application, Workshop* (pp. 87-90).

McDonald, J. T., Kim, Y. C., & Yasinsac, A. (2008). Software issues in digital forensics. *SIGOPS Oper. Syst. Rev., 42*(3), 29-40.

Michael E Busing, Joshua D Null, & Karen A Forcht. (2005, Winter). Computer forensics: the modern crime fighting tool. *The Journal of Computer Information Systems*. Retrieved from September 23, 2008.

OUT-LAW.COM News. (2002). Australia rules on where to sue for internet defamation. Retrieved September 5, 2008, from http://www.out-law.com/page-3184

Patzakis, J. (2008). *Kucala Enterprises, Ltd. v. Auto Wax Co., Inc.* Retrieved October 25, 2008, from http://www.forensicexams.org/index.php?option=com_content&task=view&id=1079&Itemid=176.

Patzakis, J., & Limongelli, V. (2003). *Evidentiary authentication within the Encase*. Retrieved October 24, 2008, from http://www1.stpt.usf.edu/gkearns/Articles_Fraud/EEEauthentication.pdf.

Peisert, S., Bishop, M., & Marzullo, K. (2008). Computer forensics. *SIGOPS Oper. Syst. Rev., 42*(3), 112-122.

Radcliff, D. (2008). *Computer forensics faces private eye competition - projects security*. Retrieved November 17, 2008, from http://www.baselinemag.com/c/a/Projects-Security/Computer-Forensics-Faces-Private-Eye-Competition/.

Reith, M., Carr, C., & Gunsch, G. (2002). *An examination of digital forensic models*. Retrieved October 26, 2008, from https://www.utica.edu/academic/institutes/ecii/publications/articles/A04A40DC-A6F6-F2C1-98F94F16AF57232D.pdf.

Robbins, J. (2008). *An explanation of computer forensics* . Retrieved October 24, 2008, from http://computerforensics.net/forensics.htm.

Robert J Benson. (2004, November). The Increasing Significance of Computer Forensics in Litigation. *Intellectual Property & Technology Law Journal*. Retrieved from October 26, 2008

Rothstein, B. J., Hedges, R. J., & Wiggins, E. C. (2007). *eldscpkt.pdf (application/pdf Object).* Retrieved October 26, 2008, from http://www.fjc.gov/public/pdf.nsf/lookup/eldscpkt.pdf/$file/eldscpkt.pdf.

Ryder, K. (2002). *Computer Forensics - We've Had an Incident, Who Do We Get to Investigate*?. Retrieved October 20, 2008, from http://www.sans.org/reading_room/whitepapers/incident/652.php.

Schauwecker, P. (2006*). Electronic Discovery and the Environmental Litigator*. Retrieved October 26, 2008, from http://www.bdlaw.com/assets/attachments/154.pdf.

Scott, C. (2003). *Computer Sleuth: Beating down the evidence trail with computer forensics*. Retrieved November 14, 2008, from http://www.thefreelibrary.com/Computer+Sleuth%3a+Beating+down+the+evidence+trail+with+computer...-a099012632.

Srinivasan, S. (2006). Security and Privacy in the Computer Forensics Context. *In Communication Technology, 2006. ICCT '06. International Conference* on (pp. 1-3).

Stahlberg, P., Miklau, G., & Levine, B. N. (2007). Threats to privacy in the forensic analysis of database systems. In Proceedings of the *2007 ACM SIGMOD International Conference on Management of Data* (pp. 91-102). Beijing, China: ACM.

Takahashi, I. (2004). Legal system and computer forensics business. *In Applications and the Internet Workshops, 2004. SAINT 2004 Workshops. 2004 International Symposium* on (pp. 74-77).

Thomas, D. S. (2004). *Legal methods of using computer forensics techniques for computer crime analysis and investigation*. Retrieved September 15, 2008, from http://www.iacis.org/iis/2004_iis/PDFfiles/ThomasForcht.pdf.

Turnbull, B. (2008). The adaptability of electronic evidence acquisition guides for new technologies. In Proceedings of the *1st International Conference on Forensic Applications and Techniques in Telecommunications, Information, and Multimedia Workshop* (pp. 1-6). Adelaide, Australia: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

Venzie, J. C. (2007). *The new era of e-discovery.* Retrieved October 26, 2008, from http://www.cfma-portland.org/portals/0/images/publications/db%20articles/the%20new%20era%20of%20e-discovery.pdf.

Walker, C. (2006). *Computer forensics: bringing the evidence to court.* Retrieved August 23, 2008, from http://www.infosecwriters.com/text_resources/pdf/Computer_Forensics_to_Court.pdf.

Walker, C. A., & McCurdy, M. R. (2002). *The dangers of destroying documents in the normal course of business.* Retrieved October 26, 2008, from http://www.fwlaw.com/documents.html.

Wegman, J. (2004). *Computer forensics: admissibility of evidence in criminal cases.* Retrieved November 12, 2008, from http://www.cbe.uidaho.edu/wegman/Computer%20Forensics%20AA%202004.htm.

WestLaw. (2006). *American Law Reports*. Retrieved September 15, 2008, from http://www.charleshogshead.com/INNSOFCOURT/PROGRAM_MATERIALS/ElectronicDiscovery/ALR_6th_2006_6.pdf.

Whitehead, A. (2005). *Computer forensic: seizing the evidence*. Retrieved November 12, 2008, from http://free-backup.info/computer-forensic-siezing-the-evidence.html.

Wilson, N. (2008). Forensics in cyber-space: the legal challenges. In Proceedings of the *1st International Conference on Forensic Applications and Techniques in Telecommunications, Information, and Multimedia Workshop* (pp. 1-6). Adelaide, Australia: ICST

Wolff, P., & Grady, J. (2002). *State v. Cook, 149 Ohio App.3d 422, 2002-Ohio-4812*. Retrieved October 23, 2008, from http://bulk.resource.org/courts.gov/states/Ohio.Ct.App.02/2002-ohio-4812.pdf.

Xie, M., Yin, H., & Wang, H. (2006). An effective defense against email spam laundering. In Proceedings of the *13th ACM Conference on Computer and Communications Security* (pp. 179-190). Alexandria, Virginia, USA: ACM.

Yasinsac, A., Erbacher, R., Marks, D., Pollitt, M., & Sommer, P. (2003). Computer forensics education. *Security & Privacy, IEEE, 1*(4), 15-23.

Yeager, R. (2006). Criminal computer forensics management. In Proceedings of the *3rd Annual Conference on Information Security Curriculum Development* (pp. 168-174). Kennesaw, Georgia: ACM.

Zorzi, D. (2000, March). *DZ Law -- Internet Law Review.* Retrieved October 22, 2008, from http://www.delzottozorzi.com/internetlawreview3/jurisdictionissue.html .

COPYRIGHT

Validating digital evidence for legal argument

Richard Boddington
Murdoch University
r.boddington@murdoch.edu.au

Valerie Hobbs
Murdoch University
v.hobbs@murdoch.edu.au

Graham Mann
Murdoch University
g.mann@murdoch.edu.au

Abstract

Digital evidence is now common in legal cases, but the understanding of the legal fraternity as to how far conventional ideas of evidence can be extended into the digital domain lags behind. Evidence determines the truth of an issue but its weight is subject to examination and verification through existing forms of legal argument. There is a need for a practical 'roadmap' that can guide the legal practitioner in identifying digital evidence relevant to support a case and in assessing its weight. A vital, but sometimes under estimated stage is that of validating the evidence before evaluating its weight. In this paper we describe a process by which the validation of relevant evidence required for legal argument can be facilitated, by an interrogative approach that ensures the chain of reasoning is sustained.

Keywords

Legal argument, digital evidence, weight of evidence, validation of evidence

INTRODUCTION

In this paper we examine the investigative and legal processes involved in preparing digital evidence for use in legal argument and suggest that evidence taken at face value may be injudicious unless its validity is established before it can be used. Validation requires confidence about inferences drawn from the evidence - can that evidence be relied upon in a legal argument? Validating digital evidence requires verification of relevant parts of the digital domain where the evidence is created, processed and transferred, including the evidence file itself, application and operating programmes and the hardware platform. While techniques of digital forensics aid in preserving and locating potential evidence from a crime scene, the extent to which this may be trusted and used as evidence in a particular legal argument still needs to be determined. We suggest that validation of digital evidence, a difficult task for the investigator, poses an even greater challenge to legal practitioners when constructing legal arguments. Legal practitioners may be unaware of the full nature and significance of digital evidence that is more technically complex compared to conventional forms of evidence.

In the past courts may have been inclined to accept the weight of digital evidence based on expediency and intuition, or if confused by technical issues have dismissed the case out of hand; however, there is the likelihood of increased legal challenges that cast doubt on the weight of the evidence in the future (Ahmad, 2002, Pospesel, Howard, & Rodes, 1997, Schneier, 2000, Whitman, 2005, Tapper, 2004, Whitcomb, 2002). This is evident by the growth in computer-based crime that has increased reliance on digital evidence, both as partial evidence in otherwise conventional legal cases, or where the evidence exists entirely in digital form (Etter, 2001a, Thompson & Berwick, 1998, Palmer, 2001, Cohen, 2006). Digital evidence exists in complex technical environments, unfamiliar territory for most legal practitioners who have difficulty determining how far conventional ideas of evidence can be extended into the digital domain (Etter, 2001b, Losavio, Adams & Rogers, 2006, Caloyannides, 2001, Edwards, 2005).

Evidence used in legal cases proves facts that are in dispute and the weight that may be attached to the facts is examined and tested by various forms of legal argument (Anderson & Twining, 1991, Tapper, 2004). Legal argument can be a complex, convoluted process taking in a broad range of evidentiary issues; technically complex digital evidence used in constructing compelling legal arguments makes the process significantly more challenging for the legal practitioner (Caloyannides, 2001, p. 3, Tapper, 2004, pp. 30-31, Mohay, 2003, Wall & Paroff, 2004, Yasinsac, Erbacher, Marks, Pollitt & Sommer, 2003). Few legal practitioners have sufficient technical expertise to analyse digital evidence in case preparation and is difficult for them to present it in simple comprehensible terms to judges and juries; what may seem a potentially successful case based on straightforward legal argument can turn a into a needless failure (Yasinsac et al, 2003). Moreover, developing legal arguments can be frustrated if unskilled use is made of the digital evidence, with unanticipated and often detrimental outcomes. For example, when presenting a legal case based on what appears to be convincing digital evidence, the case can collapse if the defence can show that the security integrity of the network is defective and shows contamination or alteration of the digital evidence it is supposed to protect. Consequently, if the validity of the evidence can be established its weight in legal argument is enhanced; however if its validity is uncertain or invalidated then weight of the evidence is diminished or negated.

THE INVESTIGATIVE AND LEGAL DOMAINS

Figure 1 highlights the processing of digital evidence in the investigative and legal domains. The investigation domain consists of the stages taken by investigators in evidence preservation, location, selection and validation that precede the stages in the legal domain that involve legal practitioners constructing and then presenting legal arguments. This paper focuses on the validation stage, at the interface between the location and selection of evidence by the investigator and its subsequent use by the legal practitioner. We examine the challenges presented to the legal practitioner on receipt of digital evidence and describe a process by which they may assess the validity of the evidence within the context of their argument.

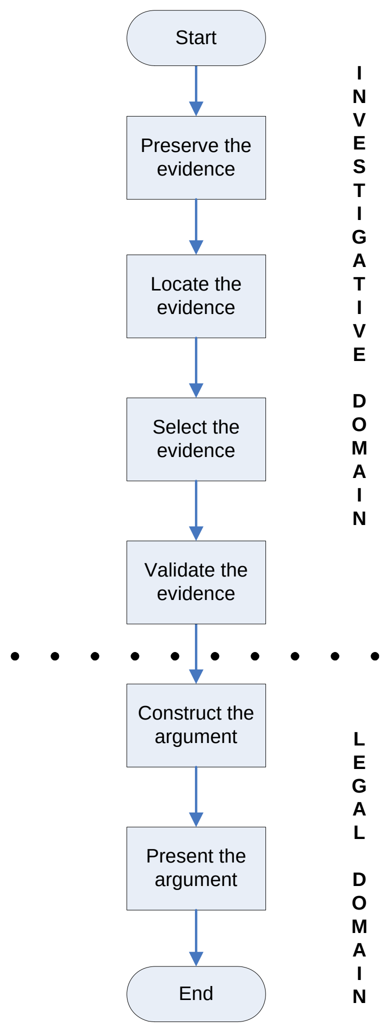


Figure 1. Evidence processing stages in the investigative and legal domains

The stages of the investigation stage shown in Figure 1 commence with the evidence preservation stage that recognises the fragility of digital evidence. Digital evidence can easily be altered, damaged, or destroyed by improper handling or improper examination and so the preservation stage attempts to stabilise and isolate the evidence scene to prevent contamination that damages its admissibility and weight (Ashcroft, 2001, Carrier & Spafford, 2003). The location stage involves locating and identifying the digital evidence for the given class of crime or violation that supports or refutes hypotheses about the incident, using various technical tools and investigative processes to accomplish this (Carrier et al, 2003). During the evidence selection stage the investigator scrutinises the evidence to determine what events occurred in the system and their significance and probative value to the case (Carrier et al, 2003).

During the validation stage the evidence is tested to determine its validity, namely if the assertion drawn from the digital evidence can be verified. For example, the assertion that an email message was deleted would require confirmation of the existence of the deleted file; that it was deleted at a specific time; that this information was not altered by system processes; and so forth. Whatever security measures exist on the host computer they are not always helpful to the investigator as they are more often intended for auditing and monitoring of the overall integrity of records rather than for specifically validating digital evidence (Carrier, 2005a). During the validation stage the investigator may revisit the location and selection stages to seek verification of validity issues and to develop new lines of investigation as circumstances dictate (Carrier et al, 2003).

Inordinate amounts of time and resources are required to collect and analyse digital evidence and the sheer volume of the cases and the time required to process them can have a negative effect on the capacity of investigators - and later legal practitioners - to analyse and present a complete reconstruction of the evidence (Ó Ciardhuáin, 2004). Failure to locate all available digital evidence occurs because the location of relevant evidence is not always evident to the untrained enquirer who may be relying solely on intuition (Cohen, 2006). While a technically astute and assiduous investigator can identify and analyse much relevant evidence, time constraints and the uniqueness of the crime scene may nevertheless produce incomplete identification of all that should be located, consequently denying examination and analysis of crucial facts (unamed, 2000). Incomplete scrutiny of the available evidence during the validation stage of the investigative process and failure to validate the evidence at that point is where the investigation can fail (Cohen, 2006). Carrier (2005a) points out that whatever security measures are used, they are more often used to assist in the auditing and monitoring of the overall integrity of records rather than directly evaluating the evidentiary integrity of digital information.

False evidence too can be generated upon which unreliable arguments are propounded by those unfamiliar with the true nature of the digital domain (Koehler & Thompson, 2006, Diaconis, 1989). Koehler et al (2006) caution against endeavours to locate circumstantial evidence that seem to support reasonable and compelling argument may well be unreliable because they are purely coincidental and nothing more. Moreover, investigators may miss evidence and worse still, resort to 'cherry-picking' when choosing or omitting evidence to gain legal advantage: the absence of evidence does not necessarily show evidence of absence - a common phenomenon of the digital domain (Koehler et al, 2006, Berk, 1983, Flusche, 2001).

There is error in every analysis method and the reliability of any particular test remains an issue for forensic investigators (Palmer, 2002, Cohen, 2006). A range of different factors can affect the validity of the evidence, including collection tools missing, failure to report exculpatory data, evidence taken out of context and misinterpreted, misleading or false evidence, failure to identify relevant evidence, system and application processing errors, and so forth (Palmer, 2002, Cohen, 2006). Because of the complexity of the digital domain prosecution cases often fail during trial where incompetency is apparent in reconstructing the case and where validation issues are raised (Cohen, 2006). The evidence collated and processed during the investigative stages is then presented to the legal practitioner who must test each piece of evidence to determine its weight in the legal argument and its suitability for use to prove or disprove the case (Ashley & Rissland, 1985, Perelman and Olbrechts-Tyteca, 1969). A more explicitly defined and repeatable process would be useful for the legal practitioner who may then have more confidence in the evidence derived during the validation stage.

Research to date has focussed on providing investigators with the means to preserve, locate and select digital evidence (Daum & Lucks, 2005, Lenstraand & de Weger, 2005, Schneier, 2004). For the legal practitioner, the research has attempted to enhance analysis of the weight of evidence as part of structuring legal arguments, but with limited adoption of such processes (Tillers, 2005). Computer and network security and digital forensics research provides documentation about the properties of digital evidence but it does not explain in a legal context helpful to the legal practitioner (Spenceley, 2003, Mohay, 2003). The validation of the evidence, however, is largely dependant on the skill and knowledge of investigators.

There is some ongoing legal debate calling for a replacement of conventional forensic identification science that relies on untested assumptions and intuition - including digital forensics - with sounder scientific analysis (Saks & Koehler, 2005, Tobin & Thompson, 2006, Mohay, 2003). Most writings on the examination and analysis of digital evidence focus on the preservation of evidence and the chain of evidence, with scant mention of the properties of the evidence itself, which may reflect the comparatively recent emergence of digital evidence and cyber forensics (Slade, 2004, Mohay, 2003). Compounding this deficiency is the inefficacy of conventional security processes to preserve digital evidence, for that is not their intended role (Caloyannides, 2001, Rowlingson, 2004, p. 2). Such processes are more often used as forensic tools to investigate a compromise of record integrity or as part of data recovery processes but do go some way towards identifying the evidence and reconstructing a timeline of events (Carrier, 2005a, Egan & Mather, 2004). However, a lack of recognition and acknowledgement by designers, owners and custodians of digital information as to its potential evidentiary importance means that computer system designs fall short of protecting evidence, sometimes preventing it being used as key exhibits (Rowlingson, 2004, p. 2).

Figure 2 lists broad areas of research and practice reported in the literature, highlighting the deficit in the area of validating digital evidence.

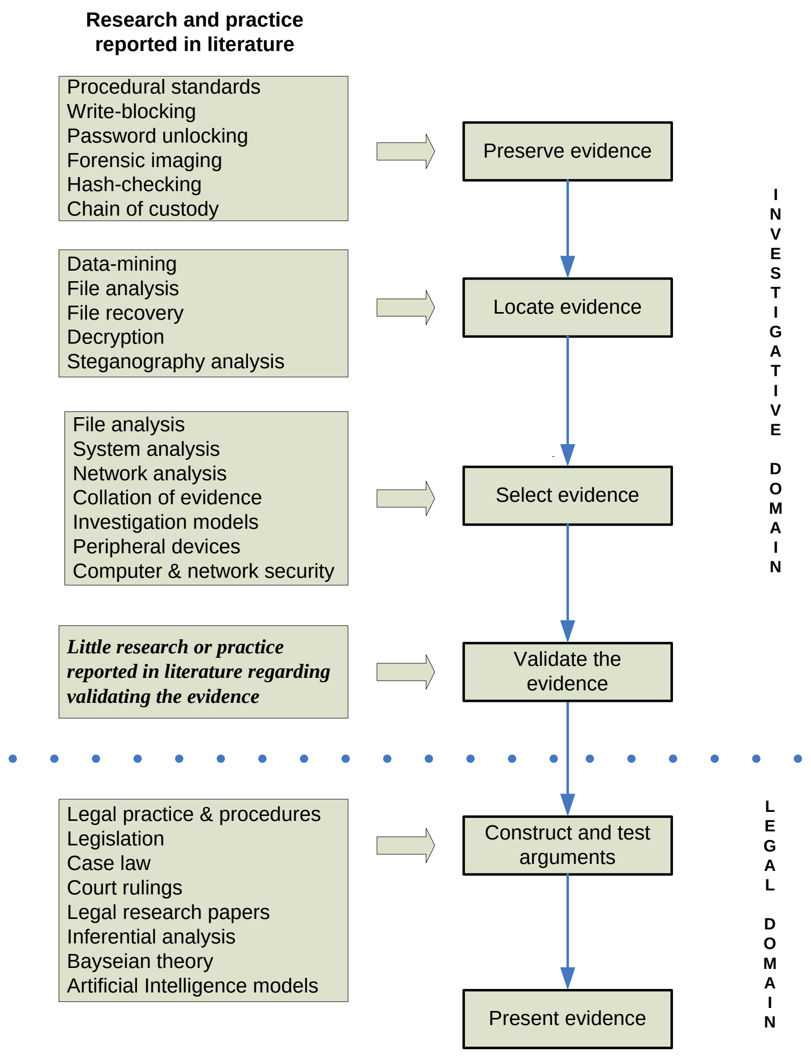


Figure 2. The gap in the research and practice reported in the literature.

Research and practice in the new field of cyber forensics is late in offering practical pedagogical models, leaving investigators to rely solely on their investigative skills and technical knowledge (Yasinsac at al, 2003). For the legal practitioner without firsthand technical skills, some form of practical 'roadmap' is needed that can prompt him or her to identify all pertinent digital evidence relevant to a case and help assess the weight of the evidence more effectively. Research attempts to help legal practitioners through such processes as computer-assisted analysis of legal arguments based on evidence reconstruction, and theories including probability theory and inferential analysis (Silverstone & Sheetz, 2007, Huygen, 2002). To date, however, attempts to present the inference processes in diagrammatic form has tended to confuse the legal practitioner rather than promoting a better understanding of the dynamics of digital evidence (Tillers, 2005).

In criminal cases the prosecution, usually a law enforcement agency, has the advantage of government investigators with experience and resources; resources not always available or affordable to the defence: the defence team relies on an outline of the prosecution's case and forensic evidence images so that it can prepare a defence (Mercuri, 2005). Even with some technical help, the defence team may have little understanding of the properties of digital evidence and may not have clear understanding of the relationship between key digital evidence and potential corroborating evidence that could be used to its advantage in developing legal counter-arguments. In civil cases, there may be a more equitable allocation of technical expertise; nonetheless, the legal practitioners would still need additional help with the complexity of digital evidence. An experienced investigator would be unwise to 'second guess' the legal practitioner but may have difficulty explaining the significance and relationship of the various pieces of evidence. The legal practitioner may take the evidence at face value but suffer its eventual overturn by a more technically astute legal opponent. It is during this validation stage that we believe the legal practitioner needs some prompting to minimise this risk.

In this paper, we are concerned specifically with the validation of the evidence, rather than the subsequent process of interpretation. We describe a process of methodical interrogation of the digital evidence that establishes whether it is valid and therefore suitable for use in legal arguments. The process we propose also offers relevant prompts guiding the legal practitioner to supplementary evidence that may corroborate, negate or offer alternative hypotheses about the validity of the evidence.

THE NATURE OF EVIDENCE AND LEGAL ARGUMENT

Evidence used in legal cases may consist of witness testimony, hearsay, documents and things, and proves facts that are in dispute through directly proving the ultimate fact without relying on other evidence to prove any intervening, penultimate steps (Anderson & Twining, 1991, Tapper, 2004). Evidence is also used to prove the plausibility of facts from which facts that are being disputed, may be understood - most notably, circumstantial evidence (Tapper, 2004). Digital evidence shares many common features with conventional forms of evidence yet it is its technical properties that tend to confound the legal practitioner.

The nature of digital evidence

Although electronic evidence is defined as information of investigative value relating to a broad range of devices and data formats (Ashcroft, 2001), a formal legal definition of digital evidence is elusive, but is generally accepted to be information held in digital form that has some probative value (Carrier et al, 2005b, Pollitt, 2001). Digital evidence typical sought in legal cases includes system logs, audit logs, application logs, network management logs, network traffic capture, and file system data (Sommer, 1998).

Digital evidence is often considered superior to conventional paper evidence being easier to locate and process, and also contains useful data containing details of key dates, times and a history of the file, and, because of its persistency in recording key data, can provide evidence that a defendant may prefer not to exist (Caloyannides, 2001, Janes, 2000). Digital evidence tends to provide metadata about itself prior to the fact, more so than paper-based evidence, and this can provide valuable information relating to a crime such as linking a defendant to an offence and showing evidence of intent, ability and opportunity leading up to the commission of the crime (Janes, 2000, Flusche, 2001).

Several authors contend that digital evidence is not fundamentally different from conventional forms of evidence but is problematic because of its volatility, the complexity of the digital domain, large datasets, and rapid changes to technology that require current technical understanding that is certainly beyond the capability of most legal practitioners (Sommer, 2000, Mercuri, 2005).

Other authors point out that there seems little difference between digital evidence and physical evidence as both forms are required to establish the commission of an offence and link the crime and the victim, or provide a link between the offence and the perpetrator (Carrier et al, 2003, Saferstein, 2000). Similarly, in the digital domain there is much merit in using conventional, crime scene investigation techniques; again showing a degree of fundamental similarity between the domains. Carrier et al (2003) provide helpful definitions to put this in better perspective by suggesting that the computer, computer hardware and peripherals are physical evidence, while the data in memory held in these devices is digital evidence.

Circumstantial evidence, which includes digital evidence, is used to construct inferences that indirectly prove the ultimate fact in a legal case (Anderson, 1991) but before it may be admitted or given any credence in legal cases, it must meet additional legal conditions and conform to courtroom conventions (Caloyannides, 2001, p. 3, Tapper, 2004, pp. 30-31). Circumstantial evidence is probabilistic in nature, often challenging and confounding observers attempting to determine the truth of an issue because the examination processes used are poorly defined (Fiske & Taylor, 1991, Nisbett, Krantz, Jepson & Kunda, 1983, Nisbett & Ross., 1980). Digital evidence is analogous to the more conventional forms of circumstantial evidence, most notably documentary evidence, and both forms are subject to the same degree of legal scrutiny afforded to direct evidence tendered by a human witness (Caloyannides, 2001, p. 3, Tapper, 2004, pp. 30-31).

Inherent differences between digital and conventional evidence exist as digital evidence is more easily altered than conventional forms and such manipulation is sometimes not evident or even possible to detect (Caloyannides, 2001). Digital evidence is mutable – it may be altered far more easily than physical records – and consequently is more susceptible to unauthorised manipulation, making it problematic to validate its admissibility and weight (Schneier, 2000, Mattord & Whitman, 2004, Akester, 2004).

Inaccuracies in attribution of authorship and the content of digital evidence occur frequently and affect legal argument as to the completeness, correctness, validity and faithfulness to an original source, thereby raising doubts as to the worth of the evidence (Akester, 2004, p. 436). More disturbing is that even in the absence of any obvious irregularity of the software platforms examination of any material of evidentiary value does not in itself attest to the accuracy or integrity of the evidence (Spenceley, 2003, pp. 130-131). The more pessimistic argue that it cannot be assumed that there is a low risk of inaccuracy in computer output due to application failures (Spenceley, 2003, pp. 130-131).

Representing legal argument

Legal argument relies on evidence that proves or disproves a case; based on the available evidence the defendant is guilty or innocent of a crime. Legal practitioners use logical chains of inferences linking one piece of evidence to another with the strength of each inference used to determine the weight of a case (Silverstone et al, 2007). The persuasiveness that flows from the combined evidence presented in a legal case is used to enable adjudicators and juries establish proof of guilt or innocence of the accused party (Silverstone et al, 2007). Further discussion of legal argument and its forms is beyond the scope of this paper.

Legal arguments are based on logical probabilities that collectively prove the case and are constructed from the simplest logic possible and may be mapped, for example by a timeline of reconstructed events, or through inferential analysis processes (Silverstone et al, 2007). The use of such processes displayed in graphic form makes the evidence collected more readily comprehensible with relevant evidence arranged as coherent discreet lines leading to the ultimate probandum.

It is unusual for legal cases to rely solely on circumstantial evidence; direct evidence such as witness testimony may corroborate, refute or obfuscate whether the defendant accessed the computer, etc. Therefore, an inferential analysis should include wherever possible evidence of direct evidence, physical evidence such as a fingerprint linking the defendant to the computer as well as the digital evidence. Locating this supplementary evidence, often intuition-based, helps develop argument and strengthen the overall weigh of available evidence. The weight of the evidence depends on the various relationships between penultimate probanda and the reliability of each probanda (Silverstone et al, 2007).

Figure 3 shows a simple chain of evidence based on apparent or available evidence consisting of unprocessed facts from which tentative legal argument can be constructed.

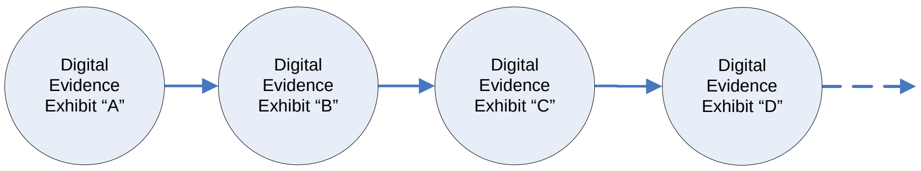

Digital Evidence Exhibit "A"
Digital Evidence Exhibit "B"
Digital Evidence Exhibit "C"
Digital Evidence Exhibit "D"

Figure 3. Chain of Evidence: Before validation of the evidence

This amount of preliminary evidence is readily comprehensible to legal practitioners but is most likely incomplete. While experienced investigators may identify the less than obvious leads or seek expert advice where their technical expertise fails, explaining the complexity of the digital evidence located to the legal practitioner may be difficult. If the investigator is diligent, has sufficient technical and investigative expertise and skills, and is dedicated to seeking all relevant evidence then the legal practitioner will be well served. But the legal practitioner must be able to determine whether enough evidence has been located and whether the validity of the digital of evidence has been satisfactorily described and determined.

A PROCESS FOR VALIDATION OF DIGITAL EVIDENCE

The interrogation process

Figure 4 outlines the basic validation interrogation process where exhibit B, taken from the chain of evidence example in Figure 3 requires validation.

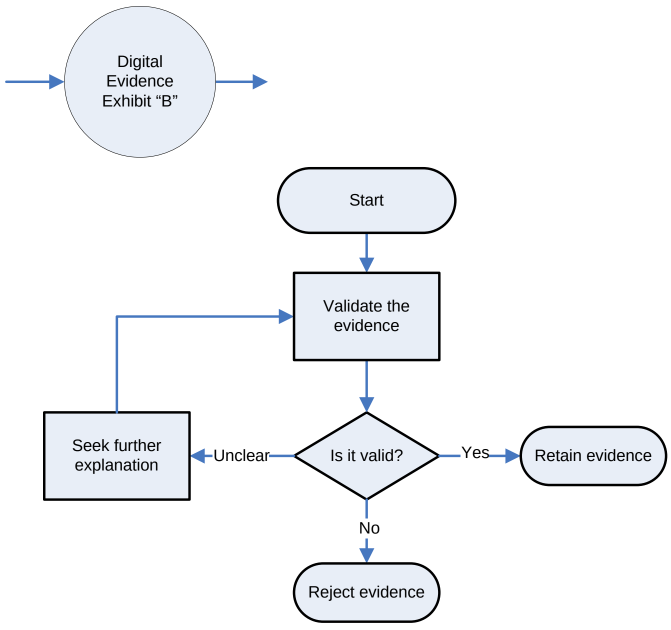


Figure 4. Chain of Evidence: Showing the validation process of digital evidence exhibit "B".

A series of prompts determines if the evidence is valid. Each prompt requires a response of ‘yes’ (the evidence can be considered valid), ‘no’ (the evidence is invalid) or ‘unclear’, (suggesting that further explanation should be considered). The “yes” and “no” outcomes are considered definitive, the “yes” indicating that the evidence can be retained and a “no” indicating that the evidence should be rejected. An “unclear” result is inconclusive and requires a further explanation to provide clarification to establish if the evidence is valid. Where further explanation continues to be inconclusive, a decision to terminate the process is required and at that point, the expertise of the legal practitioner will be required to retain or to reject the evidence based on the available validation evidence.

Each piece of digital evidence considered for inclusion in legal argument is judged on the weight of inference of at least one assertion used, for example, whether the existence of the deleted email file does infer the view that there was an attempt to conceal evidence. In other words, an assertion claims deliberate deletion of the email file with intent to remove all trace of the evidence. As evidence-based assertions are contestable, it is critical to establish their validity.

Figure 5 shows a graphic decomposition of assertions provided by a digital evidence exhibit and the systematic process required to determine its validity. Each assertion is evaluated to determine whether it is confirmed or negated by other available evidence. Each circle in Figure 5 represents an assertion underpinning the evidence, for instance “the defendant accessed a file on a computer”. The primary assertion 1 requires confirmation or negation, provided by the secondary assertions 2 and 3. For example “the deleted email file existed” requires validation. The file metadata may or may not confirm the assertion and the assertion the metadata provides may itself need further confirmation by other assertions and so forth until the interrogation is considered sufficiently strong to support the primary assertion.

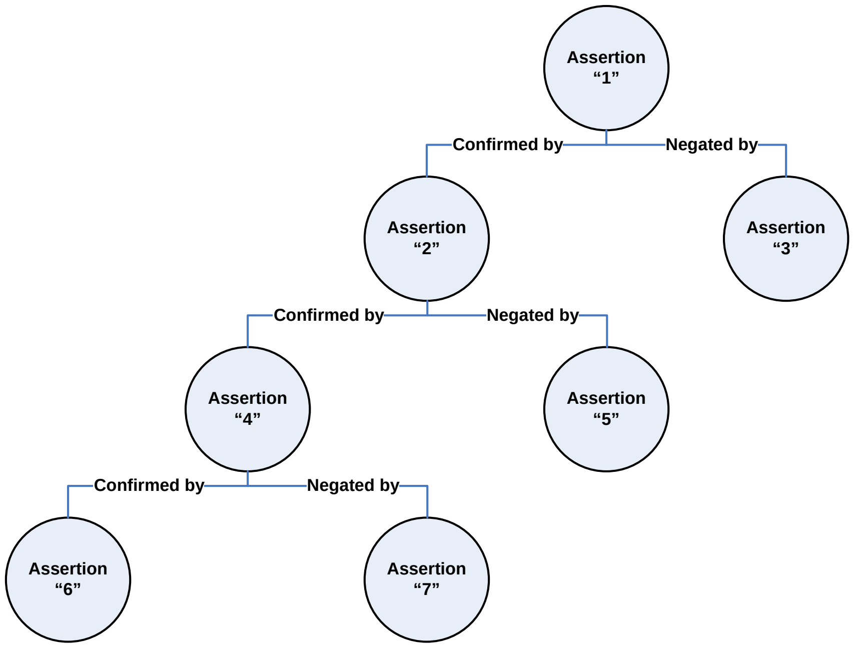


Figure 5. Decomposition of the evidence through validation process

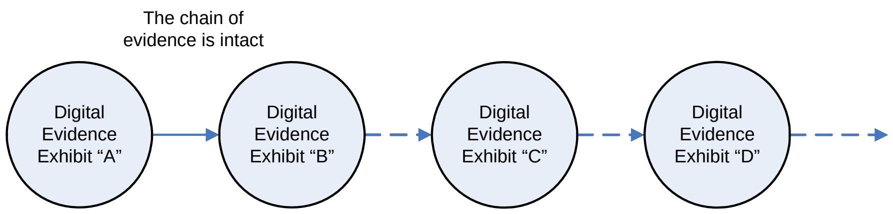

The chain of evidence is intact
Digital Evidence Exhibit "A"
Digital Evidence Exhibit "B"
Digital Evidence Exhibit "C"
Digital Evidence Exhibit "D"

Figure 6 shows an example of a chain of evidence where the evidence has been validated.

Figure 6. Chain of evidence: After validation of digital evidence exhibit "B" is achieved.

Conversely, the primary assertion may be negated from the outset or at a later point, breaking the chain of evidence as shown in Figure 7.

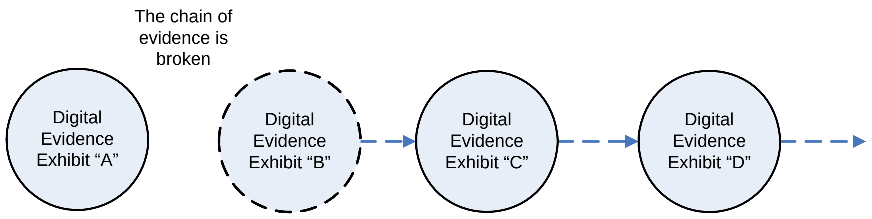

The chain of evidence is broken
Digital Evidence Exhibit "A"
Digital Evidence Exhibit "B"
Digital Evidence Exhibit "C"
Digital Evidence Exhibit "D"

Figure 7. Chain of evidence: After validation of digital evidence exhibit "B" is negated.

Example

We present a hypothetical case to demonstrate the validation process. Consider a case involving a defendant accessing a computer and sending a threatening email to another party, then deleting the email in an attempt to conceal the evidence on the computer. A neighbour witnessed the defendant inside the room at the time of the suspected offence, and police recovered the defendant's and other persons' fingerprints from the computer keyboard, as shown in Figure 8.

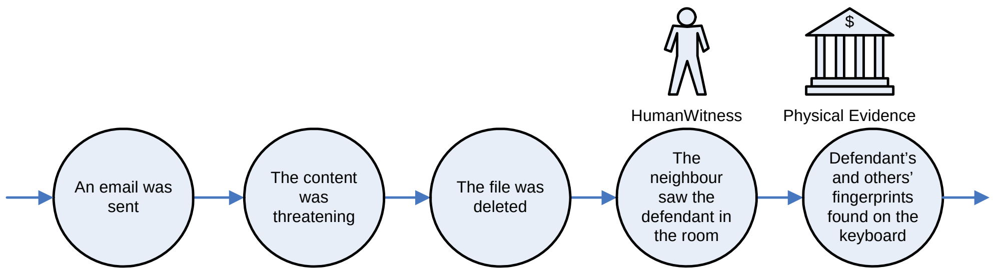

HumanWitness
Physical Evidence
$
An email was sent
The content was threatening
The file was deleted
The neighbour saw the defendant in the room
Defendant's and others' fingerprints found on the keyboard

Figure 8. Hypothetical case showing digital, human and physical evidence

If we look at one piece of the evidence, that the file was deleted, it would be prudent to find out the processes involved and whether it was possible to link the date and time of the deletion to the defendant's known presence in the room. In our hypothetical example, we seek explanations about key properties of the file evidence. We need to find out the nature of the pertinent evidence, in this case the email application properties and from that, attempt to validate the date and time of the file deletion, and then view the outcomes of that examination.

If we decompose the hypothetical case, it shows that the validation of the evidence can be a lengthy and complex process. In Figure 9 we examine the deleted file and drill down through each sub-set of evidence that provides assertions attesting to the validity of the evidence at the higher level. We expect to reach a conclusion that is acceptable in the legal practitioner's opinion as validation of the primary assertion corroborated by supporting evidence.

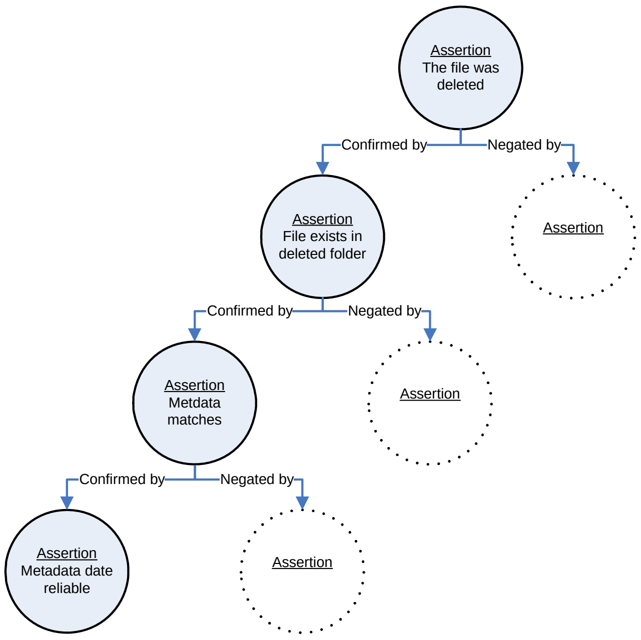


Figure 9. Hypothetical case decomposition showing the validation of assertions

Figure 10 shows an alternative path in which the decomposition established that the evidence was invalid through the absence of corroborating evidence at the fourth secondary assertion.

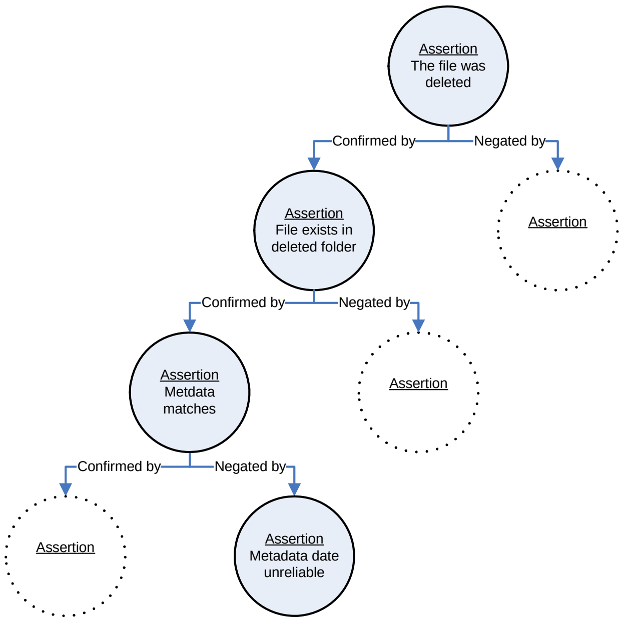


Figure10. Hypothetical case decomposition showing the invalidation of assertions

Figure 11 shows an alternative scenario where an anomaly existed about the validity of email file metadata matching the time of the suspected deletion. This required further explanation confirming modification of the metadata by virus scanning activity.

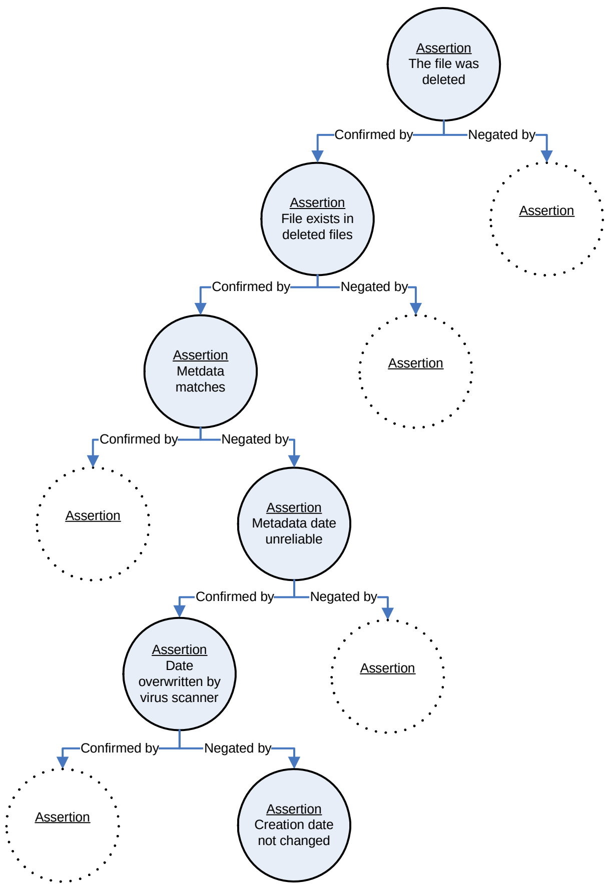


Figure 11. Hypothetical case decomposition showing the validation of assertions by drilling down through negative assertions

Further explanation showed that the virus scanning activity did not alter the actual date and time of deletion of the email file and thus validity is sustained. This scenario emphasises the importance of searching for evidence that negates an assertion as well as seeking confirmation.

Although potentially many questions about the evidence exist in our hypothetical and relatively simple example, the number of questions may increase exponentially in complex cases. The process of decomposing the original hypothetical evidence uncovered additional evidence clarifying the truth of the original bland assertion about the file deletion. While this was a positive outcome, one path of validation did identify a negative assertion but this was later nullified by other evidence, thus demonstrating the complexity of the digital domain.

Using the decomposition of the simulated result in Figure 11 the chain of evidence gained more pieces of validated evidence to replace the simple assertion that a deleted file existed. The decomposition ended with the assertion confirming the validity of the creation date of the deleted file and for the sake of brevity, we have not decomposed the process further. So, accepting the new evidence, the chain of evidence may be modified with validated evidence that is also more complete as shown in Figure 12.

The validation process described has identified a possible weakness in the evidence regarding a critical time, the date and time of the file deletion. It is an important part of the legal argument to be able to state categorically the exact time of deletion, but future counter-argument can be dispelled because it can be shown that only partial modification occurred which does not weaken the assertion about the deletion time. The process has validated the evidence providing clarification that allows the legal practitioner to gain a far greater understanding of the strengths and weaknesses of the evidence available.

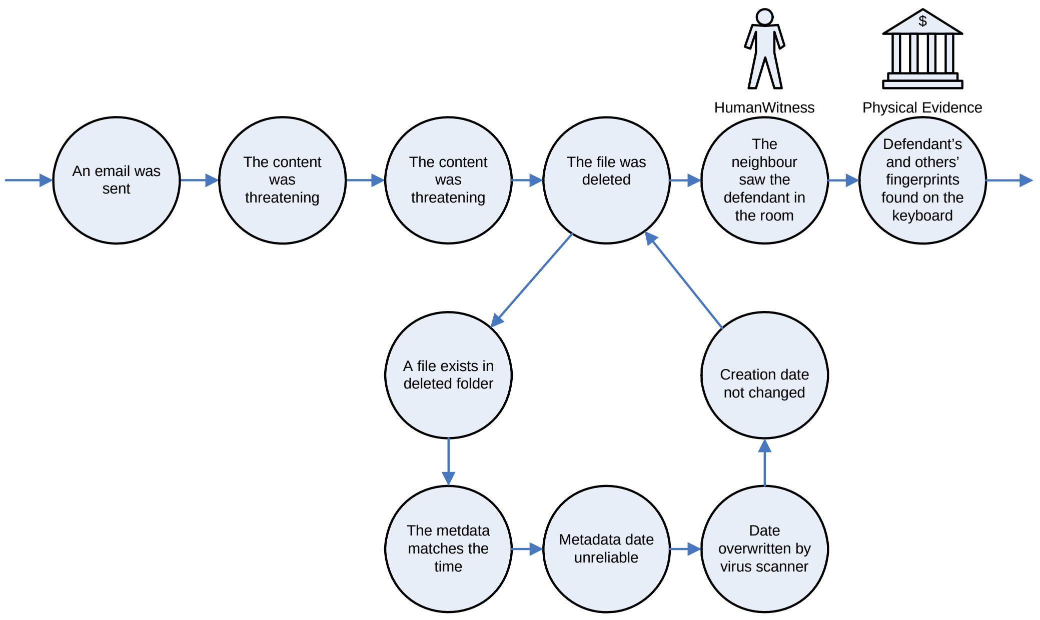

HumanWitness
Physical Evidence
$
An email was sent
The content was threatening
The content was threatening
The file was deleted
The neighbour saw the defendant in the room
Defendant's and others' fingerprints found on the keyboard
A file exists in deleted folder
Creation date not changed
The metdata matches the time
Metadata date unreliable
Date overwritten by virus scanner

Figure 12. Hypothetical case: modified chain of evidence after validation

The search for validation shown in the hypothetical example assumes that the enquirer has both investigative and technical skills but the legal practitioner may be unable to know what questions to ask to test the validity of the evidence. In the next section we describe how we may assist the legal practitioner in seeking relevant information by providing prompts at each step of the process.

An interrogation checklist

To assist the legal practitioner, it is useful as part of the validation interrogation process to provide prompts that firstly provide explanation about the properties of digital evidence, and secondly alert the legal practitioner when further validation of the digital evidence is prudent. Providing an interrogation checklist of digital evidence properties that offers a suite of prompts would enable the legal practitioner to make more discerning judgement on the weight of evidence knowing that is has been validated or invalidated, or needs more research to a point of reasonable termination of endeavour.

An interrogation checklist, which supplements the validation interrogation process, supplies prompts to direct enquiry to seek facts confirming or negating the validity of the evidence. The enquirer has two options, to seek confirmation of validity or to seek negation. These two options offer separate search patterns for the legal practitioner who may seek validation of incriminating evidence, or wishes to develop an alternative hypothesis as part of a counter-argument. We suggest that this system has the advantage of being more inclusive because it would cover a broader range of potential evidence overlooked in the selection and validation stages of the investigation process shown in Figure 1.

Table 1 provides a small example of the proposed checklist based on the hypothetical case. A set of categories - “subject”, “assertion”, several “prompts” and “known issues” – guides the legal practitioner to identify and locate corroboratory evidence to validate each piece of evidence. Using the example of the hypothetical case, the email is the subject of the validation process and the occurrence of file deletion, the assertion. The first prompt provides the enquirer with a list of file locations to commence a search. The second prompt suggests what tool should be used, such as the email application, and the third prompt provides a range of information that should be sought with the tools suggested and at the location suggested. The fourth prompt suggests conditions that could assist in validating the assertion.

Subject	**Assertion**	**Prompt** **Where to search?**	**Prompt** **How to search?**	**Prompt** **What to seek?**	**Prompt** **How do we know if : Yes / No / Unclear?**	**Known issues**
Email	[Email was] Copied	Application	Search tool	Metadata	Date/time match	Metadata can be falsified
					No date/time match	
					Date/time missing	
	[Email was] Deleted	**Email trash**	**Application**	**Metadata**	**Date/time match**	**Retention affected by storage limitations**
					No date/time match	
					Date/time missing	
		Recycle bin	**Windows Explorer** **or** **Forensic tool**	**File contents**	**Intact**	**File can be overwritten**
					Not evident	
					Part missing	
					Intact	**File can be unrecoverable**
					Not evident	
					Part missing	
				File header	**Intact**	**File can be overwritten**
					Not evident	
					Part missing	
					Intact	**File can be unrecoverable**
					Not evident	
					Part missing	
		Drafts	**- ditto -**	**- ditto -**	**- ditto -**	**- ditto -**

Table 1. Checklist entries showing prompts to assist in validating an email file deletion

In this sample, the assertion can be checked against the metadata of the email file to compare the data and time available with the known time of the offence. This prompt provides “Date/time match”, “No date/time match”, and “Date/time missing”. The final column “Known issues”, provides supplementary information about previously identified validation issues.

Subject	Assertion	Prompt	Prompt	Prompt	Prompt	Known issues
		Where to search?	How to search?	What to seek?	How do we know if : Yes / No / Unclear?	
Deleted email metadata	**Falsified**	**Inside the file properties table**	**Application**	**Authorship details**	**Present**	**Not possible to detect manual alteration of the properties using the parent application**
					Not present	
					Cannot resolve if present	
		The hard drive	**Forensic tool**	**File attribute modification application**	**Traces of the application**	
					No traces of the application	
					Unclear if traces of the application	
		Inside the application		**Other copies of the file**	**Metadata match**	
					Metadata mismatch	
					Metadata irregular	

Table 2. Checklist entries relevant to the metadata of an email file.

The information supplied through Table 1 may still be an inconclusive result but will provide additional prompts to direct further searches in a different part of the checklist to locate more information to assist the validation process. This is shown in Table 2 where a series of further prompts point to possible scenarios that may relate to the deleted email file. The validation prompt offers an extra suite of prompts such as “metadata mismatch” or “traces” and so forth.

Although it is outside the scope of this paper to develop fully the checklist we plan further research to test its feasibility and usefulness to the legal practitioner and possibly investigators in validating digital evidence. Formulating a database of digital evidence properties that can link back to the evidence in a given context would be especially useful in enhancing understanding of the evidence validation in a wide range of cases.

CONCLUSION

We have presented a practical process that can assist legal practitioners in validating digital evidence through a process of guided questioning. We suggested that the process can be supported by a checklist of appropriate prompts, and presented a hypothetical example of how the questioning and checklist of prompts might be used in practice. We suggest that such a process could be of great value to legal practitioners as it makes explicit a vital stage in the investigation of digital evidence that can easily be overlooked or underestimated. Further research is planned to progress the checklist further, with the aim of developing a generic model based on an ontology of the digital evidence field. Research will also focus on developing an appropriate representation for the process, so that it is usable as a practical tool for legal practitioners in validating digital evidence.

REFERENCES

Ahmad A. (2002) The forensic chain of evidence model: Improving the process of evidence collection in incident handling procedures. *The 6th Pacific Asia Conference on Information Systems*.

Akester, P. (2004) Internet law: authenticity of works: authorship and authenticity in cyberspace. *Computer Law & Security Report,* 20**,** 436-444.

Anderson, T., & Twining, W (1991) *Analysis of evidence: How to do things with facts based on Wigmore's Science of Judicial Proof,* Evanston, IL, Northwestern University Press.

Ashcroft, J. (2001) Electronic crime scene investigation: A guide for first responders. Washington, U.S. Department of Justice.

Ashley, K., & Rissland, E. (1985) Toward modelling legal argument. University of Massachusetts.

Berk, R. A. (1983) An introduction to sample selection bias in sociological data. *American Sociological Review,* 48**,** 386 - 398.

Caloyannides, M. A. (2001) *Computer forensics and privacy,* Norwood, Minnesota, Artech House.

Carrier, B. (2005a) *File system forensic analysis,* Upper Saddle River, New Jersey, Addison-Wesley

Carrier, B., & Spafford, E. H. (2003) Getting physical with the digital investigation process. *International Journal of Digital Evidence*.

Carrier, B. D., & Spaford, Eugene. H. (2005b) Automated digital evidence target definition using outlier analysis and existing evidence. *Digital Forensic Research Workshop.* New Orleans.

Cohen, F. (2006) Challenges to digital forensic evidence. New Haven, Fred Cohen & Associates.

Daum, M., & Lucks, Stefan. (2005) Attacking hash functions by poisoned messages: The Story of Alice and her boss. Bochum, CITS Research Group, Ruhr-Universität Bochum.

Diaconis, P., & Mosteller, F. (1989) Methods for studying coincidences. *Journal of the American Statistical Association,* 84**,** 853 - 861.

Edwards, K. (2005) Ten things about DNA contamination that lawyers should know. *Criminal Law Journal,* 29**,** 71 - 93.

Egan, M., & Mather, Tim (2004) *The executive guide to information security: Threats challenges and solutions,* Indianapolis, Addison-Wesley:Symantec Press.

Etter, B. (2001a) Computer crime. *4th National Outlook Symposium on Crime in Australia - New Crimes or New Responses.* Canberra, Australian Institute of Criminology.

Etter, B. (2001b) The forensic challenges of e-crime. *Australasian Centre for Policing Research,* 3**,** 1-8.

Fiske, S. T., & Taylor, S. E (1991) *Social cognition* New York, McGraw-Hill.

Flusche, K. J. (2001) Computer forensic case study: Espionage, Part 1 Just finding the file is not enough! *Information Security Journal,* 10**,** 1 - 10.

Huygen, P. E. M. (2002) Use of Bayesian Belief Networks in legal reasoning. *17th BILETA Annual Conference.*

Janes, S. (2000) The role of technology in computer forensic investigations. *Information Security Technical Report,* 5**,** 43 - 50.

Koehler, J. J., & Thompson, William. C. (2006) Mock jurors' reactions to selective presentation of evidence from multiple-opportunity searches. American Psychology-Law Society/Division 41 of the American Psychological Association.

Lenstraand, A., & De Weger, Benne. (2005) On the possibility of constructing meaningful hash collisions for public keys. *Australasian Conference on Information Security and Privacy 2005.* Brisbane, Lucent Technologies, Bell Laboratories, and Technische Universiteit Eindhoven.

Losavio, M., Adams. J., & Rogers, M. (2006) Gap Analysis: Judicial experience and perception of electronic evidence. *Journal of Digital Forensic Practice,* 1**,** 13 - 17.

Mattord, H. J., & Whitman, M. E. (2004) *Management of information security,* Boston, Thomson learning.

Mercuri, R. (2005) Challenges in forensic computing. *Communications of the ACM* 48**,** 17 - 21

Mohay, G. M. (2003) *Computer and intrusion forensics,* Boston, Artech House Inc.

Nisbett, R. E., & Ross, L. (1980) *Human inference: Strategies and shortcomings of social judgment,* Englewood Cliffs, NJ, Prentice Hall.

Nisbett, R. E., Krantz, D. H., Jepson, C., & Kunda, Z (1983) The use of statistical heuristics in everyday inductive reasoning. *Psychological Review,* 90**,** 339-363.

Ó Ciardhuain, S. (2004) An extended model of cybercrime investigations. *International Journal of Digital Evidence,* 3.

Palmer, G. L. (2001) A road map for digital forensic research. *First Digital Forensic Research Workshop (DFRWS).* Air Force Research Laboratory, Rome Research Site, Digital Forensic Research Workshop.

Palmer, G. L. (2002) Forensic analysis in the digital world. *International Journal of Digital Evidence,* 1.

P**e,** C., & Olbrechts-Tyteca, L. (1969) *The new rhetoric: A treatise on argumentation,* Notre Dame, Indiana, University of Notre Dame Press.

Pollitt, M. M. (2001) Report on digital evidence. *13th INTERPOL Forensic Science Symposium* Lyon, France, INTERPOL.

Pospesel, H., & Rodes (Jnr), Robert. E. (1997) *Premises and conclusions: Symbolic logic for legal analysis,* New Jersey, Pfrentice-Hall, Inc.

Rowlingson, R. (2004) A ten step process for forensic readiness. *International Journal of Digital Evidence,* 2.

Safterein, R. (2000) *Criminalistics: An introduction to forensic science*, Pearson.

Saks, M. J., & Koehler Jonathan. J (2005) The coming paradigm shift in forensic identification science. *Science,* 309**,** 892 - 895.

Schneier, B. (2000) *Secrets and lies: digital security in a networked world,* New York, Wiley Computer Publishing.

Schneier, B. (2004) Opinion: Cryptanalysis of MD5 and SHA: Time for a new standard: Crypto researchers report weaknesses in common hash functions. *Computerworld.*

Silverstone, H., & Sheetz, Michael (2007) *Forensic accounting and fraud investigation for non-experts,* New Jersey, John Wiley & Sons, Inc.

Slade, R. (2004) *Software forensics: Collecting evidence from the scene of a digital crime,* New York, McGraw Hill.

Sommer, P. (1998) Intrusion detection systems as evidence: Recent advances in intrusion detection. London School of Economics & Political Science.

Sommer, P. (2000) Digital footprints: Accessing computer evidence. *Criminal Law Review,* 61 - 78

Spenceley, C. (2003) Evidentiary treatment of computer-produced material: a reliability based evaluation. Sydney, University of Sydney.

Tapper, C. (2004) *Cross & Tapper on evidence,* London, LexisNexis Butterworths.

Thompson, D. E., & Berwick, Desmond. R. (1998) Minimum provisions for the investigation of computer based offences. Payneham, South Australia, National Police Research Unit.

Tillers, P. (2005) Picturing factual inference in legal settings. *Gerechtigkeitswissenschaft: Kolloquium aus Anlass des 70: Geburtstages von Lothar Philipps.* Berlin.

Tobin, W. A., & THOMPSON, WILLIAM. C (2006) Evaluating and challenging forensic identification evidence. *Champion Magazine.*

Unamed (2000) The virtual horizon: meeting the law enforcement challenges: developing an Australasian law enforcement strategy for dealing with electronic crime: scoping paper. *Police Commissioners' Conference - Electronic Crime Working Party 2000.* Adelaide., Australasian Centre for Policing Research.

Wall C., & Paroff, Jason. (2004) Cracking the computer forensics mystery. *UtahBar Journal,* 17.

Whitcomb, C. M. (2002) An historical perspective of digital evidence: A forensic scientist's view. *International Journal of Digital Evidence,* 1.

Whitman, M. E., AND Mattord, H. J. (2005) *Principles of information security,* Boston, Massachusetts, Thomson Learning.

Yasinsac, A., Erbacher, R. F., Marks, D. G., Pollitt, M. M., & Sommer, P. M. (2003) Computer forensics education. *IEEE Security & Privacy,* 1, 15 - 23.

COPYRIGHT

Survey and future trends of efficient cryptographic function implementations on GPGPUs

Adrian Boeing
School of Computer and Information Science
Edith Cowan University
a.boeing@ecu.edu.au

Abstract

Many standard cryptographic functions are designed to benefit from hardware specific implementations. As a result, there have been a large number of highly efficient ASIC and FPGA hardware based implementations of standard cryptographic functions. Previously, hardware accelerated devices were only available to a limited set of users. General Purpose Graphic Processing Units (GPGPUs) have become a standard consumer item and have demonstrated orders of magnitude performance improvements for general purpose computation, including cryptographic functions. This paper reviews the current and future trends in GPU technology, and examines its potential impact on current cryptographic practices.

Keywords

Cryptography, Graphics Processing Unit, GPU, RC4

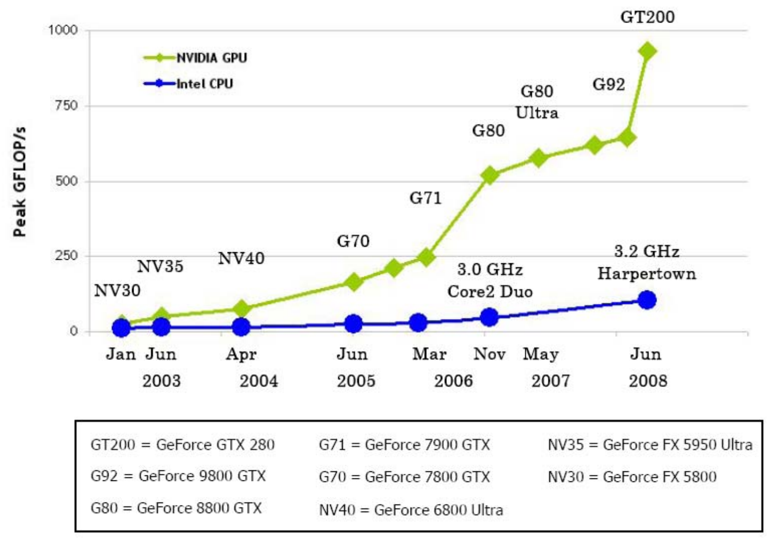
INTRODUCTION

Graphics Processing Units (GPUs) have become a standard consumer item that provide high performance parallel processing. GPUs have demonstrated orders of magnitude performance improvements for cryptographic functions, similar or superior to the performance gains achieved by Application Specific Integrated Circuit (ASIC) approaches.

Although the high performance capabilities of ASIC systems are widely known, the difficulty of developing and obtaining these systems has hindered their acceptance. Thus, ASIC based hardware cryptography has typically only been available to a few large organizations.

Many cryptographic standards that can be efficiently cracked with hardware solutions are still in common use. These may be in use due to poor administration or to enhance the performance of the system. Vinadsius (2006) indicates that many network administrators choose to employ insecure security standards such as WEP (37%), or not apply security to their networks at all (57%). Internet server administrators often choose to employ lower security standards in order to gain performance (Lowenthal, 2001). Encryption key sizes of less than 40 bits are typical (Lowenthal, 2001). These choices are often made as it has been considered unlikely that an attacker would own the required hardware resources to dedicate to an attack.

In recent years graphics card processors have become standard consumer items, offering vast performance increases for certain problem tasks over traditional CPUs. The peak floating point performance of nVidia GPUs and Intel CPUs is illustrated in a graph in Figure 1. Cope et. al. (2005) compared the performance of GPUs with Field Programmable Gate Arrays (FPGAs) and found the GPU outperformed the FPGAs for a certain signal processing tasks.



GT200 = GeForce GTX 280	G71 = GeForce 7900 GTX	NV35 = GeForce FX 5950 Ultra
G92 = GeForce 9800 GTX	G70 = GeForce 7800 GTX	NV30 = GeForce FX 5800
G80 = GeForce 8800 GTX	NV40 = GeForce 6800 Ultra	

Figure 1 - Floating point performance of nVidia GPUs and Intel CPUs in billions of floating point operations per second. **(nVidia, 2008)**

In the remained of the paper GPU technology will be reviewed and the future trends of stream processing will be investigated. Cryptographic implementations on GPUs will then be discussed and contrasted to CPU and FPGA implementations. Finally, the RC4 algorithm will be investigated in detail and the conclusions will be presented.

GPU TECHNOLOGY REVIEW

General Purpose Graphics Processing Units (GPGPUs) are a type of stream computers that offer price effective high performance computing (Owens, et al., 2007). Unlike traditional CPUs, GPGPUs perform calculations on streams of data. This is similar to a Single Instruction Multiple Data (SIMD) vector processor, however the typical GPGPU processor is composed of a large number of independent processing elements, or threads. As a result the GPU can be seen as both a SIMD and Signal Program Multiple Data (SPMD) device. The exact implementation of the stream computing device is vendor specific and thus the capabilities of each vary.

The common feature of GPUs are a large number of processor cores. Each core will then contain a number of general purpose vector processors and execution contexts, an optimized memory processing unit (or texture unit), standard CPU components (such as instruction decoding) and specialized functionality for rendering graphics (such as the rasterizer and image blend units). This design enables the GPU to execute a large number of threads concurrently. The cache design for each GPU is carefully chosen to enhance performance for streaming data applications.

There are four major vendors that are producing streaming processors. The traditional GPU manufacturers are nVidia and ATI (purchased by AMD in 2006 (AMD, 2006)). The other manufacturers produce more general processors that are highly optimized for graphics tasks. These are the CELL processor, produced by Sony, Toshiba and IBM, and the Larrabee GPU produced by Intel.

nVidia was an early developer of GPUs for consumer systems, and pioneered a number of GPGPU technologies. The traditional target market for nVidia was accelerating rasterized graphics rendering, however modern nVidia GPUs target image processing, physics calculations and financial applications. The GTX 280 (nVidia), is the current top-end consumer GPU available from nVidia. It consists of thirty processing cores, each core containing an eight-wide SIMD processing unit. Each SIMD unit can perform three operations per clock cycle, and the device runs at 1.3 GHz. This results in an overall potential peak performance of just under one trillion operations per second (Fatahalian, 2008). This is the equivalent computational power as the worlds fastest super computer from 1996, the Sandia Labs ASCI Red (TOP500.Org, 2008). It is worth emphasising that the retail price for this hardware is currently a few hundred dollars.

nVidia also provides support for a number of GPU programming environments, as well as providing their own language to enable high performance general computing for the GPU. Compute Unified Device Architecture (CUDA) is a language environment similar to C that provides extended functionality to program the GPU.

ATI is another GPU manufacturer with a strong history of applying general purpose processing techniques to GPU technology. ATI was acquired by AMD in 2006, and have since focused on combining GPU and CPU technology onto a single chip (AMD, 2006). Similar to nVidia's traditional GPUs AMD focuses on graphics processing as a primary purpose of their hardware, and design the GPU accordingly. AMD has provided double-precision floating point number support to their GPUs to encourage high performance scientific uses for their GPUs.

The current top end GPU model from ATI/AMD is the Radeon 4870. It contains ten processing cores, each with 16 SIMD units. Each SIMD unit can complete ten operations per cycle, and the chip runs at 750 MHz. This provides a total peak performance of 1.2 trillion operations per second (Fatahalian, 2008), almost 30% faster than nVidia's already impressive offering.

Similarly to nVidia, AMD provides support for programming their GPUs in a number of graphics contexts. AMD also provides its own environments, APIs and languages that support its GPU, this includes Brook+, Close To the Metal, Compute Abstraction Layer, and have announced support for OpenCL and Microsoft's DirectX 11 Compute Shader. Brook+ and OpenCL are both similar technologies to nVidia's CUDA, they are all C-like language environments with extensions to support the stream-based programming model through the use of compute kernels.

Both Intel and the STI (Sony, Toshiba, IBM) technologies are aimed at performing, or supporting graphics processing, however are also intended to be far more applicable to general purpose computing. Intel's upcoming Larrabee GPU technology is based on x86 technology, thus making it directly compatible with most existing PC software (Seiler, et al., 2008). STI's CELL processor technology is designed to be a more general processor that operates alongside the GPU, rather than aimed at displacing it.

The CELL processor is available in a wide range of configurations. The most readily available CELL processor is inside the Playstation 3. The PS3 CELL processor contains one power PC CPU and eight Synergistic Processor Elements (SPE) cores. Each SPE has four SIMD units and each unit can complete two operations per second. The SPEs run at 4000 MHz, giving a total performance of 0.25 trillion operations per second.

Whilst the SPE peak performance values appear far less impressive than the performance of the competing GPU technologies, the actual performance difference in an application may not be as great. This is due to the difficulty in programming the GPUs in such a way as to achieve the optimal performance.

Intel has announced the architecture for their new GPU technology scheduled for release in 2009. Some of the exact details on the chips architecture are presently unavailable (Seiler, et al., 2008) (Fatahalian, 2008). The Larrabee chip contains an unknown number of processors, however some experts estimate the likely number of processors for their top end model to be 32 cores (Fatahalian, 2008) and 24 cores at launch (Gruener & Polkowski, 2007). Each processor is based on the Pentium x86 design, however contains a new vector processor design. Each vector processor contains sixteen processing units, with each unit capable of completing two operations per second. The exact clock speed of Larrabee is also unknown at this point, however industry sources report that it will range from 1.7 to 2.5 GHz (Gruener & Polkowski, 2007). This gives a total of around 2 trillion operations per second. This figure is comparable to the expected performance projections for AMD and nVidia GPUs.

Technology	Specifications	Peak billion operations per second
nVidia GTX 280	30 (cores) * 8 (SIMD units) * 3 (operations per unit) * 1300 MHz (clock speed)	936
ATI/AMD Radeon 4870	10 (cores) * 16 (SIMD units) * 10 (operations per unit) * 750 MHz (clock speed)	1200
Sony, Toshiba, IBM CELL	8 (SPE cores) * 4 (SIMD units) * 2 (operations per unit) * 4 GHz (clock speed)	256
Intel Larrabee (predicted)	32 (cpu cores) * 16 (SIMD units) * 2 (operations per unit) * 2 GHz (clock speed)	2000
Intel, Core 2 E7200	2 (CPU cores) * 4 (SIMD units) * 2 (operations per unit) * 2.53 GHz (clock speed)	40

Table 1 – Peak performance characteristics of computing technology

To provide a benchmark figure, the current Intel Core 2 E7200 CPU provides roughly 40 billion operations per second, 50 times less than the projected performance of their GPU offering.

The implications of Intel's entrance to the GPU market and AMD's consolidation with ATI is that both manufacturers standard CPUs will be integrated with GPU technology. This places an extremely large processing power at the hands of every consumer PC.

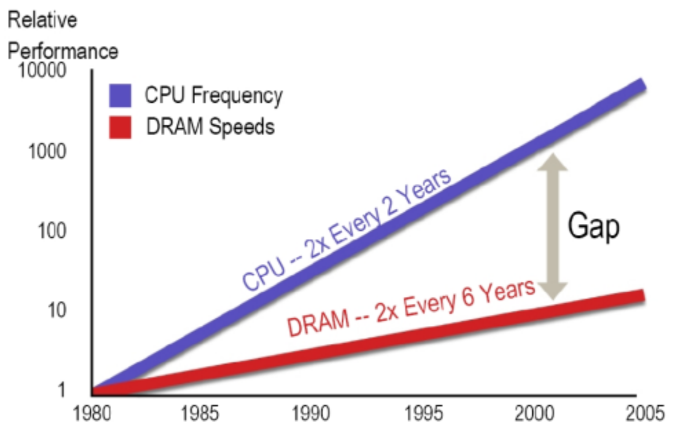


Figure 2 Trends of CPU and Memory performance increase (Mayo, 2008)

The processing tasks that GPU processors are aimed at tend to be highly parallel (such as rasterization). Therefore GPUs can gain performance in each generation by increasing the number of cores. For example, the nVidia 8600GT has 4 multiprocessors, the 9800GX2 has 16 multiprocessors, and the GTX280 has 30 multiprocessors. As with other integrated circuit technology GPUs follow the benefits from improved processes, as predicted by Moore's Law. Thus, we can reasonably expect GPU FLOP performance to continue to exponentially increase in the future at a very high rate.

Since GPUs are not aimed at general processing tasks we can expect that the memory processing capabilities of the GPU will not be expanded at the same rate. It is likely that the architectural structure of GPUs will follow other stream processors, such as the Cell's SPEs. Current GPUs have limited memory caches (eg: 16kb for a nVidia GeForce 9800GX2) compared to cache sizes upwards of four megabytes for typical CPU's. The CELL SPEs have 256KB local memory (Flachs, et al., 2005). Memory technology does not increase at the same rate as CPU technology (See Figure 2), so it is highly likely that GPU caches will remain small and limit the performance of general purpose GPU computing in the future.

REVIEW OF GPU CRYPTOGRAPHY

Applying cryptographic functions on hardware designed for graphics processing is not a recent idea. An early application of cryptography to GPUs was in 1999, where a custom built GPU architecture “PixelFlow” was used to do a brute force cryptanalysis of the DES and RC4 ciphers (Kedem & Ishihara, 1999). This demonstrated that devices designed for graphics processing could also be leveraged for cryptanalysis.

A common approach to improving the performance of cryptographic functions is to implement them on Application Specific Integrated Circuits (ASICs), such as Field Programmable Gate Arrays (FPGAs). Jarvinen et al. (2005) present an overview of 32 different AES implementations on Xilinx Virtex FPGAs (Jarvinen, Skytta, & Tommiska, 2005). The throughput of the implementations reviewed ranged from 23.57 Gbps to 0.215 Gbps, this reflects the range in the implementations performance depending on the skill of the implementers and the design goal (eg: minimize power consumption, etc.). Similar results were presented for common cryptographic algorithms such as IDEA, MD5, and SHA.

Moss et al. (2007) implemented a 1024bit modular exponentiation algorithm (a key part of the RSA public key cryptosystem) on a nVidia 7800-GTX using the OpenGL Shading Language. The GPU program was written to employ the 24bit mantissa of the floating point values and the GPU special texture-memory. Five versions of the GPU algorithm were tested:

- an un-optimized 12bit-half-word version (A)
- a texture-memory optimized half-word version (B)
- a texture-memory optimized 24bit-full-word version (C)
- a texture-memory optimized smooth-word version, (D)
- a texture-memory optimized smooth-word version with decomposed-modulus operations. (E)

The performance of their system is given in **Table 2**. These results indicate that memory optimizations provided greater performance enhancement than the algorithmic calculation changes.

CPU	17ms
GPU A	21.5ms
GPU B	12.5 ms
GPU C	11.5ms
GPU D	9.8ms
GPU E	5.7ms

Table 2 – Performance of 1024 bit modular exponentiation in ms **(Moss, Page, & Smart, 2007)**

An early AES GPU implementation was presented by Cook, et al in 2004. The AES implementation was tested on an nVidia Gefore3 TI 200, ATI Radeon 7500, and a nVidia TNT 2 GPU. The peak performance for the XOR component of their system was 105MBps, 41.5MBps, and 38.4 MBps respectively. Whilst the CPU performance of a 1.8Ghz Pentium 4 processor was 139 MBps. The AES implementation could not run completely on the GPU, and still employed the CPU for some processing. The peak CPU performance measured for the AES program was 64MBps, whereas the best GPU performance obtained was 1.53MBps with the Geforce3.

In 2007, Harison and Waldron (2007) implemented an AES cipher on an nVidia GeForce 6600GT and an nVidia GeForce 7900GT using OpenGL Pixel Buffer Objects. The AES XOR operation and memory access patterns were investigated. Three XOR approaches were tested:

- Calculating the 8bit by 8 bit XOR operation via a 65kb lookup table
- Decomposing the XOR operation into 4bit sections, thus reducing the lookup table size
- Using the final render phases internal XOR operation and render buffers

Three AES memory access techniques were tested:

- Multi Input Gather, where 4 independent texture blocks operate simultaneously
- Single Input H-Gather, where 4 consecutive texture elements from one block are read
- Single Input S-Gather, where 4 texture elements that are organized within a 2D section are read from one block

Technique	8bit XOR	4 bit XOR	Native
Multi Input	25.86	39.23	108.86
S-Gather	26.06	39.18	N/A
H-Gather	25.99	39.08	N/A

Table 3 – AES results for a GeForce 7900GT in Mbytes/s

The major limiting factor from the Harison and Waldron AES implementation was the use of native XOR operation, and was not limited by the memory access speeds. This differs from Cook et al., likely due to the newer GPU used which provided greater general purpose capabilities.

Manavski (2007) also implemented an AES cipher in CUDA and tested the performance on an nVidia 8800GTX compared to an OpenGL implementation on the ATI X1900. The performance was compared to the AES algorithm implemented in the OpenSSL library on a 3.0 GHz Pentium 4 CPU.

The CUDA language enabled the full use of 32bit XOR operations, removing the need for XOR lookup tables present in OpenGL based implementations. Precomputed T-box values were stored in the GPU's faster constant memory, and the input and output buffers of the routines were stored in the GPU's fast shared memory. As a result of these optimizations, a 19.6 fold speedup over the CPU performance was reported.

The MD5 cipher has also been implemented on GPUs. Tzeng and Wei (2008) implemented an MD5 cipher for the purpose of generating white noise in computer graphics applications. The MD5 cipher was selected over other cryptographic algorithms such as AES since it has higher memory access requirements than MD5 (Tzeng & Wei, 2008). Two versions of the MD5 algorithm were implemented, a reference version and a memory optimized version with loop unrolling.

The system processed 1024x1024 keys in 3821.5ms for the CPU version, 229.7ms for a GPU implementation, and 6.3ms for a GPU optimized version. The large speedups were gained due to the suitability of the MD5 algorithm to GPU processing. This leveraged the low memory requirements of the MD5 algorithm and the high parallel computation speed of the GPUs multiprocessor architecture.

Commercial and open-source products have been released which provide GPU-based MD5 cryptography. A benchmark of competing GPU and CPU MD5 products was completed by Aleksandrovich (2008). Six different GPU MD5 implementations were benchmarked, the best achieving 350 million keys searched per second.

There have been a number of different cryptographic algorithms implemented on GPUs. The performance characteristics of the reviewed implementations is summarized in **Table 4**. It is clear that the newer GPUs outperform earlier implementations, and that the GPU implementations speeds increase drastically when a new programming technology is applied (eg: CUDA). Some algorithms such as the MD5 algorithm received very large speedups on GPUs. Likewise the AES performance increased dramatically when the programmability of the GPU increased. These algorithms are computationally intensive, and few memory intensive algorithms have been investigated.

Author	Algorithm	Manufacturer	GPU Model	Model Year	API	GPU Speedup
Moss et al.	RSA-MOD	nVidia	GeForce 7800GTX	2005	OpenGL	3.77
Cook et al.	AES-XOR	nVidia	GeForce 3TI 200	2001	OpenGL	0.76
Cook et al.	AES-XOR	ATI	Radeon 7500	2000	OpenGL	0.30
Cook et al.	AES-XOR	nVidia	TNT 2	1999	OpenGL	0.28
Harison et al.	AES	nVidia	GeForce 7900GT	2006	OpenGL PBO	2.34
Harison et al.	AES	nVidia	GeForce 6600GT	2004	OpenGL PBO	0.98
Manavski.	AES	nVidia	8800GTX	2006	CUDA	19.60
Tzeng et al.	MD5	nVidia	GeForce 8 Series (Unspecified)	2006	GLSL	606.59
Boeing.	RC4	nVidia	GeForce 9800GX2	2008	CUDA	8.57

Table 4 – GPGPU Cryptography implementations. GPU speedup is given relative to the baseline CPU performance used in the original article.

RC4 CRYPTOGRAPHY

The RC4 cipher is commonly used to encrypt data communications, such as the Wired Equivalent Privacy (WEP) and Wi-Fi Protected Access (WPA) protocols. It is also used in Microsoft's Remote Desktop Protocol (Microsoft, 2008), and other standard desktop products such as Word and Excel (Schneier, 2005). The RC4 algorithm is a memory intensive cipher.

Recently, Kwok and Lam (2008) presented a highly parallelized FPGA based RC4 implementation that achieved $1.07\text{x}10^7$ keys per second, requiring approximately 28.5 hours to break a 40-bit RC4 encryption (Kwok & Lam, 2008). Simulation results for a high end FPGA chip indicated the system could achieve more than twice the given performance. In 2002 Tsoi et. al. presented a comparison of RC4 CPU implementations and their FPGA implementation, the FPGA managed to search $6.06\text{x}10^6$ 56 bit RC4 keys per second, approximately fifty times faster than the fastest CPU implementation they tested.

There have not been many RC4 implementations implemented on GPUs due to the high memory requirements of the algorithm, making it a difficult algorithm to speed up on the GPU architecture.

The RC4 algorithm operates in three steps. The initial step is the key-scheduling algorithm which initializes the data structures required for the encryption process. This is a 256 byte "S" permutation array. It is initialized with a sequence of values from 0 to 255, and then permuted according to the given RC4 key. The pseudo code for the key-scheduling algorithm is given in **Listing 1**.

```
for i from 0 to 255
    S[i] := i
endfor
j := 0
for i from 0 to 255
    j := (j + S[i] + key[i mod keylength]) mod 256
    swap(S[i],S[j])
endfor
```

Listing 1 - The RC4 key-scheduling algorithm

The next step is to generate the random byte stream with the Pseudo-Random Generation Algorithm (PRGA). In each iteration, the PRGA increments i, adds the value of S pointed to by i to j, exchanges the values of S[i] and S[j], and then outputs the value of S at the location S[i] + S[j] (modulo 256). Each value of S is swapped at least once every 256 iterations. The final step is to XOR the output of the PRGA with the input plaintext.

```
i := 0
j := 0
while GeneratingOutput:
    i := (i + 1) mod 256
    j := (j + S[i]) mod 256
    swap(S[i],S[j])
    RandomByte = S[(S[i] + S[j]) mod 256]
    output =  RandomByte XOR input[i]
endwhile
```

Listing 2 - The RC4 pseudo-random generation algorithm

Unlike other encryption algorithms the RC4 algorithm makes heavy use of pseudo-randomly swapping bytes within a memory buffer. This implies the RC4 algorithm will be largely memory bound, rather than computation bound.

The RC4 algorithm was implemented in CUDA 2.0 on a nVidia GeForce 9800GX2. Four versions of the algorithm were implemented. A naive RC4 implementation for the GPU as well as three optimized versions. The first optimization was to employ the GPU's multiprocessors faster shared memory to store the S array. This provided a significant speed up. The third optimization was to align the memory structures to fit within the GPU's cache, and finally some of the mathematical operations of the RC4 algorithms were optimized, such as employing bitwise operations rather than the modulo operation.

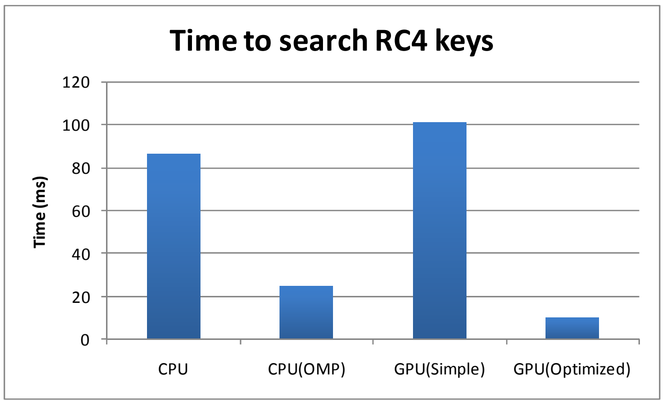


Figure 3 – Performance of RC4 key search on CPU and GPU architectures for searching 65 thousand RC4 keys

The CPU RC4 implementation achieved a 3.5x speedup when optimized for a four processor system. The naive GPU implementation was in fact slower than the CPU RC4 version. Simply porting code directly from one platform to another did not achieve a speedup, illustrating the need for GPU-specialized optimizations due to the nature for the GPU processing cores and memory system. After optimization the GPU achieved a ten fold speedup over the CPU RC4 implementation.

The impact of the different optimizations is illustrated in **Figure 4**. The largest performance gain came from placing the RC4 permutation array into the GPU's shared memory. The impact of the memory system relative to the integer maths performance of the system was investigated by varying the block size. The block size represents the number of RC4 permutation arrays that would be stored in one of the GPU's multiprocessors.

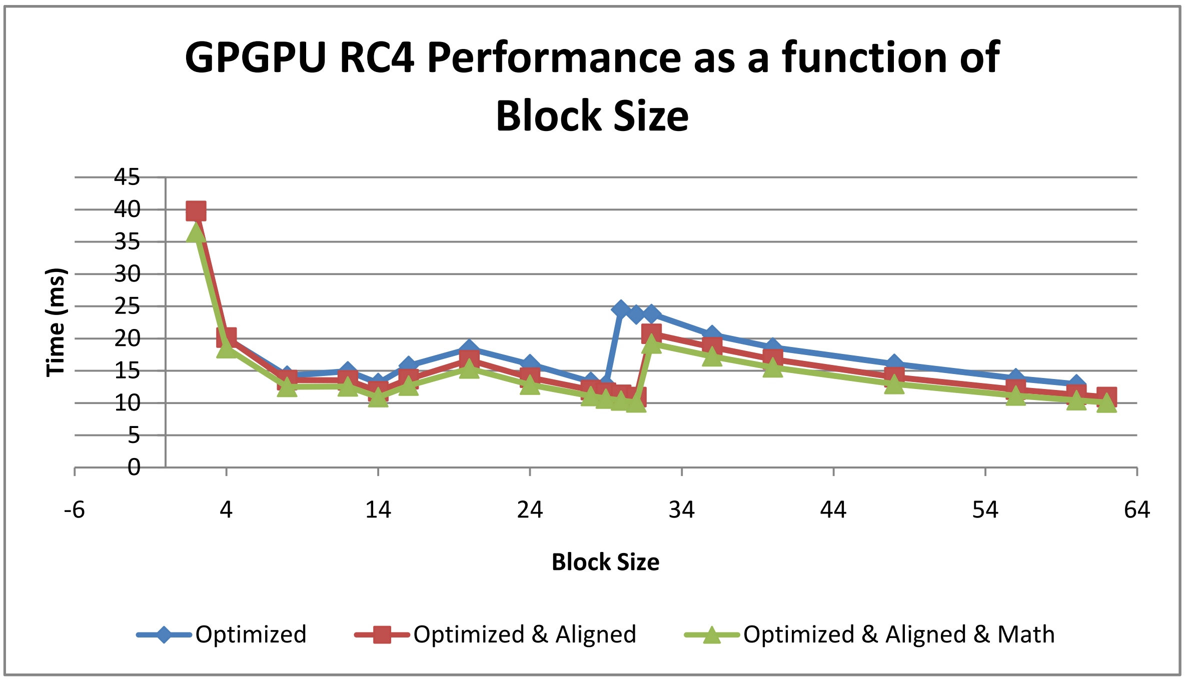


Figure 4 – GPU performance as a function of the problem block size

The results show a clear relationship between the block size and the performance of the system. A noticeable jump occurs at a block size close to 32. This indicates the memory accesses of the RC4 algorithm is a key factor in determining its performance.

The large influence of the memory characteristics can be further drawn from the performance gains from aligning the memory structures relative to optimizing the maths operations. Compacting the memory structure again provided a 14% increase in performance, whereas optimizing the mathematical operations provided only an 8% increase in performance over the memory optimized version. Overall the GPU performed 6.5x10^6 RC4 key searches per second. This is approximately 40% slower than current state-of-the art FPGA implementations (Kwok & Lam, 2008) .

CONCLUSION

GPGPUs are a very good platform for the highly parallel calculations used in cryptographic key searches. Very high performance gains have been seen for computationally intensive ciphers such as MD5 and AES. The GPGPU performance has been comparable or superior to more traditional FPGA based acceleration approaches. The overall performance gain of the RC4 cryptographic implementation was lower than that of other cryptographic algorithms. The results indicate that this is largely due to the memory intensive nature of RC4. The RC4 algorithm operates on 128 byte buffers, whereas the AES algorithm operates on only 4x4 byte buffers.

The prevalence of GPU technology and the indication by CPU manufactures to incorporate the technology on-chip will result in very high performance cryptographic accelerators to be ubiquitous. Cryptographic standards will need to be altered to keep up with the pace of computing technology. New cryptographic functions and standards should be memory intensive in order to limit the achievable performance gains by future GPU and stream computing technology.

REFERENCES

Aleksandrovich, S. M. (2008, 07 25). *MD5 bruteforce benchmark.* Retrieved 10 24, 2008, from http://3.14.by/en/read/md5_benchmark

AMD. (2006, July 24). *AMD and ATI to Create Processing Powerhouse.* Retrieved August 1, 2008, from http://www.amd.com/us-en/Corporate/VirtualPressRoom/0,,51_104_543~110899,00.html

AMD. (2006, October 25). *AMD Completes ATI Acquisition and Creates Processing Powerhouse.* Retrieved 09 1, 2008, from AMD News Room: http://www.amd.com/us-en/Corporate/VirtualPressRoom/0,,51_104_543~113741,00.html

Cope, B., Cheung, P., Luk, W., & Witt, S. (2005). Have GPUs made FPGAs redundant in the field of Video Processing? *Field-Programmable Technology*, (pp. 111-118).

Fatahalian, K. (2008). Running Code at a Teraflop: How GPU Shader Cores Work. *ACM SIGGRAPH 2008 - Course Notes: Beyond Programmable Shading.*

Flachs, B., Asano, S., Dhong, S., Hotstee, P., Gervais, G., Kim, R., et al. (2005). A streaming processing unit for a CELL processor. *IEEE International Solid-State Circuits Conference*, (pp. 134-135).

Gruener, W., & Polkowski, D. (2007, June 1). *Intel set to announce graphics partnership with Nvidia.* Retrieved 09 1, 2008, from tgdaily: http://www.tgdaily.com/content/view/32282/118/1/1/

Harrison, O., & Waldron, J. (2007). AES Encryption Implementation and Analysis on Commodity Graphics Processing Units. *Workshop on Cryptographic Hardware and Embedded Systems. 4727*, pp. 209-226. Springer-Verlag.

Jarvinen, K., Skytta, M., & Tommiska, J. (2005). Comparative survey of high-performance cryptographic algorithm implementations on FPGAs. *IEEE Proceedings on Information Security , 152* (1), 3-12.

Kedem, G., & Ishihara, Y. (1999). Brute Force Attack on UNIX Passwords with SIMD Computer. *Brute Force Attack on UNIX Passwords with SIMD Computer*, (pp. 93-98). Washington.

Kwok, S., & Lam, E. (2008). Effective Uses of FPGAs for Brute-Force Attack on RC4 Ciphers. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems , 16* (8), 1096-1100.

Lowenthal, B. (2001, January 19). *Best Practices in HTTP Security.* Retrieved October 1, 2008, from Oracle: www.oracle.com/technology/products/ias/pdf/best_practices/security_best_ practices.pdf

Manavski, S. A. (2007). CUDA compatible GPU as an efficient hardware accelerator for AES cryptography. *Proc. IEEE International Conference on Signal Processing and Communication*, (pp. 65-68).

Mayo, K. (2008, September 23). *Multi-Core Computing: Why you need to think in parallel.* Retrieved 10 24, 2008, from http://www.ivec.org/pages/tiki-download_file.php?fileId=120

Microsoft. (2008, 09 25). *Remote Desktop Protocol.* (Microsoft) Retrieved 10 23, 2008, from MSDN: http://msdn.microsoft.com/en-us/library/aa383015.aspx

Moss, A., Page, D., & Smart, N. P. (2007). Toward Acceleration of RSA Using 3D Graphics Hardware. In *Cryptography and Coding.* Springer Berlin / Heidelberg.

nVidia. (2008). *nVidia CUDA Programming Guide 2.0.*

nVidia. (n.d.). *NVIDIA GeForce GTX 280.* Retrieved 09 01, 2008, from http://www.nvidia.com/object/geforce_gtx_280.html

Owens, J. D., Luebke, D., Govindaraju, N., Harris, M., Krüger, J., Lefohn, A. E., et al. (2007). A Survey of General-Purpose Computation on Graphics Hardware. *Computer Graphics Forum , 26* (1), 80-113.

Schneier, B. (2005, 1 18). *Microsoft RC4 Flaw.* Retrieved 10 23, 2008, from Schneier on Security: http://www.schneier.com/blog/archives/2005/01/microsoft_rc4_f.html

Seiler, L., Carmean, D., Sprangle, E., Forsyth, T., Abrash, M., Dubey, P., et al. (2008). Larrabee: a many-core x86 architecture for visual computing. *ACM Transactions on Graphics , 27* (3).

TOP500.Org. (2008, 9 1). *Sandia National Laboratories ASCI Red.* Retrieved 9 1, 2008, from Top 500 Supercomputer Sites: http://www.top500.org/system/4428

Tsoi, K. H., Lee, K. H., & Leong, P. H. (2002). A Massively Parallel RC4 Key Search Engine. *Proceedings of the 10th Annual IEEE Symposium on Field-Programmable Custom Computing Machines* , p. 13.

Tzeng, S., & Wei, L.-Y. (2008). Parallel White Noise Generation on a GPU via Cryptographic Hash. *i3D* (pp. 79-88). ACM.

Vindasius, A. (2006). Security state of wireless networks. *The 10th International Conference ELECTRONICS*, *7*, p. 4. Kaunas.

COPYRIGHT

The Malware Analysis Body of Knowledge (MABOK)

Craig Valli and Murray Brand
School of Computer and Information Science
Edith Cowan University
c.valli@ecu.edu.au
mbrand0@student.ecu.edu.au

Abstract

The ability to forensically analyse malicious software (malware) is becoming an increasingly important discipline in the field of Digital Forensics. This is because malware is becoming stealthier, targeted, profit driven, managed by criminal organizations, harder to detect and much harder to analyse. Malware analysis requires a considerable skill set to delve deep into malware internals when it is designed specifically to detect and hinder such attempts. This paper presents a foundation for a Malware Analysis Body of Knowledge (MABOK) that is required to successfully forensically analyse malware. This body of knowledge has been the result of several years of research into malware dissection.

Keywords
Malware Analysis Body of Knowledge, MABOK, Malware, Digital Forensics

INTRODUCTION

Aycock (2006, p.1-12) defines malware as "software whose intent is malicious, or whose effect is malicious". Analysis of malicious software is essential for computer security professionals and digital forensic analysts and is emerging as an important field of research (Masood, 2004). Malware is often targeted at organizations and is increasingly using anti-forensics techniques to prevent detection and analysis. Anti-Forensics is described by Rogers (2006) as "attempts to negatively affect the existence, amount, and/or quality of evidence from a crime scene, or make the examination of evidence difficult or impossible to extract". Kessler (2007, p1) extends this definition in a practical sense by saying "anti-forensics, then, is that set of tools, methods, and processes that hinder such analysis". The movement towards the employment of anti-forensic techniques in malware could be attributed to increases in penalties arising from prosecutions and also from the illicit financial gain that can now be achieved from employing malware nefariously (Larsson, 2007). Commercial Anti-Virus (AV) software is often limited in its ability to detect and remove malware. It is highly unlikely to detect new malware that is unleashed on the internet, corporate intranet or that has been customized to target specific networks. It is also unlikely to detect malware that has been customized to target specific networks (Masood, 2004).

It is undeniable that there is a digital arms race between malware developers and malware researchers. As soon as a technique is developed by one side, the other side implements a counter measure. Two of the major trends discussed by Symantec (2008, p.49) is that attackers are increasingly motivated by financial gain and that there are indications that malware development is becoming increasingly commercialised and developed by professionals with extensive software engineering abilities. Another trend is that malware has an increasing variety of techniques available to hinder the forensic analyst (Falliere, 2007; Ferrie, 2008; Yason, 2007). This can include detection of the tools used by the forensic analyst and prevention of analysis via anti-debugging, anti-disassembly, anti-emulation, anti-memory dumping, incorporation of fake signatures and code obfuscation. Signature based detection of malware is dependent upon an analyst having already analysed the malware and extracted a signature as well as the end user having updated their malware signature file. Heuristics are dependent upon correct implementation. Although these techniques go some way in protecting a system they are far from infallible and only of minor assistance to the forensic analyst, especially if the malware is new or has been customised. The increasing availability of high speed network Internet connections has also enabled the rapid production and dissemination of the malware. All of these factors are contributing to increasing numbers of network borne malware with respect to volume, variety and complexity.

Security professionals in the field need to know how to determine if they are the target of an attack and how to eradicate or mitigate threats from their systems. This process of threat reduction can be assisted if security professionals have up to date methodologies and skill sets at their disposal.

The purpose of this paper is to present a Malware Analysis Body of Knowledge (MABOK) that can be used as a framework for competency development and assessment for the field of malware analysis. This body of knowledge has been the result of several years of research into malware analysis. Over 1000 samples of malware collected from a honeypot have been successfully analysed and the MABOK is an attempt to map the knowledge needed to accomplish successful dissection of malware.

THE PROBLEM WITH MALWARE ANALYSIS

The spectrum of malware that represents a real threat is expansive. A non exhaustive list includes rootkits, worms, bots, trojans, logic bombs, viruses, phishing, spam, spyware, adware, keyloggers and backdoors. No computing platform or environment is immune to these threats. Traditionally, malware is thought of as a virus or worm that has a single function or payload. The resulting countermeasure for traditional malware has been the employment of a removal tool that was initiated by signature detection or by recognition of heuristics defined by specific behaviours. These tended to be like the malware they were responding to in that they were unitary or singular in purpose.

Modern network borne malware is increasingly multi-partite in nature incorporating several infection vectors and possible payloads in the one instance. Signature based systems that rely on file hashing or similar functions that uniquely identify malware based on file contents are increasingly failing due to the mass customisation allowable with the use of frameworks such as *MetaSploit* (Metasploit LLC, 2008). Furthermore, anti-forensic techniques are widely deployed to obfuscate infection, hinder detection and retard eventual removal of the malware. This increasing complexity and entropy makes modern malware analysis a significant undertaking that takes considerable time, expertise and requires an extensive knowledge domain either in an individual or in coverage provided by a team of analysts.

Two fundamental techniques available to the analyst are static and dynamic analysis. Static analysis does not execute the code and the code is analysed via disassemblies, call graphs, searches for strings, library calls, and reconstruction of data structures, enumerations and unions within the code. This analysis technique is very time consuming and easily hindered by anti-forensics in the form of code obfuscation, packers and protectors which are increasingly being used by malware authors.

Dynamic analysis, in contrast, does run the code and the analyst observes its behaviour and interaction with the host and network via mechanisms such as registry, file and network monitoring tools. This technique is generally much easier to conduct than static analysis but is also easily hindered by malware that can detect the use of an emulation environment such as *VMware* (VMware, 2008) or the use of debugging tools such as *IDA Pro* (Hex-Rays, 2008). By detecting the use of these tools and environments, the malware can change its behaviour. Once detected, the malware can decide not to run its true payload and can run in a deceptive mode that makes it look like much less of a threat. It can delete itself together with any evidence, or if it is running with the appropriate privileges, damage or destroy the system that it is being run on or attached to (Smith & Quist, 2006). Zeltser (2007) uses an iterative and recursive technique that incorporates both the static and dynamic analysis techniques to extract the full functionality of the code in a recursive and iterative technique that spirals into the analysis from the higher level view to the more detailed view. This technique also facilitates the opportunity to discover and mitigate anti forensic techniques as the analysis process proceeds.

ANALYSIS PROCESS

A high level and simplistic view of the malware analysis process is depicted in figure 1 below. It shows malware as one of two inputs to the analysis methodology process which produces a report as an output. The generated results also feedback into the analysis methodology via an assessment process which can be used to adjust the methodology dynamically, or as a process improvement mechanism. Legal and ethical constraints serve as a bounding constraint to the process.

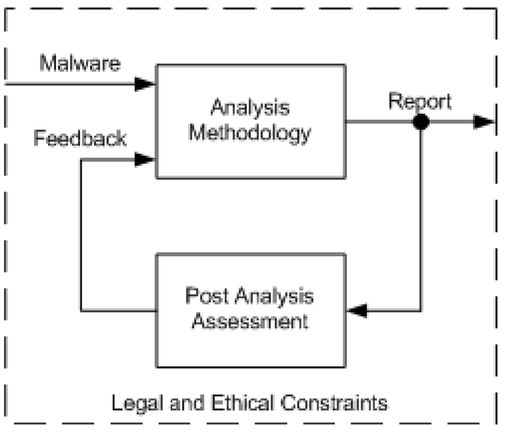


Figure 5 : Simplistic High Level View

An intermediate level view, as depicted in figure 2 below, must take additional constraints and inputs into consideration as well as the influence of a learning taxonomy. The malware can be collected from various sources and may have to be collected in a forensically sound manner. Collection can be influenced from the feedback mechanism of the post analysis assessment as well as from a learning taxonomy. The tools that are applied in the analysis may also have to be forensically acceptable and the analyst needs to understand their use and their limitations. The analyst will also require considerable cognitive abilities in malware ontology, programming, anti-forensics, environmental configurations and malware analysis. The learning taxonomy is central to the analysis process and both influences and is influenced by the collection process, analysis process and post analysis assessment. The MABOK traces directly to the learning taxonomy central to the malware analysis process.

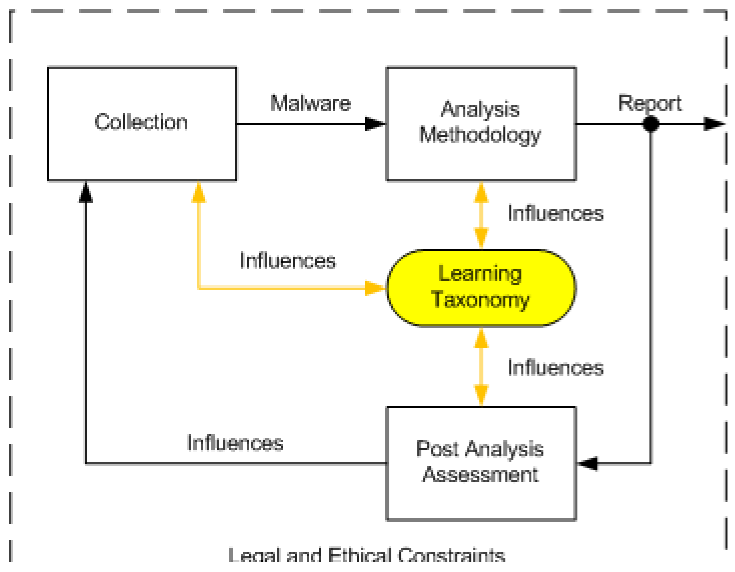


Figure 6 : Intermediate View

KNOWLEDGE DOMAIN

The knowledge domain proposed for the MABOK is presented in figure 3 below. The highest level stratum lists the categories of the MABOK whilst the branches beneath each category list the subject areas of each category in no particular order. The MABOK is a knowledge base required to perform effective malware analysis. The skill set represented in the MABOK may not reside within a single person but maybe reflected in a team.

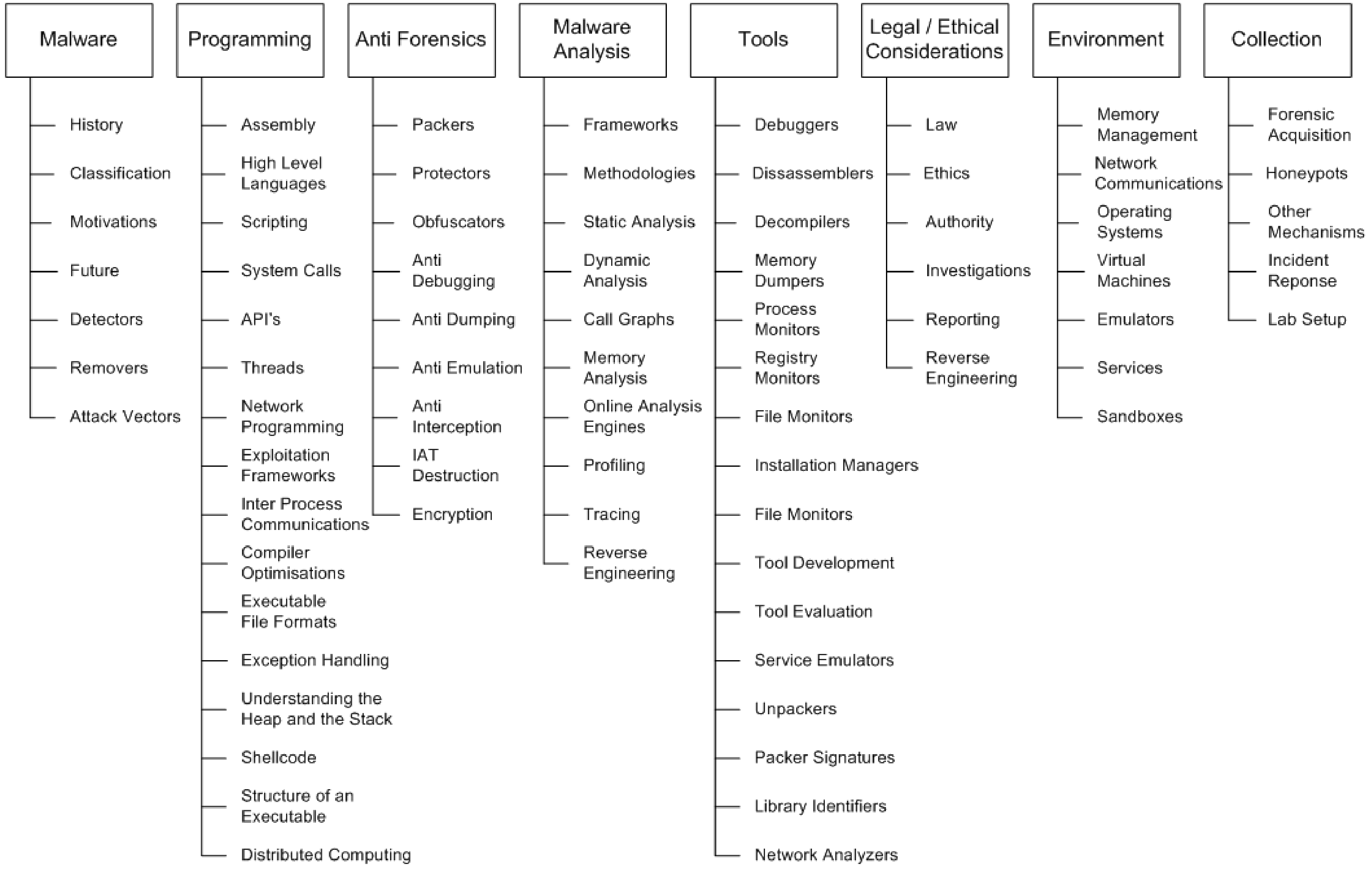


Figure 7 : Malware Analysis Body of Knowledge

Malware

Malware has an extensive history, and can be applied in a wide variety of malicious applications. It is important for the analyst to understand the motivations behind malware, including how it is applied and how it is detected. The effectiveness (or lack of) malware detection is also a very important consideration as well as detection techniques.

Programming

Programming skills are vital for in depth analysis of malware. Systems level programming, high level languages, scripting and even assembly language programming are important skills required to understand how malware is implemented and how it takes advantage of vulnerabilities. It is also an important skill set for the development of customised tools and for scripting disassemblers and debuggers. The poser of being able to script debuggers and disassemblers should not be underestimated in a malware analysis context. Many analysis tools now also allow additional functionality to be added by allowing users to write customised Dynamic Link Library (DLL) plugins or scripting languages such as *IDAPython* (Erdélyi, 2008) which integrates *IDA Pro* scripting with the *Python* (Python Software Foundation, 2008) scripting language.

Producers of malware also develop and utilise advanced programming techniques and technologies such as distributed computing to enable a competitive advantage over detection software and techniques. Therefore, it is imperative that a malware analyst also be well versed in cutting edge technologies and techniques.

Anti-Forensics

Often, static analysis of network based malware such as worms and bots cannot even start until the malware has been unpacked and the Original Entry Point (OEP) has been reached. Until then, the code appears as benign through obfuscation techniques incorporated into the malware by the authors of the code. Malware can also incorporate anti-debugging, anti-memory dumping, Import Address Table (IAT) destruction, anti-emulation and encryption to hinder the malware analyst. These techniques can be mitigated, but a thorough understanding of their use must be mastered to be able to achieve competent analysis of malware.

Malware Analysis

An adaptive, eclectic choice of techniques is required for analysis of malware. Various frameworks and methodologies such as static and dynamic analysis exist for the malware analyst to analyse malware such as *PaiMei* (Amini, 2008). Static analysis is the examination of source code logic and behaviours, whereas dynamic analysis is the monitoring and observation of the code as it executes. Both techniques have strengths. Obfuscation of code may render static analysis null and void. However, dynamic execution of that code segment may reveal the next code sections required for further static analysis. Other common software engineering techniques, such as profiling, tracing and debugging are also available, applicable and have utility in malware analysis. The diversity of malware modus operandi requires a range of approaches and techniques to perform successful dissection and analysis of the malware. The skills needed to perform competent analysis are profound, highly technical and are at the cutting edge of computer science.

Tools

A plethora of tools are available to the analyst including debuggers, disassemblers, de-compilers, memory dumpers, unpackers as well as many other tools common to the discipline of software engineering. All of these tools require niche expertise and a thorough understanding of the principles of their operation and the computers they execute on.

However, whether or not the tools are forensically sound and their use acceptable in a court of law is a matter that needs to be seriously considered. Some useful tools are available from hacking and software cracking sites that would not be considered forensically sound without considerable validation or black box testing.

Such tools could contain trojans and could easily hide a malicious purpose. They may not be forensically acceptable without significant due diligence on the part of the person or organisations using these types of tools. Other software cracking or reverse engineering sites have scripts for debuggers that can be easily and readily examined. These scripts are useful to extract the known algorithm for dealing with particular packers or to mitigate particular anti-forensic techniques used by creators of such software.

Legal/Ethical Considerations

Consideration must be given to legal and ethical issues in the analysis of malware. From a legal perspective, analysis of malware may require correct handling, preservation and presentation of evidence appropriate for a court of law. It may also include consideration of disclosure of private data that has been uncovered during the course of analysis. This private data could include, but not limited to, usernames, passwords, and personal particulars such as date of birth, address, relationships and financial data such as credit card numbers or bank account details.

Furthermore, there is an ethical consideration to ensure that the analysis process is secure and that the spread of malware is not a possibility. Also in the case of extraction or identification of a particular new malware, the responsible sharing of this knowledge is also a matter for ethical consideration.

Environment

Analysis of malware will typically require configuring a complete virtual environment suitable for it to run in, not only from an operating systems perspective, but also the inclusion of network infrastructure and services. Modern malwares are increasingly network borne and network enabled. So it may be necessary to provide an environment in which the malware can utilize commonly used services such as Domain Name System (DNS) server, Simple Mail Transfer Protocol (SMTP) server or an Internet Relay Chat (IRC) server. Establishment of this style of environment allows for the malware initiating communications with these services to allow the dynamic capture of target data to assist in the dynamic analysis of malware. This type of environment may be supported by a virtualised environment using commercial virtualisation environments such as *VMWare* or *Virtual PC* (Microsoft, 2007).

It should be noted that because malware can contain the ability to detect these virtualised environments as a result of their hardware and software fingerprints, the ability to configure real systems and devices may need serious consideration. This will require the configuration of a particular computing host environment, or network device or other system administrative tasks in order to achieve this. This type of environment would need strict control and isolation to prevent the spread of malware.

Collection

Malware can be collected in a variety of ways. It can be forensically acquired during incident response, through a honeypot or through the course of research. The handling of the collected malware may see a need to be handled in a forensically acceptable manner to enable the malware to be admissible in court.

CONCLUSION

Malware analysis is becoming an important field of specialisation for forensic analysts. Authors of malware are becoming increasingly profit driven and are incorporating techniques to make their code as stealthy and undetectable as possible. Malware is being written by professional programmers who are very knowledgeable in their craft. They have a very good understanding of digital forensic methods and endeavour to make forensic analysis as difficult as possible.

The knowledge domain required to competently analyse malware is very broad. This paper has presented a brief introduction to a Malware Analysis Body of Knowledge that would be suitable for establishing a framework for competency development and assessment for the field of malware analysis and for incorporation into academic curricula. A learning taxonomy is central to the malware analysis process and eight domain areas were identified. These areas include malware, programming, anti-forensics, malware analysis, tools, legal and ethical considerations, environment and collection. The application of Blooms Taxonomy to this body of knowledge is the next phase of development.

REFERENCES

Amini, P. (2008). PaiMei.

Aycock, J. (2006). *Computer Viruses and Malware*. New York: Springer

Erdélyi, G. (2008). IDA Python.

Falliere, N. (2007). Windows Anti-Debug Reference. Retrieved October 1, 2007 from http://www.securityfocus.com/infocus/1893

Ferrie, P. (2008). *Anti-Unpacker Tricks*. Retrieved October 8, 2008 from http://www.datasecurity-event.com/uploads/unpackers.pdf

Hex-Rays. (2008). IDA Pro.

Kessler, G. (2007). *Anti-Forensics and the Digital Investigator*. Retrieved May 04, 2008, from http://scissec.scis.ecu.edu.au/conference_proceedings/2007/forensics/01_Kessler_Anti-Forensics.pdf

Larsson, L. (2007). *Meeting the Swedish Bank Hacker*. Retrieved April 14, 2007 from http://computersweden.idg.se/2.2683/1.93344

Masood, S. G. (2004). *Malware Analysis for Administrators*. Retrieved 17 March, 2007 from http://www.securityfocus.com/infocus/1780
Metasploit LLC. (2008). Metasploit.
Microsoft. (2007). Virtual PC.
Python Software Foundation. (2008). Python.

Rogers, M. (2006). *Panel session at CERIAS 2006 Information Security Symposium*. Retrieved October 8, 2008, from http://www.cerias.purdue.edu/symposium/2006/materials/pdfs/antiforensics.pdf
Smith, S., & Quist, D. (2006). Hacking Malware: Offense is the new Defense. Retrieved July 24, 2007 from http://www.offensivecomputing.net/dc14/valsmith__dquist_hacking_malware_us06.pdf

Symantec. (2008). *Symantec Global Internet Security Threat Report. Trends for July-December 07*. Retrieved October 8, 2008 from http://www.symantec.com/en/uk/business/theme.jsp?themeid=threatreport

Yason, M. (2007). The Art of Unpacking. Retrieved Feb 12, 2008 from https://www.blackhat.com/presentations/bh-usa-07/Yason/Whitepaper/bh-usa-07-yason-WP.pdf
VMware. (2008). VMware.

Zeltser, L. (2007). *Reverse Engineering Malware: Tools and Techniques Hands-On*. Bethesda: SANS Institute.

COPYRIGHT

Forensic Acquisition and Analysis of the TomTom One Satellite Navigation Unit

Peter Hannay
SECAU – Security Research Centre
Edith Cowan University
p.hannay@ecu.edu.au

Abstract

Global Positioning Systems are becoming increasingly pervasive. The forensic acquisition and analysis of these units is of great interest as it has the potential to yield historic locational data for these units. Analysis of the TomTom one satellite navigation unit has resulted in a method to reliably extract historic data from these devices in a forensically sound manner.

Keywords

Global Positioning System, GPS, forensic methodology, digital forensics, GPS forensics, satellite navigation system, satnav, satnav forensics, TomTom, TomTom forensics.

INTRODUCTION

The first satellite navigation network became active in 1960. The network was known as TRANSIT and was used by US Navy Polaris submarines as a means of determining their current location (Parkinson, 1997). Since 1960 there have been a number of advances in satellite navigation technology as new networks are brought online and others shut down (Theiss, Yen, & Ku, 2005). The most commonly used satellite navigation systems at the moment are NAVSTAR and Glonnas, operated by the United States and Russian governments respectively (Polischuk & Kozlov, 2002). Each new satellite navigation network has allowed for increased accuracy, the technology is now at the level where it can be used for automotive navigational assistance, these satellite navigation devices are becoming increasingly common, as such the need for a method of forensically analysing these devices is becoming more critical.

A method for the analysis of the TomTom range of satellite navigation devices has been developed. This method allows for historic locational data to be retrieved from a TomTom device, in a way that does not alter the contents of the device or internal storage and allows for historic locations to be retrieved and presented in a reliable manner. It is important that this information be able to be retrieved as it has the prospective to provide information about the location of vehicles at a given time, this historic locational information has great potential in assisting with criminal investigations.

DEVELOPMENT & METHODOLOGY

When developing a methodology for the forensic analysis of these devices, a number of needs were considered, firstly the need to determine what information could be recovered, the need to recover this information in a repeatable manner and the need to recover this information without altering any of the data on the device being analysed (HB171, 2003). In order to do this a method to acquire the device had to be developed, following this a method to analyse the acquired data was needed.

Acquisition

In order to determine how to properly acquire the device, the device was analysed in terms of its components to establish where the relevant data is likely to be located. In this case there were several storage components, including flash memory, a GPS receiver module and an SD card. One of the goals of acquisition was to be able to acquire the device in a non-invasive manner, however the device does not provide a way to read the internal flash memory or GPS module without opening the case, as such the SD card was the only option that remained.

Once the SD card had been selected it was quite trivial to determine a way to read this media without impacting on the forensic integrity of the device by changing any of the data contained within. An SD card reader/writer was modified in order to render it unable to write to the SD card (Hannay, 2007). This modification was accomplished by simply bending a strip of metal located within the reader that detected the read/write state of the card, the result of which is that it would always detect the card as having been set to read only, this modification is shown in Figure 1 below.

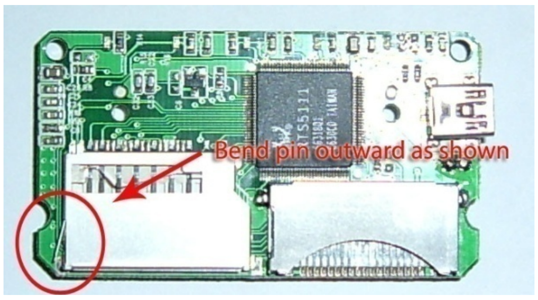


Figure 8. SD Card Reader with Read Only Modification

With the read only device constructed and tested, a procedure to perform the acquisition was developed based on the same method often used to forensically acquire hard disks. In this case the SD card was inserted into the reader and the device copied with a Linux utility known as ‘dd’. The method used to acquire the device is shown in Figure 2 below, commands entered have been bolded, whilst output has been made italic. The process (as shown in Figure 2) involves calculating the hash of the SD card, making a bit-stream copy via the ‘dd’ utility and then calculating the hash of the acquired data to ensure it matches the original exactly (ACPO, 2003).

```
$ md5deep /dev/disk2
f2452e9b6a984bef855c2b31aff099dc  /dev/disk2
$ dd if=/dev/disk2 of=test.dump
246016+0 records in
246016+0 records out
125960192 bytes transferred in 69.951924 secs (1800668 bytes/sec)
$ md5deep -l test.dump
f2452e9b6a984bef855c2b31aff099dc  test.dump
```

Figure 9. Method used to Acquire SD Card

The aforementioned method of acquisition was evaluated in order to ensure that it is not possible to alter the original copy of the data. In order to perform this evaluation acquisitions were performed several times, the device was connected to windows and linux systems and several attempts were made to directly write to the SD card. In each case it was not possible to contaminate the data on the SD card in any way. It should be noted however that the SD card reader modification may not work on all SD card readers, and as such, those wishing to make use of this procedure should ensure that they thoroughly evaluate any hardware used for forensic acquisition.

Analysis

In order to decipher any data that is acquired it was necessary to determine how historical location data was stored, and in the process, determine how to decode this data. The analysis process was developed in a series of stages, the first of which was data collection. Data collection involved establishing a baseline and acquiring this baseline, the baseline image was acquired from a TomTom One unit that had no favourite, home or any other locations set. A number of SD cards then had this baseline copied to them so that there was a point for comparison for each test.

The test procedure for data collection involved driving a series of predefined routes, each time the device was placed into a different mode of operation. It should be noted that each SD card was used for a single test and then acquired, this way it was possible to determine the impact each individual test had on the data acquired.

The data once acquired was compared to the original baseline in order to determine which files had been modified and what exactly had changed. In each case a single file was modified, the contents of this file were examined in order to determine what data could be recovered. A binary analysis was then conducted in order to determine the exact contents of this file and how it could be decoded.

RESULTS

Each location was presented within the file as a distinct record, each contained four coordinates and an ASCII representation of the location. It should be noted however that the information available depends heavily on the mode of operation that the TomTom One device was in at the time of use, a summary of these modes and the data recoverable in each instance is shown in Table 1 below. When examining these coordinates the first three sets form a triangle, this indicates the area which the device was in at the time the record was created. The fourth coordinate is used internally by the device so that it can navigate to that location again, this fourth coordinate is normally a point on a road nearby and can indicate the direction from which the location was reached.

Table 5 - Operational Modes of TomTom device and recoverable information

Operational Mode	Coordinants retreived	Coordinants consistant with destination	Textual Description Accurate	Known if destination coordinants reached	Path of movement known
Display current location only	No	N/A	N/A	N/A	N/A
Nav without reaching destination	Yes	Yes	Yes	No	No
Nav and reach destination	Yes	Yes	Yes	No	No
Nav and move in opposite direction	Yes	Yes	Yes	No	No
Search without enabling nav	Yes	Yes	Yes	No	No
Create favourite location	Yes	Yes	N/A	No	No

In order to validate the data recovered from the TomTom device, a second GPS unit was used during testing, this unit logged the vehicles precise coordinates throughout the testing process. These coordinates were compared to those recovered from the TomTom device, a sample comparison can be seen in Figure 3 below. It can be seen that there is a slight margin of error as the exact location (indicated by the green arrows) is slightly outside the area marked by the coordinates retrieved from the TomTom device (indicated by the surrounding crosshairs). The fourth navigational coordinate can also be seen to the lower right.

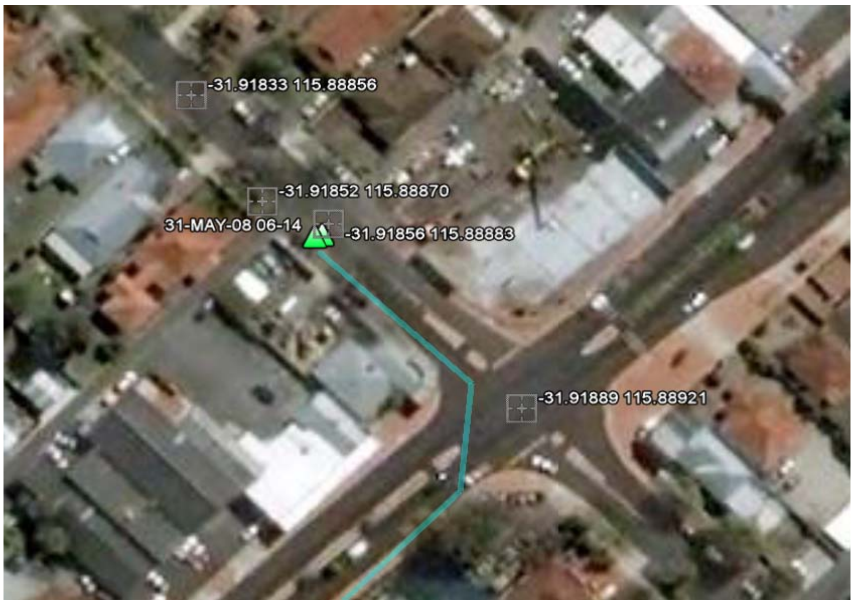


Figure 10 - Image overlaying locations retreived from TomTom with those taken from a secondary GPS unit

The acquired results are of significance as they demonstrate the data which can be recovered from a TomTom GPS unit, this information is of value as it allows forensic investigators to understand the scope of what can be determined by a forensic analysis of these devices. In this case the understanding that whilst a historic location may be present in the records of the TomTom device that does not mean that the device has actually been to that location, this type of understanding is critical when undertaking investigations involving satellite navigation based evidence.

ONGOING RESEARCH

Research into TomTom devices is currently ongoing, there are a number of alternate sources of information within the device, including onboard flash memory and the GPS receiver module itself. These sources of information will be examined in order to discover what further information may be discovered from the device. In addition to the research being conducted on the TomTom devices, a wide range of other devices is also being explored, these include automotive satellite navigation units as well as phones and other location aware devices.

CONCLUSION

Satellite navigation systems are becoming of increased forensic interest as their usage becomes increasingly widespread. There are a number of devices on the market currently, however these devices often use proprietary methods of encoding historic locational data and as such there is the need to understand specific file formats and encoding methods in order for forensic analysis to take place. In order to address this need research into the methodology for the acquisition and subsequent analysis of these devices has been conducted.

This research has lead to understanding of the file formats used by the TomTom One device. As such it is now possible to decode the historic locational data stored within these devices in a forensically sound and non invasive manner. It should be noted however that the data that can be recovered depends heavily on the mode of operation the TomTom One device was in at the time of use, with some modes leaving little to no trace of locational data and others providing a full set of coordinates.

Research into satellite navigation forensics is being continued, with focus towards on-board flash storage and GPS receiver modules. The aforementioned research is being conducted on TomTom devices, other automotive satellite navigation units as well as phones and other location aware devices.

REFERENCES

ACPO (2003). Good Pracice Guide for Computer based Electronic Evidence 3.0. Retrieved 16 Oct, 2007, from http://www.acpo.police.uk/asp/policies/Data/gpg_computer_based_evidence_v3.pdf

Hannay, P. (2007, 3rd December). *A Methodology for the Forensic Acquisition of the TomTom One Satellite Navigation System–A Research in Progress.* Paper presented at the 5th Australian Digital Forensics Conference.

HB171 (2003). *HB171: Guidelines for the management of IT evidence : handbook.* Sydney: Standards Australia.

Parkinson, B. W. (1997). Origins, evolution, and future of satellite navigation. *Journal of Guidance, Control, and Dynamics, 20*(1), 11-25.

Polischuk, G. M., & Kozlov, V. I. (2002). THE GLOBAL NAVIGATION SATELLITE SYSTEM GLONASS: DEVELOPMENT AND USAGE IN THE 21ST CENTURY. *34th Annual Precise Time and Time Interval Meeting*, 151-160.

Theiss, A. K., Yen, D. C., & Ku, C.-Y. (2005). Global Positioning Systems: an analysis of applications, current development and future implementations. *Computer Standards & Interfaces, 27*(2), 89-100.

COPYRIGHT

Preventing the Acquisition of Data from Virtual Machine based Secure Portable Execution Environments

Peter James[1]
Secure Systems Ltd
pjames@securesystems.com.au

Abstract

A Virtual Machine (VM) based secure Portable Execution Environment (PEE) provides a safe and secure environment that can be loaded into a host PC and an application executed with a degree of confidence that the application is separated, protected and little or no forensic evidence remains after the application has executed. A VM based secure PEE is characterised as a USB storage device containing a VM with a trusted guest operating system and application(s) which is stored in a protected partition, strong authentication to only allow an authorised user to load the VM into the host PC, and full storage device encryption to protect the confidentiality of the contents of the device. Secure PEEs provide an opportunity for organisations to issue a portable device to an individual (to perform a secure transaction on an available host PC) with the reduced risk to the organisation that neither malicious software (resident on the host PC) will infect the secure PEE device, nor sensitive data remnants (resulting from the transaction) will remain on the host PC hard disk drive after the secure PEE device has been removed.

A VM based secure PEE significantly reduces the opportunity to use dead forensic analysis techniques to acquire evidence of the occurrence of a transaction. However, VM based secure PEEs are susceptible to the acquisition of data through monitoring software and live forensic techniques. This paper considers the mechanisms that can be used to prevent various monitoring and live forensic techniques acquiring data from a VM based secure PEE. An attack scenario is presented to provide the context for the analysis of VM based secure PEE device vulnerabilities and why it is important that such a device would be required to counter hostile monitoring and forensic analysis. An overview is given of the security mechanisms provided by the type of VM based secure PEE under consideration and how those mechanisms combine to limit the opportunity for data acquisition through dead forensic techniques. The vulnerabilities of VM based secure PEEs with respect to malicious software and live forensic techniques are enumerated and discussed. A comprehensive set of countermeasures are proposed and analysed. The paper concludes by considering the most appropriate countermeasures to include in a VM based secure PEE to prevent the live acquisition of data...

Keywords

Secure Portable Execution Environments, Virtualisation, Virtual Machine Vulnerabilities, Securing Virtual Machines, Digital Forensics.

INTRODUCTION

The convenience provided by public Internet access centres has increasingly resulted in such centres being used to perform sensitive Internet transactions by individuals unaware of the associated risks. Given the totally open access to the PCs in these Internet access centres no level of confidence can be assumed in PC security; it is readily conceivable that such PCs have been compromised by malicious software which can exploit any Internet transactions. In addition PCs often "considered safe" (e.g. the home PC), which are used for a variety of tasks and by a number of different users often lack best practice security (e.g. anti-virus & anti-spyware software and modem/router enabled firewall capabilities). These "considered safe" PCs may contain malicious software unbeknown to the user which may be able to exploit Internet transactions. An option considered by some organisations is to issue employees and/or customers with secure portable execution environments (secure PEEs) to be used to perform sensitive Internet transactions on PCs for which no level of trust can be assumed. Secure PEEs provide trusted functionality to reduce the opportunity of malicious software exploiting sensitive data processing or an Internet transaction.

[1] Peter James is registered on a Professional Doctorate programme at the School of Computer & Information Science at Edith Cowan University. Peter is the Managing Director of Secure Systems Ltd.

In this paper a secure PEE device is considered to be a portable Universal Serial Bus (USB) storage device containing a trusted operating system and application that can be uploaded into a PC and used to perform a transaction with a high degree of confidence that the transaction will not be exploited. A secure PEE device will also provide space to store data, strong authentication to prevent unauthorised access, device encryption to protect the confidentiality of information on the device and partitioning with differentiated access rights to separate and protect the secure PEE.

In *"Secure Portable Execution Environments: A Review of Available Technologies"* (James 2008) a comprehensive review and analysis of secure PEE technologies and products is given. The paper finds that a secure PEE that utilises a bootable OS provides a strong secure PEE that is resistant to malicious software. However, booting a USB device often requires a user to change the PC Basic Input Output System (BIOS) boot order which can be an unfriendly and sometimes a non-trivial activity. An alternative to a bootable OS that requires no interaction with a PC BIOS is for a secure PEE to utilise a virtual machine (VM) with a guest OS, i.e. a VM based secure PEE.

A VM based secure PEE is simpler to load and execute than a bootable OS based secure PEE because the user can load and execute the VM when the secure PEE device is plugged into a host PC running a fully booted OS. A VM provides an abstract execution environment separate from the physical PC. There are a number of different types of VM (Smith 2005). The type of VM considered in this paper is one that runs within (on top of) the PC operating system; often referred to as a type 2 hosted VM. A "guest" OS is hosted within the type 2 VM and the application is executed within the guest OS. However, a VM based secure PEE may be susceptible to any malicious software that maybe resident on the host PC OS and also susceptible to certain data acquisition techniques, e.g. live forensics and monitoring software. In this paper techniques are considered to limit the opportunities of malicious software, monitoring software and live forensic techniques to acquire data from a VM based secure PEE.

The optimal features for a secure PEE device are considered in *"Secure Portable Execution Environments: A Review of Available Technologies"* (James 2008). It will be assumed that the type of secure PEE considered in this paper will have all of the protection measures of the "secure PEE device 4" defined in James 2008; in summary these protection measures are:

- **Authentication:** The device will prevent access to its contents and the VM can not be loaded until an authorised user has entered the correct authentication credentials. The most convenient approach to authenticate a secure PEE device is to plug it into a PC (that has a booted and executing OS) and an authentication application is uploaded from the secure PEE device. Through the authentication application a user authenticates with the secure PEE device.
- **Device Encryption:** The secure PEE device will be fully encrypted using on-the-fly encryption to preserve the confidentiality of the secure PEE and data residing on the USB device. In this paper it will be assumed that on-the fly-encryption will be implemented in hardware.
- **Swap space (virtual memory/page file) and space for temporary files:** The VM guest OS will require swap space. The usual default approach is to create the swap space on the host PC hard disk drive (HDD). Also applications executing on the guest OS may write temporary files to a 'temp' directory/folder on the host PC HDD. To prevent data remnants residing on the host PC HDD, following the use of a secure PEE, the device will be configured to:
    - provide swap space (virtual memory/page file) for the secure PEE VM guest OS; and
    - ensure the secure PEE VM guest OS and application(s) write all temporary information to available allocated space on the secure PEE device.
- **Partitioning with Differentiated Access Rights:** The secure PEE will support storage partitioning and role based differentiated access rights to partitions to preserve the integrity of the VM and any stored data on the secure PEE device. Such partitioning allows separation and isolation to be achieved. In addition to the provision of a partitioning capability the secure PEE device will also allow a partition to be defined as Read-Only. A Read-Only partition will be used to protect both the integrity of the VM (from malicious software) and 'valued' data.

Figure 1 presents a conceptual model of the configuration for the secure PEE device to be considered in this paper. The secure PEE device will be configured with three partitions. The first partition containing the VM will be set to Read-Only, to protect the VM's integrity. The second partition will have Read-Write access and will be used as swap space for the VM guest OS and for any temporary files created by applications. A separate third partition with Read-Write access will be used to separate and protect any user generated data.

The secure PEE configuration presented in Figure 1 provides strong security to protect against the acquisition of data from dead forensic analysis techniques. This configuration and the aforementioned protection measures are however, unable to protect a VM based secure PEE from malicious software once the VM has been loaded into a PC.

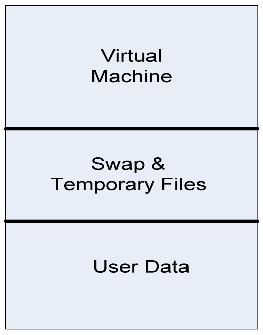


Figure 1

In this paper techniques are considered to limit the opportunities of malicious software, monitoring software and live forensic techniques to acquire data from a VM based secure PEE. Through the presentation of a threat model and an attack scenario the proposition for the identification of additional protection measures for a VM based secure PEE is given. The vulnerabilities of VMs are enumerated and discussed. A set of VM protection measures are presented and an improved VM based secure PEE proposed.

The following terms are defined in James 2008 and are restated below as they are used throughout this paper:

- **secure PEE device:** the secure platform/infrastructure consisting of a USB mass storage device configured with a secure PEE, secure storage space and possibly hardware based security mechanisms/technologies (e.g. encryption and secure partitions if available).
- **secure PEE:** the trusted OS, trusted application(s), security technologies (e.g. authentication and software encryption) and the appropriately configured hardware security mechanisms (if any) of the secure PEE device.
- **secure PEE OS:** the trusted OS component of the secure PEE.
- **VM based secure PEE:** a secure PEE that utilises a VM to host the secure PEE OS.
- **trusted OS:** an OS that has been acknowledged as secure by the supplier and users. To be considered trusted the OS may have a reduced set of hardened functionality and/or been subjected to independent rigorous evaluation and testing.
- **PEE:** a portable execution environment that does not necessarily have any security technology nor has been specifically configured to be secure.
- **portable storage device:** a USB flash device (often know as a thumb drive or pen drive) or a USB HDD packaged in a portable enclosure.

THREAT MODEL AND ATTACK SCENARIO

Threats

The secure PEE protection measures discussed above provide countermeasures to the following threats:

- Acquisition of sensitive data remnants (resulting from an application storing temporary information) residing on a host PC's HDD following the completion of a network transaction through the use of dead forensic techniques; and
- As a result of loss or theft, unauthorised access is gained to the VM and sensitive data held on the device.

However, a VM based secure PEE is susceptible to the following threat:

- Malicious software or monitoring software (inserted by a hostile user or organisation) is able to capture information through the application of
    - memory acquisition including the use of live forensic techniques
    - keyboard logging
    - screenshot capture; and
    - reverse engineering.

Attack Scenario

Numerous different attack scenarios are possible. The attack scenario presented in this paper provides the context and rationale for the analysis of VM vulnerabilities and the identification of tools and techniques to be applied to limit the exploitation of the identified vulnerabilities when a VM is used in a secure PEE. Rather than consider attack options possible from an easily accessible and open Internet access centre, the attack scenario presented below is focussed on a very specific area of use.

VM based secure PEE devices are issued by an organisation to its employees to enable work to be performed remotely using a virtual private network (VPN) connection over the Internet. The VM based secure PEE will connect (through the VPN) to remote servers that contain sensitive/classified information, also the work performed by the employee using the VM based secure PEE may result in the generation of sensitive/classified data.

An employee will use the VM based secure PEE device and its applications on remote host PCs that the employee considers to be safe and secure. The host PCs are executing the Windows OS. However, due to the nature of the business conducted by the organisation it is likely that technically sophisticated and capable hostile organisations will attempt to acquire data through:

- *Embedding monitoring and data acquisition software on the remote PCs that can transmit data to the hostile organisation.*
- *Performing dead forensic analysis on the remote host PC subsequent to the use of the VM based secure PEE.*
- *Performing live forensic data acquisition either (possibly) during or subsequent to the use of the VM based secure PEE.*

The attack scenario may appear infeasible or extreme but for certain organisations (e.g. government agencies) considering home and/or remote working for employees such an attack scenario needs to be considered as hostile organisations (e.g. foreign intelligence agencies) would use any means available to gain access to sensitive/classified data. It is conceivable that such hostile agencies will have access to sophisticated monitoring and dead & live forensics tools and techniques to enable the analysis of:

- the host PC used for processing (assuming access to the PC by the attacker is possible); and
- a captured VM based secure PEE device and the host PC used for processing.

HOW THE VM BASED SECURE PEE LIMITS THE OPPORTUNITY FOR DEAD FORENSIC ANALYSIS

The protection measures of the VM based secure PEE device considered in this paper provide strong mechanisms to protect the confidentiality of data and the integrity of the VM based secure PEE. These protection measures limit both the opportunity to acquire data using dead forensic analysis techniques when the device is at rest and to compromise the integrity of the VM when the device is operational.

Dead forensic analysis is the analysis of a digital device once processing has stopped and power has been removed. The analysis involves the examination of any aspect of the digital device's storage media to acquire data using a tool set that does not require the OS of the device to be booted. The protection measures of the VM based secure PEE device limit the opportunity to acquire data from both the device itself and the host PC HDD.

To assist the reader to appreciate the capabilities of the type of VM based secure PEE proposed in this paper an overview is given below of how the device's protection measures limit both the opportunity to:

- acquire data using dead forensic analysis techniques; and to
- compromise the integrity of the VM.

Encryption

Full hardware based device encryption prevents the acquisition of meaningful data from the VM based secure PEE storage medium. Access to decrypted data is only possible once the device has been powered and successfully authenticated. Even if the device can be successfully dismantled to bypass the authentication mechanism and allow access to the storage medium, time consuming brute force attacks would be required. Whilst brute force attacks are possible, if a strong cryptographic algorithm like AES is used then significant computing resources (i.e. very high end supercomputing centres) and large amounts of time are required to identify the cryptographic key(s) and thus acquire data.

Unlike software based encryption, hardware based encryption does not store any encryption key(s) in the host PC's random access memory (RAM) and therefore it is not possible to acquire the secure PEE device key(s) from memory. Recent research has shown it is sometimes possible to acquire keys from a PC's RAM (Haldermany et al 2008) after the PC has been powered off, however in practice such key recovery is difficult in the extreme (Hannay et al 2008).

Strong Authentication

The authentication component of a VM based secure PEE prevents access to the contents of the device and the loading of the VM into the host PC until a user has entered the correct authentication credentials. Once successful authentication has occurred access to the contents of the device is possible with the device performing on-the-fly encryption/decryption so the user can read all available data (subject to partitioning access controls). The strong authentication prevents the application of dead forensic tools (running on the host PC) accessing any data on the VM based secure PEE; as access to data is blocked until the authentication process has been completed.

Swap Space/Virtual Memory

Configuring the VM guest OS of the secure PEE to write its swap file direct to the device and also to configure applications that run under the control of the VM guest OS to write any temporary data to the device prevents data remnants residing on the host PC HDD after completion of a VM based secure PEE session. Any analysis of the host PC HDD will not reveal data generated during the use of a VM based secure PEE; although the host PC OS logs may indicate a VM was loaded and executed.

VM & Guest OS State Information

The VM and guest OS may generate data on their respective states that is required to be stored so that when another VM based secure PEE session is initiated the system state from the previous session is restored. Like the swap space and other temporary data, the files that store the VM and guest OS system state can be stored on the VM based secure PEE device; to avoid leaving data remnants on the host PC HDD.

Partitioning

Partitioning allows the separation and protection of software and data; a partition need only be mounted as required and differentiated access rights ensure only authorised access is permitted to a partition. As shown in Figure 1 the VM based secure PEE considered in this paper utilises three partitions. The first partition containing the VM will have Read-Only access to prevent any malicious software running on the host OS compromising the integrity of the VM.

The second partition will have Read-Write access and will be used as swap space for the VM guest OS, for any temporary files created by applications and system states. As identified above using a dedicated partition for all temporary data generated by the VM guest OS significantly limits the opportunity to acquire data remnants from the host PC HDD upon completion of a VM based secure PEE session.

A separate third partition with Read-Write access will be used to separate and protect any user generated data. A partition dedicated to storing user data ensures the partition containing the VM can be set to Read-Only without preventing user generated data being stored on the device.

ACQUIRING DATA FROM A VM BASED SECURE PEE

The type of VM considered in this paper utilises the OS of the host PC as an execution platform. The host PC OS can provide a platform for malicious/monitoring software. For a host PC for which no level of trust can be assumed the possibility of the host PC OS supporting attacks (through embedded malicious software) on the VM and its guest OS (& applications) is feasible. Techniques like keyboard logging and screenshot capture are typical of the type hostile acquisition methods used.

Attackers will, if logistically possible, utilise live forensic techniques to acquire sensitive data. Live forensic analysis involves the acquisition of data from a PC whilst it is still executing and therefore enables data to be acquired from the PC's memory. Another advantage of live forensic analysis is that any mounted HDD partitions that are encrypted when the data is at rest (i.e the PC is powered off) will provide full access to the plain text data when the PC is executing.

The work performed by (CSIRO 2008), (Ferrie 2007) and (Ormanday 2007) on VM vulnerabilities have provided excellent information sources to support and supplement the author's experience and knowledge.

Detecting the Presence of a VM Based Secure PEE

An attacker or embedded malicious software needs to avoid detection and therefore should only launch an attack if a VM based secure PEE has been identified. The following methods could be used to identify if a VM has been loaded and is executing:

- ***Using Standard OS Features:*** Malicious software could query any or all of the standard Windows Autorun, Task Bar, Task Manager or Windows data structures to identify if a VM is executing.
- ***Tools to Detect Processes:*** A VM could be detected by a range of specialist detection (e.g. Process Explorer) that have been created to identify executing processes; particularly if the processes are using concealment techniques.
- ***Checking OS Capabilities:*** Malicious software could search:
    - the Windows registry for entries that may identify the presence of a VM.
    - the list of loaded Windows DLLs to identify DLLs used by a VM.
    - the list of installed drivers to identify drivers used by a VM.

Memory Acquisition

The memory management function of an OS:

- controls access to, and the allocation of, the PC RAM
- implements a virtual memory system (often known as swap space or paging) which extends and optimises the use of the PC RAM by using part of the PC HDD; and
- provides memory isolation for processes.

As outlined above, to prevent virtual memory pages being placed on the host PC HDD, a VM based secure PEE device allows the VM guest OS to be configured to utilise a dedicated partition on the device for virtual memory pages. Whilst the VM based secure PEE device prevents the VM's guest OS leaving sensitive data remnants (held in virtual memory pages) on the PC HDD, the device can do nothing to protect the guest OS virtual memory from a live attack when the VM based secure PEE is executing.

As the host PC OS provides the execution environment for the VM, the PC memory (RAM and virtual memory) utilised by the VM will therefore be managed and controlled by the host PC OS. Although OS' are designed to provide process isolation (i.e. a process should not be able to interfere with the memory space of another process) it is possible for a hostile OS to interfere with a process' allocated memory; the hostile OS (or an executing hostile process) could access and modify or acquire the contents of the memory. To be successful in modifying or acquiring the contents of the memory used by the VM the attacker would need to have both a detailed knowledge of how the OS implements memory management and the design and structure of the VM and the executing application. Information on how to modify and acquire memory is becoming increasingly available due to gamers publishing details on memory hacks for games.

Techniques to modify and acquire the contents of a PC's memory include:

- ***Inserting Malicious Memory Management Software:*** An attacker, who has access to the host PC OS and has a detailed knowledge of the OS memory management architecture and design, may be able to rewrite the memory management software to allow access to the VM's allocated memory. For such an attack to be successful a detailed knowledge of the use of memory (RAM and virtual memory) by the VM is required. The advantage of embedding malicious memory management software in the OS kernel is that the user will be unaware of its existence, the disadvantage is that the attacker needs undetected access to the host PC and a highly sophisticated knowledge of the OS memory management and VM used in the secure PEE.
- ***Utilising the Memory Management API:*** The OS virtual memory management API, which allows developers to allocate, release and modify virtual memory could be used by malicious software to capture the contents of the VMs allocated memory. The advantage of using the virtual memory API is that it is a published interface that malicious software can utilise, the disadvantage is that a well written VM or guest application can use access rights on virtual memory pages to prevent malicious software acquiring data.
- ***Memory Hacking Tools:*** Generated predominately by gamers, the increasing number of available memory hacking tools can be used to gain access to memory; such tools include:
    - Memory Hacking Software – MHS (Spiro 2008): allows a user to search and change data via a graphical interface.
    - Tsearch (Corsica Productions 2008): provides a similar capability to MHS.
    - FU Rootkit (FU Project 2008): allows kernel data structures to be accessed via a device driver that gets installed.

  The advantage of memory hacking tools is that they are readily downloadable from various web sites, however the main disadvantage is that it is difficult for the attacker to use these tools without detection as concurrent access (with the VM based secure PEE user) to the host PC is required.
- ***Memory Dump:*** The standard OS memory dump upon process termination can reveal sensitive data being processed by the VM. Malicious software can cause the VM process to terminate and a process memory dump to occur. An advantage of causing memory dumps to occur is that relatively simple malicious software can cause a dump, however the main disadvantage is knowing when to trigger the memory dump to acquire sensitive data; causing multiple dumps to occur (i.e. numerous memory dumps will increase the probability of acquiring sensitive data) will make the user suspicious and/or end the VM based secure PEE session due to the multiple process terminations.
- ***Firewire Access:*** Although an overt action, it is possible to acquire the contents of a PC's RAM using a Firewire direct memory access function (Woodward et al 2008). The technique involves connecting another PC to the target PC via a Firewire port and executing a Firewire memory access tool. The tool will allow the contents of the target PC's memory to be copied to the PC running the Firewire memory access tool. The advantage of this approach is that the whole contents of the RAM can be acquired. The disadvantage is that it is an overt action that would be extremely difficult to conceal from the user.
- ***Cold Boot Memory Access:*** As discussed above, recent research (Haldermany et al 2008) has shown it is possible to acquire the contents of a PC's RAM after the PC has been shutdown if the acquisition is performed soon after the poweroff has occurred. To work successfully the memory needs to be kept cold. The advantage of this approach is that it is possible to analyse the PC after the VM based secure PEE has been used and in theory acquire potentially sensitive information. The disadvantages include being able to gain access to the PC in a timely fashion to acquire meaningful data and the need to have cold RAM.

Keyboard Logging

Keyboard logging is most often used to obtain authentication credentials and is implemented by malicious/monitoring software executing on the host PC OS and intercepting keyboard input which is either:

- stored in an unused area of the host PC HDD and retrieved by the attacker at a later time; or
- transmitted directly to the attacker as the keyboard input is received from the malicious/monitoring software.

Attackers are able to embed malicious software into a process and receive input from a keyboard due to the way OS's allow the interception of messages (using hooks) sent by concurrently executing processes. Attackers also use the information gathering capabilities available in application programming interfaces (APIs) to understand the capabilities of executing processes. Attackers are able to abuse the process communication and API features of OS' to introduce keyboard logging capabilities. Keyboard logging can be implemented in the following ways:

- ***Changing Device Drivers:*** A device driver is part of the OS kernel and has unrestricted access to the host PC hardware. A device driver based keyboard logger is able to communicate directly with the keyboard and capture all input. An advantage of this type of key logger is that it is very hard to detect its presence within the OS kernel. However, disadvantages include:
    - the requirement for administrator privilege and access to the OS kernel to be able to install the device driver (into the kernel); and
    - due to the low level nature of the key logger it is not possible to determine the context of the input, i.e. all input is captured and filtering authentication credentials (or other sensitive information) from other input at the device driver level is not possible.
- ***Utilising Application Programming Interface Hooks:*** The Windows OS provides the capability, through API hooks (Ivanov 2002), to intercept inter-process messages. The API hook allows a malicious/monitoring dynamic link library (DLL[2]) to be inserted into a process which can intercept and forward messages to a keyboard logger process. The large amount of documentation on the use of API hooks, available on the Internet, enables an attacker to readily implement this type of key logger. Another advantage is that under certain circumstances it is possible to introduce an API hook based key logger without access to the host PC. A disadvantage in the use of an API based key logger is that it can be detected as it is an executing process.
- ***Exploiting Debug Code Injection:*** Debug code injection is a Windows facility provided to enable debugging of executing programs (where the source code is not available). The technique involves modifying the program's binary code (in the PC RAM) through the injection of interceptor code to output specific information. Debug code injection can be exploited by attackers to inject key logging code. Like API hooks an advantage of debug code injection based key loggers is that lots of documentation exists to enable an attacker to build such a key logger; Microsoft supports the Detours DLL to enable code injection. An important disadvantage is that the attacker is required to understand the executing process into which the key logger code is to be injected.
- ***Replacing Dynamic Link Libraries:*** Replacing a DLL with a DLL with the same name is another approach to installing a key logger. The key logger is implemented within a DLL function that a process utilises in 'good faith'. An advantage is that a malicious substituted DLL is hard to detect, however a disadvantage for the attacker is that every function in the original DLL needs to be present in the malicious replacement DLL.

The following techniques can be used to prevent a key logger process/application from being detected:

- ***Covert Existence:*** Methods include removing the process from the taskbar, hiding the application window, hiding tracing information of the application installing and preventing the process being listed in the task manager.
- ***Alternative PC:*** The key logger can be installed on an alterative networked PC.
- ***Renaming:*** Methods include modifying the key logger process' signature and renaming files used by the key logger.

[2] A DLL provides a capability to allow multiple programs to share a set of library functions. In the case of a VM based secure PEE the library function is linked to the VM at run-time with the host PC OS performing the binding.

Screenshot Capture

A captured image of the screen can be used to acquire data from a VM based secure PEE. Windows provides a number of capabilities to capture and store screen images, also it is possible to use the processing capabilities of a PC graphics card to acquire screenshots. Screenshot capture can be implemented in the following ways:

- ***Changing Graphics Card Device Drivers:*** A PC uses a specialist graphics card to manage and present the output on the PC screen. A graphics card device driver based screenshot capturer can capture all screen output and save as bit streams in a file(s). To successfully implement such a malicious device driver requires the attacker to have a detailed knowledge of the host PC's graphic card design. An advantage of this type of screenshot capture is that it is very hard to detect its presence within the OS kernel. However, a major disadvantage is that due to the low level nature of the device drivers it is not possible to determine the context of the output and therefore large volumes of screen displays will be captured.
- ***Standard Print Screen Capability:*** Windows provides a capability that allows the print screen key (present on almost all keyboards) to be pressed and an image of the screen is captured and placed on the Windows 'clipboard'. Malicious software can use the print screen capability by emulating the key press, once a screenshot image has been written to the clipboard the malicious software can use the standard clipboard API to move the image to a file. An advantage in using the print screen capability is that it is a reliable existing feature that can be easily implemented. A disadvantage is that the capability can be disabled.
- ***Graphics Device Interface (GDI+):*** The Windows GDI+ API is responsible in the Windows OS for managing the presentation of output to the screen. GDI+ presents each screen as a pixel map. The GDI+ API can be used by malicious software to make copies of the pixel maps and move the map to an alterative part of the PC RAM and then be exported as an image to a file. An advantage of using the GDI+ based screenshot capture is that it cannot be disabled. A disadvantage is that non-trivial malicious software needs to be developed (in comparison with the print screen approach) to exploit the capability.
- ***DirectX API:*** The DirectX multimedia library is a comprehensive capability used for presenting videos and graphics. DirectX uses buffers to store the screen display before sending it to a graphics card for display on the screen. Malicious software can use the DirectX API to retrieve the contents of the buffers, convert into a bitmap and export as an image to a file. Like GDI+, the DirectX capability cannot be disabled, but similarly it requires relatively complex malicious software to produce useful output and of course can only be exploited if the VM guest application utilises DirectX.

Reverse Engineering

Reverse engineering a VM based secure PEE will enable an attacker to understand how the VM (and its OS and applications) work by reconstructing the source code from binary code. Once a VM (and its OS and applications) has been reconstructed into source code the attacker can identify vulnerabilities and compile a plan of attack. Reverse engineering a VM based secure PEE will obviously require a highly skilled and experienced attacker.

Reverse engineering is generally performed by a range of techniques including:

- debug software to step through the VM code.
- emulation software which will allow snapshots of the state of the VM to be taken and analysed
- process memory dumps that can be analysed.
- forensic analysis and memory hacking tools to understand how the VM utilises the HDD and memory.
- packet sniffing software to analyse network traffic.

The advantage of reverse engineering is that it enables the attacker to build a comprehensive understanding of the VM based secure PEE implementation to enable attacks to occur. The disadvantages include the need to acquire a VM based secure PEE device to reverse engineer, and of course reverse engineering by itself does not allow data to be acquired; the technique needs to be used in collaboration with other techniques to mount a successful attack.

PREVENTING THE LIVE ACQUISITION OF DATA FROM A VM BASED SECURE PEE

It has been shown above that there are potentially a number of exploitable vulnerabilities in VMs that the protection measures of the secure PEE device cannot counter. However, a range of techniques exist that can be constructed into a set of countermeasures to address the vulnerabilities. The work performed by CSIRO (CSIRO 2008) provided valuable input in the identification of many of the techniques presented.

Preventing the Detection of a VM

If the presence of a VM based secure PEE executing on a host PC can be successfully hidden then an attacker and/or embedded malicious software will be ineffective. Most of the techniques to prevent detection counter the detection techniques identified above. Techniques to hide the presence of a VM include:

- ***OS Features Avoidance:*** The VM should avoid execution by Autorun and prevent an icon appearing in the Task Bar. Using an application and process naming convention that has no association to VMs may avoid detection by both the Task Manager and searching through Windows data structures.
- ***Hiding from Scanning/Monitoring Tools:*** Preventing the detection of a VM by sophisticated scanning/monitoring tools may be difficult and will depend upon the capabilities of each tool. For instance, the freeware tool Process Explorer (SysInternals 2006) provides detailed information about a process icon, command-line, full image path, memory statistics, user account and security attributes. Some of these process attributes would be hard to conceal by the VM.
- ***OS Capability Avoidance:*** The VM should be constructed to avoid placing entries in the Windows registry, and where possible not use the host PC OS DLLs. However, avoiding the use of installed OS drivers is extremely unlikely, for obvious reasons.

Memory Acquisition

The following techniques can be used to prevent or limit the opportunity from memory acquisition based attacks:

- ***VM and Memory Address Obfuscation:*** Obfuscation is a technique used to prevent the reverse engineering of software. Software obfuscation is implemented by creating hard to interpret code by masking the language syntax and grammar. The principles of obfuscation can be used to make it difficult for an attacker to locate sensitive data in memory. Obfuscation techniques can be applied as follows:
  - *Memory Address Obfuscation:* By changing (randomising) the address space of memory an attacker cannot acquire meaningful data from consecutive memory space. Address obfuscation can be implemented by the guest OS virtual memory management system. The disadvantages are the degraded performance (due to retrieving data from non-consecutive memory locations) and complexity to implement.
  - *VM Obfuscation:* By implementing a virtual memory management system within the VM (in addition to the host PC OS and guest OS virtual memory management systems) the level of complexity for the attacker and obscurity of data will increase. The disadvantages of VM obfuscation are both the complexity to implement and the performance degradation due to multiple virtual memory management systems operating simultaneously.
- ***Complex Data Structures:*** Approaches to making data structures complex and therefore difficult for an attacker to understand include:
  - *Mixing variables:* The methodology involves placing parts of one variable into another. Records are kept of where each of the various parts of the variables are located to enable reassembly of the correct value to occur. The disadvantage of this approach is that the attacker may be able to read the variables before they are mixed or obtain the record of mixing locations and then perform reassembly; there is also the degradation in performance due to mixing and reassemble of variables.
  - *Assigning the wrong data type:* By storing integers as strings, strings as integer arrays, etc, it is possible to make some memory hacking tools become confused and possibly crash. The disadvantage is that the sophisticated hacker is able to circumvent this technique.

- ***Memory Encryption:*** Encryption can be used to protect the contents of memory. Whilst encryption is probably the strongest mechanism available to prevent the attacker gaining meaningful data from memory, the cryptographic algorithm and encryption keys must be stored in plaintext in memory to enable execution. The attacker therefore may be able to acquire the algorithm and keys and reverse engineer the VM. Performance is also likely to be an issue.
- ***Preventing Overflow Attacks:*** Causing memory to overflow is a technique used by attackers to allow inserted malicious code to execute, i.e. code is inserted into available process memory space, then the code is able to execute when a buffer overflow is invoked. Overflow attacks can be prevented by setting the virtual memory pages with no execution rights. The advantage of this technique is that it is a well documented technique, however a disadvantage is that an OS has the capability to mark memory pages as “no execute”.

Keyboard Logging

The following approaches can be used to prevent or limit the opportunity from keyboard logging based attacks:

- ***On Screen Keyboard:*** A very popular technique to counter keyboard logging is to use an on screen keyboard (the keyboard is a screen image) where keystrokes are entered by pointing to the respective character with the mouse pointer and “clicking”; other methods (e.g. a stylus) can be used to select the required character on the screen. An on screen keyboard does not mean that it has to be used for all keyboard input, for instance the on screen keyboard could be used just for the input of authentication credentials; a number of Internet banking applications use this approach. The on screen keyboard can be implemented as:
    - a feature in the VM.
    - a feature in the guest OS; or
    - part of an application.
- ***Embedded Authentication Credentials:*** A technique to protect authentication credentials from capture is to embed them in a string of random text (Herley et al 2006). This technique can be used with passwords and personal identification numbers (pin) as follows:
    - when the password/pin dialogue box appears on the screen then perform the following steps:
        1. move out of password/pin dialogue box focus
        2. type any random input
        3. move into focus for the password/pin dialogue box and type the next character of the password/pin
        4. repeat the above steps 1 to 3 until the password/pin has been entered.

  Any keyboard logger will gather a large string of characters, as the keyboard logger will capture all input, however the password/pin dialogue box will only receive the characters entered whilst in focus. Any parsing of the input by malicious software will prevent the password/pin being identified.
- ***Preventing the use of malicious API Hooks:*** A technique that works to counter an API hook based key logger is to enter as the very first hook in the hook list a trusted hook to a trusted DLL. The trusted DLL will be responsible for passing only input to the VM and/or its guest OS. The trusted hook will only work if it is the very first hook.
- ***Simple Encryption:*** To counter a keyboard logger an application expecting input can issue a simple encryption key to the user which is used to encrypt the input. The keyboard logger would intercept meaningless characters but the application would be able to decrypt the input because it has the encryption key. The encryption key would need to be very simple to enable the user to calculate and enter data in a timely manner.

Screenshot Capture

The following approaches can be used to prevent or limit the opportunity of data acquisition from screenshot capture:

- ***Disabling the Standard Print Screen Capability:*** Windows allows the print screen capability to be disabled. However, this approach will not prevent sophisticated attackers as disabling does not prevent GDI+ and DirectX based malicious software.
- ***Use of Graphic Card Overlays:*** An approach to counter GDI+ and DirectX based screenshot capture is to use the overlay capabilities available in most PC graphics cards. By using the graphics card API code can be written that bypasses the OS and directly modifies the screen space to produce an overlay. An overlay allows graphics to be placed over the graphics being displayed by an application but the overlayed graphics cannot be captured by screenshot loggers because the functions performing the overlay are not part of the OS.
- ***Scrambling Screen Output:*** To prevent a screenshot logger capturing useful information the output on the screen can be scrambled with the exception of a small restricted area. The user accesses only the restricted area of the screen for sensitive operations. Whilst a screenshot capturer will capture the restricted area, if this area is kept small it may be difficult to identify meaningful information amongst the scrambled data.

Reverse Engineering

The main countermeasure adopted to prevent reverse engineering is software obfuscation (Ogiso et al 2003). As outlined above obfuscation is the process making software hard to read. Software obfuscation techniques include software compression, keyword substitution and the use/non-use of whitespace to mask language syntax and grammer.

In theory encryption could be used to encrypt the whole VM executable with only the encrypt/decrypt algorithm in plaintext, however in practice such an approach would require the cryptographic algorithm to execute under the control of the host PC OS and could therefore be susceptible to attack and the cryptographic algorithm compromised.

Detection of Malicious Software

Performing checks to detect malicious software could be performed as an alternative, or as an additional measure to the use of the aforementioned complex countermeasures. The issue with using detection as the sole countermeasure is that only known malicious techniques will be identified; also kernel level malicious software is unlikely to be detected. The best strategy is to use detection techniques to support other countermeasures. The following detection techniques could be used:

- ***Detection of Malicious DLLs:*** The host PC OS DLLs and API hooks (used by the VM based secure PEE) could be monitored by checking the modules loaded prior to execution and comparing against known modules for the DLL. Any identified unknown modules can be treated as malicious and action taken. This detection technique would require a DLL examination tool to be executed prior to loading the VM.
- ***Use of Anti-Virus/Anti-Spyware Tools:*** Prior to loading the VM, anti-virus and anti-spyware tools can be executed to identify and remove any known malicious software. The tools would need to be appropriate to identifying the type of malicious software that can subvert VMs, which may mean specific bespoke tools are required.

The advantage of the above two detection techniques is that an untrusted host PC OS can be identified before the VM is loaded. However, the disadvantages include:

- the time required to perform OS scanning and DLL examination
- the tools will only be as good as the database of known problems
- the inconvenience of having to determine the course of action to take if an untrusted environment is identified, i.e. is it feasible to remediate the host PC OS or should an alternative host PC be used?

An alternative detection technique is:

- ***In Parallel Monitoring:*** This technique involves a detection process(es) running in parallel with the VM. Such a process could execute on the host PC OS or within the VM, and would monitor:
    - any unexpected changes to the VM
    - if another process attempts to access the VM
    - the commencement of any suspicious processes on the host PC OS.

The advantage of this technique is that it does not require time consuming pre-processing detection software to be executed before the VM can be loaded. The disadvantage of parallel monitoring is that it is conceivable that the VM could be exploited before the detection process has identified any malicious software.

AN IMPROVED VM BASED SECURE PEE THAT LIMITS THE OPPORTUNITY FOR LIVE ACQUISITION OF DATA FROM A VM BASED SECURE PEE

As the specific brand of VM has not been explicitly stated the reader may have assumed a commercial off the shelf (COTS) product would be used, which would also be the author's desired approach. Access would be required to the VM source code to implement many of the countermeasures proposed in this paper. To preserve its intellectual property most commercial organisations do not publish a products source code. Therefore to achieve the desired security for a COTS VM the product vendor would be required to change the VM. Alternative approaches to using a COTS VM could include:

- developing a bespoke VM that implements all of the required countermeasures; or
- using a freeware VM and adding the countermeasure, however most freeware requires any added functionality to be published and made freely available – which would obviously then be available to an attacker.

In proposing an improved VM based secure PEE the commercial and logistical viability of implementing the required countermeasures are not consider. Observations may be made on the feasibility of implementing a countermeasure, otherwise it is assumed that all required functionality is implementable.

Summary of Attack Scenario

Important aspects of the attack scenario described above can be summarised as:

- The issuer of the VM based secure PEE has to assume the host PC is not secure even though the user may consider it to be secure.
- If the user considers the host PC to be safe, then it will most likely be located in an environment where physical access will be difficult. However, it must be assumed the attacker is able to gain physical access and have sufficient time to examine the HDD using dead forensic tools, install malicious software and possibly use live forensic tools if the PC has been left powered on.
- The host PC will be connected to the Internet and therefore malicious software can be pushed to the host PC.

An Improved VM Based Secure PEE

Table 1 below summarises the set of countermeasures that can limit the opportunity to perform live acquisition of data from a VM based secure PEE. For each countermeasure:

- confirmation is given on whether the countermeasure will be used in an improved VM based secure PEE; and
- comments provided on the effectiveness, useability and implementation.

The decision to include a countermeasure is based upon:

1. does the countermeasure address a vulnerability that could be exploited within the given attack scenario.

2. the level of inconvenience the implemented countermeasure will cause the user, i.e. will the VM based secure PEE now be slower and more difficult to use.
3. in practice could the countermeasure be implemented, ignoring the commercial and logistical viability.

Required Countermeasures for Improved VM Based Secure PEE		
Countermeasure	**To Be Used**	**Comments with respect to effectiveness, useability and implementation in an improved VM Based Secure PEE**
Preventing the detection of a VM – OS features avoidance	**Yes**	Relatively simple to implement and effective.
Preventing the detection of a VM – hiding from monitoring tools	No	Counters an unlikely attack, also difficult to implement.
Preventing the detection of a VM – OS capability avoidance	**Yes**	Even if a bespoke VM is developed it maybe infeasible to avoid the use of DLLs & registry entries by a VM.
Memory acquisition – VM & memory address obfuscation	**Yes**	Only possibly if a bespoke VM developed.
Memory acquisition – complex data structures	No	Complex to implement and unlikely to provide an effective measure.
Memory acquisition – memory encryption	No	Likely to be difficult to implement and resultant implementation slow to execute.
Memory acquisition – preventing overflow attacks	**Yes**	For COTS product it may not be possible to implement.
Keyboard logging – on screen keyboard	**Yes**	Can be built into secure PEE application.
Keyboard logging – embedded authentication credentials	**Yes**	Implemented through user procedure.
Keyboard logging – preventing the use of malicious API hooks	**Yes**	For COTS product it may not be possible to implement.
Keyboard logging – simple encryption	No	Likely to be complex for user and easy to break by attacker.
Screenshot capture – disabling the print screen capability	**Yes**	Can be implemented with guest OS
Screenshot capture – use of graphic card overlays	No	As graphic card APIs will differ any overlay countermeasure would not be portable across a arrange of different host PCs.
Screenshot capture – scrambling screen output	**Yes**	Can be implemented by the secure PEE application.
Reverse engineering - obfuscation	**Yes**	Only possible if a bespoke VM developed.
Reverse engineering - encryption	No	Whilst in theory this countermeasure may be possible, in practice it is likely only parts of the binary can be encrypted.
Detection of malicious software – detection of malicious DLLs	**Yes**	Implement within a detection application.
Detection of malicious software – use of anti-virus tools	**Yes**	Likely to require bespoke tools as standard anti-virus & anti-spyware are unlikely to find all the malicious software designed to attack a VM based secure PEE.
Detection of malicious software – in parallel monitoring	**Yes**	Likely to require bespoke tools as standard anti-virus & anti-spyware are unlikely to find all the malicious software designed to attack a VM based secure PEE.

Table 1

CONCLUSION

A VM based secure PEE device provides a secure platform for portable computing, however it is susceptible to attack and the live acquisition of data due to the VM's reliance upon the underlying host PC OS. In this paper a range of countermeasures have been proposed that can address the VM's vulnerabilities. An improved VM based secure PEE that incorporates countermeasures that are effective, usable and implementable (in theory) has been proposed. The field of virtualisation is developing rapidly. It is possible that VM vendors may consider implementing some of the countermeasures identified in this paper for security applications.

Future work could include a detail review of available VMs to determine the VM most likely to be suitable for use in a secure PEE either because it has some of the countermeasures required or could readily be changed to implement the required countermeasures. Other work could include a proof of concept of the countermeasures in a VM based secure PEE

REFERENCES

Corsica Productions (2008). "Tsearch - a memory scanner/debugger utility." Retrieved October, 2008, from http://duckduckgo.com/TSearch.

CSIRO (2008). "Virtual Machines: An Initial Analysis of Threats and Remedial Actions."

Ferrie, P. (2006) Attacks on Virtual Machine Emulators. Volume, DOI:

FU Project (2008). "FU Rootkit." Retrieved October, 2008, from http://www.rootkit.com/project.php?id=12.

Hannay, P. W., A (2008). Cold Boot Memory Acquisition: An Investigation into Memory Freezing and Data Retention Claims. The 2008 International Conference on Security & Management, Las Vegas, Navada.

Herley, C. F., D (2006). How To Login From an Internet Cafe Without Worrying About Keyloggers. Symposium on Usable Privacy and Security CMU.

Ivanov, I. (2002). "API Hook Revealed." Retrieved October, 2008, from http://www.codeproject.com/KB/system/hooksys.aspx.

J. Alex Haldermany, S. D. S., Nadia Heningery, William Clarksony, William Paulx, and A. J. F. Joseph A. Calandrinoy, Jacob Appelbaum, and Edward W. Felten (2008). Lest We Remember: Cold Boot Attacks on Encryption Keys. Proc. 2008 USENIX Security Symposium.

James, P. (2008). Secure Portable Execution Environments: A Review of Available Technologies. 6th Australian Information Security Conference, Perth.

OGISO, Y. S., M SOSHI, A MIYAJI (2003). "Software Obfuscation on a Theoretical Basis and Its Implementation." IIEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences Vol.E86-A (1): 176-186.

Ormandy, T. (2007) An Empirical Study into the Security Exposure to Hosts of Hostile Virtualized Environments. CanSecWest Volume, DOI:

Smith, J. (2005). "The Architecture of Virtual Machines." Computer 38(5): 32-38.

Spiro, L. (2008). "Memory Hacking Software (MNS)." Retrieved October, 2008, from http://www.memoryhacking.com/.

SysInternals. (2006). "Process Explorer." Retrieved October, 2008, from http://live.sysinternals.com/.

Woodward, A. H., P (2008). Forensic implications of using the FireWire memory exploit with Microsoft Windows XP. The 2008 International Conference on Security & Management, Las Vegas, Navada.

COPYRIGHT

Extraction of User Activity through Comparison of Windows Restore Points

Damir Kahvedžić
Computer Science and Informatics
University College Dublin,
Ireland
Damir.kahvedzic@ucd.ie

Dr Tahar Kechadi
Computer Science and Informatics
University College Dublin,
Ireland
Tahar.kechadi@ucd.ie

Abstract

The extraction of past user activity is one of the main goals in the analysis of digital evidence. In this paper we present a methodology for extracting this activity by comparing multiple Restore Points found in the Windows XP operating system. We concentrate on comparing the copies of the registry hives found within these points. The registry copies represent a snapshot in time of the state of the system. Differences between them can reveal user activity from one instant to another. This approach is implemented and presented as a tool that is able to compare any set of offline hive files and present the results to the user. Investigative techniques are presented to use the software as efficiently as possible. The techniques range from general analysis, in which areas of high user activity are pinpointed, to specific techniques, where user activity relating to specific files and file types is found.

Keywords

Registry Restore-Points User Activity Reconstruction

INTRODUCTION

System Restore is a process in Microsoft Windows that monitors key system changes on a user's computer. Whenever a change that could jeopardise the system's stability is detected, System Restore copies the core system files and stores them in a hidden directory ("C:\System Volume Information_restore{GUID}") in the files system before allowing the change to take place. If the subsequent change results in an unstable system the user can simply reload the last know good configuration and undo the damaging changes. Typically a large number of these time points, called Restore Points, are found in the system. They present a snapshot of the state of the system at that point in time. Differences between them can highlight significant user activity that could be useful to a criminal investigation.

A number of different files are monitored by System Restore and are archived in the restore point (Microsoft (2007)). The user can specify which file types they want to monitor by modifying a particular system parameter. However, left as default, the system restore archives the Registry, COM+ database, the IIS metabase and other specific file extensions. A log of the changes is also stored.

The most important of these are the registry hive files. The registry is a central hierarchical database used in Microsoft Windows operating systems to store information that is necessary to configure the system for the users, application, and hardware devices. It provides a single location where installed programs, user profiles and settings can be stored and managed. Analysing different values in the registry not only reveals currently installed programs and the state of the operating system but would also give clues to recent opened files, folders and network connection and other user activity.

This paper presents a new methodology for extracting user activity from Windows Restore points. We design and implement a forensic registry analysis tool that is able to compare different Restore Points and hives on a file by file and key by key basis. The results are presented to the investigators to help them in determining how the system was used between the snapshots. A methodology for using this tool in the most efficient way is presented. Initially we focus on extracting the differences between the registry hives however the same technique can be applied to other sets of files stored in the Restore Points.

Restore Point Creation

The number of registry Restore Points stored on the file system varies from computer to computer. The factors that influence their creation include how often new drivers and large programs are installed, if the computer is powered on constantly and whether the user has turned off restore point creation. Regardless of these factors, Restore Points are likely to be found in the majority of the file systems. Once created, they store an exact copy of the active hives of the system registry. Honeycutt, J. (2002) described when Restore Points are created.

On Schedule: The default is 24 hours.

On Program Installation: The system may be backed up when a user installs a program that uses a particular type of installer.

On Update: The system is backed up just before an update to the operating system takes place.

On System Restore: The system is backed up before a system is restored using one of the Restore Points.

On Driver Installation: Device drivers affect system stability so the system is backed up for security.

On User Request: Users can create manual restore points whenever they chose.

The Restore Points are kept on disk for up to 90 days and are deleted after this time. As a result, it is not uncommon to find many different copies of the state of the system in a typical forensic investigation. Although these Restore Points are extremely useful to roll back unwanted system wide changes, they are also an invaluable forensic resource to provide insight into the state of the system at a given time.

Test Setup

Throughout the paper we will use a test file system to illustrate the investigative techniques on a practical example. The system is a typical home computer using the Windows XP operating system. The computer configuration and other relevant data are shown below. All settings dictating the frequency of the restore point creation were left in their default values. Nevertheless, the creation of the Restore Points was not done at regular intervals. The differences between the Restore Point creations varied from a single day to 10 days. In this case study the Restore Points found in the system had a total time range of 2½ months. In the rest of the paper, the Restore Points will be referred to with respect to its creation name (i.e. RP11) and the number of days after the oldest hive they were created (i.e. RP11 (25 days)). In this way a perspective is maintained on the time range between two Restore Points.

Computer Manufacturer	Dell
Operating System	Windows XP SP2
Number of Restore Points	17
Time Range of Restore Points	2 months
Frequency of Use	Light to Medium Use

Table 6: Test system configuration

The above computer system is used to demonstrate how user activity can be recreated with the comparison of the Restore Points. Initially we focus on the registry hives within these points. The validity of the activity found was confirmed by interviewing the owner of the above system. The demonstrations use the SOFTWARE and ntuser.dat hives respectively, since they hold most of the interesting evidence. Other hives can be used in a similar manner.

RPCOMPARE

RPCompare, (*R*estore *P*oint *Comparer*), is a tool developed to address the issues raised in comparing Restore Points. It is designed to compare the points in an offline forensic environment and present the differences to the user. It does not use the WMI interface or any inbuilt Windows functions. The tool includes techniques that can carry out the registry comparisons on a number of different abstractions depending on the time limits of the user. Initially, it includes techniques to find information on which files are stored in the points and in particular can find all the differences between the registry hives throughout these points. It can highlight any of the keys and values that have been deleted, added or modified in the interim. The methodology for using the best technique and the potential advantage of using them is described in subsequent sections.

The rest of this section will detail the comparison function used for comparing registry branches and individual registry keys. The latter scenario is an invariably slower technique but captures all of the changes between the hives and can therefore give a more complete result.

RPCompare takes any number of similar hives and compares either their keys or the values with each other. Any keys or values that are found are extracted and tagged with Added, Modified or Removed with respect to the more recent registry. The results are then presented to the user for further analysis.

Registry Comparison Function

At first, RPCompare uses the naming conventions of the Restore Points to order the points. Each Restore Point is named RP*xx* with *xx* being an integer numbering the Restore Point (Bunting, S. (2008)). To compare the registry keys themselves, RPCompare utilizes the ``Last Written Time" values present in all of the registry keys. The value is updated by Windows whenever an operation to write or modify the key's data is carried out on the data of that key. Each key, including the root contains this information. The time written to the value depends on the system clock, which can be manipulated by the user to show an inaccurate time. For the purposes of this study, we assume that the time is an accurate reflection on the real time, and that the time is consistent throughout the restore points.

The comparison function used in RPCompare is recursive and traverses the length of each hive tree comparing every node's time values. All the relations are with respect to the earlier node. If a node has a different time value to its corresponding node in the next hive then that node is tagged as modified. If it does not exist in the next hive then it has been removed. Finally, if a new node has been found in the new hive then it has been added. Values are compared only for those keys that have been tagged as modified.

Performance

Since RPCompare compares every key to its corresponding key in another hive, the complexity of its execution is (n*m), where n and m are the number of keys in the hive. In the worst case if the key is the root of the hive, the whole hive will be compared. In our tests, comparison of a mature hive, such as a SOFTWARE hive, is very time consuming; as such we present a number of investigative techniques to concentrate on specific branches of the hive or to limit the time range of the comparisons.The next section details these techniques. They have been developed to give the investigator a progressively detailed view of the data. We classify the techniques into two categories; those that attempt to find large scale differences in the system such as installations/uninstallations of programs and those techniques that attempt to recreate the minute steps of the user. The latter techniques involve the processing of Most Recently Used (MRU) lists and other private attribute information that can highlight how the user used the system. A special processing needs to be carried out on these types of registry entries to understand their meaning and their relationship with the user.

LARGE SCALE HIVE COMPARISON

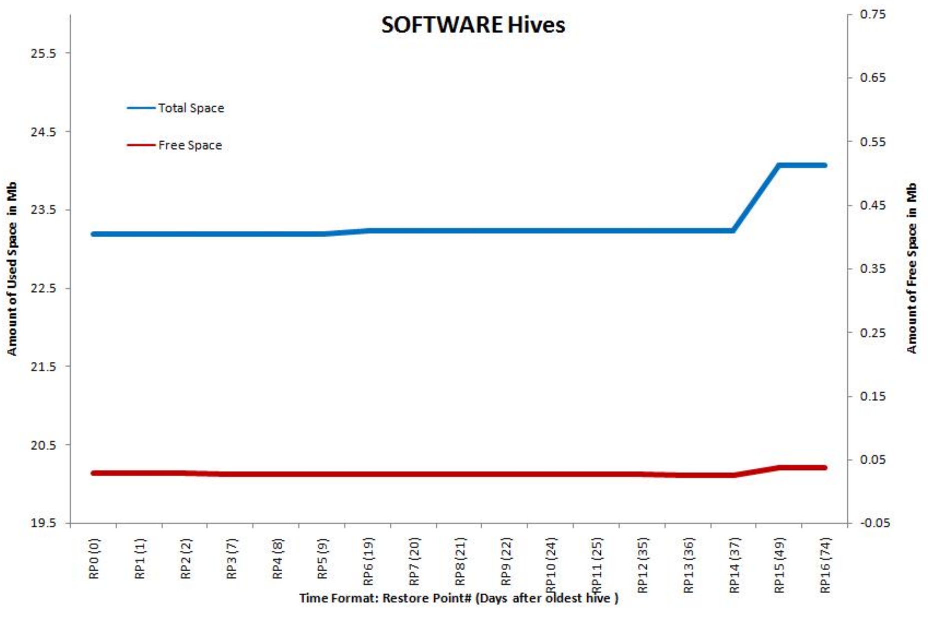
The investigator may need to expose large spots of activity and illustrate what was the general use of the system over a long period of time. User activity such as installation and uninstallation of programs and the addition or deletion of user accounts can be detected by comparing the registries of the system taken before and after the activity. The following technique illustrates a method on how the registries are compared in a progressively detailed manner. The process involves highlighting the areas of large activity by comparing file sizes first and then selectively comparing entire hives, then branches and finally values. The progressively detailed comparisons allow the investigators to streamline the comparison process and avoid spending too much time on complete hive comparisons.

Registry Size Comparisons

The first procedure entails comparing the sizes of the hives against each other. The differences can highlight some important changes that have occurred between time points in a relatively quick manner. Although small changes may not be noticeable, a large change to the hive, such as a program installation, can be easily spotted by comparing the hive sizes. The results can highlight a time point of high activity and bring it to the attention of the investigator for further investigation.

Variations in sizes highlight additions to the registry but it cannot show activity that *removes* keys from the hives. The extraction of unallocated space in the hive can be used to accomplish this aim. Whenever a key or a set of keys is deleted, due to an uninstallation or registry cleaning for example, the space left by the removed keys is marked as empty and is kept for future use (Russinovich, M.). The hives never shrink to compress this space and therefore do not reveal the uninstallation in its file size. In order to highlight this fact, RPCompare calculates the amount of free space in the registry alongside the total amount of used space. Sharp increases in the total amount of unallocated space signify large scale removal of keys.

Figure 1 shows the sizes of the hives in the case study. The size of the SOFTWARE hive increased at the latter part of the graph; between RP14 and RP15 which relate to 37th day and 49th day after the oldest restore point. Similarly, in the ``ntuser.dat'' file, the total space rises sharply between RP5 and RP6 or the 9th and 10th day after the oldest hive. Deallocated space has largely remained constant except between RP0 and RP1 and RP13 and RP14. The investigator can therefore narrow the range of the comparison for closer investigation.



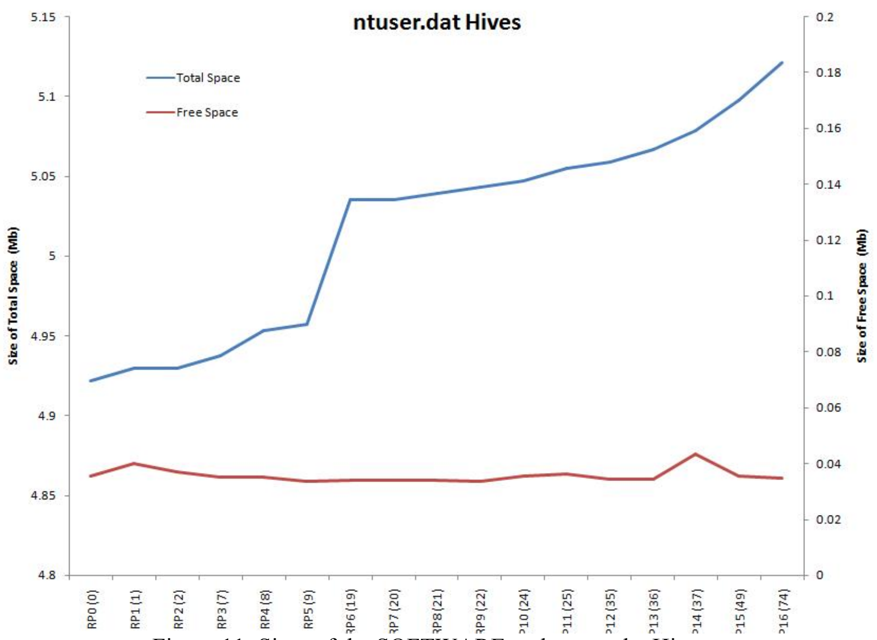
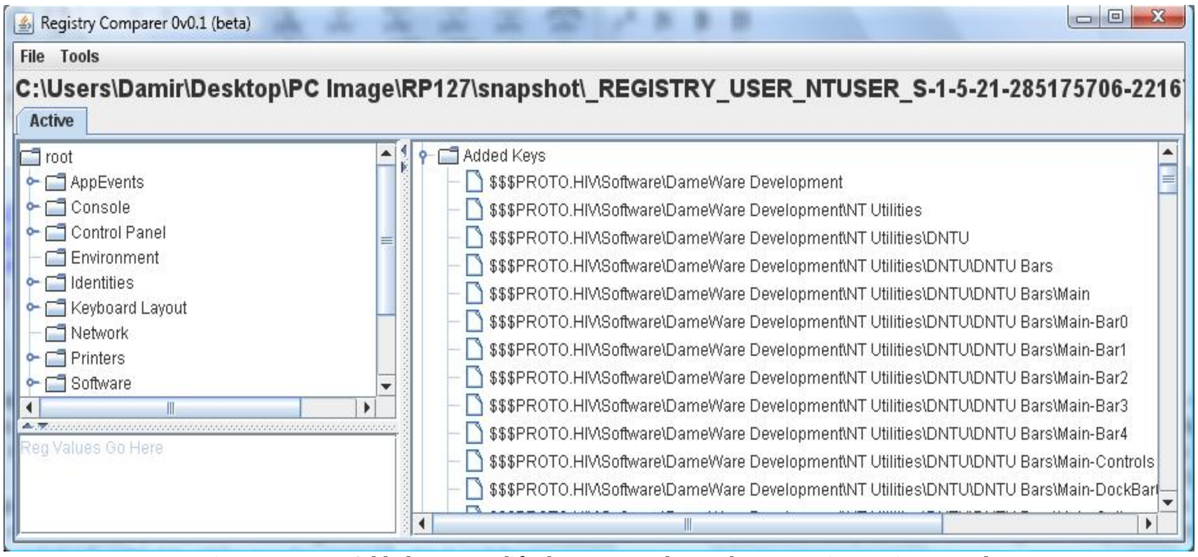


Figure 11: Sizes of the SOFTWARE and ntuser.dat Hives

RPCompare was executed on the SOFTWARE and ``ntuser.dat'' hives to recover the keys that were added at the above times. In the case of the SOFTWARE hive, RPCompare found that keys relating to the installation of the .NetFramework were responsible for the size increase in the 10 day time range between RP14 and RP15.

In the case of the ``ntuser.dat'' hive, the size of the hive began to increase at RP3 (7 days after the first restore point) with a huge size increase at RP6 and a steady increase thereafter. Comparison of RP5 and RP6 resulted in the identification of 143 Added, 111 Modified, 4 Removed keys. This result is shown in the Registry Comparer window of the program as shown in Figure 2. A majority of added keys are related to a DameWare (DameWare 2008) program. Upon further investigation, this program was found to be a PC remote control utility. Although in this case the installation was for innocent use, if the investigator is looking for a particular type of criminal activity this can be seen as vital evidence. The progressive increase in hive size from RP12 was attributed to new keys being added in the ``ShellNoRoam'' branch of the hive. These keys store window positioning preferences for each folder in the file system and are discussed further on in this paper. New ``ShellNoRoam'' keys indicate creation of new folders in the file system.


Registry Comparer 0v0.1 (beta)

File Tools

C:\Users\Damir\Desktop\PC Image\RP127\snapshot_REGISTRY_USER_NTUSER_S-1-5-21-285175706-2216

Active

root
AppEvents
Console
Control Panel
Environment
Identities
Keyboard Layout
Network
Printers
Software

Reg Values Go Here

Added Keys
$$$PROTO.HIV\Software\DameWare Development
$$$PROTO.HIV\Software\DameWare Development\NT Utilities
$$$PROTO.HIV\Software\DameWare Development\NT Utilities\DNTU
$$$PROTO.HIV\Software\DameWare Development\NT Utilities\DNTU\DNTU Bars
$$$PROTO.HIV\Software\DameWare Development\NT Utilities\DNTU\DNTU Bars\Main
$$$PROTO.HIV\Software\DameWare Development\NT Utilities\DNTU\DNTU Bars\Main-Bar0
$$$PROTO.HIV\Software\DameWare Development\NT Utilities\DNTU\DNTU Bars\Main-Bar1
$$$PROTO.HIV\Software\DameWare Development\NT Utilities\DNTU\DNTU Bars\Main-Bar2
$$$PROTO.HIV\Software\DameWare Development\NT Utilities\DNTU\DNTU Bars\Main-Bar3
$$$PROTO.HIV\Software\DameWare Development\NT Utilities\DNTU\DNTU Bars\Main-Bar4
$$$PROTO.HIV\Software\DameWare Development\NT Utilities\DNTU\DNTU Bars\Main-Controls
$$$PROTO.HIV\Software\DameWare Development\NT Utilities\DNTU\DNTU Bars\Main-DockBar

Figure 12: 143 Added, 111 Modified, 4 Removed Keys between Time Points 5 and 6

Registry Branch Comparisons

Once a branch of a hive has been found and suspected to contain evidence, RPCompare can take the root of that branch and compare it with similar branches in other hives. Because comparing a single branch is much faster than comparing the whole tree hierarchy, the investigator can concentrate on particular aspects of the hive at any given time relatively efficiently.

Returning to the DameWare example above, the DameWare Development key was compared to all the hives that were created after its installation. None of the hives reported any differences. This suggests that the program was rarely used. Upon further investigation it was found that the program was installed as a trial and not actively utilized. Progressively detailed key or branch comparisons can also be carried out if the investigator deemed it to be necessary.

USER ACTIVITY EXTRACTION

The registry contains many important locations that can be directly associated to the user and the way that the system was used at the time of the registry snapshot. The ``Most Recently Used'' (MRU) lists store evidence of files names, programs and other information that has been opened by the user in the recent past. They have been particularly highlighted as highly valuable pieces of user activity (Honeycutt, J. (2002), ForensicMatter.com (2008)). These locations are widely known and are actively analysed in most investigations. The investigator may look manually at the MRUs in the Restore Points but this can be extremely laborious. RPCompare can extract an MRU key, compare it across different Restore Points and extract the user activity that the MRUs held. This section elaborates on the MRUs and how they can be processed to gain understanding of user activity.

Registry MRU Management

The MRU key is a standard in Windows that store the most recently used items in the system. Each MRU `listens' for particular user activity and updates its content if this activity occurs. They store two types of values, a value for each of the entries and an index value, the MRU value, which stores a list of the entries in order of most recent.

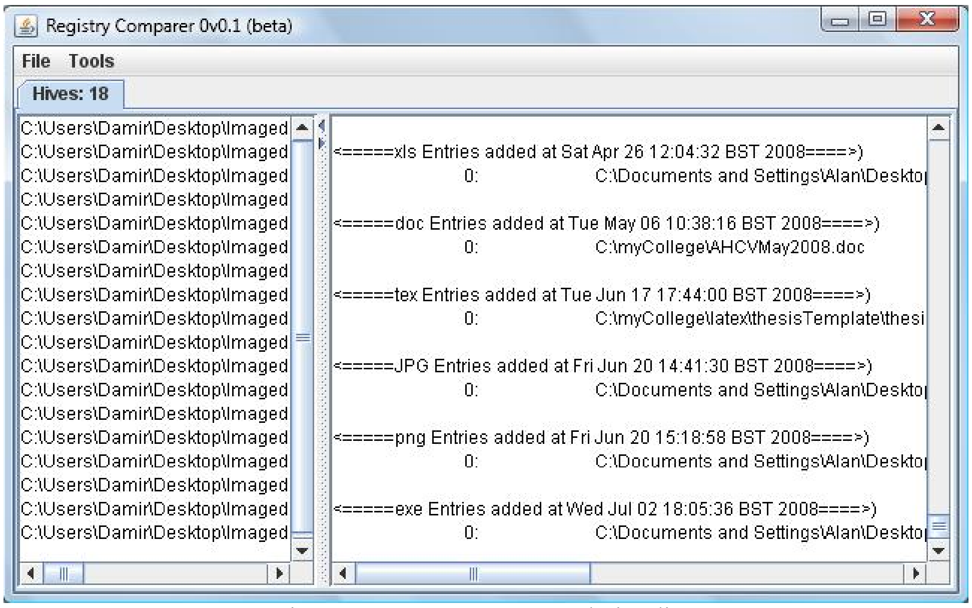
For example, the ``HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU'' key stores the most recent files opened in the Windows open dialogue. The subkeys of the key store entries for specific file extensions. Any new file opened is captured as a value in the extension's MRU with an unused index as its name and the filename as its data. The index is placed at the head of the MRU list signifying that it is the most recent. Only a limited number of indexes exist, so if there are none free, the oldest entry in the list is removed and its index is given to the new entry. If a command has been executed and is already present in the MRU key, then its index is simply upgraded in the MRU list. No new values are created.

RPCompare and the MRU Timeline

In order to compare the MRU keys correctly, RPCompare contains an algorithm to combine the MRU lists and disregard any reoccurring differences. Therefore, if only one new entry is found, only the new command will be highlighted in the report. In this way the analyst can get a clear history of the list without being confused by the other repetitive values.

RPCompare was executed on the `OpenSaveMRU' key to extract the user activity held in this MRU. A timeline of the different MRU's timestamps can be created to illustrate this more easily. Figures 3 and 4 show the different perspectives. Figures 3 show a textual representation (obscured for privacy) while Figure 4 shows a timeline where peaks indicate higher amounts of new MRU entries and therefore more user activity.

Very few new MRU entries are created even for an extended time range. This low user activity indicates that the computer system was used very lightly. This corroborates the stated system specifications in the Test Setup Section. Although only OpenSaveMRU was analysed, RPCompare can aggregate other MRUs from other locations in the registry in a similar manner. The more timestamps that are collected the more accurate the MRU time line becomes.



```
C:\Users\Damir\Desktop\Imaged    <=====xls Entries added at Sat Apr 26 12:04:32 BST 2008====>)
C:\Users\Damir\Desktop\Imaged              0:                  C:\Documents and Settings\Alan\Deskto
C:\Users\Damir\Desktop\Imaged
C:\Users\Damir\Desktop\Imaged    <=====doc Entries added at Tue May 06 10:38:16 BST 2008====>)
C:\Users\Damir\Desktop\Imaged              0:                  C:\myCollege\AHCVMay2008.doc
C:\Users\Damir\Desktop\Imaged
C:\Users\Damir\Desktop\Imaged    <=====tex Entries added at Tue Jun 17 17:44:00 BST 2008====>)
C:\Users\Damir\Desktop\Imaged              0:                  C:\myCollege\latex\thesisTemplate\thesi
C:\Users\Damir\Desktop\Imaged
C:\Users\Damir\Desktop\Imaged    <=====JPG Entries added at Fri Jun 20 14:41:30 BST 2008====>)
C:\Users\Damir\Desktop\Imaged              0:                  C:\Documents and Settings\Alan\Deskto
C:\Users\Damir\Desktop\Imaged
C:\Users\Damir\Desktop\Imaged    <=====png Entries added at Fri Jun 20 15:18:58 BST 2008====>)
C:\Users\Damir\Desktop\Imaged              0:                  C:\Documents and Settings\Alan\Deskto
C:\Users\Damir\Desktop\Imaged
C:\Users\Damir\Desktop\Imaged    <=====exe Entries added at Wed Jul 02 18:05:36 BST 2008====>)
C:\Users\Damir\Desktop\Imaged              0:                  C:\Documents and Settings\Alan\Deskto
```

Figure 3: OpenSaveMRU textual Time line

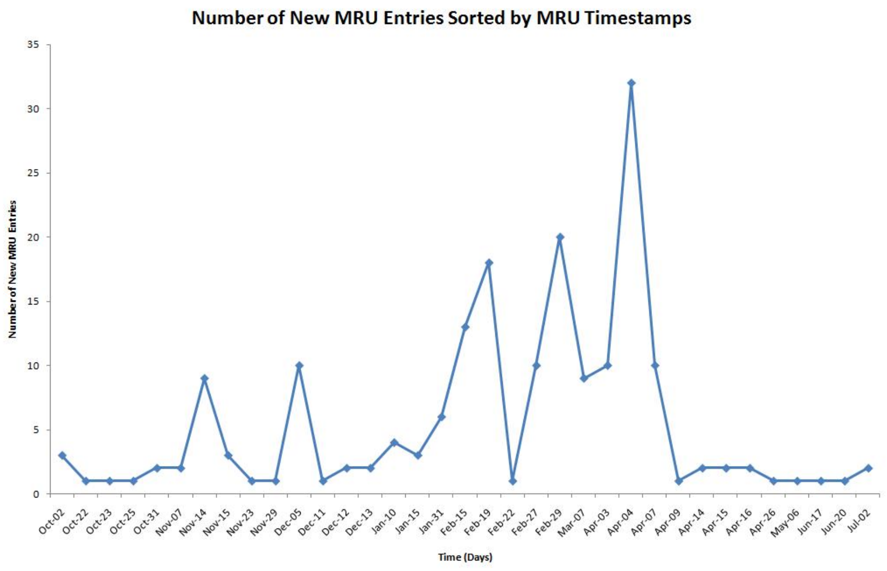
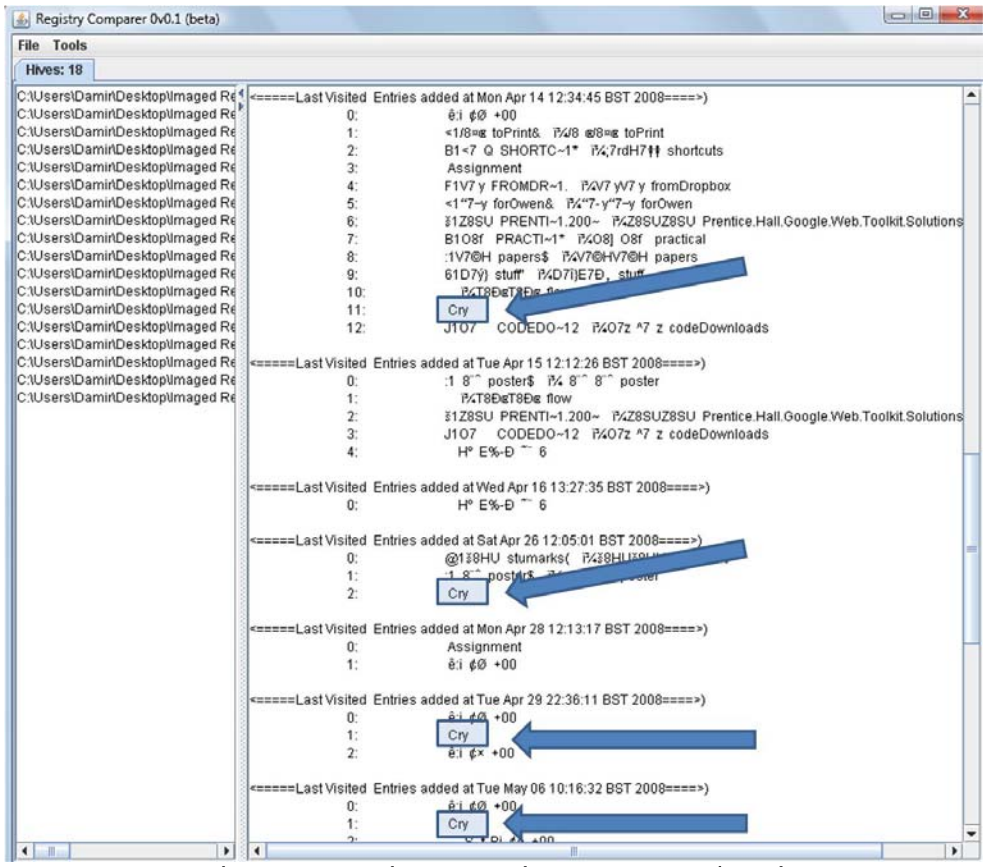


Figure 4: OpenSaveMRU Graphed Timeline



```
C:\Users\Damin\Desktop\Imaged Re
C:\Users\Damin\Desktop\Imaged Re
C:\Users\Damin\Desktop\Imaged Re
C:\Users\Damin\Desktop\Imaged Re
C:\Users\Damin\Desktop\Imaged Re
C:\Users\Damin\Desktop\Imaged Re
C:\Users\Damin\Desktop\Imaged Re
C:\Users\Damin\Desktop\Imaged Re
C:\Users\Damin\Desktop\Imaged Re
C:\Users\Damin\Desktop\Imaged Re
C:\Users\Damin\Desktop\Imaged Re
C:\Users\Damin\Desktop\Imaged Re
C:\Users\Damin\Desktop\Imaged Re
C:\Users\Damin\Desktop\Imaged Re
C:\Users\Damin\Desktop\Imaged Re
C:\Users\Damin\Desktop\Imaged Re
C:\Users\Damin\Desktop\Imaged Re
C:\Users\Damin\Desktop\Imaged Re
```

```
<=====Last Visited  Entries added at Mon Apr 14 12:34:45 BST 2008====>)
        0:        ê:i ¢Ø +00
        1:        <1/8¤ɛ toPrint&  ï¼8 ɐ/8¤ɛ toPrint
        2:        B1<7 Q SHORTC~1*  ï¼;7rdH7ↆↆ shortcuts
        3:        Assignment
        4:        F1V7 y FROMDR~1.  ï¼V7 yV7 y fromDropbox
        5:        <1“7–y forOwen&  ï¼“7-y“7–y forOwen
        6:        ៛1Z8SU PRENTI~1.200~  ï¼Z8SUZ8SU Prentice.Hall.Google.Web.Toolkit.Solutions
        7:        B1O8f  PRACTI~1*  ï¼O8] O8f  practical
        8:        :1V7©H papers$  ï¼V7©HV7©H papers
        9:        61D7ÿ) stuff'  ï¼D7ï)E7Ð, stuff
        10:         ï¼T8ÐɛT8Ðɛ flo
        11:       Cry
        12:       J1O7   CODEDO~12  ï¼O7z ^7 z codeDownloads

<=====Last Visited  Entries added at Tue Apr 15 12:12:26 BST 2008====>)
        0:        :1 8¨ˆ poster$  ï¼ 8¨ˆ 8¨ˆ poster
        1:          ï¼T8ÐɛT8Ðɛ flow
        2:        ៛1Z8SU PRENTI~1.200~  ï¼Z8SUZ8SU Prentice.Hall.Google.Web.Toolkit.Solutions
        3:        J1O7   CODEDO~12  ï¼O7z ^7 z codeDownloads
        4:          H° E%-Ð ¯ 6

<=====Last Visited  Entries added at Wed Apr 16 13:27:35 BST 2008====>)
        0:          H° E%-Ð ¯ 6

<=====Last Visited  Entries added at Sat Apr 26 12:05:01 BST 2008====>)
        0:        @1៛8HU stumarks(  ï¼៛8HU
        1:        :1 8¨ˆ post r$
        2:        Cry

<=====Last Visited  Entries added at Mon Apr 28 12:13:17 BST 2008====>)
        0:        Assignment
        1:        ê:i ¢Ø +00

<=====Last Visited  Entries added at Tue Apr 29 22:36:11 BST 2008====>)
        0:        ê:i ¢Ø +00
        1:        Cry
        2:        ê:i ¢× +00

<=====Last Visited  Entries added at Tue May 06 10:16:32 BST 2008====>)
        0:        ê:i ¢Ø +00
        1:        Cry
```

Figure 5: Subfolder MRU list of the Desktop folder. Highlighting `Cry' folder Access

Shell Bag MRU

It was seen in the Registry Size Comparison Section that a large number of ShellNoRoam keys are continuously created and were responsible for the size increase in `ntuser.dat' hive. These keys store positional information of windows for each folder in the file system. Each folder (bag) has its own MRU storing information on which of its subfolders was accessed most recently. RPCompare is able to process this information and present the user with user activity with respect to folders in a similar manner to the OpenSaveMRU. However, since every folder of the system has its own MRU, these keys contain much more information than the OpenSaveMRUs. It allows the investigator to highlight which folders where opened by the user and which were most widely used over a period of time.

Figure 5 show a sample of the timeline of the `Desktop' folder in the case study. It shows the activity relating to all its subfolders that were opened over a 3 week period. The folder `Cry' is highlighted in 4 out of 6 Restore Points. This indicates that the screen position of the folder `Cry' was changed at least 4 times in that time period. A user must have opened this folder and repositioned its window. This signifies significant user activity. If an investigator is looking for activity relating to a suspiciously named folder, this can be seen as vital incriminating evidence.

MRUs for other folders are found elsewhere in the registry in a similar format as the Desktop folder shown above. Using RPCompare, the investigator can extract any user activity relating to any folder in the system. The investigator can combine the evidence found in the Bag MRUs with those MRUs relating to the files themselves to paint an ever more detailed picture of user activity.

RELATED WORK

Registry comparison softwares have existed for a number of years in the registry analysis fields (RegDiff (Ver3.3), WinDiff (Ver5.1) (2001)). Exact registry specifications have not been published by Microsoft. Therefore, one of the techniques to see what the function is of any key is to take snapshots of the registry before and after known activity and analyse the difference (Microsoft (2008), Honeycutt, J. (2002)). However, these programs are limited in either the focus of the comparison and on what hives they operate on. Most of the registry comparison softwares are limited to comparing .REG files; ASCII versions of the binary registry hives. Each time a snapshot is taken, the relevant hive is exported with the inbuilt Microsoft RegEdit32 tool. This added step is undesirable in the forensic community where the goal is to avoid any modification of the original file. Other tools (RegDiff (Ver3.3)) can compare only the active registries and rely on commands provided by Microsoft Win32 API to extract relevant registry keys. Therefore they are unsuitable for offline registry analysis.

The tools above can only compare two hives at any one time and are not designed for digital forensic investigations. RPCompare differs to these programs since it has a digital forensic focus and aims to extract meaning out of the differences with respect to user activity. It can parse any offline registry hive even when it is extracted from a live system independently from any API.

Investigators have for a long time acknowledged the value of analysing the registry for evidence (Carvey, H. (2005), Carvey, H. (2007)). Guides have been published to explain to the investigators which keys are the most relevant to particular investigations (ForensicMatter.com (2008)). Most current forensic suites, EnCase (Encase (Ver6.8) (2008)) or Forensic Toolkit (FTK (Ver1.62.1) (2008)), contain registry parsers that can parse any registry hive files and present the contents to the investigator for analysis. However, the forensic analysis of the restore points has been treated the same as the analysis of the active registry. Namely, the investigator must open the hive files manually and access the different registry keys.

Research into analysis of the registry with respect to retrieval of deleted data has been done by Morgan, T.D. (2008) and Y. Kim et al (2008). The latter concentrated on retrieving still active keys not deleted by uninstalled programs. These clues are highly dependent on the uninstallation process of the software and may not reveal much information.

Research into Restore Points with respect to forensics has only been tackled recently (Bunting, S. (2008), Carvey, H. (2006), Harms, K. (2006)). Harms, K. 2006 has illustrated how the information stored in the Restore Points can be used to uncover evidence of a system intrusion. However, the author concentrates on the analysis of the ``change.log'' file only. This file is created at every Restore Point and tracks all files saved throughout the restore process. The registry hive files are not analysed.

CONCLUSION:

This paper presented a new approach for extracting user activity in a digital investigation. Namely, it focuses on comparisons of the Restore Points in general and the registry hives stored within them in particular. Differences between them can highlight changes in user activity that can be useful in digital investigation. We introduce a tool, RPCompare; an offline, self contained and integrated environment that can compare Restore Points and registry hives and present the user with the differences in a clear and logical interface. We also present a methodology using this tool to streamline the investigative process. Two techniques were presented in particular. The first focused on the registry in its entirety and attempted to ascertain time points of high user activity. This activity included what software was installed and removed and which keys were added or deleted relating to this activity. The technique, based on comparisons of hive size as well as content, was structured in a series of progressively detailed comparisons which highlighted areas of user activity with progressively higher levels of accuracy. The technique guided the investigator away from time consuming wholesale hive comparison and into much more efficient selective hive and branch comparison.

The second technique focused on the user trail itself and recovered and analysed the Most Recent Used (MRU) keys of the hives. The analysis of the MRUs requires specific processing for them to be investigated properly. User activity was extracted with respect to `file open' MRUs as well as `folder access' MRUs to get a complete user trail. In particular it was shown how RPCompare can reveal which folders and files the user accessed more frequently. Using both the timestamps of the MRUs and the hives, RPCompare presented an informative account of how the system was used by the user in a clear and useful manner to the investigator. This evidence can be extremely useful to any cybercrime investigation.

FUTURE WORK

RPCompare will be further enhanced to streamline the techniques presented above and to add new functionality in the comparison function. In this paper, we have concentrated mainly on the registry hives found within the points. Further work needs to be done to allow the software to utilise other similar files in the Restore Points and gather more information on what has changed between one point and another.

As described in the Large Scale Registry Comparison section, the investigator progressively narrows down their analysis of the registries by focusing on less and less branches. At the start of this process, a large number of differences may be returned that may be irrelevant to the investigation. In the DameWare scenario for example, a large number of HKLM\Software\Microsoft\Windows\ShellNoRoam\Bags\ keys were present. These keys store positional information on windows that the user has opened. Although it may be relevant to the investigator to parse these and extract useful information from them, in finding traces of added \ removed programs, these keys were not relevant. Future development of RPCompare will include filters to remove unwanted keys from the results or mark them as being irrelevant to the case.

Microsoft has enhanced the MRU list standard by storing the values in the MRU key in binary format, MRUListex. This allows the MRUListex keys to contain much more information than a single `Run' command or a filename. The data contained appears to be different for each type of key and is used extensively in Vista. In the HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\LastVisitedMRU key in Windows XP for example, every value stored in the key contains the filename of the document as well as the program that executed it. In Windows Vista, the same key is renamed to LastVisitedPidlMRU and evidently stores much more information. Processing of these MRUListex in Windows Vista has not been implemented in the RPCompare software.

REFERENCES:

Bunting, S. (2008) University of Delaware Police Computer Forensics Lab. *Restore Point Forensics*. URL http://128.175.24.251/forensics/restorepoints.htm, Accessed Sep 2008

Carvey, H. (2005). "The Windows Registry as a Forensic Resource". *Digital Investigation*, 2(3), pp201-205

Carvey, H. (2006). "Restore Point Forensics". URL http://windowsir.blogspot.com/2006/10/restore-point-forensics.html, Accessed Oct 2008

Carvey, H. (2007). "Registry Analysis", in *Windows Forensic Analysis DVD Toolkit*. Syngress Press, pp125-189

DameWare (Ver6.0). (2008) Dame Ware Development. URL http://www.dameware.com. Accessed: Sep 2008

Encase (Ver6.8) (2008) Guidance Software Digital Investigations URL http://www.guidancesoftware.com/, Accessed 28/Mar/2008.

ForensicMatter (2008). *Forensicmatter.com: Registry Hives*. Available at URL http://www.forensicsmatter.com/registry_hives.php, Accessed 19/May/2008.

FTK (Ver1.62.1) (2008) Access Data, URL http://www.accessdata.com/, Accessed 31/Mar/2008.

Harms, K. (2006). "Forensic Analysis of System Restore Points in Microsoft Windows XP". *Digital Investigation*, 3(3), pp151-158

Honeycutt, J. (2002) *Microsoft Windows XP Registry Guide*. Microsoft Press

Microsoft (2007). "Monitored File Extensions". URL http://msdn.microsoft.com/en-us/library/aa378870(VS.85).aspx, Accessed Oct 2008

Microsoft (2008). "How to use WinDiff to Compare Registry Files". URL http://support.microsoft.com/kb/171780, Accessed Sep 2008

Morgan, T.D. (2008) "Recovering Deleted Data from the Windows Registry". *Proceedings of Digital Forensic Research Workshop 2008*, pp33-42

RegDiff (Ver3.3). Available at URL http://p-nand-q.com/download/regdiff.html, Accessed Sep 2008

Russinovich, M. "Inside the registry". URL http://technet.microsoft.com/en-gb/library/cc750583.aspx. Accessed: Sep 2008

Y. Kim et al (2008) "Suspects' Data Hiding at Remaining Registry Values of Uninstalled Programs". *Proc. Of The 1st Int. Conference on Forensic Applications And Techniques In Telecommunications, Information, And Multimedia And Workshop*, 2008.

WinDiff (Ver5.1) (2001). Microsoft, Available at URL http://www.grigsoft.com/download-windiff.htm, Accessed Sep 2008

COPYRIGHT

Trouble in Florida: The Genesis of Phishing attacks on Australian Banks

Stephen McCombie
Cybercrime Research Lab, Macquarie University
mccombie@ics.mq.edu.au

Abstract

Today Phishing of Internet banks is a well know problem and globally is responsible for more than US$3 billion in fraud annually. To date there has been limited research into the individuals and groups responsible for these attacks. Considerable anecdotal evidence exists to suggest that transnational organised crime groups are involved in Phishing. The involvement of these groups, particularly those operating out of Eastern Europe, is of concern given their sophistication and resources. Earlier work by CRL@mq looked at a month of Phishing against one Australian financial institution and clustering indicative of a small number of groups being responsible was seen. To get a better picture of the nature of the groups behind Phishing we now look back to the genesis of attacks against Internet banks. The first attacks against Australian banks started in March 2003 and were in fact the first attacks of this kind against Internet banks globally. We examine these incidents as a case study and look at the individuals and organisations involved. The circumstances behind these attacks are *clearer now than might be imagined given none of the perpetrators were indentified at the time. We then briefly examine how much Phishing has changed in the intervening 5 years.*

Keywords

Computer crime case studies, cybercrime, Phishing, money laundering, e-crime, e-fraud, Internet banking fraud.

INTRODUCTION

Phishing is a well-known problem, accounting for as much as 1 out of every 281 Internet email messages in September this year (Messagelabs 2008). Gartner estimated that annual losses from Phishing attacks in the US alone went from USD$928 Million in 2005 (Litan 2005) to USD$3.2 Billion in 2007 (Gartner 2007). APACS, the UK payments association, reported UK online banking fraud was GBP£21.4 million in the first six months of 2008 (APACS 2008). Phishing attacks today are so frequent and numerous it is difficult to understand their true scope or to understand the actors behind them except in isolation. Earlier work at the Cybercrime Research Lab @ Macquarie University (CRL@mq) looked at Phishing against one Australian financial institution in July 2006 and examined the archival data available in that case (McCombie 2008). In that case study some clear indicators of a discrete number of attackers being involved in multiple was observed. That archival study examined data that covered just one organisation in one country over one month and as such a tiny portion of the total. Given that, this work is aimed at looking at an earlier time when Phishing was not an everyday occurrence against financial institutions, was little known and therefore relatively discrete. This time is very late 2002 to the middle of 2003. Examining archival material and other work from this period we get a picture of the circumstances behind this early Phishing and some insight into how and why it began the way it did. Surprisingly the participants behind the scenes may be easier to identify than we would expect given no one has been arrested for these early attacks. However at that time the nature of the problem was little known and certainly not well understood. What now seem rather suspicious associations may have been completely missed by responders and law enforcement at the time.

The rise of Phishing has seen the "Black Hat" hacker community in recent years transformed from a culture based largely on youthful exploration to one focused on criminal profit. With that shift markets for "Phishing" tools, for "Botnets", for zero day vulnerabilities and compromised credentials have been established to support this highly organised criminal trade. Spammers, malware writers, hackers and organised crime have come together as never before. Extensive efforts to facilitate the laundering of the illicit earnings of these crimes have also been observed with third parties known as "mules" utilised along with the services of various companies, such as Western Union, which perform international wire transfers. These mules, often unwittingly, act as agents to forward and launder proceeds of Internet banking fraud using their own accounts. The money is then drawn out in cash by the mule and then wired overseas.

Considerable anecdotal evidence exists to suggest that transnational organised crime groups are involved in this "Phishing". Their alleged involvement in these attacks has received extensive coverage in the press with headlines like "Dutch Botnet Trio Reportedly Connected To Russian Mob" (Keizer 2005), "Return of the Web Mob" (Naraine 2006). The US President's Identity Theft Task Force, set up to combat Phishing and other identity, theft reported in 2007,

"Law enforcement agencies also have seen increased involvement of foreign organized criminal groups in computer - or Internet-related identity theft schemes (The President's Identity Theft Task Force 2007)."

Groups from Russian Federation, the Ukraine and Romania were identified by the US Secret Service as being responsible for a number of the attacks (The President's Identity Theft Task Force 2007). The involvement of transnational crime groups, particularly those operating out of Eastern Europe, is of concern given their sophistication and resources. For example, Galeotti (2006) suggests that former members of the Russian Federal Agency of Governmental Communication and Information (FAPSI) - whose role was similar to that of the US National Security Agency - were recruited by organised crime groups as computer hackers when FAPSI was disbanded in 2003. Notably, this was around the same time Phishing became a significant problem and this case study relates. Galeotti also suggests other former USSR states such as Latvia are being used by Russian gangs to commit phishing attacks (2005). In February 2007, Microsoft's Chief Security Advisor in the UK, Edward Gibson (a former FBI Agent), was quoted by Viruslist.com saying, “it’s not the hacker crackers you have to worry about, but the Ukrainian mafia” (Kornakov 2007).

Some of the organised crime groups are believed to use legitimate enterprises they are involved in to support illegal activities. The large Russian organised crime group Tambov was believed to have used its petrol distribution company PTK’s IT division to commit phishing attacks (Galeotti 2008). Some Russian IT organisations are also suspected of being purely being vehicles for Internet crime such as the now infamous Russian Business Network (Zenz 2007).
Russian organised crime first entered the United States in numbers in the 1980s and set up significant bases in Brighton Beach New York and in Miami Florida (Friedman 2000). This case study concerns three businesses based in Florida.

To date there has been limited research into the individuals and groups behind “Phishing”. To effectively combat this problem we need to better understand the disposition and motives of these criminals. This paper aims to be a further step in delivering this important analysis to help government and industry address this problem.

PHISHING HISTORY

The term Phishing originated in 1996 to refer to a practice of tricking users into giving up their America On-Line (AOL) accounts to be used to distribute warez (pirated software) and other misuse. Originally the attacker would use instant messaging and purport to be an administrator from AOL. They would then ask users to provide their credentials. Later emails were used in a similar fashion. AOL actively policed the problem and by 2000 it all but disappeared (Ramzan 2007).

A New Type of Phishing

Starting in late 2002 a new style of Phishing attack began. The AOL phishers in the process of taking over AOL accounts had also got access to bank credit card details and they sometimes used them to use them to pay for services on the net (Ramzan 2007). Now taking the concept one step further, the target would be the banks themselves.

In 2000 despite the significant growth of Internet banking in a number of countries Internet banking fraud was virtually non-existent. Its notable that originally Commonwealth Bank of Australia’s NetBank used a fat client and National Australia Bank’s Internet Banking used client side certificates. These measures had been dropped by both these organisations by 2003.

While a number of observers have spoken of this change to Phishing most seem to indicate it started in the second half of 2003 or later (Grigg 2005)(Youl 2004)(James 2005)(Harley 2007) this research shows it was clearly happening in the first six months of 2003. The below timeline by Grigg shows Phishing switching to online banks in the end of 2003. It should be noted Grigg makes mention of two earlier attacks in his paper against e-Gold but this author was unable to find any references that support this or any other material to help understand the style of those early attacks (Grigg 2005).

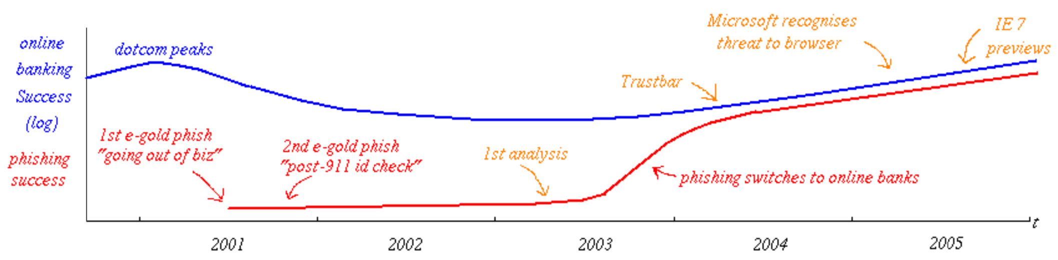


Figure 1 The Battle of Online Banking (Grigg 2005)

THE VICTIMS

E - Gold

The first victim of new style of Phishing was Florida based E-Gold not an Internet Bank per se. E-Gold, who in recent years has seen its' directors charged with money laundering (Broache 2007), is an Internet global payment provider who backs each transaction in gold. Customers hold their balances in gold rather than currency. E-Gold is believed to have had organised crime figures as customers prior to the attack and this may be part of the reason they became the first victim of this style of phishing attack. Jeffrey Taylor, U.S. Attorney for the District of Columbia, would later characterise them as having,

"Criminals of every stripe gravitated to E-Gold as a place to move their money with impunity (Department of Justice 2007)"

On Saturday 28 December 2002 during the quiet Christmas New Year period an email purporting to be from E-Gold support was spammed out to a large number of Internet users. It's said,

"Dear Valued Customer

- Our new security system will help you to avoid frequently fraud

  transactions and to keep your capitals in safety.

- Due to technical update we recommend you to reactivate your account.

Click on the link below to login and begin using

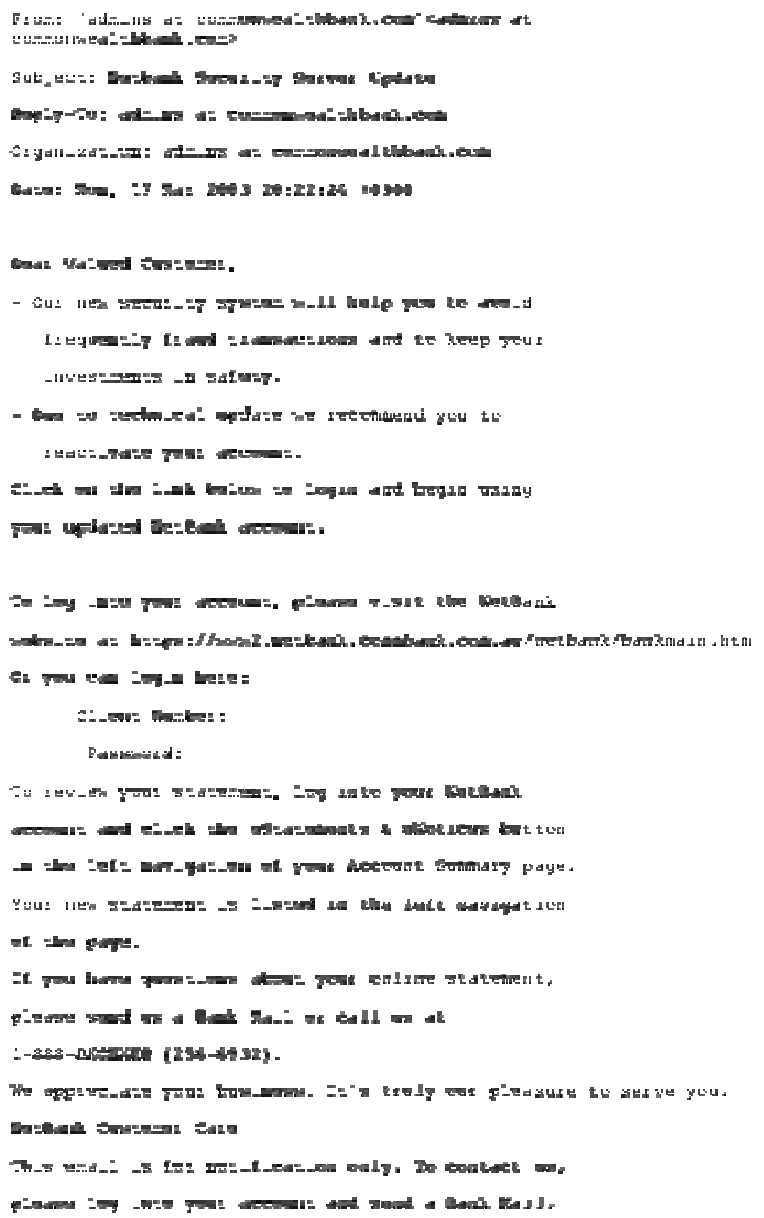
your updated e-gold account.

(Riley 2003)"

An email message like this is now a red flag to indicate a Phishing email, however despite the poor grammar, at the time it was a clever hook to get E-Gold credentials from customers. The web server hosting the Phishing page belonged to the IP range of 3d Wizards Hosting in Winter Park Florida on the address https://64.46.113.69/login.htm. The https certificate for that page belonged to cyberinvestigation.net, allegedly issued by ebizhostingsolutions.com (Riley 2003). E-Biz Hosting Solutions used some of the IP space of 3d Wizards and were also located in Winter Park Florida. One of the email samples seen by the author seems to have originated from a system at lsanca1-ar13-4-60-133-139.lsanca1.dsl-verizon.net [4.60.133.139] (Riley 2003). This appears to be a compromised system in the USA belonging to Verizon's DSL network.

Commonwealth Bank of Australia

The next victim was as different an organisation from E-Gold as one could find. Commonwealth Bank of Australia (CBA) formerly a wholly government owned bank in Australia. It is the largest Australian bank with a $58.2 billion market capitalisation as of October 2008 (Zappone 2008). The one thing it did share in common with E-Gold is its early presence on the Internet and its more advanced functionality for users to transfer their money. On Monday 17 March 2003 an email was sent out purporting to be from "admins at Commonwealth Bank". It used much of the same text as the attack on E-Gold and was again hosted on an IP belonging to 3d Wizards in Winter Park Florida on the address http://64.46.113.74/netbank/bankmain.htm. The Head of Security of the CBA was the former head of the Electronic Services Section of the Australian Federal Police (AFP) and he took no time in getting the AFP involved in investigating the matter. AFP agents from the Sydney office were assigned and in conjunction with NSW Police started an investigation. The law enforcement response was to follow the money. When compromised credentials were used and money transferred to a Croatian man recruited on a Croatian community website in Tasmania to be what would be later referred to as a "money mule". He was arrested by Police picking up the proceeds of one compromised account at a branch but as with money mules today was not able to identify the ultimate beneficiary of the fraud (Colley 2003). At the same time an apparent good citizen, Kevin Searle, who posted using the name Wombat to the news.admin.net-abuse.email newsgroup detailing the attack. He had contacted CBA indicating that this site was hosted on Florida and he also alerted Sydney Police and the Florida Computer Crimes Unit. Searle later told his story to Sam Varghese from the Sydney Morning Herald (Varghese 2003).

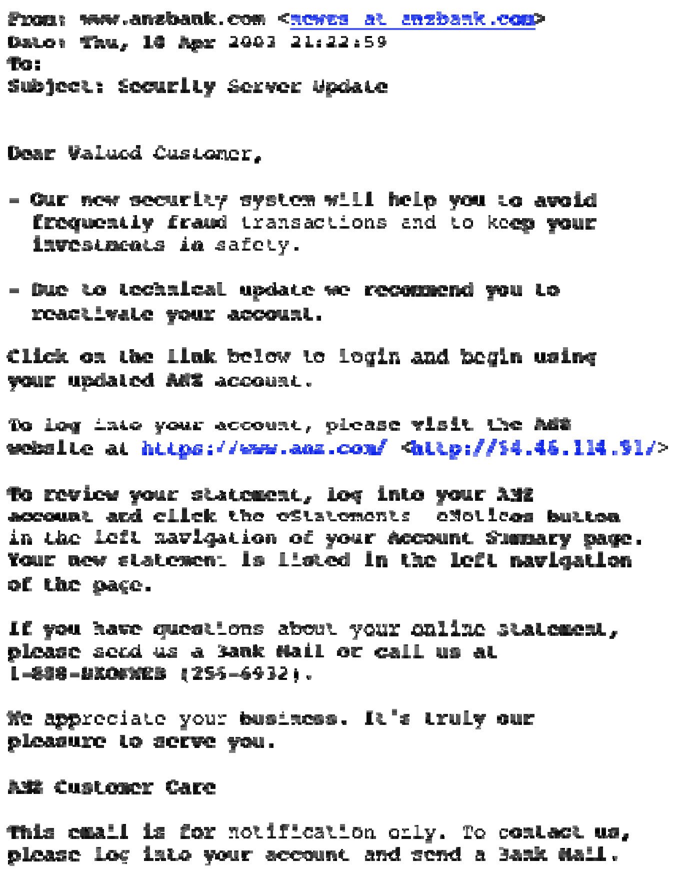


```
From: "admins at commonwealthbank.com" <admins at
commonwealthbank.com>

Subject: Netbank Security Server Update

Reply-To: admins at commonwealthbank.com

Organization: admins at commonwealthbank.com

Date: Sun, 17 May 2003 20:22:24 +0300


Dear Valued Customer,

- Our new security system will help you to avoid

   frequently fraud transactions and to keep your

   investments in safety.

- Due to technical update we recommend you to

   reactivate your account.

Click on the link below to login and begin using

your updated NetBank account.


To log into your account, please visit the NetBank

website at https://host2.netbank.commbank.com.au/netbank/bankmain.htm

Or you can login here:

      Client Number:

       Password:

To review your statement, log into your NetBank

account and click the "Statements & Notices" button

on the left navigation of your Account Summary page.

Your new statement is listed in the left navigation

of the page.

If you have questions about your online statement,

please send us a Bank Mail or call us at

1-888-[illegible] (256-4932).

We appreciate your business. It's truly our pleasure to serve you.

NetBank Customer Care

This email is for notification only. To contact us,

please log into your account and send a Bank Mail.
```

Figure 2 CBA email 17 May 2003 (Searle 2003)

ANZ

The Commonwealth Bank incident was publicised in the Australian and International media. Other Australian banks started to look at their vulnerability to similar attacks. They did not have to wait long. On 10 April 2003 another Phishing email was sent, this time targeting ANZ bank and coming from from "newzs at anzbank.com". ANZ Bank (Australia and New Zealand Bank) is Australia's third largest bank. The samples seen by the author originated from 0x50a104ef.virnxx9.adsl-dhcp.tele.dk [80.161.4.239] and d141-107-221.home.cgocable.net (d141-107-221.home.cgocable.net [24.141.107.221], which appear to be compromised systems in Denmark and the USA. The site was again hosted by 3d Wizards in Florida at the address http://64.46.114.91/ and used similar text the attacks of CBA and E-Gold. On its ftp port the server at that IP responded as server2013.ebizhostingsolutions.com. Another good citizen informed ANZ and passed on details of the hosting company and Adam Kling from E-Biz Hosting Solutions as a contact. ANZ contacted Adam Kling and asked for the site to be removed, which happened a few days after.

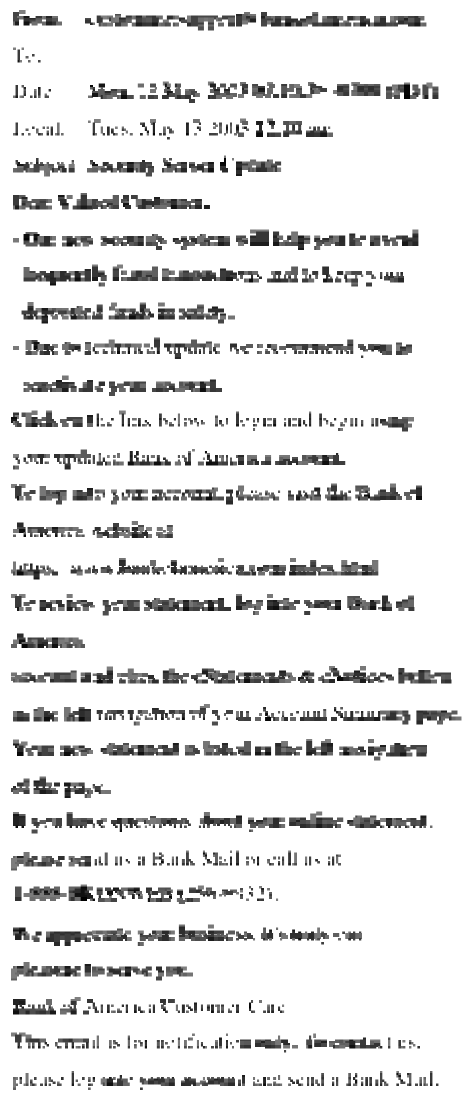


```
From: www.anzbank.com <news at anzbank.com>
Date: Thu, 10 Apr 2003 21:22:59
To:
Subject: Security Server Update


Dear Valued Customer,

- Our new security system will help you to avoid
  frequently fraud transactions and to keep your
  investments in safety.

- Due to technical update we recommend you to
  reactivate your account.

Click on the link below to login and begin using
your updated ANZ account.

To log into your account, please visit the ANZ
website at https://www.anz.com/ <http://64.46.114.91/>

To review your statement, log into your ANZ
account and click the eStatements  eNotices button
in the left navigation of your Account Summary page.
Your new statement is listed in the left navigation
of the page.

If you have questions about your online statement,
please send us a Bank Mail or call us at
1-888-BKONWEB (256-6932).

We appreciate your business. It's truly our
pleasure to serve you.

ANZ Customer Care

This email is for notification only. To contact us,
please log into your account and send a Bank Mail.
```

Figure 3 ANZ Phishing Email 10 April 2003 (Scheid 2003)

Bank of America

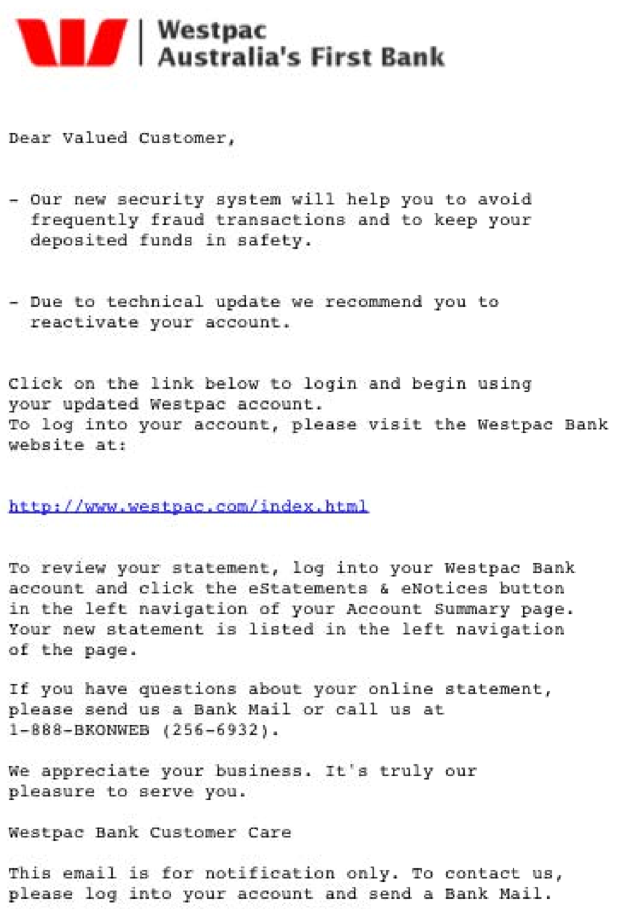
While other Australian Banks became increasingly concerned about a potential attack the next Phishing incident moved offshore. On 12 May 2003 a Phishing email was sent out targeting Bank of America. It again used similar text to the attacks on E-Gold, CBA and ANZ. The site this time was hosted by Verio a large hosting provider registered in Colorado and Florida at the address http://198.173.235.126/index.htm.



From: [illegible]
To:
Date: Mon, 12 May 2003 [illegible]
Local: Tues, May 13 2003 12:[illegible] am
Subject: Security Server Update
Dear Valued Customer,
- Our new security system will help you to avoid
  frequently fraud transactions and to keep your
  deposited funds in safety.
- Due to technical update we recommend you to
  reactivate your account.
Click on the link below to login and begin using
your updated Bank of America account.
To log into your account, please visit the Bank of
America website at
https://www.bankofamerica.com/index.html
To review your statement, log into your Bank of
America
account and click the «Statements & Notices» button
on the left navigation of your Account Summary page.
Your new statement is listed on the left navigation
of the page.
If you have questions about your online statement,
please send us a Bank Mail or call us at
1-800-[illegible].
We appreciate your business. It's truly our
pleasure to serve you.
Bank of America Customer Care
This email is for notification only. To contact us,
please log into your account and send a Bank Mail.

Figure 4 Bank of America Phishing Email 12 May 2003 (Jennings 2003)

Westpac
Australia's fourth-largest bank at this time was Westpac but was the second most popular on-line bank which had watched the recent events against its competitors Commonwealth Bank and ANZ closely. On 4 July 2003, US Independence Day they become subject of a Phishing attack. Again the same text was used as the previous banks and a site on IP space managed by 3d Wizards was involved using a domain belonging to E-Biz Hosting Solutions at the address http://d308902.website29.ebizdns.com/login.htm. A Westpac graphic was used in the html version of the email. Westpac reported the matter to the Australian Federal Police who were already engaged in the earlier ANZ and Commonwealth Bank incidents. Contact was made with E-Biz Hosting administrators via ICQ who turned out to be in Ukraine and the site was shut down after two days.


Westpac
Australia's First Bank

Dear Valued Customer,

- Our new security system will help you to avoid frequently fraud transactions and to keep your deposited funds in safety.

- Due to technical update we recommend you to reactivate your account.

Click on the link below to login and begin using your updated Westpac account.
To log into your account, please visit the Westpac Bank website at:

http://www.westpac.com/index.html

To review your statement, log into your Westpac Bank account and click the eStatements & eNotices button in the left navigation of your Account Summary page. Your new statement is listed in the left navigation of the page.

If you have questions about your online statement, please send us a Bank Mail or call us at 1-888-BKONWEB (256-6932).

We appreciate your business. It's truly our pleasure to serve you.

Westpac Bank Customer Care

This email is for notification only. To contact us, please log into your account and send a Bank Mail.

Figure 5 Westpac Phishing Email 4 July 2003 (Clapperton 2003)

Date	Victim	Subject	Phishing Site IP
28-Dec-02	e-Gold	Security Server Update	64.46.113.69
17-Mar-03	CBA	Netbank Security Server Update	64.46.113.74
11-Apr-03	ANZ	Security Server Update	64.46.114.91
12-May-03	Bank Of America	Security Server Update	198.173.235.126
4-Jul-03	Westpac	Security Server Update	64.46.100.64
4-Jul-03	ANZ	Security Server Update	64.46.113.208

Table 1 Selected List of Phishing Attacks 28/12/2002 to 4/7/2003

Other Phishing Attacks in this Period

Two other Internet banks had Phishing attacks during this period. On 19 May 2003 after the Bank of America incident there was an attack on Citibank using a site at http://209.97.63.225/cgi-bin/webforms.pl (Rohrich 2003). Also in May there was an attack on First Union Bank part of Wachovia Corporation another large US Bank (Fisher 2003). There are limited details of these attacks available so it is unknown whether they are related to the six phishing attacks described above.

THE HOSTING COMPANIES

3d Wizards

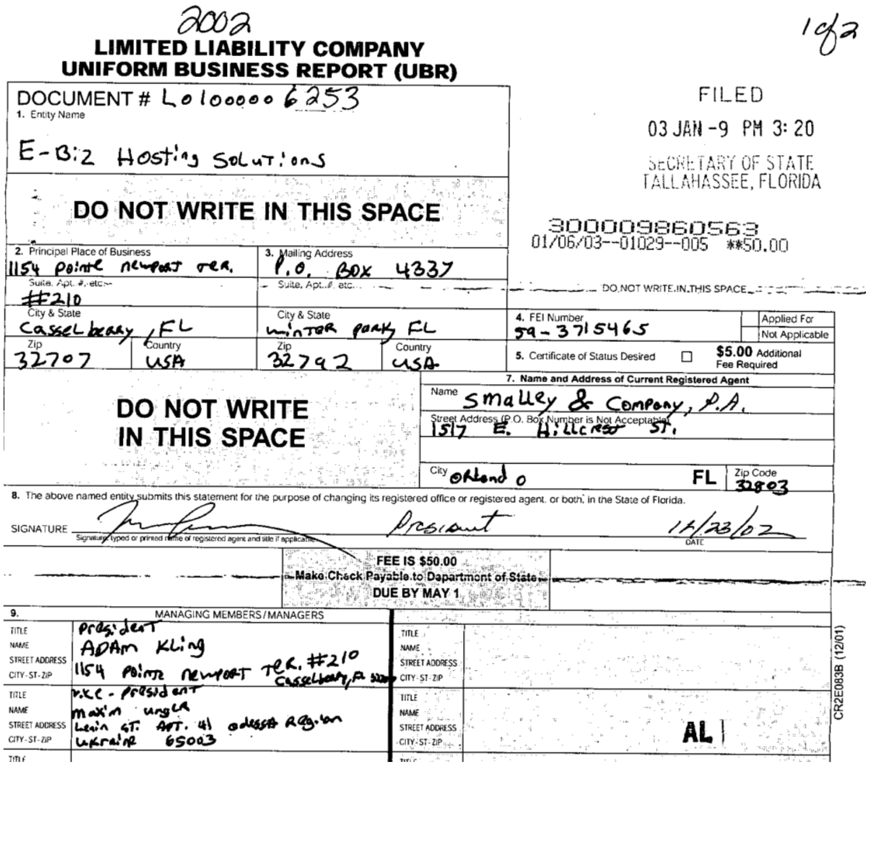
3d Wizards owned the IP space for five of the Phishing sites in this period and were part of DataColo, which was managed by Carlos Rego. The company was also known as Relio Ltd. It was based in Winter Park Florida.

DataColo

DataColo owned the larger block in which 3d Wizards block resided and similarly was managed by Carlos Rego. It was also based in Winter Park Florida.

E-Biz Hosting Solutions

E-Biz Hosting Solutions is also based in Winter Park Florida. It uses 3D Wizards IP space and was the domain owner of the domain used in the Westpac and both ANZ sites and appeared to have issued the https certificate for the e-Gold web site. It may well have used the other IPs associated the attacks that were part of 3d Wizards hosting space but this is unable to be confirmed. Adam Kling is listed in various documents as the President but the Vice-President is listed as Maxim Unger from Odessa Ukraine. Alex Mosh also from Odessa Ukraine is mentioned as CTO and employee in a number of newsgroup postings and is described in more detail below. A number of other Ukrainians or expatriate Ukrainians also seem to be associated with E-Biz Hosting Solutions in admin and sales roles according to Internet posts, including Tim Rogovets, Constantin Pogorelov and Kate Foteva.



2002

1 of 2

LIMITED LIABILITY COMPANY
UNIFORM BUSINESS REPORT (UBR)

DOCUMENT # L0100000 6253

1. Entity Name: E-Biz Hosting Solutions

DO NOT WRITE IN THIS SPACE

FILED
03 JAN -9 PM 3: 20
SECRETARY OF STATE
TALLAHASSEE, FLORIDA

300009860563
01/06/03--01029--005 **50.00

2. Principal Place of Business: 1154 pointe newport ter.	3. Mailing Address: P.O. BOX 4337
Suite, Apt. #, etc.: #210	Suite, Apt. #, etc.
City & State: Casselberry, FL	City & State: winter park FL
Zip: 32707 Country: USA	Zip: 32792 Country: USA

DO NOT WRITE IN THIS SPACE

4. FEI Number: 59-3715465 ☐ Applied For ☐ Not Applicable

5. Certificate of Status Desired ☐ **$5.00** Additional Fee Required

7. Name and Address of Current Registered Agent
Name: smalley & Company, P.A.
Street Address (P.O. Box Number is Not Acceptable): 1517 E. Hillcrest ST.
City: orlando FL Zip Code: 32803

DO NOT WRITE IN THIS SPACE

8. The above named entity submits this statement for the purpose of changing its registered office or registered agent, or both, in the State of Florida.

SIGNATURE [illegible] President 11/23/02
Signature, typed or printed name of registered agent and title if applicable DATE

FEE IS $50.00
Make Check Payable to Department of State
DUE BY MAY 1

9. MANAGING MEMBERS/MANAGERS

TITLE	President	TITLE	
NAME	ADAM Kling	NAME	
STREET ADDRESS	1154 Pointe newport ter. #210	STREET ADDRESS	
CITY-ST-ZIP	Casselberry, FL [illegible]	CITY-ST-ZIP	
TITLE	vice - president	TITLE	
NAME	maxim unger	NAME	
STREET ADDRESS	Lenin ST. APT. 41 odessa Region	STREET ADDRESS	
CITY-ST-ZIP	ukraine 65003	CITY-ST-ZIP	AL
TITLE		TITLE	

CR2E083B (12/01)

Figure 6 Florida Department of Commerce Filing for E-Biz Hosting Solutions (http://www.sunbiz.org)

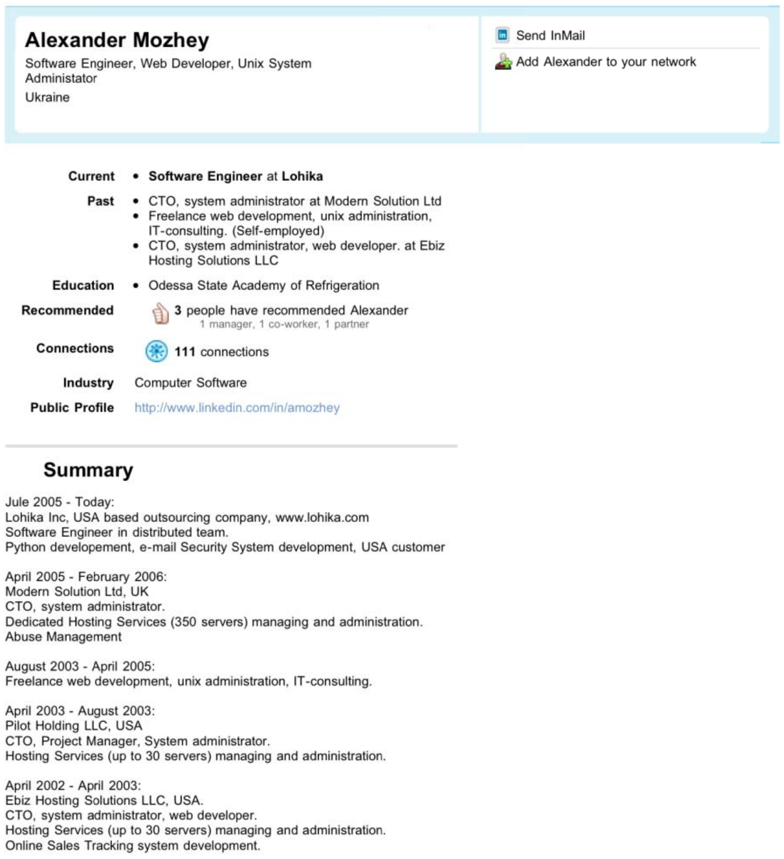
THE INDIVIDUALS

Adam Kling

Adam Daniel Kling is listed as the President of E-Biz Hosting Solutions. On a number of the incidents 3d Wizards administrators and other upstream providers gave his name and contact number to responders and he was contacted to shut both the ANZ Phishing sites down. He appears to be a resident of Florida. How he came to be working with Maxim Unger, Alex Mozhey (see below) and a number of others from the Ukraine is unknown.

Alex Mozhey, Alex Blood, Alex Mosh, Alex Polyakov

Listed in a number of Internet news postings as an employee and CTO of E-Biz Hosting Solutions is Alex Mosh. Alex Mosh is listed on the spamhaus Register of Known Spam Organisations (ROKSO) top ten list as of spammers, currently No.3 as of 6 October 2008 (http://www.spamhaus.org/statistics/spammers.lasso). In 2007 he was listed No.1. He has a number of aliases including Alex Blood and Alex Polyakov. The name Alex Polyakov is a Russian spy character from John Le Carre's novel Tinker, Tailor, Spy, which may explain its use. Alex Mosh used an ICQ address when working for E-Biz Hosting Solutions, which now is used by an Alex Mozhey who lists in his linked-in profile that he indeed worked for E-Biz Hosting Solutions as CTO. In his profile Mozhey also lists being the CTO for Pilot Hosting, which is also associated with Alex Mosh and listed frequently by ROKSO in connection with spamming. Mozhey and Mosh are likely to be the same person.



Alexander Mozhey
Software Engineer, Web Developer, Unix System Administator
Ukraine

Send InMail
Add Alexander to your network

Current	• **Software Engineer** at **Lohika**
Past	• CTO, system administrator at Modern Solution Ltd • Freelance web development, unix administration, IT-consulting. (Self-employed) • CTO, system administrator, web developer. at Ebiz Hosting Solutions LLC
Education	• Odessa State Academy of Refrigeration
Recommended	**3** people have recommended Alexander 1 manager, 1 co-worker, 1 partner
Connections	**111** connections
Industry	Computer Software
Public Profile	http://www.linkedin.com/in/amozhey

Summary

Jule 2005 - Today:
Lohika Inc, USA based outsourcing company, www.lohika.com
Software Engineer in distributed team.
Python developement, e-mail Security System development, USA customer

April 2005 - February 2006:
Modern Solution Ltd, UK
CTO, system administrator.
Dedicated Hosting Services (350 servers) managing and administration.
Abuse Management

August 2003 - April 2005:
Freelance web development, unix administration, IT-consulting.

April 2003 - August 2003:
Pilot Holding LLC, USA
CTO, Project Manager, System administrator.
Hosting Services (up to 30 servers) managing and administration.

April 2002 - April 2003:
Ebiz Hosting Solutions LLC, USA.
CTO, system administrator, web developer.
Hosting Services (up to 30 servers) managing and administration.
Online Sales Tracking system development.

Figure 7 Alexander Mozhey's Linked-in profile (http://www.linkedin.com)

Mosh's ROKSO record also connects him with money laundering or money mules and now acknowledged as a key part of Phishing. Mosh's ROKSO record lists website Verimer-australia.com used in 2005 for recruiting money mules in Australia and is connected with the entities and pseudonyms used by Alex Mosh.

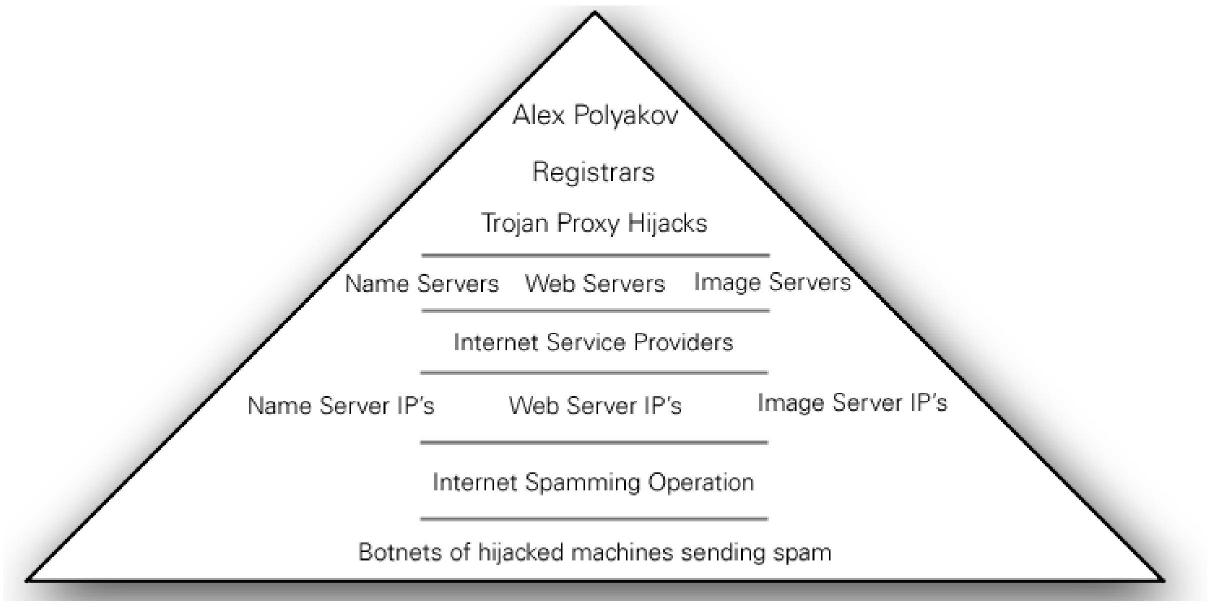
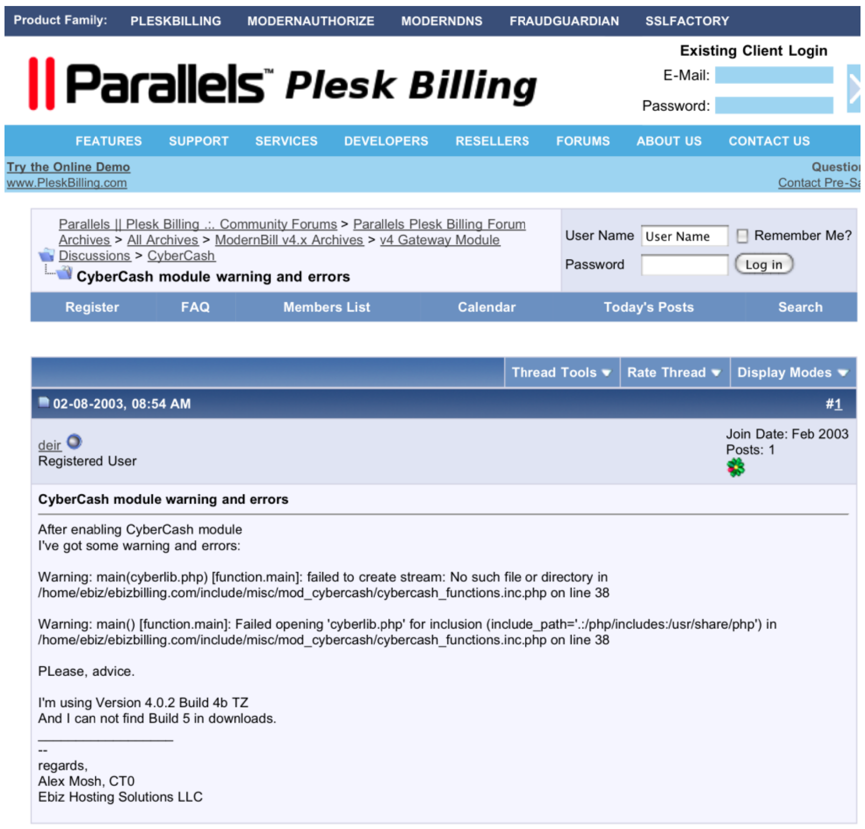


Figure 8 Alex Polyakov Internet Operation (http://spamtrackers.eu/wiki/index.php?title=Alex_Polyakov)

Mozhey in his linked-in profile amongst his skills are, "Good knowledge of Payment/Billing Systems, CC (credit card) processing, Merchant Gateways". He also indicates past experience in "Abuse management". Both he names Alex Mozhey and Alex Mosh are also connected with the nickname Deir that uses the same ICQ address and in some places Mozhey's actual name. Deir is a member a Parallels Forum. Below Deir signs himself as Alex Mosh CTO Ebiz Hosting Solutions LLC in that forum.


Product Family: PLESKBILLING MODERNAUTHORIZE MODERNDNS FRAUDGUARDIAN SSLFACTORY
Parallels Plesk Billing
Existing Client Login
E-Mail:
Password:
FEATURES SUPPORT SERVICES DEVELOPERS RESELLERS FORUMS ABOUT US CONTACT US
Try the Online Demo
www.PleskBilling.com
Questio
Contact Pre-S
Parallels || Plesk Billing .:. Community Forums > Parallels Plesk Billing Forum Archives > All Archives > ModernBill v4.x Archives > v4 Gateway Module Discussions > CyberCash
User Name
Remember Me?
Password
Log in
Register FAQ Members List Calendar Today's Posts Search
Thread Tools Rate Thread Display Modes

CyberCash module warning and errors

02-08-2003, 08:54 AM #1

deir
Registered User

Join Date: Feb 2003
Posts: 1

CyberCash module warning and errors

After enabling CyberCash module
I've got some warning and errors:

Warning: main(cyberlib.php) [function.main]: failed to create stream: No such file or directory in /home/ebiz/ebizbilling.com/include/misc/mod_cybercash/cybercash_functions.inc.php on line 38

Warning: main() [function.main]: Failed opening 'cyberlib.php' for inclusion (include_path='.:/php/includes:/usr/share/php') in /home/ebiz/ebizbilling.com/include/misc/mod_cybercash/cybercash_functions.inc.php on line 38

PLease, advice.

I'm using Version 4.0.2 Build 4b TZ
And I can not find Build 5 in downloads.

--
regards,
Alex Mosh, CTO
Ebiz Hosting Solutions LLC

Figure 9 Posting by Alex Mosh to Parallels Forum (http://forums.modernbill.com/member.php?u=757)

Carlos Rego

Carlos Rego was the CEO of 3dWizards Hosting and DataColo and in 2003 lived in Florida. He has a blog and uses the handle nullmind. Amongst his postings he refers to the day in September 2003 when the FBI came to the DataColo office apparently in connection with the aforementioned Phishing incidents.

“Today the FBI came by the office to pickup some logs on a scammer that was hosting with us, after taking his site down we kept all the info and logs on him .. I hope they catch the sucker. Basically the user had a fake e-gold site, he would send emails out to people saying they need to verify their e-gold accounts, people then would go to HIS site and enter their details and pin numbers :p ouch ..

Null (http://nullmind.com/2003/09/)”

Rego only mentions E-Gold but it is believed this FBI visit was also a result of an international mutual assistance application from the AFP on behalf of the Australian banks impacted by these early Phishing attacks. According to Carlos’s linked-in profile and Internet news items since leaving DataColo he has worked for Comodo, Positive Software and successful virtualisation software maker Parallels. All these organisations seem to have strong links to Russia and/or Ukraine. For instance Parallels CEO Serguei Beloussov studied for his Ph.D. in Computer Science at the Moscow Institute of Physics and Technology and the company has development centres in Russia and Ukraine. There is nothing suspicious in this but clearly Rego has a large degree of contact with Ukrainians and Russians in his business life.

Again it is not known how Rego who was born in Portugal and now lives in the United Kingdom came to be working with these individuals from Eastern Europe.

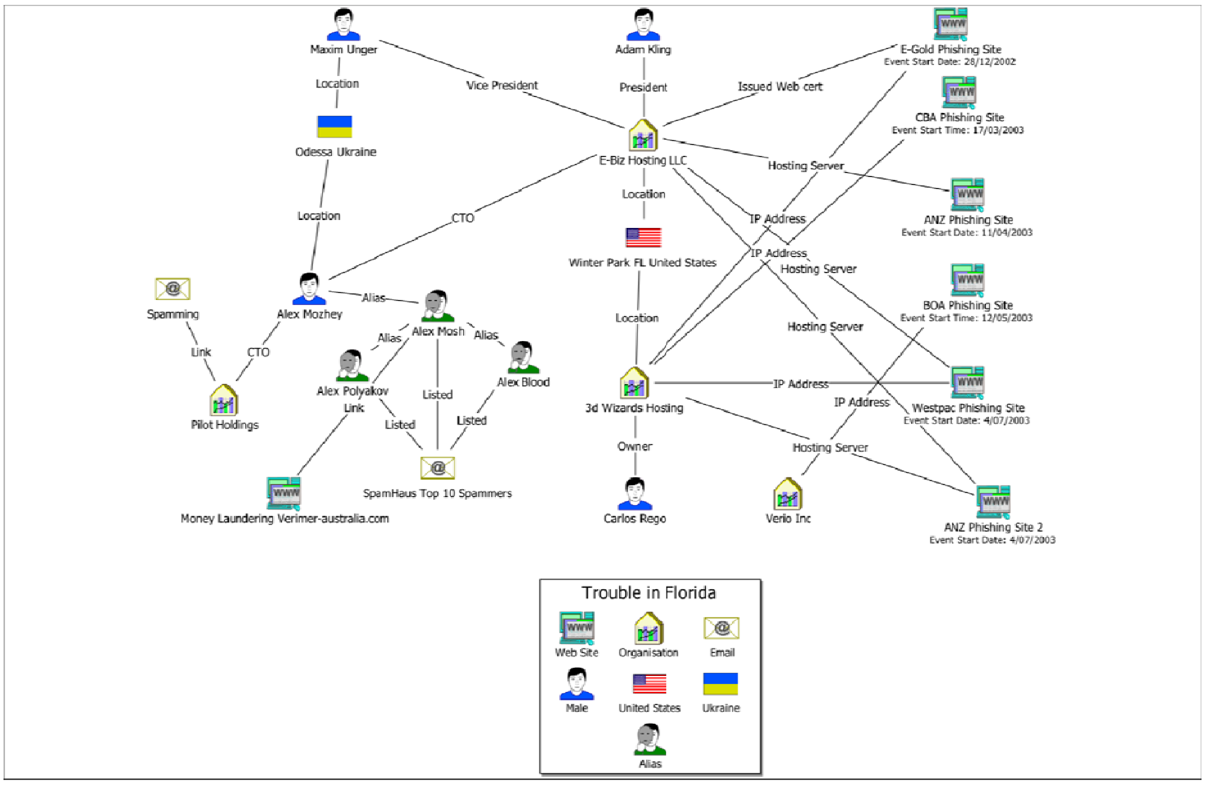


Figure 10 Relationships with Internet Bank Phishing Attacks Late 2002 to July 2003

WHAT HAPPENED AFTER JULY 2003?

Detailed figures on Phishing attacks were only collected towards the end of 2003. Judging from press reports and the documented histories of Phishing attacks; they did increase in numbers from August to the end of 2003 with more brands being targeted, including numerous UK and US Internet Banks. The earliest statistics from APWG Anti Phishing Working Group (APWG) show 21 phishing incidents in the month of November 2003 (APWG 2004). The phishing sites at this time were primarily located at large web hosting providers whose systems were apparently compromised and used to set up the sites. This method continued for some years even being the main method observed during the examination of phishing attacks in July 2006 on one Australian financial institution (McCombie 2008). The number of attacks increased into 2004 and has continued to increase to date, see below for the most recent figures.

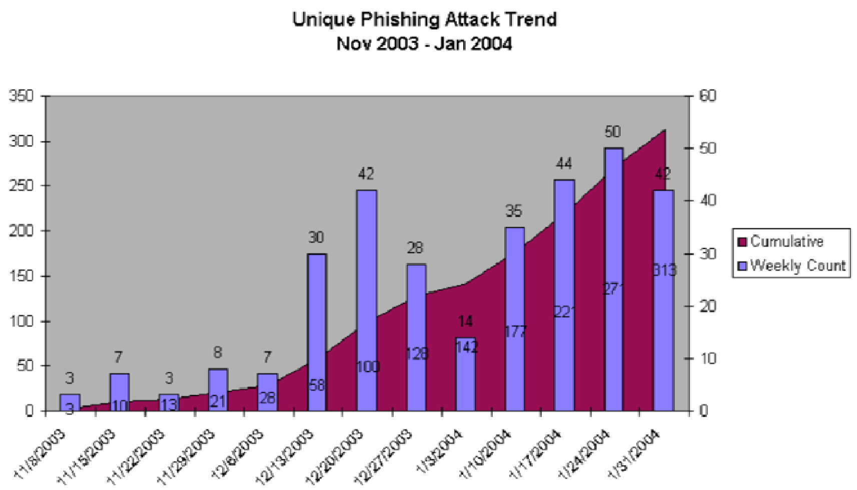


Figure 11 Unique Phishing Attack Trend Nov 2003 to Jan 2003 (APWG 2004)

HOW IS PHISHING DIFFERENT IN 2008?

Today many aspects of Phishing have changed. Phishing sites are now almost always found on Botnets. While Botnets certainly existed in 2003 they were far less common. Their use provides greater redundancy and is also more resilient to take down requests by the victim banks and their service providers. There is also a greater use of password stealing malware (crimeware as it is now described) to compromise users of Internet banks, which is again delivered using Botnets. Since 2004 the significance of crimeware has grown. For the month of March 2008 APWG reported 356 new unique password stealing malicious code applications (APWG 2008).

The ability of the Phishing sites to dupe unwitting users has reduced over time as user education and the shear volume of Phishing emails made knowledge of Phishing mainstream. However the attacks continue as they rely on only a small rate of success. In 2006 APACS the UK payments association working on behalf of the banking industry commissioned research agency Canvasse Opinion from Experian to poll a representative sample of 1,835 adults aged 18 and over, who have access to the Internet across the United Kingdom. Their results were,

"If we extrapolate for the 15.7 million people (in the UK) who regularly use the Internet to access their current, savings and credit card accounts as:

- 3.8% (an estimated half a million people) said they would still respond to an unsolicited email asking them to follow a link and re-enter personal security details, supposedly from their bank, unwittingly giving fraudsters access to their account (this is slightly down from 4% in 2004)."

Despite the fact that at the time of this survey Phishing had been widely known for 3 years a return of 3.8% shows us why Phishing sites still appear, in fact, APWG reported over 25,000 unique Phishing attacks attacking 139 different brands in February 2008 alone (APWG 2008).

WHAT CAN WE LEARN FROM THIS CASE STUDY

While not conclusive this case study shows there is some evidence to support the thesis that East European groups involved in spamming branched into Phishing and other online crime in 2003. Further research into the involvement of East European IT companies in on-line crime is needed. The trend in traditional Eastern European organised crime and indeed other transnational organised crime to move illegal profits into legitimate enterprises may well have extended to the cybercrime area but further work is needed to confirm this. Regardless there is clearly availability of IT skills within Eastern Europe to support both legal and illegal IT businesses and the challenge for those countries and the broader European community is to ensure organised cybercrime groups do not get a foot hold in legitimate industries.

Why did Australian Banks figure so significantly in these attacks? One likely reason is that Australian Internet banks had much greater functionality for payments than those in the US and most of the rest of the world at that time. Westpac for instance actually allowed Overseas Telegraphic Transfers (OTTs) to overseas banks direct from their Internet Banking in 2003. This allowed phishers to move the money straight from compromised accounts to banks in Eastern Europe. So Australian Internet banks were indeed world leading but in ways that were not intended.

CONCLUSION

Further work is required to better understand these early attacks but we hope this will start further research in this area. The author would have liked to interview more individuals involved but many were either unreachable or unable to comment on the events so this case study has been developed looking mostly at news reports and archival material available on the Internet from a number of sources and from the author's personal knowledge of events. While this approach has its shortcomings it was felt this case study was worth relating even on this limited information. We hope in future research to conduct further interviews with those involved and obtain more archival data on the organisations involved for more in depth analysis of these events.

REFERENCES

APACS (2008) *APACS announces latest fraud figures*. Retrieved 20 March 2007 from http://www.apacs.org.uk/APACSannounceslatestfraudfigures.htm

APWG (2004) *Phishing Attack Trends Report January, 2004.* Retrieved 9 October 2008 from http://www.antiphishing.org/reports/APWG.Phishing.Attack.Report.Jan2004.pdf

APWG (2008) *Phishing Activity Trends Report Q1/2008.* Retrieved 9 October 2008 from http://www.antiphishing.org/reports/apwg_report_Q1_2008.pdf

Broache, A. (2007) *E-Gold charged with money.* Retrieved 9 October 2008 from http://news.cnet.com/2100-1017_3-6180302.html

Clapperton, D. (2003) *[Oz-ISP] Westpac online banking scam in progress.* Retrieved 15 October 2008 from http://archive.humbug.org.au/aussie-isp/1057285342.54415.28.camel%40inferno

Colley, A. (2003) *NetBank suspect nabbed in Sydney.* ZDnet Australia. Retrieved 9 October 2008 from http://m.zdnet.com.au/120273072.htm

Department of Justice (2007) *Digital Currency Business E-Gold Indicted For Money Laundering And Illegal Money Transmitting.* DOJ press release. Retrieved 9 October 2008 from http://www.usdoj.gov/criminal/cybercrime/egoldIndict.htm

Fisher, D. (2003) *First Union Hoax on the Loose.* Retrieved 15 October 2008 from http://www.eweek.com/c/a/Messaging-and-Collaboration/First-Union-Hoax-on-the-Loose/

Friedman, R. (2000) Red Mafiya: How the Russian Mob has invaded America, New York. Penguin Putnam

Gartner (2007) *Gartner Survey Shows Phishing Attacks Escalated in 2007; More than $3 Billion Lost to These Attacks*. Retrieved 9 October 2008 from http://www.gartner.com/it/page.jsp?id=565125

Galeotti, M. (2005). *Russian mafiya become more active in Eastern Europe*. Jane's Intelligence Review - June 01, 2005

Galeotti, M. (2006). *The Criminalisation of Russian State Security*. Global Crime Volume 7 (Number 3-4): August-November 2006.

Galeotti, M. (2008) Interview with the Author.

Grigg, I. (2005) *GP4.3 - Growth and Fraud - Case #3 – Phishing.* Retrieved 9 October 2008 from http://www.financialcryptography.com/mt/archives/000609.html

Harley, D. (2007) *A Pretty Kettle of Phish*. Retrieved 9 October 2008 from http://www.eset.com/download/whitepapers/Phishing(June2007)Online.pdf

James, L. (2005). Phishing Exposed, Rockland MA Syngress Publishing.

Jennings, I. (2003) [fraud?] Security Server Update. Retrieved 15 October 2008 from http://groups.google.com.au/group/news.admin.net-abuse.sightings/browse_thread/thread/b2cbf3154a916d14/41aabb11fdcc8067?hl=en)aabb11fdcc8067

Keizer, G. (2005). *Dutch Botnet Trio Reportedly Connected To Russian Mob.* Retrieved 24 January 2007 from http://www.techweb.com/article/showArticle.jhtml?articleId=173600331&pgno=1

Kornakov (2007) *Gibson offers sneak peek into his world.* Retrieved 2 March 2007 from http://www.cambridge-news.co.uk/business/news/2007/02/06/ca10f0fb-fa50-4e49-b8d4-51b8c359075a.lpf

Litan, A. (2005). *Increased Phishing and Online Attacks Cause Dip in Consumer Confidence.* Gartner Research. Gartner.

McCombie, S., Watters, P.A., Ng, A. & Watson, B. (2008) *Forensic Characteristics Of Phishing – Petty Theft or Organized Crime?*, Proceedings of the 4th International Conference on Web Information Systems and Technologies (WEBIST), Madeira, Portugal.

Naraine, R. (2006) *Return of the Web Mob.* Retrieved 20 March 2007 from http://www.eweek.com/article2/0,1895,1947561,00.asp

Ramzan Z. (2007) *A Brief History of Phishing: Part I*, Retrieved 9 October 2008 from https://forums.symantec.com/syment/blog/article?message.uid=306505

Rohrich R. (2003) *CRIME Fwd: Your account is On Hold.* Retrieved 15 October 2008 from http://lists.jammed.com/crime/2003/05/0044.html

Riley D. (2003) *Security Server Update.* Retrieved 15 October 2008 from http://groups.google.com/group/news.admin.net-abuse.sightings/browse_thread/thread/c3c46036499f48f7/95565cf69675334d?hl=encf69675334d

Searle, K. (2003) Netbank Security Server Update (Commonwealth Bank scam Australia) host in FL. Retrieved 15 October 2008 from http://groups.google.com/group/news.admin.net-abuse.email/msg/11f128a770befb15?hl=en

Scheid E., (2003) *FW: Security Server Update.* Retrieved 15 October 2008 from http://mailman.anu.edu.au/pipermail/link/2003-April/049438.html

Schultz, E. (2003) *Email hoaxes continue to deceive users.* In Computers & Security,Volume 22, Issue 5, July 2003, Pages 368-377

The Presidents Identity Theft Task Force. (2007) *Combating Identity Theft: A Strategic Plan.* Retrieved 10 May 2007 from: http://www.idtheft.gov/reports/StrategicPlan.pdf.

Varghese, S. (2003) *NetBank scam: why didn't Commonwealth Bank do the obvious?* Sydney Morning Herald. Retrieved 9 October 2008 from: http://www.smh.com.au/articles/2003/03/19/1047749811735.html

Youl, T. (2004) *Phishing Scams: Understanding the latest trends.* Retrieved 9 October 2008 from http://www.fraudwatchinternational.com/pdf/report.pdf

Zenz, K. (2007) *Uncovering Online Fraud Rings: The Russian Business Network.* Retrieved 9 October 2008 from http://labs.idefense.com/intelligence/researchpapers.php

COPYRIGHT

Email 'Message-IDs' helpful for forensic analysis?

Satheesaan Pasupatheeswaran

School of Computer and Information Science
Edith Cowan University
Perth, Western Australia
spasupat@student.ecu.edu.au
ptheesan@yahoo.com

Abstract

Finding the source of spoofed email is a challenging task for forensic investigators. Header of an email has several fields that can be used for investigation. An investigator can easily understand the evidences embedded within most of the header fields of an email, except the message-id field. Therefore, there is a need to understand how message-ids are constructed and what useful information can be recovered from them. The immediate aim of the analysis is to find the message-id construction mechanism of 'Sendmail' mail transfer agent (MTA) version 8.14 and how the findings can be used successfully in forensic analysis. Source code of the 'Sendmail' MTA is made use of during analysis. This analysis will uncover several information that will help to find email source and validate other email header fields also. The drawbacks in message-id based forensic analysis also discussed here.

Keywords

E-mail header, message-id, msgid, sendmail forensics, e-mail forensics, e-mail header analysis, network forensics.

INTRODUCTION

An electronic mail consists of two parts, the header and the body. The header part carries information that is needed for email routing, subject line and time stamps while the body contains the actual message/data of an email. The header and the body are separated by a blank line. The header contains several mandatory and optional fields (Resnick, 2001). In order to uniquely identify each email all mail transfer agents (MTAs) use some sort of unique identifier. This identifier is referred to as 'Message-ID'. Message-ID field is inserted into a header either by mail user agent (MUA) or the first MTA. Even though the Message-ID is optional as per RF2822 it recommends using it. Sendmail is one of MTA that handles email delivery and relaying process. Sendmail uses message-id for tracing emails and for logging process ids (Costales, Janse, Abmann, & Shapiro, 2007, p1160). Sendmail recommends including message-id in emails and also it recommends setting relevant macros in its configuration file in order to implement compulsory checking of message-ids (Costales et al, 2007, p776). Unlike spoofing other fields in the header, spoofing message-id needs special knowledge. Only technical envy spammers can spoof the message-id cleverly. So deep analysis on message-ids may reveal some sort of information that will open a window to trace the source of an email. Also the message-id will help to find a particular email log entry within a log file of email server.

Like conventional mail service, when e-mail is routed from source to destination all intermediate relay servers (SMTP) insert their stamp at the beginning of the header. This stamping procedure helps to trace the email if such a demand arises. The stamp consists of three fields known as 'From', 'SMTP ID', and 'For'(Klensin, 2001) . Figure 1 shows an email header that passed through several MTAs. Each MTA inserted a unique-id in the header of email (Al-Zarouni, 2004). There are several IDs in the header field of an email that may help to trace the source of the email but this discussion is limited to sendmail message-id only. Analyzing intermediate SMTP IDs is beyond the scope of the discussion. However this paper briefly discusses intermediate SMTP-Ids also.

```
Received: from search.org ([64.162.18.2]) by sgiserver1.search.org with SMTP (Microsoft
Exchange

Internet Mail Service Version 5.5.2650.21)

id K9HBB4C4; Mon, 21 May 2001 09:47:01 -0700

Received: from web14506.mail.yahoo.com ([216.136.224.69]) by SEARCH.ORG

with SMTP (IPAD 2.52) id 3579700; Mon, 21 May 2001 08:47:23 -0800

Message-ID: <20010521164640.85785.qmail@web14506.mail.yahoo.com>

Received: from [216.104.228.118] by web14506.mail.yahoo.com; Mon, 21 May 2001 09:46:40
PDT

Date: Mon, 21 May 2001 09:46:40 -0700 (PDT)

From: <can_do1@yahoo.com>

Subject: check out this e-mail header

To: todd@search.org
```

Figure1: Email header with several identifiers (ID)

Message-ID

RFC 2822 states that each email must have a globally unique identifier. This must be included into the header of an email. The RFC 2822 also defines the syntax of message-id. It should be like a legitimate email address and it must be included within a pair of angle brackets. According to RFC 2822, message-id can appear in three header fields. They are 'message-id header', 'in-reply-to header' and 'references header'. But message-id of the present email must be included against the 'message-id' header.

Sendmail Message-ID

Sendmail Message-ID is formatted with two parts and they are connected by '@'sign. It looks like a legitimate email address. Right hand side (RHS) of the @ sign is a fully qualified domain name (FQDN) and left hand side (LHS) of the @ sign has two parts separated by '.'. The LHS part is created with date, time, process id and a few random numbers. Shown below is a sample message-id.

Message-ID: <200712141511.d872mLVW024467@cs.slt.edu>

Message-ID is always included within a pair of angle brackets. FQDN makes the MTA globally unique. The date and time with the combination of process id and special random numbers make the message unique in a particular MTA. This combination makes message-ids globally unique. Figure 1 shows sample sendmail header field (Costales et al, 2007, p7). Message-id is typed in blue bolded font.

```
From you@Here.US.EDU Fri Dec 13 08:11:44 2008

Received: (from you@localhost)

          by Here.US.EDU (8.12.7/8.12.7)
```

id d8BILug12835 for you; Fri, 14 Dec 2007 08:11:44 -0600 (MDT)

Date: Fri, 13 Aug 2008 08:11:43 Header

From: you@Here.US.EDU (Your Full Name)

Message-Id: **msgid=<200808131227.m7DCKVem009817@Here.US.EDU>**

Subject: a test ← *note*

To: you

This blank line separates body and header part.

Body part of email starts here. ⇩ Body

Figure2: Sample sendmail email structure

Message-ID generation

Sendmail message-id is defined in the following format (Costales et al., 2007, p776).

Message-id: $t.$i@$j

E.g.: 200808131227.m7DCKVem009817@Here.US.EDU

Following paragraphs discuss each part of message-id.

$t

$t macro is a current UTC date and time. This is formatted in yyyymmddhhmm. It consists of 12decimal values. In the above e.g. the $t part is **200808131227**. If it is decoded the final results will be 2008-08-13 12:27. That means the email is handed over to delivery or delivered at 12:27 on 13-08-2008 UTC (Sendmail, 2007).

$i

$i is referred as a queue id. It is generated with a special algorithm. Queue id has three different formats with respect to sendmail versions. Queue id versions are categorised as 'before V8.6', 'starting with V8.6' and 'starting with V8.10'. Format of queue-id with respect to sendmail versions are given below (Costales et al., 2007).

Before V8.6 → AApid

From V8.6 → hourAApid

From V8.10 → YMDhmsSEQpid

Following paragraphs will present a brief description about components ‘AA’ and ‘hour’ and discusses sendmail V8.14 in detail.

AA

‘AA’ is a combination of English alphabet and other characters. RHS clocks from A-Z (26 characters) and LHS clocks from A- ~ (62 characters) until it generates a unique-id. This provides more than 1600 combinations (Costales et al., 2007, p397).

AA

AB

. So on...

AZ

.So on...

~Y

~Z ← *failure*

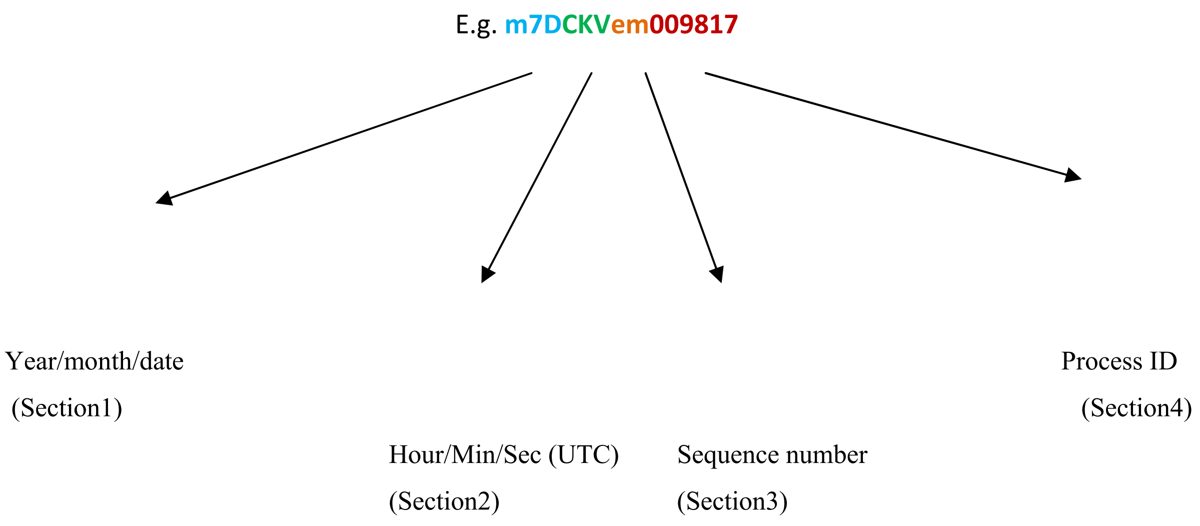
So on...

hour

This maps 24 hour clock to uppercase alphabet. The time starts at midnight and midnight 12 is mapped as A. Then 1’ o’ clock is B and so on (Costales et al., 2007, p397).

Sendmail V8.14

The message ID of V8.14 consists of three parts. The below example clearly indicates each part (Sendmail, 2007). In order to make it more understandable each group of components are named as ‘section x’ where x=1, 2, 3, 4 within brackets directly below each description.


E.g. m7DCKVem009817
Year/month/date (Section1)
Hour/Min/Sec (UTC) (Section2)
Sequence number (Section3)
Process ID (Section4)

The first eight characters can be of any combination from the characters given in table 1(Costales et al., 2007, p397). The last 6 digits are process id.

Decimal Numbers	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61
Mapping character	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	b	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	x	y	z

Table1: Mapping table

Section1

This part is current UTC time. Number of years is calculated from 1900 and then is divided by 60. The reminder is mapped to its relevant single character value (Costales et al., 2007, p397). See below example

Formula: Reminder (R) = (Current year- 1900) % 60

Reminder = (2008-1900) % 60 = 48

Map 48 in table 1$\rightarrow$ 'm'

Months January through December is numbered from 0 to 11. Therefore number 7 must be August.

Date is represented by 'D'. From the map table it is 13th.

Hence the encoded year, month and date is 2008-08-13.

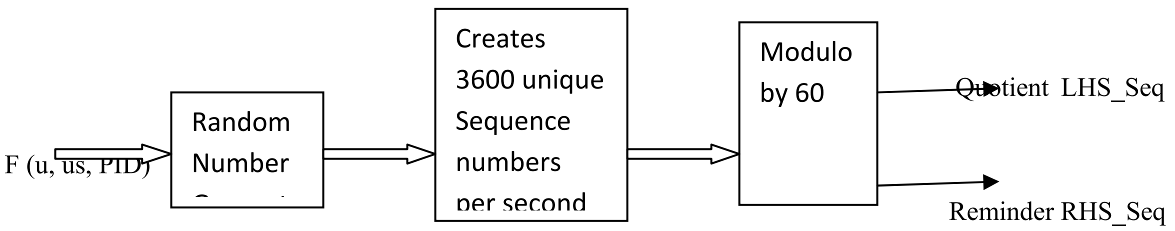
Section2

This is current UTC time. This is ordered as hour, minute and seconds. This is coded as hour, minute modulo 62 and seconds modulo 62 (Sendmail, 2007). Reverse mapping of each letter will decode the originating UTC time of the message.

C$\rightarrow$ 12, K$\rightarrow$ 20, V$\rightarrow$ 31. So the email is originated or submitted for delivery at 12:20:31 UTC.

Section3

This is referred to as sequence number. These two are generated from a random number. Right hand side number is quotient and left hand side is reminder (modulo) of a random number. Seed of the random number is created with dynamic unique numbers in order to make the best possible random number. The time period since epoch to current time is calculated in seconds and microseconds. Then the total number of seconds, microseconds and process id is summed up. This sum is used as seed for the random number generator Figure 3 shows the sequence number generation process (Sendmail, 2007).



Where F (u, us, pid) = seconds (since epoch) + microseconds+ Process-ID

Figure3: Sequence numbers generator

Section4

This is a 6 digit process identifier (PID or Process-ID). This process ID is relevant to the process that attempted to deliver the email (Hunt, 2001). Sendmail tries to make Process-ID unique for each queuing process (Costales et al., 2007).

$j

This macro represents the fully qualified domain name (FQDN). This part starts with local host name followed by a dot and other parts of domain information (Costales et al., 2007). Domain names are globally unique. In our previous example the $j part was **Here.US.EDU**. Local host name is ‘Here’ and the local domain name is ‘US.EDU’.

TRACING E-MAIL

In Message-ID generation section we found that the factors used to construct message-id themselves carried important information that can be used to trace source of an email. The following paragraphs discuss how this will help trace the source of an email message.

$j: Fully qualified domain name

FQDN contains local host name, from where the email was originated or the first sendmail MTA, and other domain information. In our previous example Here.US.EDU the first part, preceding the first dot is the local host or the first MTA server name. Right side of the first dot is other domain information. Once domain name is found then domain’s point of contact and other domain registration details can be found with readily available tools (Nelson, Philips, Enfinger, & Steuart, 2008, p484). Some of such web based tools are www.arin.net, www.internic.com, www.freeality.com and a command line tool is ‘whois’ (Mulligan, 1999, p32). Once the domain administrator is identified then forensic analysers can get her/his help to track the source with message-id.

$t: Date and time

Time is a critical factor in forensic investigation. The time part of message-id provides when the message was handed over for delivery. This time information will help to solve some of the problem stated below.

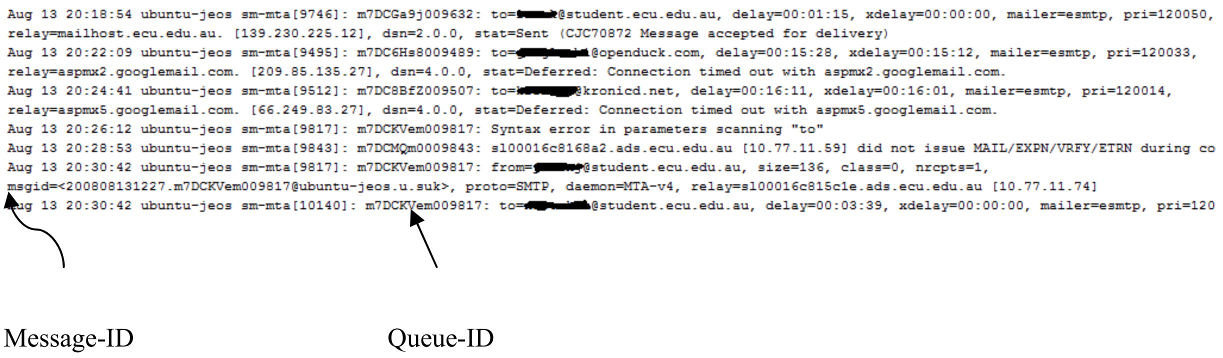
Dynamic IP addressing

In order to conserve IP address space most ISPs provide dynamic IP addresses. During investigation if IP address of the sender is found to be dynamic then the time information will help to search in the billing server who used this particular IP address at the specified time. This will help to identify the email sender. Billing servers contain session information such as period of login and allocated IP address for billing purposes. If sender used company's SMTP server then both SMTP log and DHCP log must be collected for analysis (Al-Zarouni, 2004).

Remember the time retrieved from message-id is UTC. So it is important to find out actual zone time. This can be done in several ways. Country of origin of email can be found from domain name as discussed in section $j. Once the country is known then time difference can be determined from several timing servers. Even this will help to verify the originating date and time of the email. The calculated time will help forensic analysers to check whether the source MTA is in sync with any standard time reference or not.

Email-server log file

Sendmail records all SMTP communication between servers in mail.log file. This log file contains date, host, process-id, queue-id and the log information. This log file maintains queue-id as a unique-id to distinguish each record (Costales et al., 2007, p517). By analysing the log file of source MTA with either message-id or queue-id, the expected record can be found. Time stamp and hostname/IP address found in the record can be verified with suspected email header so as to confirm the sender. Below figure shows a typical sendmail log file.



```
Aug 13 20:18:54 ubuntu-jeos sm-mta[9746]: m7DCGa9j009632: to=[redacted]@student.ecu.edu.au, delay=00:01:15, xdelay=00:00:00, mailer=esmtp, pri=120050,
relay=mailhost.ecu.edu.au. [139.230.225.12], dsn=2.0.0, stat=Sent (CJC70872 Message accepted for delivery)
Aug 13 20:22:09 ubuntu-jeos sm-mta[9495]: m7DC6Hs8009489: to=[redacted]@openduck.com, delay=00:15:28, xdelay=00:15:12, mailer=esmtp, pri=120033,
relay=aspmx2.googlemail.com. [209.85.135.27], dsn=4.0.0, stat=Deferred: Connection timed out with aspmx2.googlemail.com.
Aug 13 20:24:41 ubuntu-jeos sm-mta[9512]: m7DC8BfZ009507: to=[redacted]@kronicd.net, delay=00:16:11, xdelay=00:16:01, mailer=esmtp, pri=120014,
relay=aspmx5.googlemail.com. [66.249.83.27], dsn=4.0.0, stat=Deferred: Connection timed out with aspmx5.googlemail.com.
Aug 13 20:26:12 ubuntu-jeos sm-mta[9817]: m7DCKVem009817: Syntax error in parameters scanning "to"
Aug 13 20:28:53 ubuntu-jeos sm-mta[9843]: m7DCMQm0009843: s100016c8168a2.ads.ecu.edu.au [10.77.11.59] did not issue MAIL/EXPN/VRFY/ETRN during co
Aug 13 20:30:42 ubuntu-jeos sm-mta[9817]: m7DCKVem009817: from=[redacted]@student.ecu.edu.au, size=136, class=0, nrcpts=1,
msgid=<200808131227.m7DCKVem009817@ubuntu-jeos.u.suk>, proto=SMTP, daemon=MTA-v4, relay=s100016c815c1e.ads.ecu.edu.au [10.77.11.74]
Aug 13 20:30:42 ubuntu-jeos sm-mta[10140]: m7DCKVem009817: to=[redacted]@student.ecu.edu.au, delay=00:03:39, xdelay=00:00:00, mailer=esmtp, pri=120
```

Figure4: Typical sendmail log file

The source MTA may be maintained by ISP or it may belong to a company. Forensic analysers will require legal authorisation to access the log files.

In-Reply-To header

In-reply-to header holds message-id of original message to which it is replying to. Also this may include comma separated several message-ids, as a reply to several emails (Costales et al., 2007, p1158). Checking this header will help to find other suspicious emails.

```
In-Reply-To: <847.193925.780455@hostA.com>, <1021169802.330@HostB.co.th>,
             <200106020731.BAA20313@HostC.br.ca>
```

Figure5: An in-reply-to header with a few message-ids

References header

In case of threaded emails, continuous correspondence between parties, the reference header holds all message-ids from the first email to the last email (Loshin, 2000, p 92). Supposing the message-id of the interested email is spoofed the other message-ids will help to trace the email source. Figure 7 shows reference header with two message-ids.

Masquerade options

Sendmail and even some of the other MTAs like Microsoft exchange server support an option called masquerade. This option is used to hide the local host behind a local domain name or central email server. It usually rewrites sender address field with local domain name. Then any outgoing email will not have FQDN or $j. It might be tricky when tracing email with source address domain name. But this option does not affect FQDN of message-id (Costales et al., 2007, p600). In most of the situations analysing a message-id will help to directly locate the local host or the mail server which handled the initial delivery process. Figure 6 shows an email header with masqueraded source address.

```
Return-Path: <abcxyz@unsw.edu.au>

Received: from smtp.unsw.edu.au ([127.0.0.1])
        by localhost (snarl.comms.unsw.edu.au [127.0.0.1]) (amavisd-new, port 10025)
        with ESMTP id j2nhwsBjmdj7 for <satheesaan@slt.com.lk>;
        Mon, 28 May 2007 12:52:09 +1000 (EST)
Received: from central12-eng.eng.unsw.edu.au (central12-eng.eng.unsw.EDU.AU [129.94.131.112])
        by smtp.unsw.edu.au (8.13.6/8.13.6) with ESMTP id l4S2q9Gm010024;
        Mon, 28 May 2007 12:52:09 +1000 (EST)

Date: Mon, 28 May 2007 12:53:57 +1000                  Local host or Local mail server

Message-ID: <D9307185226FDB4FA388C58AE2A347859D34C7@central12-eng.eng.unsw.edu.au>

In-Reply-To: <C280794D.BDAC%eng.faculty@unsw.edu.au>

References: <200705250328.l4P3ShY5024597@smtp.slt.com.lk>  ←——    Reference header with
multiple msgid
            <C280794D.BDAC%eng.faculty@unsw.edu.au>

From:  abcxyz@unsw.edu.au ←——   Masquerade with local domain name

To:    <sssssss@slt.com.lk>
```

Figure 6: Reference and masquerade header

In the above figure 'From:' field has local domain name 'unsw.edu.au' but 'Message-id field' and 'Received field' have the local email server that handled the first delivery process. If investigators can provide the email server details that handled the email delivery process, it will help to speed up the process. Consequently it will reduce the burden of log file analysis.

Intermediate SMTP-ID

As emails go through intermediate MTAs (hops) each MTA insert their unique-id (SMTP-ID) on the email header. If it is necessary to analyse intermediate server log file this unique-id is important. Knowledge about intermediate smtp-id also will help to identify any fake smtp-id. Intermediate sendmail servers create a queue id as stated in section $i and use it as smtp-id (Costales et al., 2007, p826). Intermediate mail servers stamp starts with 'Received:' header. Figure6 shows an email that is routed through three sendmail MTAs. Each MTA insert their stamp with smtp-id. This smtp-id can be used in log file analysis.

```
Return-Path: <satheesaan@slt.com.lk>

Authentication-Results: mta379.mail.mud.yahoo.com from=slt.com.lk; domainkeys=neutral (no
             sig)

Received: from 203.115.19.199 (EHLO xmail.slt.com.lk) (203.115.19.199) by
             mta379.mail.mud.yahoo.com with SMTP; Fri, 23 Jun 2006 02:24:38 -0700

Received: from smtp.slt.com.lk (smtp.slt.com.lk [172.25.1.100]) by xmail.slt.com.lk
             (8.12.11/8.12.11) with ESMTP id k5N9LmIO004855; Fri, 23 Jun 2006 14:51:48 +0530

Received: from slt.com.lk (pop.slt.com.lk [172.25.1.101]) by smtp.slt.com.lk     (8.12.10/8.12.10)
             with ESMTP id k5NKrSuA013912; Fri, 23 Jun 2006 14:53:28 -0600 (GMT)

Received: from slt.com.lk (slt.com.lk [127.0.0.1]) by slt.com.lk (8.13.1/8.13.1) with ESMTP

           id k5N8rfWP021228; Fri, 23 Jun 2006 14:53:51 +0600

From:  <satheesaan@slt.com.lk>

To: camalan@celetronix.com.uk,

Subject: Fw: you've got to see this Date: Fri, 23 Jun 2006 15:53:41 +0700

Message-Id: <20060623085301.M4685@slt.com.lk>

In-Reply-To: <000701c6969f$78ff94e0$374019ac@RAIN>
```

Figure7: Intermediate SMTP-ID

The pattern of smtp-id and reverse mapping of smtp-id proves that smtp-id is a queue-id.

FAKE MESSAGE-ID

Just like spoofing other header fields of email, spoofing message-id is also possible. By observing a few email headers where the first MTA is sendmail, it is possible to make a message-id that look legitimate.

E.g.: 200808131227.m7DCKVem009817@Here.US.EDU

LHS of the dot is simply date and time and RHS of the dot contains 14 characters, first 8 characters are a combination of numbers and English alphabets and other 6 are just numbers.

So before using message-id for forensic analysis the message-id must be verified for its validity.

Message-id verification

Knowledge of sendmail message-id construction will help to verify the message-id. With the help of mapping table, $t part can be verified with first 5 characters of $i. The sequence number and process id are dynamically created characters so verifying them is difficult.

Spam identification

Spam mail filters check for empty message-id or illegal message-id pattern only. The message-id is an optional field and it also can be spoofed. So message-id cannot be a reliable spam indicator(Allman, Assmann, & Shapiro, n.d).
Spam mail senders harvest email addresses through several ways one such method is scanning USENET articles (Costale & Flynt, 2005). If any email received from known source is suspected to be sent by spammers, the suspected email can be verified by comparing message-id of the email against known good email message-id from the same source. However, checking message-id is not a consistent spam checking method because a good spammer can create same pattern of message-id.

ISSUES RELATED TO MESSAGE-IDS

No Standard algorithm

RFC2822 standard states every email should have a unique identifier and provides syntax of message-id and some suggestion to create unique identifier. However, it does not define how it should be generated. Email software developers use their own algorithm to generate message-ids. Forensic analyser or relevant technical advisor must be well informed on the different vendor message ids as he/she might come across different types of message-ids.

This drawback makes it difficult to make a tool for checking validity of message-ids. Sendmail checks message-id header, if it is blank it will insert a new message-id otherwise it will not alter the available message-id (Costales et al., 2007, p834). This vulnerability aids the successful transmission of emails with spoofed message-ids. Spoofed message-id will compromise forensic analysis results.

Open source and closed source

In case of open source softwares it is possible to find out the construction mechanism of message-ids but it will be difficult to determine the message-id construction mechanism in closed source softwares.

Identifying Source MTA

There are several MTAs in use. In order to select the suitable analysis procedure investigator must know the source MTA. If the source MTA is known it will help to verify the message-id against fake ids. Sendmail will not generate new message-id if the email already has a message-id. Some MUA also generate message-ids (Costales et al., 2007, p1159). Even the first MTA is a sendmail; the message-id might not be sendmail compatible. In this case first smtp-id will help log analysis. This area needs the special attention of researchers.

Versions

The message–id algorithm of sendmail has already changed thrice (Costales et al., 2007, p387). Therefore for analysis, continuous research and updates on message-ids is important. Determining the version of the sendmail is also necessary before start of message-id analysis.

Host time

MTA host time must be synchronized with reliable time reference. Since forensic investigation is time sensitive, if there is any difference it time it may invalidate the case in court or it may be very difficult to prove in court. There are some tools, such as NTP, STIME and GPS clock, that can be used to synchronise the host time (Al-Zarouni, 2004). Incorrect timing and time setting may cause message-id collision in the specific host itself.

Spoofed message-ids

Spoofing email message-ids is possible and it will compromise the forensic analysis. If message-id is spoofed with an earlier valid email message-id then this will change the direction of the investigation. This will create unnecessary problems and delay in the investigation. Figure8 shows an email header with spoofed message-id.

```
Return Path: <dhjmifpo@msn.com>

Received: from 200.94.239.104 (HELO 216.136.129.5) (200.94.239.104)
by mta136.mail.sc5.yahoo.com with SMTP; Sun, 27 Jun 2004 17:14:03

Received: from 162.134.15.76 by 200.94.239.104; Sun, 27 Jun 2004
20:11:02 - Message-ID: <P[20
```

Fighure8: An email header with spoofed message-id from my inbox

Headers without message-id

Some emails, especially drafted for illegal activity or spam, do not have message-ids in their headers. In such circumstances message-id forensic is not applicable. Below figure shows a successfully delivered webmail without message-id.

```
Return-Path: <good@mta463.mail.mud.yahoo.com>

Authentication-Results:  mta463.mail.mud.yahoo.com from=yahoo.com;

                  domainkeys=neutral

 Received: from 122.44.118.105 (HELO fpyd.net) (122.44.118.105) by

          mta463.mail.mud.yahoo.com with SMTP; Fri, 05 Sep 2008 16:04:41 -0700

From: <ptheghost@yahoo.com>

To: <ptheghost@yahoo.com>

Subject: Hurry.. Buy US based medications here !..save your money ! MIME-Version: 1.0 Content-
Type: multipart/mixed;boundary= "----=_NextPart_000_00CA_A0C54188.80C58DC2" Content-
Length: 690
```

Figure9: Fake header without message-id

In the above header both ‘From’ and ‘To’ header fields have same address. Thus it is confirmed that the email is a fake. Also it does not have a message-id.

International cooperation

Message-ID based forensics analysis needs log file analysis. In some occasions the source server might be located in another country. To handle this type of situation investigator needs cooperation from that foreign country to carry out the analysis successfully.

CONCLUSION:

This discussion reveals that email message-id plays an important role in email forensic analysis. The global unique feature of message-id helps to distinguish each email and so help in forensic analysis. Knowledge of message-id construction part will help to identify spoofed emails, source host, email log file analysis and time details. This paper also discussed the ways to determine fake message-ids. Beyond some of the identified weaknesses in message-id, the information that is carried by the message-id is highly important in tracing the email source.

This study is carried out only on sendmail message-id. However this area needs more study on other message-ids that are created by different email software. The key factor in message-id analysis is that the source email software must be known to the investigator in order to apply suitable methods during analysis.

REFERENCES:

Al-Zarouni, M. (2004). *Tracing E-mail Headers*. Retrieved 02-Sep-2008, from http://scissec.scis.ecu.edu.au/publications/forensics04/Al-Zarouni.pdf

Allman, E., Assmann, C., & Shapiro, G. N. *Sendmail Installation and operation Guide*. Retrieved 03-Sep-2008, from http://www.sendmail.org/doc/sendmail-current/doc/op/op.pdf`

Costales, B., & Flynt, M. (2005). *Sendmail Milters AGuide for Fighting Spam*. NJ: Addison-Wesley.

Costales, B., Janse, G., Abmann, C., & Shapiro, G. N. (2007). *Sendmail* (4th ed.). Sebastopol: O'Reilly.

Hunt, C. (2001). *Linux Sendmail Administration*. Retrieved 04-Sep-2008, from http://library.ecu.edu.au/

Klensin, J. (2001). *RFC2821:Simple Mail Transfer Protocol*. Retrieved 01-Sep-2008, from http://www.ietf.org/rfc/rfc2821.txt

Loshin, P. (2000). *Essential Email Standards: RFCs and Protocols Made Practical*. NY: Wiley.

Mulligan, G. (1999). *Removing the Spam: Email Processing and Filtering*. Reading: Addison-Wesley.

Nelson, B., Philips, A., Enfinger, F., & Steuart, C. (2008). *Guide to Computer Forensics and Investigations* (3rd ed.). Boston: THOMSON COURSE TECHNOLOGY.

Resnick, P. (2001). *RFC2822: Internet Message Format*. Retrieved 01-Sep-2008, from http://www.ietf.org/rfc/rfc2822.txt

Sendmail [Computer Software]. (2007). *Sendmail* (Version V8.14.2): Sendmail Inc.

COPYRIGHT

Data recovery from PalmmsgV001

Satheesaan Pasupatheeswaran

School of Computer and Information Science
Edith Cowan University
Perth, Western Australia
spasupat@student.ecu.edu.au
ptheesan@yahoo.com

Abstract

Both SMS and MMS data analysis is an important factor in mobile forensic analysis. Author did not find any mobile forensic tool that is capable of extracting short messages (SMS) and multimedia messages (MMS) from Palm Treo 750. SMS file of Palm Treo 750 is called PalmMgeV001 and it is a proprietary file system. A research work done to find a method to recover SMS data from PalmMsgV001 file. This paper is going to describe the research work and its findings. This paper also discusses a methodology that will help recover SMS data from PalmMsgV001. The PalmMsgV001 file is analysed using hex analysis method. Solutions were found to recover each message from every folder like Inbox, Outbox, Sentbox, Draft and Template. The research work partially contributes to improving mobile forensic analysis since the finding will be helpful to forensic tool developers. At this stage, this study will concern only the SMS part and not the MMS part.

Keywords

Mobile forensics, Palm Treo 750 SMS file, PalmmsgV001 data recovery

INTRODUCTION

Hardware and software architecture of smart phones, unlike computers, differ from manufacturer to manufacturer. Also the frequency of arrival of new models of mobile phones with new technology is increasing and this becomes a challenge to mobile forensic analysers and forensic tool developers. There is no mobile forensic tool that has the ability to analyse all brands of mobile phones all by itself (McCarthy, 2005). Beyond any argument, analysing SMS file is significant in mobile forensics because it may contain valid evidences.

Palm Treo 750 mobile phone uses a new SMS technology called threaded SMS technology. At the time of writing this paper, the author did not find any commercial or open source tool that helps to analyse message file of Palm Treo 750 also referred to as Palmmsgv001. Aim of the research is to find a method to retrieve SMS data from PalmMsgV001file. Binary analysis and binary comparison method is used during the research work. The research work is concentrated on SMS data recovery from all text message folders such as Inbox, Outbox, Sentbox, Draft and Template.

According to the ACPO (2003), any operation performed on original evidence should not alter the evidence. If alteration is necessary, it should be documented and its impact on evidence should be realized (ACPO, 2003). Due to some limitations in accessing mobile phone data acquisition tools and accessing mobile devices, the file acquisition was not done in a forensically sound manner. However, this did not significantly affect the research results because the goal of the research was not forensic analysis.

Significance

Lack of knowledge about messaging file (Palmmshv001) of Palm Treo 750 imposed limitations on mobile forensic analysis. Nowadays, communication via text messages and exchanging multimedia messages is quiet common because of its several attractive features like sending a single SMS to multiple recipient and low cost. Several crimes such as drug trafficking and pornography are also committed via text and multimedia messaging (Press, 2008). Therefore analysing messaging file during mobile forensic analysis is crucial.

Research questions

The analysis work is focused on answering the following questions.

i. How to identify an empty file?

ii. How to identify and retrieve read, unread, saved, deleted, drafted and template messages?

iii. How to identify and retrieve source of message and other relevant contact information?

iv. In some occasions there may be identical message from a particular source. If so how to distinguish messages?

SHORT MESSAGE SERVICE

SMS is an abbreviation for short message service, it first appeared in 1992 in Europe (Bodic, 2005). Since then it has become hugely popular among mobile users. SMS application is widely used for multiple purposes such as person to person communication, information services, email alerts, business cards, chat applications and other co-operate services such as sim updates, unlocking mobile to use on any network and remote monitoring (Bodic, 2005).

SMS provides a way to transfer short message between two entities via service centre (SC). SC provides relaying service for short messages. There are two types of SMS transport services. A point to point SMS service and point to Omni-point service. Point to point SMS is a service between two parties. Point to Omni-point (cell broadcast) SMS service is sending message to multiple recipients from a single source (ETSI, 1999). Point to point SMS service contains two types of basic services such as short message point to point mobile terminated (SM-MT) and short message point to point mobile originated (SM-MO).

SM-MT is a GSM service that handles delivery of a short message from SC to mobile station (MS). SM-MO is a GSM service that handles submition of short messages from MS to another mobile entity through SC. Both services are capable of providing delivery status report (ETSI, 1999). SC provides store and forward service so as to give a reliable service (Trosby, 2004).

SMS size is very limited; a single SMS size is 140 octets. It supports only 160 characters if 7 bit character encoding is used and 70 characters if 16 bit Unicode encoding is used (ETSI, 1999).

Chat Application

Chat application helps to chronologically display the communication history between two parties. Most of the mobile chat applications use SMS for message transport (Bodic, 2005).

APPARATUS

During the research process both software and hardware tools were used for data collection and analysis. Following paragraphs discuss each tool in detail.

Dopod 838 Pro smart phone

A Dopod smart phone was used for data collection. It was running on WM 5.0. With a few modifications this mobile was made to operate in threaded style SMS. These modifications make the Dopod's SMS application to replicate a Palm Treo 750. The message application file is placed in a windows folder.

'ExamDiff Pro' Software

ExamDiffPro is a hex file comparison tool. Version 4.0 of this tool was used for the research. This program supports opening two files in a window. This tool shows edited characters, deleted characters and added character in different colours in both files. It also provides a navigation facility that helps to jump to locations of modified data. It has facilities like file swapping, simultaneous scrolling of both files and hex to text/text to hex toggle ("Visual File And Directory Comparison Tool," 2008).

"UltraEdit" software

Version 14.10 of UltraEdit Professional was used in this analysis. UltraEdit is a powerful text, hex and software source code editor. During hex edit, it displays both its hex and its ASCII values. It supports toggling between hex to text and text to hex ("UltraEdit Text Editor Features," 2008).

PROCEDURE

Data collection

The Palmmsgv001 file is accessed from the desktop and copied from the mobile to the desktop by synchronising the mobile phone with the desktop.

Reference image

A reference point is necessary so as to make sure each test phase is started from the same environment. Starting each phase of test from a reference point will help to protect the file from previous data content. An empty Palmmsgv001 is kept as reference file. Each phase of test is started with empty file acquisition.

Sample data preparation

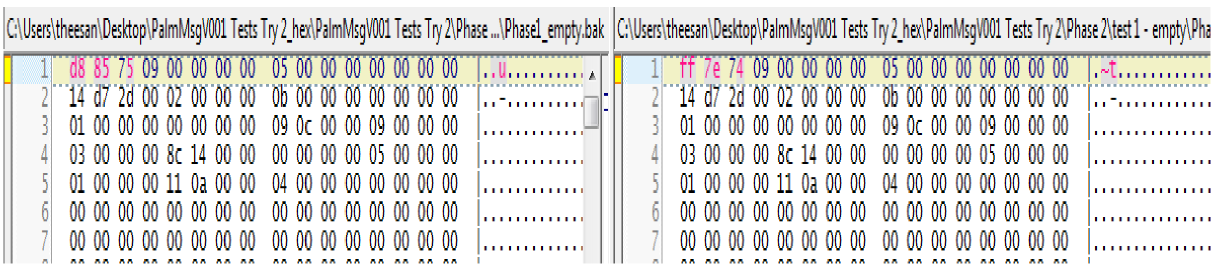
Several samples were created in a way to address the research questions.

ANALYSIS OF SAMPLE FILES

Analysis is performed in two stages. Each stage is discussed below. Analysis was done based on ETSI TS 100 901 short message standards (ETSI, 1999).

Stage1

At first empty file of each test phase was compared with the help of 'ExamDiff Pro'. The aim of this comparison was to identify header bytes.



```
C:\Users\theesan\Desktop\PalmMsgV001 Tests Try 2_hex\PalmMsgV001 Tests Try 2\Phase ...\Phase1_empty.bak
d8 85 75 09 00 00 00 00  05 00 00 00 00 00 00 00  |..u..........
14 d7 2d 00 02 00 00 00  0b 00 00 00 00 00 00 00  |..-..........
01 00 00 00 00 00 00 00  09 0c 00 00 09 00 00 00  |.............
03 00 00 00 8c 14 00 00  00 00 00 00 05 00 00 00  |.............
01 00 00 00 11 0a 00 00  04 00 00 00 00 00 00 00  |.............
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |.............
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |.............
```

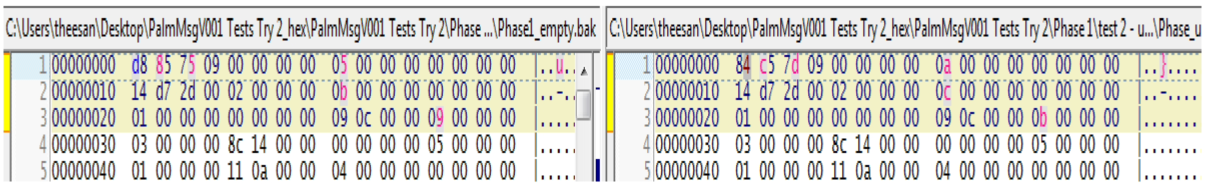
```
C:\Users\theesan\Desktop\PalmMsgV001 Tests Try 2_hex\PalmMsgV001 Tests Try 2\Phase 2\test 1 - empty\Pha
ff 7e 74 09 00 00 00 00  05 00 00 00 00 00 00 00  |.~t...........
14 d7 2d 00 02 00 00 00  0b 00 00 00 00 00 00 00  |..-...........
01 00 00 00 00 00 00 00  09 0c 00 00 09 00 00 00  |..............
03 00 00 00 8c 14 00 00  00 00 00 00 05 00 00 00  |..............
01 00 00 00 11 0a 00 00  04 00 00 00 00 00 00 00  |..............
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |..............
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |..............
```

Figure1: Comparison of two empty file

The above figure 1 shows comparison of two empty files. Bytes, those are different between the two files are coloured in pink.

It is found that the first three bytes are changing from file to file. Observations in all phases, helped to identify two pairs of identical files.

Then empty file of each phase was compared with the relevant data files that contained known data such as unread file, read file, saved file and deleted file. First few lines were observed for changes. It was found that first three lines had changed bytes. Closer observation in all phases helped to identify a byte which is used to indicate file status ,i.e. whether the file is empty or not. There were some other bytes also changed from file to file but the changes were not consistent. Figure 2 shows a screen shot of changed bytes in the first three lines.

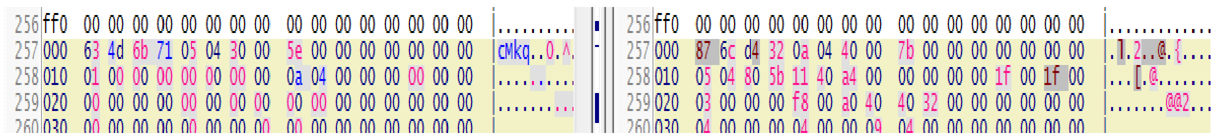


```
C:\Users\theesan\Desktop\PalmMsgV001 Tests Try 2_hex\PalmMsgV001 Tests Try 2\Phase ...\Phase1_empty.bak
00000000  d8 85 75 09 00 00 00 00  05 00 00 00 00 00 00 00  |..u..
00000010  14 d7 2d 00 02 00 00 00  0b 00 00 00 00 00 00 00  |..-..
00000020  01 00 00 00 00 00 00 00  09 0c 00 00 09 00 00 00  |.....
00000030  03 00 00 00 8c 14 00 00  00 00 00 00 05 00 00 00  |.....
00000040  01 00 00 00 11 0a 00 00  04 00 00 00 00 00 00 00  |.....
```

```
C:\Users\theesan\Desktop\PalmMsgV001 Tests Try 2_hex\PalmMsgV001 Tests Try 2\Phase 1\test 2 - u...\Phase_u
00000000  84 c5 7d 09 00 00 00 00  0a 00 00 00 00 00 00 00  |..}....
00000010  14 d7 2d 00 02 00 00 00  0c 00 00 00 00 00 00 00  |..-....
00000020  01 00 00 00 00 00 00 00  09 0c 00 00 0b 00 00 00  |.......
00000030  03 00 00 00 8c 14 00 00  00 00 00 00 05 00 00 00  |.......
00000040  01 00 00 00 11 0a 00 00  04 00 00 00 00 00 00 00  |.......
```

Figure2: File comparison shows changed bytes.

Further analysis helped to identify a set of four bytes that indicate whether the message data structure is empty or not. Figure3 shows the set of four bytes and its ASCII character in blue font.



```
ff0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |.........
000  63 4d 6b 71 05 04 30 00  5e 00 00 00 00 00 00 00  |cMkq..0.^.
010  01 00 00 00 00 00 00 00  0a 04 00 00 00 00 00 00  |.........
020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |.........
030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |
```

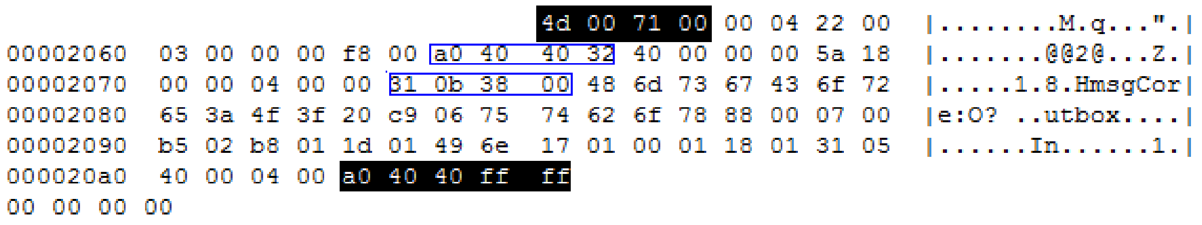
```
ff0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |............
000  87 6c d4 32 0a 04 40 00  7b 00 00 00 00 00 00 00  |.].2..@.{....
010  05 04 80 5b 11 40 a4 00  00 00 00 00 1f 00 1f 00  |...[.@.......
020  03 00 00 00 f8 00 a0 40  40 32 00 00 00 00 00 00  |.......@@2...
030  04 00 00 00 04 00 00 09  04 00 00 00 00 00 00 00  |
```

Figure3: Empty message data structure and non empty data structure

Stage2

During this stage, each sample file was opened with 'UltraEdit' in order to perform pattern analysis. Each file was manually analysed. A column that represents ASCII characters of hex values was used to identify the known data. This analysis helped to find another data structure that contains all SMS data, MMS data and relevant folders. This data structure has a special signature. Also it was found that the data structure has several small data units each units has a header and a footer. Mostly all footers have the same bytes but some differ. Also a data unit was identified that is used as a separator and it is used to separate two different sections.

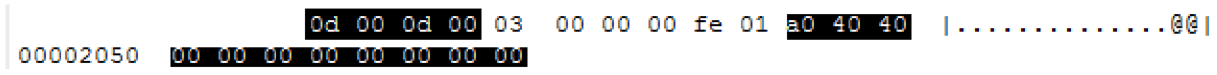
The footer was used to break the data structure into small data units, then the data structure was analysed. From the analysis it was found that, at first the data structure initialises all folders like 'Outbox', 'Sentbox', 'Savedbox', 'Draftbox' and 'Template'. Two repeating patterns were observed during each folder initialisation, those patterns were identified as start of folder and initialisation part. These patterns are included within blue coloured boxes in figure4. Figure4 shows a small data unit, its header and footer are highlighted. The header value is the signature of the data structure. This data unit includes Outbox initialization part also.



```
                                 4d 00 71 00  00 04 22 00   |........M.q...".|
00002060  03 00 00 00 f8 00 a0 40  40 32 40 00  00 00 5a 18   |.......@@2@...Z.|
00002070  00 00 04 00 00 31 0b 38  00 48 6d 73  67 43 6f 72   |.....1.8.HmsgCor|
00002080  65 3a 4f 3f 20 c9 06 75  74 62 6f 78  88 00 07 00   |e:O? ..utbox....|
00002090  b5 02 b8 01 1d 01 49 6e  17 01 00 01  18 01 31 05   |......In......1.|
000020a0  40 00 04 00 a0 40 40 ff  ff
00 00 00 00
```

Figure4: Small data unit with header and footer that initialize 'Outbox'

Figure 5 shows the separator data unit that always start with a particular 4 bytes header and ends with footer. Separator data unit consists of 21 bytes.



```
                     0d 00 0d 00 03  00 00 00 fe 01 a0 40 40   |..............@@|
00002050  00 00 00 00 00 00 00 00
```

Figure5: Separator data unit

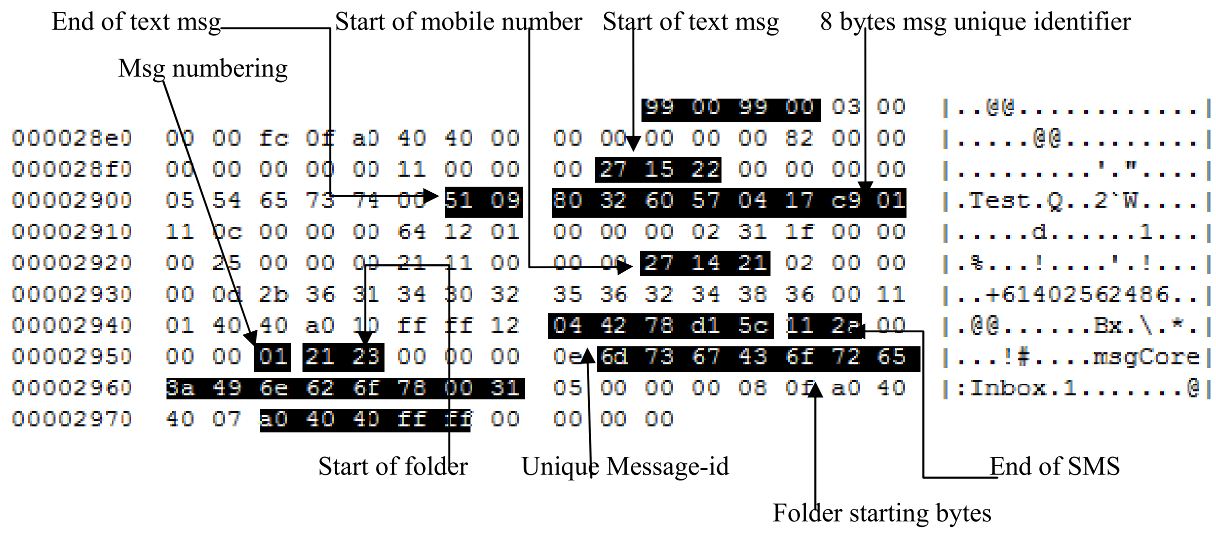
Further analysis helped to find the folder data structure. Each folder is also identified as a data unit. Folder data structures start with 4 bytes header. The header bytes are not always consistent, it takes different byte value for each messages and the reason was not found. A set of bytes, which are used to indicate the start and the end of SMS text, was identified. Similarly another set of bytes were identified, which are used to indicate the start and the end of the source mobile number. Two sets of unique bytes were identified in inbox folder, one set of the unique bytes was used to link data that is relevant to a particular SMS and the other set of bytes was used as a unique inbox message-id followed by a pattern that was used as end of SMS text part.

Further analysis helped to identify how to distinguish between messages from different folders. The information of the Folder, to which the SMS belongs to, was appended at the end of data unit.

Two different patterns, which were used to indicate starting of the source mobile number, were identified. One pattern was used to refer to the source within folders and the other pattern was used to refer the source in other data units such as contact list.

Another data unit was identified and that was used as contact list. The contact list data unit contains source mobile number, sender's first name, surname and others. The pattern used to separate each detail of the sender was also identified. A byte that was used to rank SMS data was also identified.

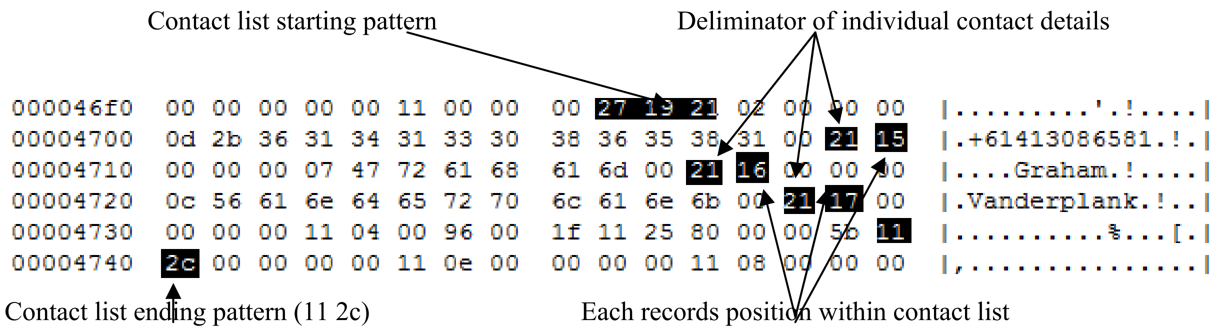
Figure6 shows message data structure of the inbox for a single SMS. Similar structure repeats for every SMS record.



```
                                           99 00 99 00 03 00  |..@@............|
000028e0  00 00 fc 0f a0 40 40 00  00 00 00 00 00 82 00 00  |.....@@.........|
000028f0  00 00 00 00 00 11 00 00  00 27 15 22 00 00 00 00  |.........'."....|
00002900  05 54 65 73 74 00 51 09  80 32 60 57 04 17 c9 01  |.Test.Q..2`W....|
00002910  11 0c 00 00 00 64 12 01  00 00 00 02 31 1f 00 00  |.....d......1...|
00002920  00 25 00 00 00 21 11 00  00 00 27 14 21 02 00 00  |.%...!....'.!...|
00002930  00 0d 2b 36 31 34 30 32  35 36 32 34 38 36 00 11  |..+61402562486..|
00002940  01 40 40 a0 10 ff ff 12  04 42 78 d1 5c 11 2a 00  |.@@......Bx.\.*.|
00002950  00 00 01 21 23 00 00 00  0e 6d 73 67 43 6f 72 65  |...!#....msgCore|
00002960  3a 49 6e 62 6f 78 00 31  05 00 00 00 08 0f a0 40  |:Inbox.1.......@|
00002970  40 07 a0 40 40 ff ff 00  00 00 00
```

Figure6: Inbox data structure

The above figure shows that the message belongs to the inbox folder. The 'message numbering' part indicates position of message in inbox folder. Figure7 below shows detail description of a contact list.


Contact list starting pattern
Deliminator of individual contact details
Contact list ending pattern (11 2c)
Each records position within contact list

```
000046f0  00 00 00 00 00 11 00 00  00 27 19 21 02 00 00 00  |.........'.!....|
00004700  0d 2b 36 31 34 31 33 30  38 36 35 38 31 00 21 15  |.+61413086581.!.|
00004710  00 00 00 07 47 72 61 68  61 6d 00 21 16 00 00 00  |....Graham.!....|
00004720  0c 56 61 6e 64 65 72 70  6c 61 6e 6b 00 21 17 00  |.Vanderplank.!..|
00004730  00 00 00 11 04 00 96 00  1f 11 25 80 00 00 5b 11  |..........%...[.|
00004740  2c 00 00 00 00 11 0e 00  00 00 00 11 08 00 00 00  |,...............|
```

Figure7: Contact list structure

Abbreviations used in coming sections are listed in below table.

ABBREVIATIONS

Abbreviation	Full form	No of bytes
CD	Contact Detail	Varies (Specify by CDL)
CDL	Contact Detail Length	1
CLSP	Contact Detail Starting Pattern	3
DA	Destination Address	Varies (Specify by MNL)
DUSP	Data Unit Start Pattern	3
ECL	End of Contact List	2
EMID	End of Message Identifier	2
EMNP	End of Mobile Number Pattern	2
EOF	End Of File	5
EOR	End Of Record	2
FL	Folder Information	Varies
IMID	Inbox Message Identifier	4
LFL	Log File Length	1
Log	Log file	Varies (Specify by LFL)
LSP	Log Start Pattern	2
MC	Message Core: xxxx	Varies (Specify by MCL)
MCL	Message Core Length	1
MD	Message Data	Varies
MID	Message Identifier	8
MN	Mobile Number	Varies (Specify by MNL)
MNL	Mobile Number Length	1
MP	Message Position	1
PB	Padded Bytes	Varies(Specified at locations wherever necessary)
SFLP	Start of Folder Pattern	2
SMIDP	Start of Message-ID Pattern	2
SMNP	Start of Mobile Number Pattern	3
SNCD	Start of Next Contact Detail	2
UD	Used Data	Varies (Specify by UDL)
UDEP	Used Data End Pattern	2
UDL	User Data Length	1
UDSP	User Data Start Pattern	Either 1 or 3 depends on folder

Table1: Abbreviations with their number of bytes

RESULTS AND ANSWERS TO RESEARCH QUESTIONS

Stage1

i. How to identify an empty file?

File status byte is stored at Offset 0x18. If its value is 0x0b then the file is empty and value of a non empty file is 0x0c. In addition to the offset 0x18, an empty files contain a unique hex pattern "63 4d 6b 71" somewhere within the file.

Stage2

From stage2 analysis, the purpose of this research is achieved. Below paragraphs discuss all findings and methods of retrieval.

ii. How to identify and retrieve read, unread, saved, deleted, drafted and template messages?

File status flag, which indicates whether file is read or not, were not identified. However, it was observed that if the file was an unread SMS it keeps the database at offset 0x1000, once the SMS is read it moves it to different location, mostly to offset 0x2000. But these offsets were not consistent, and so it is not possible to exactly identify unread and read SMS messages.

Other activities like saved, deleted, drafted and template messages were identified and recovery methods also found. Below paragraphs discuss identification and recovery process.

Reading messages from inbox

Inbox data structure is comprised of message data (MD), source address (SA) and folder information (FL). Figure8 shows main parts of inbox data structure and each part is magnified.

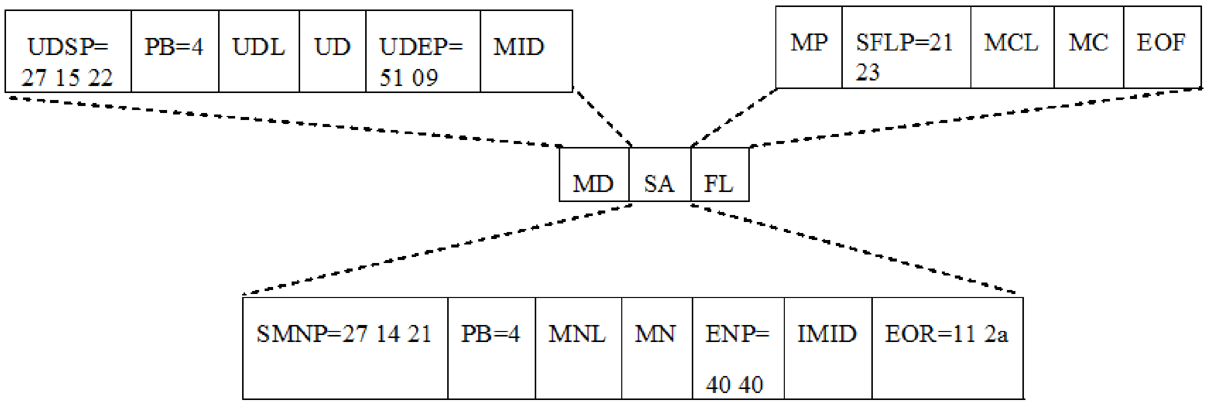


Figure8: Inbox data structure with magnified elements

a) User data (UD) starts with a hex pattern of "27 15 22" then neglect the next four bytes following it. Fifth byte denotes length of user data (UDL). The text message starts from the sixth byte. The text message ends with a hex pattern of "51 09"(UDEP).

b) After the end of text message, the next consecutive eight bytes are the unique message identifier (MID). It is used to refer this text message in other areas. This will help to identify the order of SMS and its relevant contact list data unit. And also these 8 bytes always end with a hex pattern of "c9 01". This might contain time related information but at this time it was unable to confirm.

c) After the unique message identifier, the source mobile number starts with a hex pattern of "27 14 21"(SMNP). The next four bytes are padded with zeros. Then the fifth byte indicates the mobile number length (MNL) and the next consecutive bytes are the source mobile number. This mobile number ends with a hex pattern of "40 40" (ENP).

d) A hex pattern "11 2a" is used to indicate the end of message record (EOR).

e) A set of four bytes that forms a unique inbox message identifier (IMID) was identified. This IMID is located just before the end of message record (EOR) pattern. This IMID appears for messages within the inbox only.

f) After the end of the message, the record folder information starts with a hex pattern of "21 23"(SFLP). Next three bytes are neglected. Fourth byte indicates length of message core (MCL). Message core contains name of folder.

g) The byte before the pattern "21 23" indicates the message records position in inbox (MP).

h) Inbox folder ends with a hex pattern of "a0 40 40 ff ff".

Reading message from outbox

Out box record structure consists of destination address (DA), message identifier (MID), message data (MD) and folder information (FL).

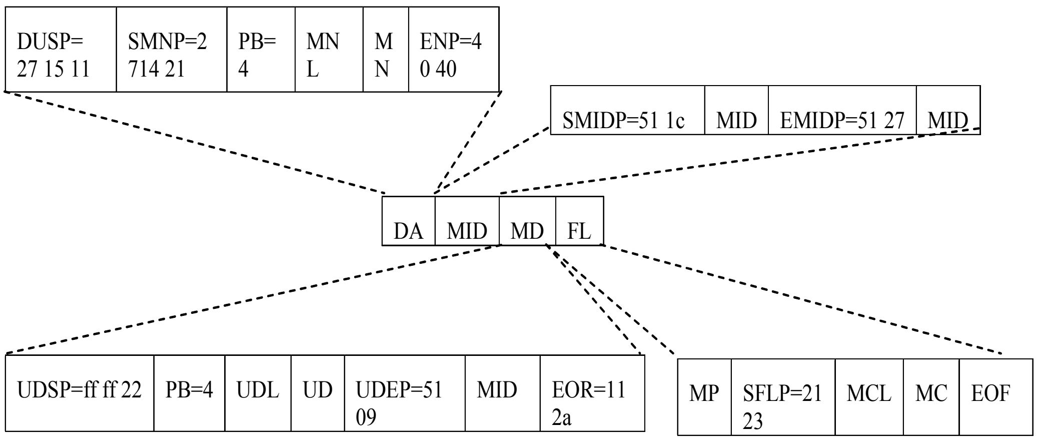


Figure9: Outbox data structure with magnified elements

a) Like inbox, destination mobile number starts with a hex pattern of "27 14 21" and fifth byte from SMNP denotes mobile number length then mobile number starts at sixth byte and ends with a hex pattern "40 40".

b) Then the unique message identifier starts with a hex pattern of "51 1c"(SMIDP) and the next set of eight consecutive bytes is the unique message identifier. This identifier ends with a hex pattern "51 27" (EMIDP).

c) Then the user data starts with a hex pattern of "ff ff 22". Next four bytes are padded with 0s or any other hex values and the fifth byte specify length of user data (UDL). The user data (UD) starts immediately after the UDL and ends with a hex pattern "51 09".

d) Then next 8 bytes are the same unique message identifier followed with the end of record pattern "11 2a".

e) Folder information starts and ends as stated in inbox part.

Reading saved message

Reading 'saved folder' is similar to reading inbox but there are a few difference like text message starting pattern and folder information are different and saved messages do not have unique inbox message identifier (IMID). Inbox data structure folder information part contains "msgcore:Inbox" but 'saved' data structure folder information part contains "msgcore:Saved". User data starting pattern for saved message is "02 22".

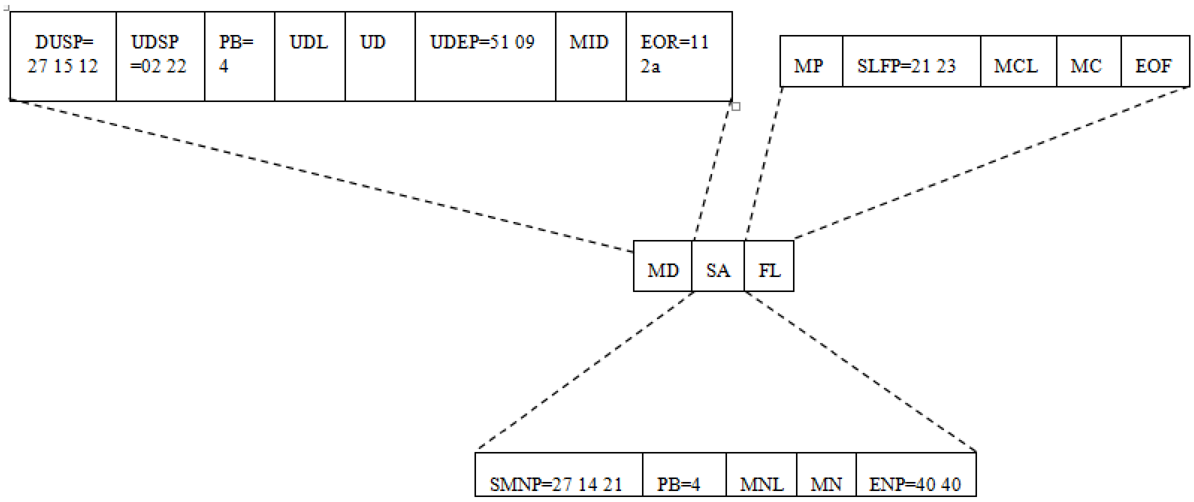


Figure10: Saved message data structure

Reading sent message

This folder contains log entry of sending attempt and relevant text message with destination mobile number. As outbox data structure, 'sent message' data structure also starts with destination mobiles number.

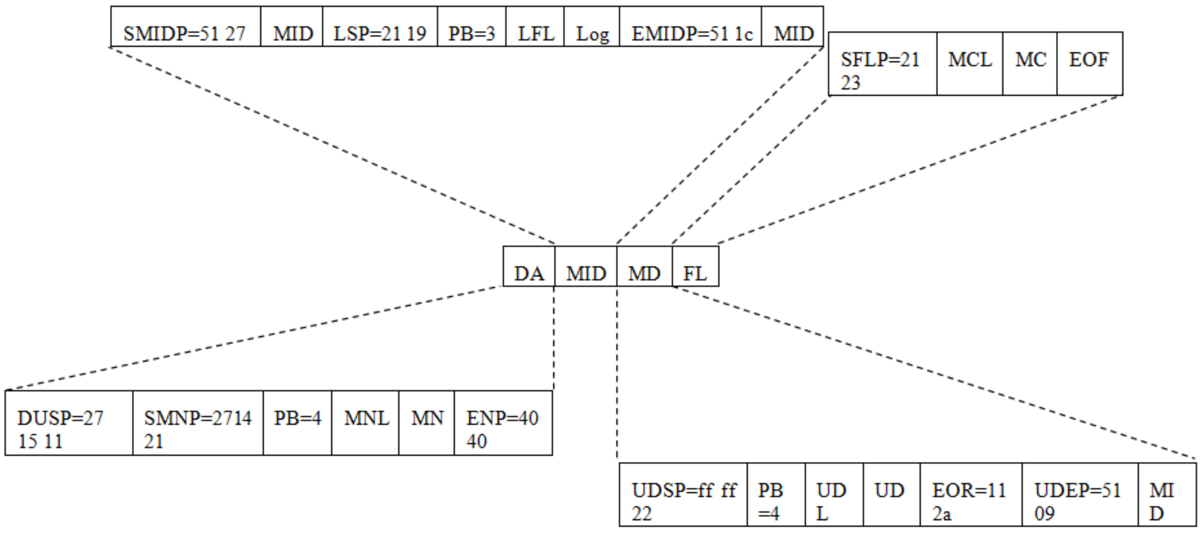


Figure11: Sentbox data structure with magnified elements

Destination mobile number start and end pattern are same as outbox structure.

a) Starting and ending pattern structure of MID of 'sentbox' are "51 27'and "51 1c" respectively. The MID section of 'sentbox' also contains log entries of sending attempts.

b) Next consecutive eight bytes from SMIDP are unique message identifier.

c) Log file starts (LSP) with a hex pattern of "21 19". Then next three bytes are padded with 0s. Fourth byte shows log file length (LFL). Log information follows immediately after LFL. Eight bytes followed by EMIDP are message-id.

d) The user data starts with a hex pattern “ff ff 22”. Next four bytes are padded with 0s or any other hex values. Fifth byte denotes length of user data and the user data starts from 6th byte.

e) End of record (EOR) pattern appears before the end of user data pattern (UDEP).

f) The 8 bytes immediately after UDEP are MID.

g) Sent box does not have a byte to indicate a message position (MP) within ‘sentbox’ folder.

Reading drafted message

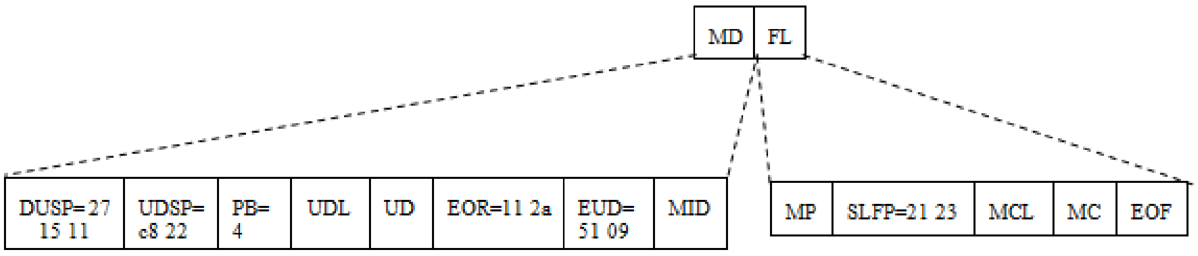


Figure 12: Drafted message format

a) Draft data structure is identified with a hex pattern “27 15 11”.

b) Similar to saved message its text UDSP is identified by the pattern “c8 22”.

c) End of text pattern “51 09” appears earlier than end of message record pattern.

d) The 8 bytes unique message identifier is placed before the starting pattern of folder information.

e) Folder information part (FL) does not have a message position byte (MP).

Reading message templates

Template data structure is exactly same as ‘draft’ data structure but it varies at folder information part only. Folder information part contains ‘msgcore: Template’.

Reading Deleted messages

Deleting operation does not actually delete text message from the message file. The message is still in the file but a flag is set to not to display the message in folders. But the flag was not identified during this research. Normal inbox reading procedure will read the deleted messages also. Following will help to identify deleted message.

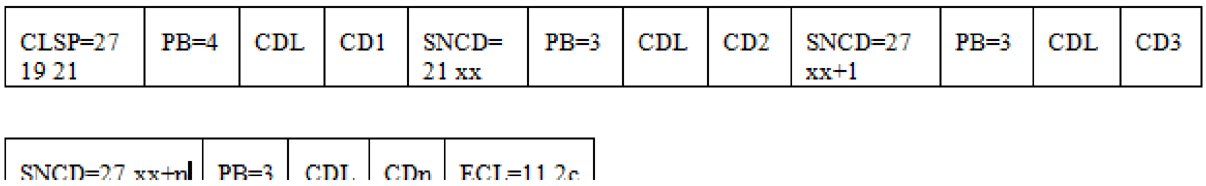
a) Deleted message is still in inbox. So reading procedure is same but it lacks the source mobile number part.

b) The folder information part is repeated twice . As usual, first folder information ends with footer then again the folder information starts with a starter byte pattern “23” and ends with footer.

Reading contact list

iii. How to identify and retrieve source of message and other relevant contact information?

Contact list or phone book contains source or destination contact details such as mobile number, first name, last name and others. The first record of contact list is mobile number.



CLSP=27 19 21	PB=4	CDL	CD1	SNCD= 21 xx	PB=3	CDL	CD2	SNCD=27 xx+1	PB=3	CDL	CD3

SNCD=27 xx+n	PB=3	CDL	CDn	ECL=11 2c

Figure 13: Contact list format

Contact list starts with a hex pattern “27 19 21” (CLSP). The next 4 bytes are padded. The fifth byte is used to specify the first contact detail length (CDL). Then first contact detail starts from the sixth byte, usually the first contact detail is a mobile number.

Next contact detail starts (SNCD) with a hex pattern of “21 xx” where “xx” increments with number of contact details. The ‘xx’ is also used to arrange the contact details. In this analysis xx started from 0x15.

Then the next three bytes are padded with zeros. Again fourth byte is CDL then second contact detail. Then again start of third contact detail if available. Otherwise contact list is terminated with a hex pattern of “11 2c”(ECL).

iv. **In some occasions there may be identical messages from a particular source. If so how to distinguish between messages?**

Unique message identifier (MID) or unique inbox message-id (IMID) of SMS record in inbox can be used to distinguish messages, even if those messages were from same source and contains same data.

PRACTICAL DIFFICULTIES

Palmmsgv001 file repeats the message database at least twice and each SMS record several times within a database. Any attempt to read palmmsgv001 file will retrieve duplicate records of each SMS. Sometimes a SMS message may undergo several operations such as save and delete; if message retrieval procedure sorts out SMS records based on folder name then inbox folder, save folder and delete folder will have the same message. It is important to mark duplicate messages during retrieval procedure.

Unique message identifier will help to tag duplicate SMS records and also it will help to mark same SMS record in all folders. To find deleted messages form recovered inbox message this unique identifier can be used.

Encountered Issues

This analysis did not help to find timing parameters. It was assumed that timing parameter must hide within unique message identifier (MID). Time parameter is an important parameter in forensic analysis. Message identifier was analysed to find any time related information. Due to time restriction, message-id was not fully analysed. Analysing the file may reveal time related information.

It was identified in most of the samples, when the SMS was unread the message database appeared at offset 0x1000. Once it is read the message database was moved to offset 0x2000. But some sample files with unread SMS showed different behaviour, it stored message database to some other location. Moreover no any special flag were found to distinguish read message from unread message or vice versa.

It was observed that with multiple SMS records, message data base did not show its signature and in some cases even it did not include the folder initialisation part. These issues, however, did not affect the aim of the research.

CONCLUSION

Thread SMS is a new technology in mobile phone SMS message service and Palm Treo 750 SMS service. The research work on palmmsgv001, message file of Palm Treo 750, revealed that it is possible to recover all message data from Palmmsgv001 message file. A methodology, based on pattern search, to recover data from PalmMsgV001 is also proposed.

This research work did not find message timing parameters. It is suspected that the unique message identifier may be constructed with time information but not resolved. Timing parameters are important parameters in forensic analysis. This emphasizes a future research to determine timing parameters. There are still several information needs to be identified such as group-id, to identify a message from same source, and flag that indicates message status.

The research work was limited with SMS recovery but the message file contains MMS (Multi Media Service) data also. Future research is needed to study MMS parameters so as to recover MMS data.

REFERENCES

ACPO (2003). Good Practice Guide for Computer based Electronic Evidence. *Journal*. Retrieved from http://www.acpo.police.uk/asp/policies/Data/gpg_cpmputer_based_evidence_v3.pdf

Bodic, G. L. (2005). *Mobile Messaging Technologies and Services: SMS, EMS and MMS* (2nd ed.): John Wiley and Sons.

ETSI. (1999). *ETSI TS 100 901 V7.4.0 (1999-12): Digital cellular telecommunications system (Phase 2+);Technical realization of the Short Message Service (SMS);*

(GSM 03.40 version 7.4.0 Release 1998). Retrieved 18 July, 2008, from http://amber.feld.cvut.cz/user/pokorny/bpdp/GSM_0340.PDF

McCarthy, P. (2005). *Forensic Analysis of Mobile Phones*. Retrieved 18 Aug, 2008, from http://esm.cis.unisa.edu.au/new_esml/resources/publications/forensic%20analysis%20of%20mobile%20p hones.pdf

Press, T. A. (2008). Dozens of college students busted in drug sting. Retrieved 28/9/2008, from http://www.msnbc.msn.com/id/21134540/vp/24487634#24487634

Trosby, F. (2004). *SMS, the strange duckling of GSM*. Retrieved 22 July,2008, from http://www.telenor.com/telektronikk/volumes/pdf/3.2004/Page_187-194.pdf

UltraEdit Text Editor Features. (2008). Retrieved 5 Aug,, 2008, from http://www.ultraedit.com/products/ultraedit/ultraedit_features.html

Visual File And Directory Comparison Tool. (2008). Retrieved 25 Aug,, 2008, from http://www.prestosoft.com/edp_examdiffpro.asp

Data Hiding in Windows Executable Files

DaeMin Shin, Yeog Kim, KeunDuck Byun, SangJin Lee
Center for Information Security Technologies (CIST),
Korea University, Seoul, Republic of Korea
{grace_rain, yeog, gdfriend, sangjin}@korea.ac.kr

Abstract

A common technique for hiding information in executable files is the embedding a limited amount of information in program binaries. The hiding technique is commonly achieved by using special software tools as e.g. the tools presented by Hydan and Stilo in (Rakan, 2004, Bertrand, 2005). These tools can be used to commit crimes as e.g. industrial spy activities or other forms of illegal data access. In this paper, we propose new methods for hiding information in Portable Executable (PE) files. PE is a file format for executables used in the 32-bit and 64-bit versions of the Windows operating system. In addition, we discuss the analysis techniques which can be applied to detect and recover data hidden using each of these methods. The existing techniques for hiding information in an executable file determine the total number of bytes to be hidden on the foundation of the size of the executable code. Our novel methods proposed here do not limit the amount of hidden code.

Keywords

Portable Executable (PE), Executable file, Program binaries, Hiding information

INTRODUCTION

It is general in hiding information to embed a limited amount of information in media such as image and music files, so the embedding methods could be under surveillance from system managers in an organization that requires the high level of security. This fact requires researches on new hiding techniques and cover objects which hidden information is embedded in. It is the result from the researches to embed information in executable files. It can be considered that Stilo and Hydan represent common techniques for the embedding information in executable files. These techniques make original files modified, so code signing techniques that guarantee the integrity of the code can be used for the detecting hidden information. But, it becomes general phenomenon making executable files as the level of using computer and computing environment has been raised. In addition, it has not been general to use code signing techniques.

Silo and Hydan modify program binaries that have been in optimization, so the performance of the program binaries may fall. In addition, the amount of information to be embedded in executable files by using Silo and Hydan is limited because these tools determine the number of bytes to be hidden on the foundation of the size of the program binaries. Therefore, it needs to carry out researches on new hiding techniques that consider the efficiency of program binaries and the amount of information to be hidden. In this paper, we examine the new methods that consider the efficiency and the amount of information to be hidden. Further we discuss the analysis techniques which can be applied to detect and recover data hidden using each of these methods.

PORTABLE EXECUTABLE (PE)

The Program Loader that is a subset of the Windows System assumes the loading executable files into a virtual memory, so the executable files have the format that the Program Loader can identify, and the format is called PE (Portable Executable). It is necessary to know the PE format and RVA which is an address type used in the PE in order to understand the new methods for hiding information in the PE, so we briefly describe the format and the address type.

RVA

RVA is a position unit in the PE, and the RVA is used as an offset from the start-address of a PE file loaded on the memory. The start-address of a file on the memory is in ImageBase that is one of the attributes of the PE file. For instance, if the ImageBase of a file is 0x00400000 and one position of the file is 0x1000(RVA), the position on the memory will be 0x00401000.

PE Format

The header of PE format starts with MS-DOS stub that is used for printing a message, “This program cannot be run in DOS mode”, if the operating system can’t identify the PE on execution time.

IMAGE_NT_HEADER located in the position after the MS-DOS stub has the information for the execution of a file, and consists of IMAGE_FILE_HEADER and IMAGE_OPTIONAL_HEADER. The IMAGE_FILE_HEA-DER has the information on the file, such as create time and machine type. The IMAGE_OPTIONAL_HEADE-R has the information on functions used in the file and on the start-address of the file on a memory, and the infor-mation is managed by IMAGE_DATA_DIRECTORY.

A PE file except the header is composed of several sections that are basic unit of code or data within a PE or COFF file (Microsoft, 2006). IMAGE_SECTION_HEADER that is located in the position following to IMAG-E_OPTIONAL_HEADER has the information on each section. The information consists of PointerToRawData, SizeOfRawData, VirtualAddress, and VirtualSize. The PointerToRawData and the SizeOfRawData respectively mean the position of each section and the size of each section on the file. The VirtualAddress and the VirtualSize respectively mean the position of each section and the size of each section on the memory. The size of each section on the file is a multiple of FileAlignment that is in IMAGE_OPTIONAL_HEADER. If the amount of the data of a section is smaller than the size of the section that is allotted on compile time, the slack space of the section occurs.

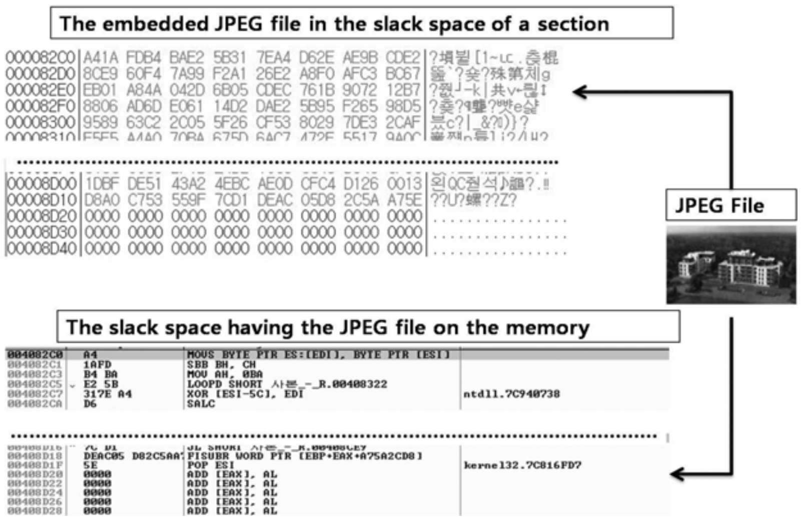
The common sections used in the PE include a .text that has program binaries, .data, .idata that has information on export and import functions, .edata, and .rsrc section. An .idata section has the information on import functions used in executable files during the period of an execution. When loading a file on a memory, the Loader refer to the .idata section to move the address of each import function used in the file on Import Address Table(IAT) which the IMAGE_DATA_DIRECTORY of the file has the position of.

NEW METHODS FOR HIDING INFORMATION

In this section, we will see two methods that embed information in executable files for hiding information. First method is the using the slack space of each section. Since slack spaces are the unused space of each section, it has an advantage that does not generate error on execution time to embed information in the slack space. The method, however, is not applicable to the case when much information has to be embedded. Second method is the embedding information in the .text section of executable files. This method increases the entire size of executable files, and the increased size affects the sections that have been located in the position following to the .text section, which makes the executable files unexecuted. Therefore, it needs extra tasks to execute clearly the executable files of which the information has been embedded in the .text section, but this method has advantages that can maintain the original program binaries that have been optimized on compile time, and do not limit the amount of information to be hidden

Embedding in Slack Spaces

Slack spaces are filled with 0x00 by less than the FileAlignment (Goppit, 2005). The upper picture in figure 1 shows that an encrypted jpeg file has been embedded in a slack space. The slack space is from 0x82C0 to 0x8FF0, and the first part that is not filled with 0x00 in the upper picture is the embedded jpeg file, and the following part that is filled with 0x00 is the slack space. The under picture in figure 1 shows the slack space loaded on the memory. It doesn’t generate error on execution time to embed information in slack spaces for hiding information. We have to know the start position of the slack space of each section in the executable file and the encrypt key that is used for encrypting information to extract hidden information.


The embedded JPEG file in the slack space of a section
JPEG File
The slack space having the JPEG file on the memory

```
000082C0|A41A FDB4 BAE2 5B31 7EA4 D62E AE9B CDE2|?塡뷥[1~ㄸ.춯棍
000082D0|8CE9 60F4 7A99 F2A1 26E2 A8F0 AFC3 BC67|鑑`?솟?殊箕체g
000082E0|EB01 A84A 042D 6B05 CDEC 761B 9072 12B7|?쮜┘-k|共v←릥↕
000082F0|8806 AD6D E061 14D2 DAE2 5B95 F265 98D5|?춏?¶뺠?뺫e샾
00008300|9589 63C2 2C05 5F26 CF53 8029 7DE3 2CAF|븄c?|_&?)}?
........................................................................
00008D00|1DBF DE51 43A2 4EBC AE0D CFC4 D126 0013|왼QC젪석♪讍?.‼
00008D10|D8A0 C753 559F 7CD1 DEAC 05D8 2C5A A75E|??U?螺??Z?
00008D20|0000 0000 0000 0000 0000 0000 0000 0000|................
00008D30|0000 0000 0000 0000 0000 0000 0000 0000|................
00008D40|0000 0000 0000 0000 0000 0000 0000 0000|................
```

```
004082C0   A4            MOVS BYTE PTR ES:[EDI], BYTE PTR [ESI]
004082C1   1AFD          SBB BH, CH
004082C3   B4 BA         MOV AH, 0BA
004082C5 ⌄ E2 5B         LOOPD SHORT 사본_-_R.00408322
004082C7   317E A4       XOR [ESI-5C], EDI                        ntdll.7C940738
004082CA   D6            SALC
........................................................................
00408D18   DEAC05 D82C5AA FISUBR WORD PTR [EBP+EAX+A75A2CD8]
00408D1F   5E            POP ESI                                  kernel32.7C816FD7
00408D20   0000          ADD [EAX], AL
00408D22   0000          ADD [EAX], AL
00408D24   0000          ADD [EAX], AL
00408D26   0000          ADD [EAX], AL
00408D28   0000          ADD [EAX], AL
```

Figure 13: Embedding a JPEG file in the slack space of a section.

Embedding in .Text section

This section introduces the method for embedding information in a .text section, and figure 2 shows the flow diagram of the method. First of all, we have to secure the enough space in a .text section in which information has to be embedded for hiding it, and then modify the section table that has the size and the position of each section, and the IMAGE_OPTIONAL_HEADER that has the size of the entire file. Once the information by 4096 byte has been embedded in the .text section, the PointerToRawData of both .rdata and .data section located in the position following to the .text section are moved back by 4096 byte due to the embedding. It is needed to modify the positions information of the sections in the section table. In addition, since the entire size of the executable file has been changed, we have to modify in the IMAGE_OPTIONAL_HEADER both the SizeOfCode that has the sum of all section sizes and the SizeOfImage that is the enough size which the executable file needs on the memory to be loaded. Otherwise, the executable file may have the mark of the embedding because of the difference between the entire sizes of the file with the size information that exists in the IMAGE_OPTIONAL_HEADER.

Then, we have to modify the values of the Import Table that has been moved back, the position of the table, the values of the IAT that has the addresses of the import functions used in the executable file, and the position of the IAT. The Import Table has the information of the functions used in the executable file. The Loader on execution time locates the address of each function in the IAT by the referring to the Import Table.

Finally, we need a procedure that confirms whether or not absolute addresses used in the .text section change due to the embedding data. If the absolute addresses have changed, we have to make the absolute addresses modified applicably to refer to the reference positions prior to the embedding, for which we can use a .reloc section. The absolute address decided on compile time is the reference position on the memory, and it is used in global variables, function pointers, flow control instructions (break, continue, goto, throw), CALL, JMP, and so on.

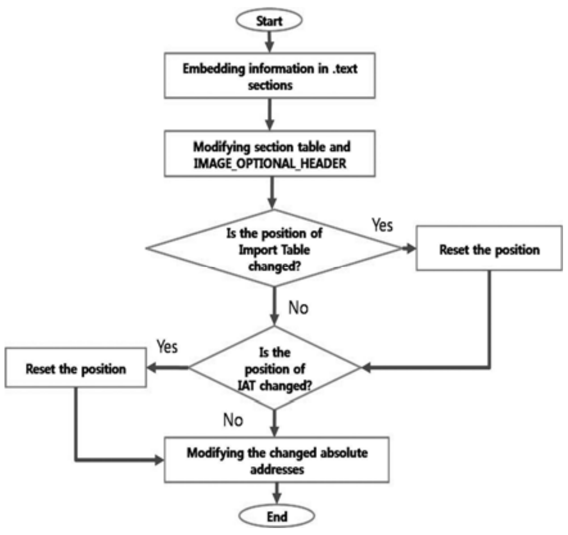

Start
Embedding information in .text sections
Modifying section table and IMAGE_OPTIONAL_HEADER
Is the position of Import Table changed?
Yes
Reset the position
No
Is the position of IAT changed?
Yes
Reset the position
No
Modifying the changed absolute addresses
End

Figure 14: The flow diagram for the embedding process in a .text section

The .reloc section (Microsoft, 2006) is the array of the basic relocation block shown in figure 3. The Page RVA of the block is the start position of the program binaries which this relocation block applies to. The Block Size of the block is the size of this block. The last array consisting of 2byte offsets has the position of each absolute address on the file, and each offset is a distance from the Page RVA to each absolute address. Therefore, we can modify changed absolute addresses on the embedding time by using the .reloc section if the embedding data in the executable file makes the absolute addresses changed.

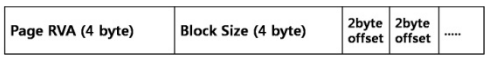


Figure 15: Basic relocation block

Although this method has a disadvantage, which requires additional tasks for a clear execution due to deliberately modifying the .text section, it does not limit the amount of information to be hidden. In addition, it does not make differences in the execution result between original files and the embedded files.

If an executable file that has already existed is used for hiding information, it can be easily detected by code signing techniques or by comparing with attributes such as the file size of same files. Thus, it makes sense that the executable file that information is to be embedded in has to be unique.

It is most cases to embed encrypted and compressed information in a .text section in order to have a difficulty in recovering the information when the information is detected. It can be, therefore, used as a detecting method to verify that the format of program binaries consists of instructions and the operands of them by applying disassemble softwares to the executable file, so it requires that the method embedding information in a .text section have a countermeasure against discriminating the information from the program binaries by using disassemble softwares.

Embedded information can be translated into the format of program binaries by using a method that inserts the embedded data in the operands of instructions. The process of the method is shown in figure 4, and figure 5 shows the embedded information that is on the black underlines after the process. As a result, it can highly lower a possibility that the embedded data is discriminated from the program binaries by using disassemble softwares to translate the embedded data into the form of program binaries.

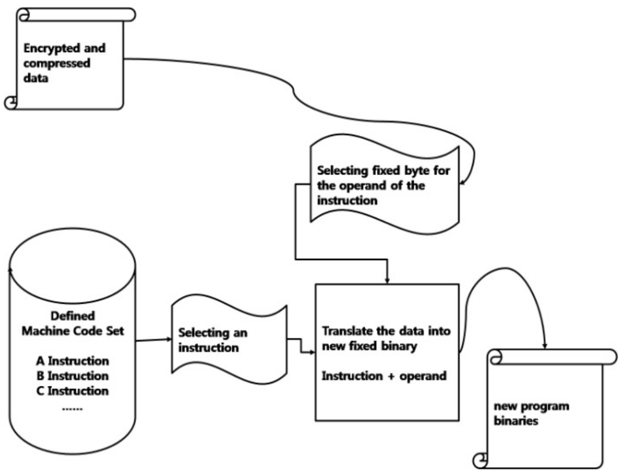
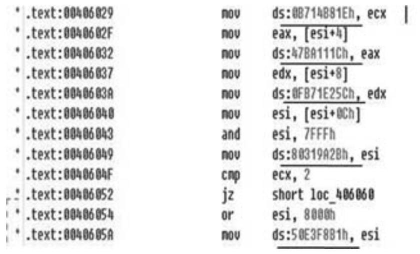


Figure 16: The process for translating embedded data into new program binaries.



```
.text:00406029      mov     ds:0B714B81Eh, ecx
.text:0040602F      mov     eax, [esi+4]
.text:00406032      mov     ds:47BA111Ch, eax
.text:00406037      mov     edx, [esi+8]
.text:0040603A      mov     ds:0FB71E25Ch, edx
.text:00406040      mov     esi, [esi+0Ch]
.text:00406043      and     esi, 7FFFh
.text:00406049      mov     ds:80319A2Bh, esi
.text:0040604F      cmp     ecx, 2
.text:00406052      jz      short loc_406060
.text:00406054      or      esi, 8000h
.text:0040605A      mov     ds:50E3F8B1h, esi
```

Figure 17: The embedded data after the process in figure 4.

Encrypt key and the start position of embedded data in a .text section are needed in an extraction process. In the process, the first of all, the .text section in which information is embedded is parsed from the executable file. Then, the embedded data is extracted from the parsed .text section by using previously shared key and the start position of the hidden information.

ANALYSIS TECHNIQUES

In this section, we discuss analysis techniques which can be applied to detect and recover information hidden using proposed here new methods. Figure 6 shows a flow diagram for the analysis process.

Each data of sections is in fixed format as e.g. program binaries or relocation blocks in a relocation section. Hidden information in section slacks is generally encrypted and compressed, so it can helps decide whether or not each section has hidden information to compare the format of each section and the values of each section.

Program binaries are composed of several basic blocks that consist of instructions. The starting position of all basic blocks in program binaries must be referred to by instructions relating to jump as e.g. CALL, JMP, JNZ. Otherwise, the basic blocks of which the instructions in the program binaries don't refer to the starting position may be unused parts that have hidden information in the file, so it is used as an analysis technique that can be applied to detect the hidden information in the .text section to confirm that the starting position of all basic blocks is referred to by the instructions relating to jump.

The reference range of the operands of instructions relating to jump in program binaries is limited within the memory area assigned to the file on execution time, but since hidden information is encrypted and than used as the operand of the instructions, the reference range of the operands can be out of the memory area assigned to the file on execution time. It is, therefore, used for an analysis technique to confirm that the reference range of the operand of all instructions is within the memory area assigned on execution time.

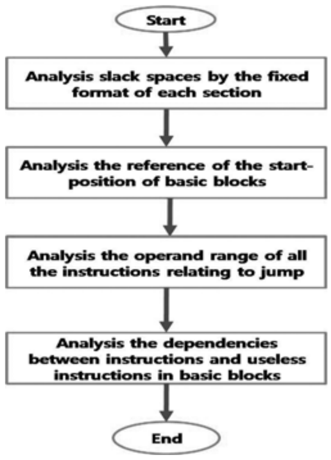

Start
Analysis slack spaces by the fixed format of each section
Analysis the reference of the start-position of basic blocks
Analysis the operand range of all the instructions relating to jump
Analysis the dependencies between instructions and useless instructions in basic blocks
End

Figure 18: Analysis techniques for the detection.

Typically a compiler's back-end goes through 4 phases, instruction selection, register allocation, instruction scheduling, code layout. Instruction scheduling phase finds all possible orders for an instruction process considering a dependency between instructions, and than selects one order of them. Therefore, dependencies between instructions in a basic block exist (Bertrand, 2005). But, it is difficult to make embedded binaries exist with dependencies between instructions on the embedding time due to randomly selected instructions which random information is to be embedded in as operands, so it can be useful to confirm whether or not dependencies between instructions exist in program binaries when analysing suspicious executable files that may have hidden information.

In addition, even though program binaries that are decided on compile time have been in optimization, new program binaries that have hidden information can be composed of useless (suboptimal) instructions by randomly selecting instructions. Thus, it can be an analysis technique to confirm useless instructions constantly exist in program binaries.

CONCLUSION

New method proposed in this paper is the advanced result from previous researches on the embedding information in executable files for hiding information. In addition, it has the strong advantages that don't limit the amount of information to be hidden and don't alter optimized program binaries for the hiding information.

Although analysis techniques that we discussed may be helpful in detection and recovery of the hidden data, the process is time consuming without automated softwares. Thus, in the future, we have to carry out the researches on automated software development and recovery techniques from detected information.

ACKNOWLEDGEMENT

This research was supported by Bio R&D program through the Korea Science and Engineering Foundation funded by the Ministry of Education, Science and Technology (M10640030004-08N4003-00410).

REFERENCES

Rakan El-Khalil and Angelos D. Keromytis. (2004) Hydan: Hiding Information in Program Binaries, In *Proceedings of the 6th International Conference on Information and Communications Security (ICICS)*, Pages 187-199.

Bertrand Anckaert, Bjorn De Sutter, Dominique Chanet, and Koen De Bosschere. (2005) Steganography for Executable and Code Transformation Signatures, In *Proceedings of the 7th Information Security and Cryptology (ICISC)*, Pages 431-445

Microsoft Corporation. (2006) Microsoft Portable Executable and Common Object File Format Specification Revision 8.0 May 2006

Goppit. (2005), Portable Executable File Format-A Reverse Engineer View. *Code Breaker Journal*, vol 2, No 3, August 2005

Subverting National Internet Censorship - An Investigation into existing Tools and Techniques

Jason Smart, Kyle Tedeschi, Daniel Meakins, Peter Hannay & Christopher Bolan
School of Computer and Information Science
Edith Cowan University

Abstract

The announcement of a trial of a National level internet filter in Australia has caused renewed interest in the arena of internet censorship. Whilst details on the schemes being tested have been fairly sparse the announcement of the trial itself, has drawn wide condemnation from privacy advocates throughout the world. Given this announcement it was decided to test and compare three of the most popular free tools available that allow for the bypassing of internet censorship devices such as those used within China. Tests were conducted using three software packages, Freegate, GPass and GTunnel which were analysed through packet capture to determine their likely effectiveness against the speculated methods to be employed by the Australian trials. The tests clearly showed that all three applications provide an easy means of subverting any likely filtering method with GPass and GTunnel the more suitable candidates as Freegate still allowed for plain-text DNS requests.

Keywords

Censorship, Gpass, Gtunnel, FreeGate, filtering, internet filtering, government censorship, bypass filtering

INTRODUCTION

Internet censorship in Australia is governed by a tangle of laws and regulation at both Federal and State/Territory level which is attributable to the lack of censorship and control powers granted to the Government in the Australian Constitution (EFA, 2008). Specifically the current legislation is focused as follows (*ibid)*:

- Commonwealth Level - focused on Internet Content Hosts (ICH) and Internet Service Providers (ISP), but no regulation is specified for content creators or end users. This level of legislation allows powers for the Government or an appointed regulator to order providers and hosts to remove hosted content that is deemed “objectionable” or “unsuitable for minors”.
- State / Teritory Level – focused on both ISPs/ICH and user level and differs from state to state and often allow for the prosecution of users for “making available” material that is deemed by legislation to be “objectionable”. Beyond this some jurisdictions also apply a penalty for the viewing or downloading of such content.

Recently, it has been noted that “[Australia’s] net censorship laws are more akin to those in totalitarian regimes than to those, if any, in other countries purporting to be Western democracies” (Libertus, 2008). Such claims are backed by Electronic Frontiers Australia who charge that “following extensive criticism by EFA and other organisations and individuals, it [Australian censorship] remains a draconian scheme unlike any existing or proposed laws in countries similar to Australia” (EFA, 2008). Whilst this issue may be seen as a relatively new reaction to the announced plans of the Federal Government (Marshall, 2008), a lot of the issues seen today relate back to the amendment of the Freedom of Information Act in 2003 (Comlaw, 2008). Whilst claiming to assist in the removal of objectionable content on the internet the result of the bill was to grant the government wider and unilateral powers in what could be deemed “objectionable” (EFA, 2008).

As alluded to early the current plans of the Federal Government to trial and eventually impose mandatory content filtering at the ISP level has drawn a wide rage of criticism and concern (Dudley-Nicholson, 2008; EFA, 2008; Libertus, 2008). With such concerns abounding it seemed to be an apposite time to explore and contrast freely available tools used to bypass internet censorship regimes in other countries and investigate their usefulness and speculate on their effectiveness against future Australian filtering schemes.

SELECTION OF FILTER BYPASS SOFTWARE

To carry out this investigation it was decided to select three of the most popular freely available programs that allowed for the bypassing internet content filters (Global Internet Freedom [GIF], 2008). The three chosen software applications were GTunnel, FreeGate and GPass, each of which functions in a slightly different manner, using different infrastructure or methods of bypassing internet content filters (GPass, 2008; Garden Networks, 2008; Dynamic Internet Technologies, 2008).

GPass

GPass is an anti-censorship application created by The World's Gate Inc designed to bypass censorship methods used to filter out internet content (GPass, 2008). Initially released in July 2006, GPass is now one of the five most used anti-censorship tools and to date is also only one of two (the other being FirePhoenix) that claim to offer multi-protocol support such as Web 2.0, online multimedia, and communication tools such as email and instant messaging (Wang, 2008). According to its producers, GPass is able to bypass censorship devices such as the Chinese Great Firewall allowing anonymous web surfing whilst protecting the users identity and encrypting Internet communication, utilising a proprietary method to find an available server from the GPass server farm and then establishes an encrypted tunnel on the other side of the censorship firewall (GPass, 2008). The operation of GPass is illustrated below in figure 1.

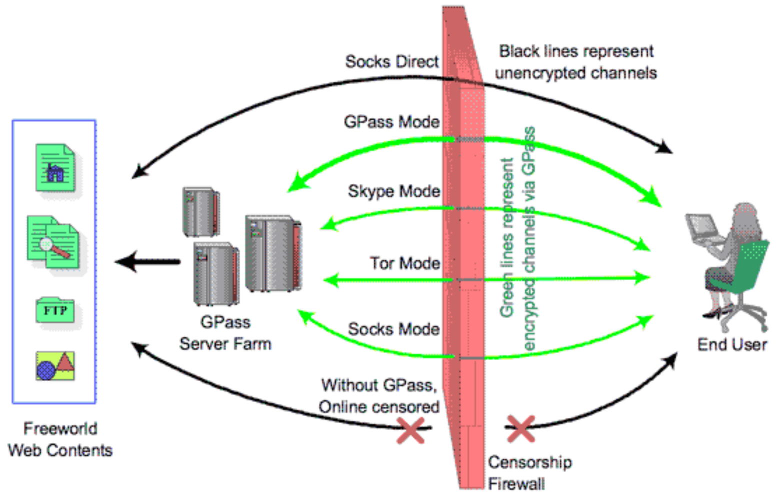

Socks Direct
Black lines represent unencrypted channels
GPass Mode
Skype Mode
Tor Mode
Socks Mode
Green lines represent encrypted channels via GPass
GPass Server Farm
End User
Without GPass, Online censored
Freeworld Web Contents
Censorship Firewall
FTP

Figure 1. Overview of GPass Operation (GPass, 2008)

GTunnel

The second toll selected for the investigation was GTunnel designed by Garden Networks for Freedom of Information, a small development house based Canada (Garden Networks, 2008). The creators of the GTunnel software claim to have created the Garden Network and its front-end application, GTunnel, to provide users with anonymity while browsing the internet (*ibid*). GTunnel functions by providing a local SOCKS v5 or HTTP proxy on the host computer, which facilitates a connection to the Garden Networks server farm, the proxy is then automatically configured inside Microsoft Internet Explorer, which will send all browsing traffic through the secure proxy tunnel thus obfuscating the host computer IP address and anonymising http traffic. GTunnel operates in two modes of operation, namely TOR based and Skype based. These options effect the network used to first make the connection, in both cases the connections pass through GTunnel's own servers prior to reaching its destination. In order to understand the consequences of each of these methods a brief introduction to the Skype and TOR networks is required.

TOR operates by routing encrypted data through a series of TOR nodes, each of these nodes is operated by volunteers who wish to support (or undermine) the TOR network (Bugher, 2007). The last TOR node before the traffic continues out to it's final destination is known as an 'exit node' the exit node is special in that it sees the traffic in the format intended to be received by the destination, this can often be unencrypted and insecure, whilst all the other nodes will only see encrypted data (*ibid*). The concept of a hostile exit node is nothing new and has in the past been used to intercept email and other passwords (Gray, 2007), GTunnel addresses the issue of hostile exit nodes by using its own servers as an encrypted proxy between TOR and the intended destination (Garden Networks, 2008).

This however does not eliminate the risk associated with hostile exit nodes, it merely shifts the trust to another party, in this case Garden Networks. Skype is a peer to peer (P2P) based voice over IP (VOIP) service that is built specifically to bypass firewalls so that customers can make calls regardless of the filtering mechanisms in place (Schmidt, 2006). GTunnel uses an unknown method to make use of the Skype network in order to bypass content filtering networks, it should be noted that Skype makes use of encryption methods that should make decoding of data whilst it is within the Skype network unfeasible (Baset & Schulzrinne, 2004).

In each of these cases Garden Networks is trusted to respect its users privacy as it is able to intercept unencrypted data passing through it's network. However the use of the TOR mode of operation should prevent Garden Networks from determining the origin of this data (this is assuming that the unencrypted data itself does not provide this information).

FreeGate

The final application, Freegate is an anti-censorship program designed by Dynamic Internet Technology (DIT-US), for use in countries where internet censorship occurs (Dynamic Internet Technologies, 2008). Dynamic Internet Technology has a strong affiliation with the United States Department of Defence for whom they created Dynaweb on which the Freegate application is based (*ibid*). The software claims to be a secure and fast way of browsing the internet in relative security having the added feature of not requiring installation on the user's system and working without altering the host computers system settings (GIF, 2008). Freegate has two separate modes one to run in proxy mode, in which it automatically sets the IE proxy settings, the other defaults to Dynaweb servers overseas where you can browse websites straight through the mirror however this option limits its multi-protocol support (Dynamic Internet Technologies, 2008). It is claimed by Dynamic Internet Technologies that due to its method of bypass that, any program that is capable of using the SOCKS v5 proxy is capable of using Freegate to hide its traffic (*ibid*).

TEST SETUP

In order to provide a suitable testing framework, virtual machines were used to ensure that the test environment remained consistent and clean throughout. The host system was running VMware Server 1.0.6 on a Windows XP SP3 system. Two identical virtual machines were created which were running Windows XP SP3 and the following applications as well as the tool being tested:

- Internet Explorer 7,
- Mozilla Firefox 3,
- VMware Tools 7.6.2,
- uTorrent 1.8,
- FileZilla Client 3.1.1.8,
- Xchat 2.8.7c,
- Windows Live Messenger 8.5.1302.1018,
- Google Talk 1.0.0.104,

VMware was configured to have only a host based virtual NIC. Therefore, the virtual NIC on the host machine acted as the gateway for the virtual machine. IP routing was then enabled on the Windows XP host system to allow Internet access to the virtual machine and Wireshark was installed on the host which was then used to capture packets originating from the VMware Virtual NIC thus ensuring all packets would be captured by the experiment. The setup is illustrated in figure 2 below.

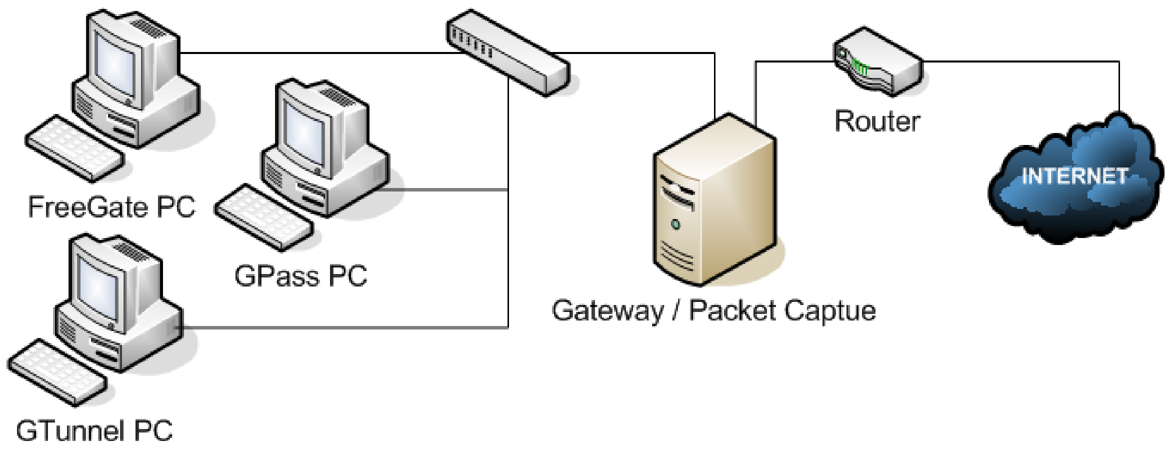


Figure 2. Experimental Setup for Testing

Before the testing was undertaken, a baseline of results were created using the same setup without the Bypass software application in order to allow an easy determination of the effects of the anti-censorship tool. Subsequently the experiments were conducted using the default setups for each application. Upon completion, these tests were then compared with the baseline tests using Wireshark 0.99.8 (Combs, 2008) and NetworkMiner 0.85 (Hjelmvik, 2008) to analyse the packet dumps and extract relevant information.

RESULTS

The results of each test were recorded in order to allow the software applications to be compared to each other and to the baseline in order to determine the effectiveness of these tools for the obsfucation of various protocols.

Table 1. Results of Testing

	Gpass	**FreeGate**	**Gtunnel**	**No Bypass Software**
Webmail	Hidden	Hidden	Hidden	Plaintext
POP3	Hidden	Hidden	Hidden	Plaintext
SMTP	Unknown*	Unknown*	Unknown*	Plaintext
Google Talk	Hidden	Hidden	Plaintext/Hidden**	Plaintext/Hidden**
Windows Live Messenger	Hidden	Hidden	Plaintext	Plaintext
IRC	Hidden	Hidden	Hidden	Plaintext
HTTP	Hidden	Hidden	Hidden	Plaintext
HTTPS	Hidden	Hidden	Hidden	Plaintext
FTP	Hidden	Hidden	Hidden	Plaintext
BitTorrent	Hidden	Hidden	Hidden	Plaintext

* Unable to be tested as none of the clients properly supported SOCKS

** Conversations were hidden whilst protocol messages were not

The results of testing are shown in Table 1 above, in each case the result of the test is listed as hidden or plaintext. Hidden specifies that the contents of the packets as shown in Wireshark and Network Miner appear as encrypted or otherwise obsfucated traffic. It should be noted that in every case where traffic was hidden in some way the protocol used appears to be SSLv3, this protocol is believed to be secure as long as the private certificate of the server is not known. If the private certificate is known it is possible to decrypt all traffic sent to and from the server in question. It may be seen that GPass and FreeGate are able to hide Google Talk and Windows Live Messenger whist GTunnel does not support these protocols. In each of the other tests all three solutions were able to hide the data sent across the network in the method specified. In each case the operational mode of the software (for example, TOR or Skype) had no impact the effectiveness of the software in any observable way.

Aside from the observations presented in Table 1, it should be noted that FreeGate sends all DNS requests in plaintext across the network. It also makes use of the US Department of Defence (DoD) DNS servers, this may prove to be a security risk as the US DoD or any host in position to intercept traffic between the source and destination of these DNS requsts could determine the exact IP address of the computer originating the request as well as the details of the requested domain name. It is conceivable that this would render FreeGate to be unsuitable for bypassing some content filtering mechanisms if these track DNS requests to determine if illict content is being accessed.

Table 2. Methods of bypassing different content filtering methods

Content Filtering Method	Filtering Bypass Method
Filtering DNS requests	Route DNS requests though an encrypted tunnel
Filtering Web page contents	Route web traffic through an encrypted tunnel
Filtering IP addresses	Route all traffic through an encrypted tunnel

In order to evaluate each of the tools usage in terms of bypassing internet content filtering, it is first important to understand how each of these content filtering methods function. There are three main types of internet content filtering, these are, filtering DNS requests, filtering web page content and filtering IP addresses. In each of these cases there is a different method of bypassing these restrictions. These filtering methods and their corresponding bypass methods are presented in Table 2.

Table 3. Results of testing bypass tools against defined bypass methods

	GPass	**GTunnel**	**FreeGate**
Route DNS requests though an encrypted tunnel	Supported	Supported	Not Supported
Route web traffic through an encrypted tunnel	Supported	Supported	Supported
Route all traffic through an encrypted tunnel	Supported	Supported	Supported

The evaluation of each tool to determine which methods of content filtering is supported was conducted and the results displayed in Table 3. These results show that both GPass and GTunnel support all the defined methods for the bypassing of internet content filters, however FreeGate sends all DNS requests openly through the internet and as such it would be possible for a third party determine the domain names on which content may have been retrieved, but not the details of which content was retrieved as this is encrypted. It would also be possible for a content filtering system to intercept and block DNS requests for domains which are considered to be objectionable or otherwise undesirable by the governing body controlling the filtering system. Based on this analysis it can be seen that both GPass and GTunnel would allow for content filtering to be bypassed entirely, however FreeGate does not meet the DNS tunnelling requirement and as such may be unsuitable for accessing prohibited content when behind an internet content filtering system.

CONCLUSION

As outlined in this paper, there is an ever increasing trend in Australia towards draconian internet censorship and the latest moves by the Australian Federal Government to trial filtering methods at the ISP levels is another chip in the ever eroding idea on free speech on the net. Yet despite the claims of the Government that such measures will make the Internet safter, it would seem that anyone with a rudimentary knowledge of search engines would be able to locate free software to bypass such censorship.

This study has found that three of the most popular free tools for ensuring internet privacy and bypassing censorship firewalls would likely function well against likely measures taken at the ISP level. This finding brings into question the need for such filtering, if the method for bypass is so simple and would actually make the detection of illegal activity from packet capture that much more difficult. Whilst it is easy to tout filtering measures as a 'cure' for objectionable material it will have little or no real effect on the access of such materials inside Australia to any but the most naive of users and will likely result in a degrading of the service speeds currently available to Australian internet users.

Unfortunately, despite an initial outcry about the testing of these measures there has been little pressure put on the Government from the general public. One has to wonder whether the public will ever stand up to such inroads into blanket censorship or sit quietly as Australia joins the list of censored nations taking its place with China Iran, and Syria.

REFERENCES

Baset, S., & Schulzrinne, H. (2004). An Analysis of the Skype Peer-to-Peer Internel Telephony Protocol. *Arxiv preprint cs.NI/0412017*.

Bugher, G. (2007, December 10th). Anonymity with TOR and its limits Retrieved November 4th, 2008, from http://perimetergrid.com/wp/2007/12/10/anonymity-with-tor-and-its-limits/

Combs, G. (2008). Wireshark (Version 0.99.8).

ComLaw (2002). Communications Legislation Amendment Bill (No. 1), from http://www.comlaw.gov.au/ComLaw/Legislation/Bills1.nsf/0/011C0CDFB2CA36BECA256F7200246D3E?OpenDocument&VIEWCAT=item&COUNT=999&START=1

Dudley-Nicholson, J. (2008). Australia's compulsory internet filtering 'costly, ineffective', from http://www.news.com.au/technology/story/0,25642,24569656-5014239,00.html

Dynamic Internet Technologies (2008). Freegate, from http://www.dit-inc.us/freegate

EFA (2008). Internet Censorship Laws in Australia, from http://www.efa.org.au/Issues/Censor/cens1.html

Garden Networks (2008). Garden Networks for Freedom of Information, from http://gardennetworks.org/

Global Internet Freedom [GIF] (2008). Global Internet Freedom Consortium – Our Solutions, from http://www.internetfreedom.org/Products-and-Services

GPass (2008). Global Pass Home Page, from http://gpass1.com/gpass/about

Gray, P. (2007, November 13th). The hack of the year Retrieved November 5th, 2008, from http://www.smh.com.au/news/security/the-hack-of-the-year/2007/11/12/1194766589522.html

Hjelmvik, E. (2008). NetworkMiner (Version 0.8.5).

Libertus (2008). Australia's Internet Censorship System, from http://libertus.net/censor/netcensor.html

Marshall, T. (2008). Minister welcomes advances in internet filtering technology, from http://www.minister.dbcde.gov.au/media/media_releases/2008/060

Schmidt, J. (2006, 15 December 2006). The hole trick - How Skype & Co. get round firewalls Retrieved November 6th, 2008, from http://www.heise-online.co.uk/security/How-Skype-Co-get-round-firewalls--/features/82481

Industrial Espionage from Residual Data: Risks and Countermeasures

Iain Sutherland
University of Glamorgan
isutherl@glam.ac.uk

Dr Andy Jones
Centre for Information & Security Systems Research, BT
SECAU – Security Research Centre, Edith Cowan University
andrew.28.jones@bt.com

Abstract
This paper outlines the possible recovery of potentially sensitive corporate information from residual data. It outlines previous work on the recovery of information contained on second hand hard disks and handheld devices and discusses the risk of individuals conducting industrial espionage by targeting specific organizations. It examines the possible avenues for an attacker to obtain a storage device, then discusses the skill level required to extract information from the storage devices and considers the potential risk to an organization from this particular avenue of attack. The paper concludes by proposing a number of possible countermeasures to enable organizations to reduce the risk of this particular form of attack.

Keywords
Industrial Espionage, Residual Data, Hand Held Device, Hard Disk Drive.

INTRODUCTION

In recent years there has been a massive migration of data from the paper based systems where issues of management and security were well understood to digital systems, where the management and control of data storage is less well defined. When the primary storage means for information was paper, the control of access to it, the storage of it and the disposal of it were well practiced processes; but these have not mapped easily into the digital domain. While there was leakage of corporate information through poor processes, accidents and oversights or malicious activity, the volume of information at risk was, in part, controlled by the sheer bulk of the medium.

The advent of the use of digital processing and storage has removed this constraint and it is now possible to store gigabytes of information on a single piece of magnetic or optical media. This coupled with the low cost and difficulty in controlling the storage media, has significantly increased the potential exposure of organisations. In this paper we address the issue of the risk to an organisation as the result of the disposal of obsolete storage and communication devices in the form of computers hard disk drives and handheld devices such as mobile telephones or RIM Blackberry devices.

The secure deletion of data from storage devices has been an issue of discussion for more than ten years (Gutmann 1996) and of some concern for those involved in data security (Garfinkel 2003). There have been a number of studies examining disk disposal practices; assessing the volume and type of residual data remaining on disks available on the second hand market (Garfinkel 2003, Sutherland *et al* 2006, Valli *et al* 2004, 2007). One such annual study currently collecting a fourth year of data has been sponsored by the Centre for Information and Security Systems Research at BT examining disposal practices in the UK, USA, Germany and Australia (Jones *et al,* 2005, 2006, and 2008a). This disk study project has been conducted in order to obtain an understanding of the amounts and types of information that remained on disks offered for sale on the second hand market through various channels including on-line auctions. Most of the studies undertaken to date have indicated that a proportion of these disks contain some residual data that was stored on the drive by the original owners. The study results have highlighted a number of interesting findings. One particular discovery of thestudy worthy of note was that in each of the annual sample sets of disks a significant number of corporate disks were recovered, some containing sensitive corporate data.

A separate, but related study, again sponsored by the Centre for Information and Security Systems Research at BT was focused on examining disposal practices for hand held devices in the UK, USA and Australia (Jones et al, 2008b). This study looked at mobile phones, RIM Blackberry devices and Personal Digital Assistants

(PDAs). While the levels of data that were recovered were different, the handheld device study reflected underlying issues exposed during the disk studies.

STUDY METHODOLOGY

The methodology employed for both the disk study and the hand held device studies were very similar. In the case of the disk study, this involved the purchase of a sample set of disks. In the case of the current UK sample set for 2008, this was in the order of more than 140 disks obtained from computer fairs, computer auctions or via on-line auction sites. The disks were purchased discretely by a number of purchasers and were obtained for the most part either singly or in small batches. These disks were obtained by British Telecommunications (BT) Limited and then supplied 'blind' to the researchers responsible for the imaging and analysis with no indication of where they had been sourced and identified only by a unique sequential serial number and in some cases a manufacturer's label.

The disks were forensically analysed with the objective of determining whether data had been securely erased from the disks or whether they still contained information that was either visible or easily recoverable. If information was present on the disks, this was then examined to determine if it was sufficient to identify an organization or individual. The analysis methods and practices used followed forensic good practice procedures as defined in the Association of Chief Police Officers (ACPO) Good Practice Guide for Computer-Based Electronic Evidence1, with each disk being forensically imaged using commercial software and then stored in secure storage areas; the analysis being undertaken on the forensic images of the original disks. The research to date from the 2005, 2006, 2007 and current 2008 studies indicate that a large proportion of the disks examined still contained information pertaining to a previous owner of the disk, much of which could be considered of a sensitive nature to the organization or individual.

	2005	**2006**	**2007**
Total Number of Disks UK and Australia	116	253	300
Commercial Data Present	60 (70%*)	42 (47%*)	51 (40%*)

** Percentage of readable disks that had not been wiped*

Table 1: Commercial data present on disks in the 2005, 2006 and 2007 surveys (based on Jones 2007)

It was expected that a significant number of inexperienced home users, unaware of the process by which information was stored and removed from the drive, might have left data on a drive when disposing of a disk. However, surprisingly each year a number of business organizations also disposed of disks in an insecure manner. This occurs despite the fact that corporate organizations can be expected to have access to better advice (e.g. Price Waterhouse Cooper (2006)) and resources than the average home user, and that data losses have been discussed heavily in the popular media (AFPS 2006, BBC News 2005, Jenkins 2005, Kerber 2006, Leyden 2004, Vance 2006a, 2006b). Table 1 outlines the percentages of disks recovered containing commercial data. Table 2 outlines the breakdown across the various areas of the study comparing the results from the disks obtained in the different regions. In each of the regions, disks containing corporate data were found within the sample sets.

	UK	Australia	Germany	North America
Total No. of Disks Analyzed	133	79	42	46
Commercial data present	19 (41%*)	22 (46%*)	2 (29%*)	8 (35%*)
Individual data present	34 (74%*)	7 (15%*)	5 (71%*)	15 (48%*)

** Percentage of readable disks that had not been wiped*

Table 2: Disks containing corporate data across the different regions (based on Jones 2007)

The results of the hand held devices study (Jones et al, 2008b) summarized in Table 3 indicate that corporate data is being lost via this route, although this is currently limited due to the number of older 1G devices in circulation. The study showed that as 2G and 3G devices have come into common use within business and our personal lives, people have started to store significant volumes of data on these devices. While there are a number of factors that have contributed to this trend that require further study, these devices are now being offered for sale on the second hand market and many of them still contain significant quantities of information.

Category	**Percentage**
Broken or not accessible due to physical problem (dead battery – no suitable lead	*82 (51 %)*
Blank – all data removed	*37 (23 %)*
Encrypted	*7 (4%)*
SIM Present	*3 (2%)*
Identifiable to user	*14 (9%)*
Identifiable to organization	*19 (12%)*
Personal Data present	*14 (9%)*
Sensitive Corporate data present	*6 (4%)*
Number of Hand Held Devices Analyzed	*161*

Table 3: Breakdown of Results from hand Held Devices (from Jones 2008b)

This paper addresses the possibility that someone with a motive to make a profit from questionable practices, may deliberately target a commercial organization by seeking to obtain second hand media or devices. In order to achieve this goal, the individual intending to conduct the industrial espionage has to first obtain the disk or device and then also be competent to extract the data from it.

INDUSTRIAL ESPIONAGE

Industrial espionage in the high-tech environment may be either focused on gaining information relating to a particular organisation or may be a more general collection of useful corporate information that can be sold to interested groups or individuals. If the target is a specific organisation, then there will be a requirement for planning and reconnaissance to determine the best attack vector to gain the required information for the lowest investment in time and effort. If the industrial espionage is of an information brokering type, where information is collected and then sold on to interested parties, then it will be unfocussed with the primary aim of gathering as much information that may have value. The latter approach will have the greatest chance of success and be the easiest to undertake. During the disk and mobile device studies, there have been a number of noteworthy discoveries of corporate data that would have been of significant value to competitor organisations. These have included business plans that were less than three months old, profit and loss sheets for each of the elements of a large multinational organisation, detailed communications between a company and its customers dealing with product faults and remedial actions and the detailed advertising strategies for a large multinational corporation with proposed expenditures in millions of dollars.

OBTAINING THE DISK OR DEVICE

The disk and hand held device studies were both aimed at providing a generalized picture of the data available on the second hand market. They did not target a specific organization or type of organization. From the experience gained during the studies, obtaining the disk or device from a particular organization is likely to be the most difficult part of the process. If an organization takes steps to effectively remove data prior to the disposal of equipment, then this successfully denies an attack via this route. If the data has not been effectively removed, there are a number of possible avenues that an attacker could explore:

- Purchasing second hand disks or devices from a disposal company or reseller. If the organisation that has disposed of the disks or devices has put in place suitable arrangements with a competent re-seller, the disks or devices should have been wiped and there would be no data available. The 2005 disk study results highlighted one reseller that was merely formatting the disks as a method of removing the data and this has since been found to be relatively common practice. As a result the majority of data waseasily recoverable, leaving any company using its services open to this form of attack.

- Targeting the company directly, buying up second hard drives, complete computer systems or hand held devices under the guise of charity or a reseller. This could be achieved most effectively if acting as a disk disposal service as this would allow direct access to a selection of drives or devices from the organisation potentially containing corporate data.

- The above method could also be applied to target users to obtain old systems by offering users a secure disposal service. The disk study results in recent years have highlighted a blurring of personal and corporate data and it is most common to find elements of both on disks and devices. This could either be as a result of an increase in the use of corporate systems for personal use or the use of home systems to carry out work activities as a result of home-working.

If the industrial espionage was targeted at a specific organisation, the adversary would probably have to carry out a significant level of research to determine the processes and procedures in use by the organisation for the disposal of obsolete equipment. The adversary would also probably need to be well funded as the success rate for obtaining disks or devices from a specific organization is likely to be fairly low. One finding of the current 2008 study is that the imaging and analysis of SCSI drives resulted in data from a number of commercial system being recovered and the selection of this type of drive may be a used as an attack vector by an adversary.

Even if the adversary has obtained a disk or device from the target organization, it may not contain the type of information that they are seeking. It should be noted that the studies indicated that most corporate disks included a wide range of information: this could be in the form of internal contact details or items such as network configuration. Some of this information might be publically available, whereas other elements could be more sensitive and may, in addition to the damage caused by the exposure of the information also even facilitate some form of network attack.

EXTRACTING DATA OF VALUE

Once a disk has been obtained, the adversary will be interested in extracting any useful and useable data. This can require varying degrees of effort depending on whether an attempt has been made to remove data. If an attempt has been made to remove the data, this may not be a significant impediment to the adversary as there is a surprising amount of software available on the internet that has been developed for forensic analysis and data recovery which they can use to assist them in acquiring the data. There are cracked copies of both of the major forensic tools FTK (AccessData 2008) and Encase (Guidance 2008) in circulation. These are both powerful tools with a range of search utilities which enhance an adversary's ability to examine a drive for useful data. There are also a number of open source tools which can be used to 'carve' (recover) deleted files from unallocated space. This can be hampered by some file systems and if encryption has been used to protect the files it can prevent data recovery. The sampled set of disks and devices indicated that the majority of corporations either do not appear to encrypt their data, or use NTFS encryption which is not sufficient to prevent data recovery.

The actual extraction of the data requires some degree of understanding of the working of a computer hard disk or handheld device and the tools required to recover data from this type of device. Although these tools often require some technical knowledge, there is extensive information readily available on the web (Wikipedia 2008), in text books (Carrier 2005, Casey 2004) and in academic papers (Fragkos 2006) to guide a user. There are in the UK around 17 undergraduate degree courses teaching forensics and presumably data recovery techniques (UCAS 2008). There are also a number of shorter, commercial training courses.

COUNTERMEASURES

This form of attack is likely to be ineffective on an organization which disposes of data in a secure manner, although even then they will be exposed to human failures and subverted staff. The risk of this form of attack being successful can be reduced by employing the following countermeasures:

- Ideally data disposal should be retained as an in house function, but where this function is outsourced there should be an audit system in place to allow the disks or devices to be randomly checked to ensure the quality of the data disposal process.

- Disk and device disposal services should be offered to employees for their home systems to ensure the company is protected against inadvertent data loss via an employees system due to an employee working on their home system. The incentive could be the protection offered to the employee's personal data.

- Encryption is an effective measure for preventing this form or recovery, subject to the limitationsrelating to strength of the encryption and the security of the key.

- Software to allow employees to access work systems from home should ensure a secure work space to prevent residual data being left on home systems.

- A company should understand the risk in terms of what may reside on a corporate disk by carrying out some form of internal review on residual data on a sample of corporate disks.

- Corporate systems should be purged of data if they are to be reused within the organization to prevent the build up (aggregation) of sensitive data on a drive or device.

- Technological solutions can be used to tamper-proof disks and devices or ensure data is wiped if theyare used in an unauthorized manner.

SUMMARY

This paper discusses the risks of an attacker successfully obtaining corporate data from an incorrectly disposed computer disk or hand held device. This paper describes some of the types of information which might be found and raises the question as the skill set and requirements that would be required by individuals to obtain useful data.

It is acknowledged that the risk of a successful attack via this method is limited, in that the adversary would have to be relatively well funded if there was a specific organizational target and that the hit rate for a specific organization depends on the type of disk or device obtained from the organization and the disposal practices in place.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the generous sponsorship of British Telecommunications (BT) who support the disk and hand held device studies each year. In addition to the authors of this paper, we would like to acknowledge the other researchers who participated in the 2007 and previous disk studies and our colleagues at Edith Cowen (Australia), Longwood (United States of America) and Glamorgan (United Kingdom) Universities.

REFERENCES

AccessData (last visited September 2008) www.accessdata.com

American Forces Press Service (2006), *Current Service Members Possibly Affected by VA Data Loss,* 6 June 2006.

Association of Chief Police Officers Good Practice Guide for Computer-Based Electronic Evidence. http://www.7safe.com/electronic_evidence/ACPO_guidelines_computer_evidence.pdf

BBC News (2005), *Data dangers dog hard drive sales*, BBC, 12 September 2005.

Carrier B, (2005) *Forensic File System Analysis,* Addison Wesley.

Casey, E., (2004) *Digital Evidence and Computer Crime; Forensic Science, Computers and the Internet*, Academic Press, Second Edition.

Guidance Software (last visited September 2008) www.guidance.com

Fragkos G. et al (2006) *An empirical methodology derived from the analysis of information remaining on second hand hard disks* in Blyth A., and Sutherland I., *WFDIA Proceedings of the first Workshop in Digital Forensics and Incident Analysis*

Garfinkel S.L, Shelat A, (2003), *Remembrance of Data Passed: A Study of Disk Sanitization Practices.* IEEE Security & Privacy, Vol. 1, No. 1, 2003.

Gutmann, P. (1996), *Secure Deletion of Data from Magnetic and Solid-State Memory, Sixth USENIX Security Symposium Proceedings*, San Jose, California, July 22-25, 1996.
Jenkins, C. (2005), *Govt data sent to auction*. The Australian, 2nd August 2005.

Jones, A., Mee, V., Meyler, C., and Gooch, J, (2005), *Analysis of Data Recovered From Computer Disks released for sale by organisations*, Journal of Information Warfare, (2005) 4 (2), 45-53.

Jones A., Valli C., Sutherland I., Thomas P. (2006) *An Analysis of Information Remaining on Disks offered for sale on the second hand market.* Journal of Digital Security, Forensics & Law. Volume 1, Issue 3.

Jones A, Dardick G., Sutherland I, Valli C., (2008a) *The 2007 Analysis of Information Remaining on Disks offered for sale on the second hand market*. Journal of Digital Security, Forensics & Law. Volume 3, Issue 1.

Jones A., Valli C., Sutherland I, (2008b) *Analysis of Information remaining on Hand Held Devices offered for sale on the second hand market.* Journal of Digital Security, Forensics & Law. - *In Press*

Kerber R (2006), *Firm will settle with state over data loss: Missing laptop had information on thousands*, Boston Globe, 12 December 2006.

Leyden, J. (2004), *Oops! Firm accidentally eBays customer database*, The Register, 7 June 2004.

Price Waterhouse Cooper (2006), *DTI Information security breaches survey* 2006, http://www.dti.gov.uk/industries/information_security Sept 2006.

Sutherland I, and Mee V. (2006) *Data Disposal: How educated are your Schools?,* 6th European Conference on Information Warfare and Security, June 2006.

UCAS – The Universities Central Administration System (last visited September 2008). www.ucas.com

Valli, C. (2004), Throwing out the Enterprise with the Hard Disk, In 2nd Australian Computer, Information and Network Forensics Conference, We-BCentre.COM, Fremantle Western Australia.

Valli C. & Woodward A., (2007) *Oops they did it again: The 2007 Australian study of remnant data contained on 2nd hand hard disk* Presented at the 5th Australian Digital Forensics Conference, Edith Cowan University Australia.

Vance A (2006a), *Ernst & Young fails to disclose high-profile data loss: Sun CEO's social security number exposed*, The Register, 25 February 2006.

Vance A (2006b), *Wells Fargo fesses up to data loss: Lightning strikes twice for HP man*, The Register, 12 May 2006**.**

Wikipedia - pages on forensics and data recovery (last visited September 2008) http://en.wikipedia.org/wiki/Computer_forensics, http://en.wikipedia.org/wiki/Data_recovery

COPYRIGHT

Virtual Environments Support Insider Security Violations

I. Swanson and P.A.H. Williams

SECAU Security Research Centre
Edith Cowan University

Abstract

This paper describes an investigation into how an employee using a virtual environment can circumvent any or all of the security, policies and procedures within an organization. The paper discusses the fundamental issues that organizations must address to be able to detect such an attack. Attacks of this nature may be malicious with intent to cause disruption by flooding the network or disabling specific equipment, or non-malicious by quietly gathering critical information such as user names and passwords or a colleague's internet banking details. Identification of potential residual evidence following an attack is presented. Such evidence may be used to speculate or verify an attack incident occurrence. Additionally, the forensic extraction of any such evidence is discussed. Finally, the paper raises the possibility of a virtual machine being used as an anti-forensic tool to destroy incriminating evidence in such circumstances.

Keywords: Anti-Forensics, computer forensics, virtual machine, security, antivirus

INTRODUCTION

The security and protection of an organizations data and infrastructure is of utmost importance to all companies regardless of size. Tens of thousands and in some cases millions of dollars are being spent each year to try to keep networks and computers free of infection and secure from outside intruders. There are numerous threats from individuals trying to compromise an organization's security from beyond their exterior firewalls, however the "attacker" working from within the organization should also be considered. Is there enough security inside to protect the organization? This paper will look at a real life situation to try and find the answers to these questions within the bounds of legality. In general, the desktop standard operating environment (SOE) is built to include basic security tools such as 'antivirus', 'sms client', 'vnc' and a locked down 'desktop firewall'. Firewalls are placed between different environments, Secure Lan, DMZ's and so on. Intrusion detection systems on servers, network monitoring, security on switches and routers, port lock downs, web content filtering and mail checking should provide adequate protection for any organization. The main problem with security measures such as these, are that they can frustrate employees if they cannot get to their favourite web sites, or their personal mail gets bounced. This in turn may turn an employee into a security risk as they search for loop hole in the system. The other scenario however and the most likely one, is that the organization already has an attacker in their environment looking to circumvent security by using a virtual machine to get round the policies and security measurers that are in place. "These intra-host threats can elude any existing security protection schemes" (Ruykhaver, 2002).

Assumptions

Several assumptions are made about the type of access to the computer and the skill level the employee may have. It is assumed that the employee has or can obtain administrator access to the machine that will be used for any attacks. It is also thought that the employee would have sufficient technical knowledge to install software and possesses the skill to use it. In this case the potential attack under investigation is in deference to the employee being able to bring in a laptop into the business environment and connecting it directly into the network, as this may be noticed.

CIRCUMVENTION

In recent years, the proliferation of virtual environments has been accelerated due to products like VMware Server becoming available to download at no cost (VMware Inc, 2008). Other advancements, such as VMplayer and pre-built 'virtual appliances', have made it straightforward to install and run a separate operating aystem with a complete set of exploit, monitoring, sniffing and cracking tools within the normal desktop environment.

Virtual machine

Virtualization provides a layer of abstraction between computing, storage and networking hardware of the host machine and the applications that run on it. This provides the virtual machine with an operating environment identical to the host on which it sits without knowing anything about the host. The virtual machine is completely isolated from the host although it shares the same hardware. The host machine only knows that an application is running and it is similarly unaware of another operating system. Figure 1 shows a before and after virtualization overview (VMware Inc, 2008).

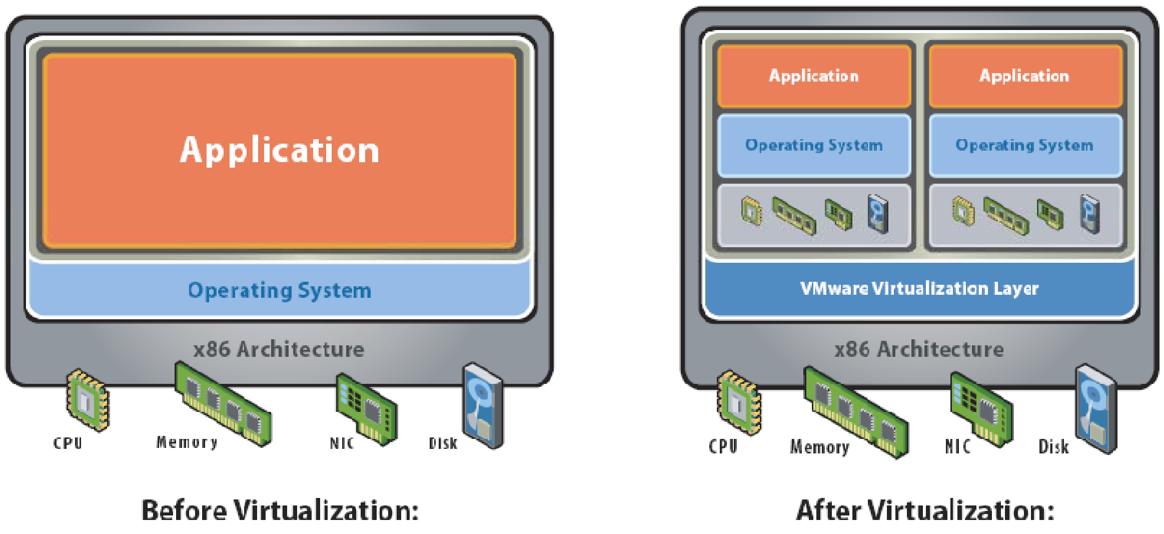


Figure 1. A virtualisation overview (VMware Inc, 2008.)

Using VMware Player and a pre-made virtual machine such as Backtrack3 (Remote Exploit, 2008), a new operating system can be booted up. The network interface card is set to bridge network address translation giving assess to explore the network with tools that are now invisible to the antivirus engine on the desktop. In normal use, tools like 'John the Ripper' (Openwall Project, n.d.) or 'dniff'(Song, 1999) would be deleted or at least quarantined by the antivirus software. Further, if the system is monitored it would raise an alarm that suspect tools were in use. Using a pre-made Linux distribution allows the use of the environment without additional complication. This creates a hole in security for the organization and any tool can be downloaded to the virtual machine undetected unless blocked by a web filter. Subsequently, this block can also be circumvented by using HTTP tunnelling or using an open SSH. In the example described below SSH is blocked so no SSH port redirects can be done. This using PingFu Iris (PingFu, 2008) a secure tunnel to the Internet is created.

Attacks

The infrastructure within the organization is now at risk from this virtual machine not only from the attacker but from the covert attacker that may now have access on the back of other software downloaded from the Internet. The host machine itself is safe from viruses, trojans and other malware, and the virtual machine is encapsulated and cannot infect the host directly, but the virtual machine can infect other devices on the network.

Backtrack3 is a comprehensive suite of the tools necessary to exploit any device not secured on the network. Attacks can be launched at specific machines or devices. Also, 'nmap' can be deployed to discover computers on the network and to see if it is detected by the security teams (Lyon, 2008).

One aim of such attacks is to listen for conversations between users and other devices or websites that use clear text user names and passwords. One method of doing this is to use Ettercap in promiscuous mode and ARP poisoning (Ornaghi & Valleri, 2005). Ettercap is a very powerful tool that is able to sniff switched networks for passwords and even redirect web sessions to different web sites. Such abilities could be used to fake internet banking sites for instance. The ARP cache poisoning attack works by poisoning the ARP cache of the target hosts (Spangler, 2003). Figure 2 below shows how ARP spoofing works.

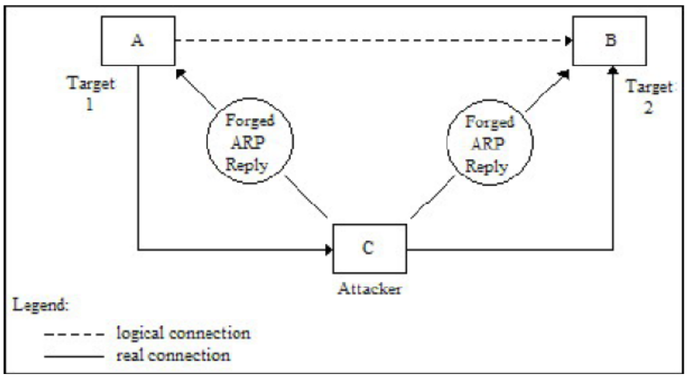


Figure 2. ARP spoofing (Spangler,2003)

Backtrack 3 has a comprehensive list of tools that can be used to exploit a computer or network. Included in the suite of tools is Metasploit Framework which has many types of exploits and pay loads that can be implemented with relative ease. Metasploit Framework is described as "a development platform for creating security tools and exploits" (Metasploit, 2008).

Trialling

Using the methods described above, a trial of the potential attacks was undertaken. By using a combination of tools built into the virtual machine's operating system it was discovered that user names and internet passwords, telnet user names, and password for routers and switches when 'SSH' was not being used, could be recovered. Potentially, malicious attacks could be run from this platform without immediate detection. HTTP sessions could be high-jacked, attacks on routers and switches, and compromises to servers holding sensitive data that the employee does not have legitimate access to, could be effected. As this was a live test and to ensure that no undue security risk was put in place the testing stopped. Whilst not definitive, it was enough to prove that an organization can be compromised by an installation of a virtual environment.

PREVENTION AND DETECTION

Pre Attack Detection and Prevention

Consideration needs to be given as to whether or not such an installation and the use of tools can be detected and therefore prevented or at least intercepted. Several scenarios for detection are possible:

- If a virtual machine were in bridged mode the DHCP server would supply a new IP address. Therefore an administrator checking the logs would see a machine name with no domain. Whilst not significant, this should warrant further investigation, as it would show that it had the same MAC address as another machine currently on the network;
- Similarly, this scenario would also hold true if the network switches were monitored. The switch port would have one MAC address with two IP addresses bound to it. As Microsoft's SMS is installed as part of the SOE, regular scans can be made of all machines which would produce a list of all software on the desktop. This could then be followed up with a visit to the offending computer;
- Network monitoring can detect the presence of an ARP poisoner or spoofer if the monitors are configured to search for such software. Also, most modern switches can be configured to avoid using ARP or to statically map IP addresses to MAC's. In addition, there are many ways to secure the switch which outside the scope of this paper.
- ArpWatch is a utility that keeps a list of IP address to MAC address mappings and will contact the administrator if any of these mappings are altered. Any malicious attacks should be picked up by the internal IDS. In this scenario, IDS is only installed on critical servers so other data servers are vulnerable.
- Service packs should be up-to-date and a policy of renewal should be in place;

- From a technical perspective, machines could be locked down so that no additional software can be loaded, although this may cause a problem of usability and raises other security issues. Also the netlogon script could contain a script that would track and report an installation on the machine; and
- Usage monitoring could identify a problem if a machine seems to be producing large amount of data.

Discovery and Identification

One detection problem is that if the attacker is not creating significant 'noise' on the network or creating any suspicion, then any activity may largely go unnoticed. Thus it is likely that an attacker may continue to gather information until discovered by accident. There are several possible reasons for lack of detection: either the security team do not have the time to investigate minor anomalies in infrastructure, or as in the scenario discussed here, the internal infrastructure is large and the management diverse, such that problems remain unnoticed for some time.

Therefore, it is worth considering that if the attacks were noticed, or that ARP spoofing was being monitored, the attacker would be discovered. Essentially this would all depend on the quality of the documentation and how well patch panels are labelled. The attacker could be traced back to the switch that originated the attacks. Consequently, the port being used could be found and traced back to the panel, and eventually back to the RJ45 plug that is attached to the wall. Obviously, there are methods to circumvent this such as running a virtual machine from a machine which belongs to an absent employee and using a virtual network to access it. Alternatively, an attacker can spoof a MAC address of a machine that is switched off, it would slow detection, perhaps long enough to try and destroy any tracks or evidence on the host computer.

Investigation

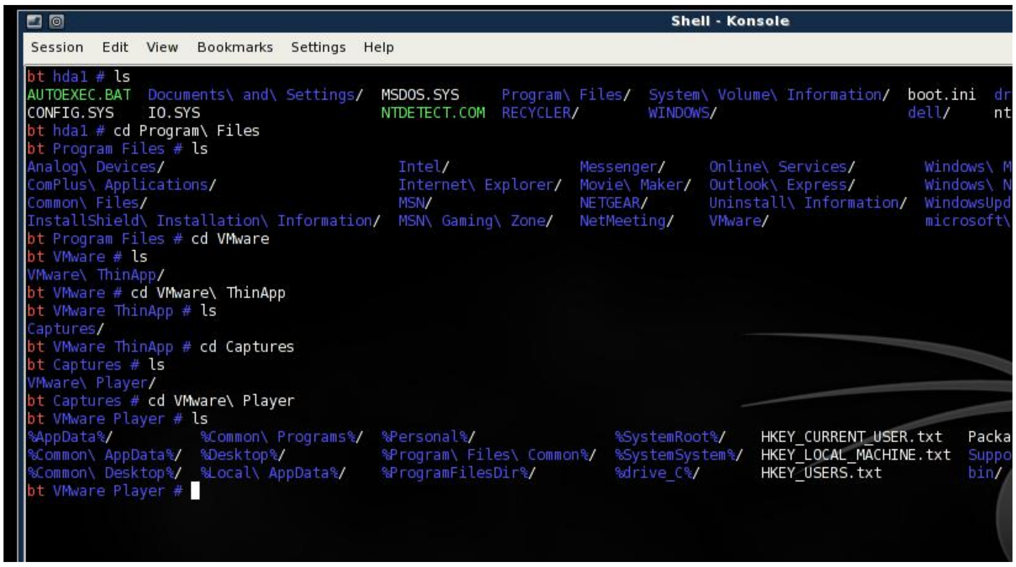
After discovery and eventual identification of the computer involved in the attacks, a problem then exists to find evidence that will prove that this machine and its user were the culprit. The computer would need to be seized for analysis.

FORENSIC EVIDENCE

Forensic evidence must be obtained in a sound manner that can be verified and must be significant enough to provide a 'no reasonable doubt' in the minds of jurors or in this case a manager's mind. In this scenario, through the trialling of the potential attack method, evidence was able to be collected as described in the subsequent sections.

Evidence

A forensically sound image was taken of the suspected attackers computer disk using a live Linux distribution. This was then mounted and searched for any suspect tools. None were found. From the image there were traces of VMware Player on the disk i.e. the empty folder was still there. Figures 3 and 4 show what was found to be on the disk.

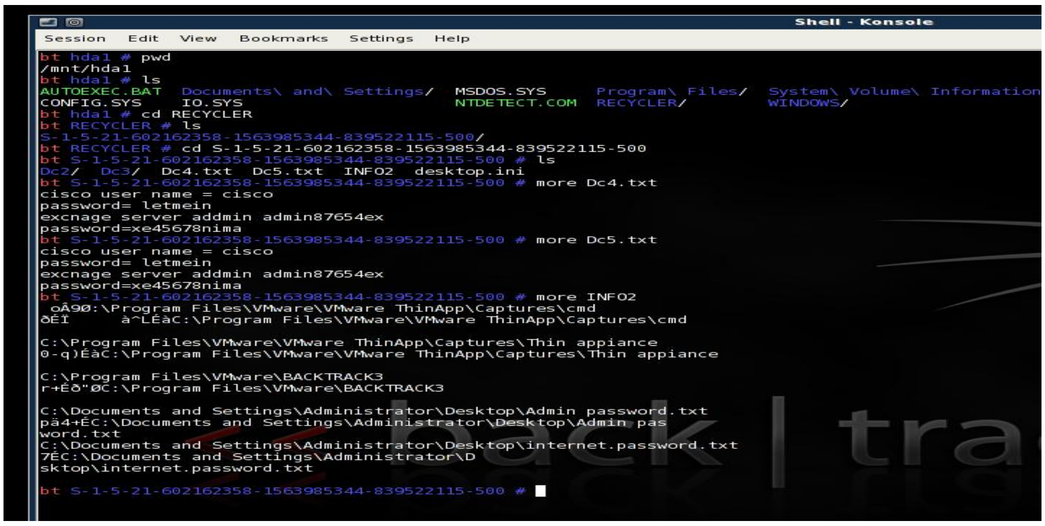


```
bt hda1 # ls
AUTOEXEC.BAT  Documents\ and\ Settings/  MSDOS.SYS     Program\ Files/  System\ Volume\ Information/  boot.ini  dr
CONFIG.SYS    IO.SYS                     NTDETECT.COM  RECYCLER/        WINDOWS/                      dell/     nt
bt hda1 # cd Program\ Files
bt Program Files # ls
Analog\ Devices/                         Intel/              Messenger/     Online\ Services/       Windows\ M
ComPlus\ Applications/                   Internet\ Explorer/ Movie\ Maker/  Outlook\ Express/       Windows\ N
Common\ Files/                           MSN/                NETGEAR/       Uninstall\ Information/ WindowsUpd
InstallShield\ Installation\ Information/ MSN\ Gaming\ Zone/ NetMeeting/    VMware/                 microsoft\
bt Program Files # cd VMware
bt VMware # ls
VMware\ ThinApp/
bt VMware # cd VMware\ ThinApp
bt VMware ThinApp # ls
Captures/
bt VMware ThinApp # cd Captures
bt Captures # ls
VMware\ Player/
bt Captures # cd VMware\ Player
bt VMware Player # ls
%AppData%/          %Common\ Programs%/  %Personal%/             %SystemRoot%/    HKEY_CURRENT_USER.txt   Packa
%Common\ AppData%/  %Desktop%/           %Program\ Files\ Common%/ %SystemSystem%/ HKEY_LOCAL_MACHINE.txt Suppo
%Common\ Desktop%/  %Local\ AppData%/    %ProgramFilesDir%/      %drive_C%/       HKEY_USERS.txt          bin/
bt VMware Player # █
```

Figure 3. Residual evidence of VMware player installation.

Figure 3 shows that although VMplayer had been uninstalled, it was previously installed or at least some thing with the same name is present. The other directory 'ThinApp' contains a capture directory which turns out to be a new version of the Player called ThinApp.

The next step was the investigation of the rest of the disk and in particular looking at the Recycle bin to see if anything has been left there. Figure 4 shows its content.



```
bt hda1 # pwd
/mnt/hda1
bt hda1 # ls
AUTOEXEC.BAT  Documents\ and\ Settings/  MSDOS.SYS     Program\ Files/  System\ Volume\ Information
CONFIG.SYS    IO.SYS                     NTDETECT.COM  RECYCLER/        WINDOWS/
bt hda1 # cd RECYCLER
bt RECYCLER # ls
S-1-5-21-602162358-1563985344-839522115-500/
bt RECYCLER # cd S-1-5-21-602162358-1563985344-839522115-500
bt S-1-5-21-602162358-1563985344-839522115-500 # ls
Dc2/  Dc3/  Dc4.txt  Dc5.txt  INFO2  desktop.ini
bt S-1-5-21-602162358-1563985344-839522115-500 # more Dc4.txt
cisco user name = cisco
password= letmein
excnage server addmin admin87654ex
password=xe45678nima
bt S-1-5-21-602162358-1563985344-839522115-500 # more Dc5.txt
cisco user name = cisco
password= letmein
excnage server addmin admin87654ex
password=xe45678nima
bt S-1-5-21-602162358-1563985344-839522115-500 # more INFO2
 oÂ9Ø:\Program Files\VMware\VMware ThinApp\Captures\cmd
ðÉÏ      à^LÉàC:\Program Files\VMware\VMware ThinApp\Captures\cmd

C:\Program Files\VMware\VMware ThinApp\Captures\Thin appiance
0-q)ÉàC:\Program Files\VMware\VMware ThinApp\Captures\Thin appiance

C:\Program Files\VMware\BACKTRACK3
r+Éð"ØC:\Program Files\VMware\BACKTRACK3

C:\Documents and Settings\Administrator\Desktop\Admin password.txt
pä4+ÉC:\Documents and Settings\Administrator\Desktop\Admin pas
word.txt
C:\Documents and Settings\Administrator\Desktop\internet.password.txt
7ÉC:\Documents and Settings\Administrator\D
sktop\internet.password.txt

bt S-1-5-21-602162358-1563985344-839522115-500 # █
```

Figure 4. Recycle bin content.

Figure 4 shows there are several files left in the Recycle bin. Using the command 'more' it is possible to read the text files contained in it. The file INFO2 contains what appears to be a deleted copy of Backtrack. Backtrack 3 was extractable from the Recycle bin and contained the files necessary to run up a virtual machine. This evidence provides the ability to investigate the virtual disk.

This scenario, although done on a test bed, outlines some of the possibilities. In the example given the size of the virtual machine is only 3.5GB. If the image were much larger there may be problems of how to retrieve it. In a virtual machine there are several files that would need examination. The following file types are those of interest however there are several more depending on the type of virtual machine and the software that was used to create it.

- .vmsn - Virtual Memory Snapshot file, file that containing the memory data at the time of the snap shot.
- .vmx - Configuration file, could also show if external media has been used

- .vmdk - Raw Disk data of the appropriate disk format.
- .log – logs of how it booted

Several tools were tested to examine the virtual machine. As per sound forensics procedures, a copy of the image was taken, a hash value calculated, and the remainder of the investigation was done on the image copy (dd image).

The VMDK files attributes were set to read only and these were used to attempt to start the virtual machine. This failed with a "not enough permissions" error. Then, forensic image mounting software, Mount Image Pro (Get Data, 2008) was used. This mounted both the dd image and the VMDK but provided little in the way of information except that it identified that it was a Linux file system. Then, Forensic Toolkit -FTK Imager from AccessData Corp was used which read the files and offered the best drill down into the file system allowing recovery of deleted files and stored file, together with the date stamps on documents etc. FTK however would not load the VMDK files as an image but would load it as a single file. This did not yield much evidence. The extract below gives some idea of when the Metasploit framework was last initialized.

> Query: "framework" <ASCII/Unicode, Case Insensitive> -- 6794 Hits in 125 Files, 26 Hits -- [Other Linux 2.6.x kernel-000001.vmdk_039] C:\BACKTRACK3\Other Linux 2.6.x kernel-000001.vmdk_039, Offset 170F055 (24178773) -- Security <<Framework>> v1.0.0 initialized. Aug 26 06:48:38 (none) kernel: CPU: L1 I cache: 32K.

This sequence required more research. The failure of this software may have been due to the restraints of the demonstration version of the software. Also, the files contained in the slack space of the image were visible. VMware also has a utility to mount virtual machines without the use of player or server software. With Vmware-Mount installed on a Windows machine it was possible to mount the image. Unfortunately, the software was not able to read the image as it did not recognize the images file system. Subsequently, Window's wanted to format it. When the image was installed and mounted on a Linux environment, the file system was visible and the files could be extracted.

With the correct tools it is possible to investigate the virtual machine by way of the VMDK files. However, it was not possible to ascertain when any of the tools within the environment had been used. The discovery of a date and time stamp would have to be performed on the virtual machine files themselves outside of the virtual environment, and this would provide the time and date it was last used. However, depending on the detection methods and logs acquired, it may be difficult to prove anything happened except on the day and time recorded on the file. However, this data may also have been tampered with before deletion.

ANTI-FORENSIC

Anti-Forensics is a term which has different connotations to different people or groups. It has been described as "attempts to negatively affect the existence, amount, and/or quality of evidence from a crime scene, or make the examination of evidence difficult or impossible to conduct" (Rogers, 2006). Further it has been described as "application of the scientific method to digital media in order to invalidate factual information for judicial review" (Brown & Lui, 2006). In consideration of the scenario posed in this paper, it is useful to consider Rogers' definition - attempting in the process to discover if a virtual machine and the tools contained inside it, can be used to hinder or cover up any evidence of what has been done or what tools have been used.

According to Rogers (2006) anti-forensic tools come in four varieties; data hiding, artefact wiping, trail obfuscation and forensic tool direct attacks. These categories are fairly self explanatory, data hiding concerns it self with hiding data i.e. stenography, hiding data in slack space etc; artefact wiping would be deleting data either by normal methods or by using propriety tools to wipe the data. Trail obfuscation and attacks against forensic tools is the main concern within the context of this paper. For instance, file dates being changed, log files being altered and the fact that using the *.vmdk it was not possible to readily access or recover data without extra tools. This can itself be considered an attack of forensic tools or at the least trail obfuscation. There are some tools that could be used from within the virtual environment to hide data or shred data so that it could not be recovered and therefore be the ultimate anti-forensic tool. As Kessler (2007) noted, "cryptography is the ultimate anti-forensic tool, and has and will continue to make digital investigations difficult or impossible". Any type of encryption, PGP, SSH SSL, whether of a disk, a file or connection will hamper or kill off an investigation. Tools like Metasploit's Sam Juicer or Timestomp can only be used from the Windows host system and could be used on the virtual disk. However, this does not make the virtual machine an anti-forensic tool on its own, it would still require input from the host system.

CONCLUSION

From this investigation it can be concluded that a virtual environment of an operating system with correct tools can indeed circumvent the security of the organization. What occurs after the virtual machine is loaded depends largely on the types of security and policing that is in place. Organizations are certainly at risk from an attack from inside, not only attacks but infections of other machines through installation of rogue software on the virtual machine. Forensic extraction of useful information needs more research. In this case it was possible to see detect that a virtual machine had been used to circumvent the security and some useful files were collected to prove what type of information the attacker was gathering. Dates and solid evidence of what exact data had been viewed was more difficult to collect, however the amount of data that was collected might be sufficient enough for an employee to be dismissed on the grounds of misconduct. The evidence collected however may not have been of a legal quality or quantity.

Further, it should be considered that virtual machine could be used as an anti-forensic tool. Particularly, this would have application in slowing the forensic process down, although the virtual machine can certainly contain tools that could stall an investigation and can get round most of the security of the organization. Further, it may prove impossible for an investigator to determine which tools were used, however this would still only slow an investigation. The fact that something has to be installed unnoticed, or unauthorised by the organisation, is what should be of concern for an organisation. Thus, organisations should be aware of this and should take measures to find or stop any unauthorised installs. This subsequently leads to the emerging problem of ThinApps or portable Apps that can be brought in on small portable devices, such as USB sticks, and run straight from the device without needing to be installed and leaving no trace of ever having touched the machine.

REFERENCES

AccessData.(2008). FTK Imager. Retrieved September 9, 2008 from http://www.accessdata.com/forensictoolkit.html.

Brown F, & Liu, V.(2006). *Bleeding-Edge Anti-Forensics.* Presentation at InfoSec World 2006. Retrieved September 6, 2008 from http://www.stachliu.com/research_conferences.html/ .

Designer, S. (2008). *John the Ripper*. Retrieved September 1, 2008 from http://www.openwall.com/john/.

GetData Pty Ltd (2008). *Mount Image Pro*. Retrieved September 1, 2008 from http://www.mountimage.com/.

Kessler, G. C. (2007). *Anti-Forensics and the Digital Investigator*. Unpublished Paper. Champlain College Burlington, VT, USA. Edith Cowan University.

Lui, V. & Stach, P. (2006). Defeating Forensic Analysis. CEIC Lecture, 6-10. http://www.metasploit.com/data/antiforensics/CEIC2006-Defeating_Forensic_Analysis.pdf

Lyon, G. (2008). *Nmap*. Retrieved September 1, 2008 from http://nmap.org.

Metsploit LLC. (2008). *Metssploit Antiforensic homempage*. Retrieved September 1, 2008 from http://www.metasploit.com/research/projects/antiforensics/.

Metasploit. (2008). *Metasploit Framework 3 (Version 3).* Retrieved September 1, 2008 from http://www.metasploit.com/framework/.

Metasploit. (2005). *Remote Rogue Network DetectionTechniques and Implementations*. Retrieved September 4, 2008 from http://metasploit.com/research/projects/rogue_network/.

Openwall Project. (n.d.). *Jack the ripper password cracker.* Retrieved September 7, 2008 from http://www.openwall.com/john/.

Ornaghi, A. (2003). *Man in the middle attacks Demos. Blackhat* [Online Document]. Retrieved September 6, 2008 from http://www.blackhat.com/presentations/bh-usa-03/bh-usa-03-ornaghi-valleri.pdf.

RemoteExploit. (2003). *BACKTRACK 3 (Version 3).* Retrieved September 1, 2008 from http://Remote-Exploit.org.

Rogers, M. (2006). *Anti-Forensics: The Coming Wave in Digital Forensics.* Retrieved September 7, 2008 from http://www.cerias.purdue.edu/symposium/2006/materials/pdfs/antiforensics.pdf.

Ruykhaver, J. (23 January 2008). *What are the security risks of virtualization*. Daily Wire. Retrieved September 1, 2008 from…

Whalen, S.. (2001,). *An Introduction to ARP Spoofing", Node99* [Online Document]. Retrieved September 5, 2008 from http://www.node99.org/projects/arpspoof/

Song, D. (1999). *Dniff*. Retrieved September 1, 2008 from http://www.monkey.org/~dugsong/dsniff/.

Spangler, R. (2003). *Packet Sniffing on Layer 2 Switched Local Area Networks*. Packetwatch Research. Retrieved September 5, 2008 from http://www.packetwatch.net/.

Valleri, M. (2005). Ettercap. Retrieved September 1, 2008 from http://ettercap.sourceforge.net/.

Vmware Inc. (2007). *Virtulization White Paper*. Journal. Retrieved September 1, 2008 from http://www.vmware.com/pdf/virtualization.pdf.

Vmware Inc. (2008). Retrieved September 1, 2008 from http://www.vmware.com.

COPYRIGHT

Malware, Viruses and Log Visualisation

Iain Swanson

SECAU Security Research Centre
Edith Cowan University

Abstract

This paper will look at the current state of visualization in relation to mainly malware collector logs, network logs and the possibility of visualizing their payloads. We will show that this type of visualization of activity on the network can help us in the forensic investigation of the traffic, which may contain unwanted pieces of cod, and may identify any patterns within the traffic or payloads that might help us determine the nature of the traffic visually. We will further speculate on a framework that could be built which would be able to finger print any type of malware, based on the theory that the basic structure of Malware code does not change, it may mutate but the internal structure stays the same. By passing it through either a current log Visualisation algorithm or a purpose built piece of visual inspection software which would output a 3D visual representation of the malware to screen or be further processed by a multipoint mapping utility similar to a finger print mapper, which would determine the base structure of the malware and categorise it. If we could finger print zero day virus by recognising visually, we may then able to detect and create an antidote to it much quicker and more efficiently than is currently being done by most antivirus vendors

Keywords

Malware, Vizualisation, Network Security,

INTRODUCTION

The amount of data collected from a network, be it logs from the router, the proxy, the web server or from a malware collector such as mwcollect, ("Mwcollect," 2008) nepenthes,(Nepenthesdev, 2008) or honeytrap ("Honeytrap," 2008) the quantity of data and size of files can be daunting. If the data has come form a large corporation the log files could run into tens of gigabytes of data, this data has to be painstakingly searched through by someone looking of anomalies or in the case of honeypots looking where the traffic came from, identify the payload and then determine if it is malicious or not. According to (Frie & Rennhard, 2008) humans are very good at picking up and detecting patterns and analyzing images rather just text. Therefore the ability to see log files as a visual representation on the data contained within in it would greatly speed up the time required to analyze log files. The fact that humans can interpret complex visual images much faster than the text contained in the logs should bring visualization to the forefront of network forensics taking much of the tedious and painful trolling through data away as the examiner should be able to pinpoint anomalies and suspicious behaviors just by looking and the image that the data makes. Taking this another step forward the possibility of looking for and visualizing a virus or malware code in the same way would be quite possible, but what does it look like? Considering that it is only code it does not look like anything unless you are in The Matrix (Wachowski & Wachowski, 1999).

Malicious malware can be abstracted and rendered into another form as with Wechsler & Wrights description of their Digital Wilit software “the binary code of the virus is rendered as a sound wave; sound then becomes the input signal for a computer screen”(Wechsler & Wright, 2000). This solution is complex but gives the reader the idea of what can be done. Very little has change since 2000 in the representation of malware, but several applications have been developed to visualize log files which can produce very accurate representations of the details contained the log files and also incorporate the type of virus or malware which was included in the traffic. This paper it will look at some of the visualization programs and the different outputs that they produce and speculate on how they could be used and/or bolted together to produce a live on screen image of what is passing over the network to our computers before an antivirus product even knows they are there or at least has decided what it is. If a better way of identifying a piece of code was quicker than the current signature base methods, then zero-day virus could be caught and dealt with almost instantly. Although the proposal is of a signature nature it is creating more of a finger print which defines the malware by its internal structure and hopefully separates itself from signature bases detectors by it speed and accuracy of detection through the image it produces.

Assumptions

To visualise log files we need a large amount of computing power at our disposal for large files,and for a representation of malware we would need to off load some graphic processing to other machines, or better graphic cards if we want to see anything that resembles a pattern or an image in reasonable amount of time. Also there is the assumption that malware code has a particular internal structure, depending what type of malware it is.

VISUALIZATION

There are several types of visualization tools that can be used today to produce a visual representation of log files, although the appliance they have come from and their file format is a hindrance at this point in time, that being said many files can be analysed in this way. A framework by (Frie & Rennhard, 2008) is tiring to correct the file format problem amongst other things, by building HMAT (Histogram Matrix) which will be able to visualize any log format presented to it. The main problem at the moment is not that visualization tools do not work, they do work, but they are hard to use and consist of several steps using other numerous applications and consume a lot time and even more computing power.

Visualization comes into its own when we are talking about large files, we can analyse a 200mb file in a hex editor with relative ease but it still does not give us a good picture of the structure of the log its contents or a relationship between data in the logs, for example groups of IP addresses are not easily seen in a hex editor. There are different variants of tools which read a log file and produce different types of graphic representations, some tools will produce a 3D graphic of the logs which can also be broken down into sub sections like scatter plots, hemisphere spatial views, Axis Views and Total 3D views, at the moment 2D views are the most prevalent for looking at log files, determining what is happening on the network and where, just by looking at the 2D topology of the log representation. Several tools were looked at during this research to understand what the output was and how it could be adapted to view live networks or malware code.

Current Visualization tools

Here is a short list of current visualization applications:

- AfterGlow (Christian & Raffy, 2007)
- TreeMap (HCIL, 2008)
- InteVis (Van Riel, 2007)
- Cytoscape (UCSD, 2008)
- GraphViz (Low, 2004)
- TNV (Goodall, 2007)
- NvisionIP (NCSA, 2004)
- Rumint (G. Conti, 2007)
- Nazar (Roth, 2008)
- Skyrails (Widjaja, 2007)
- Sequoia (Bruls, Geerlings, & Van Ham, 2002)

There are many more applications that can visualize data but for this paper, some of above applications are the ones that were looked at. Some of the above applications work together to form a representation and others are stand alone applications that can generate a graphic with out any interaction with other software, for example to produce an image from a data stream we need to format it into a *.csv file, the following steps would have to be undertaken:

```
“\tcpdump -vttttnneli ath0 | \ ./tcpdump2csv.pl "sip dip dport" | head -2000 | \../graph/afterglow.pl -c
color.properties -e 2 | neato -Tgif -o test.gif;”
```

This would generate a representation of the log generated by tcpdump, as can be seen here it is not the easiest or quickest way to get an image, also several different applications were involved, tcpdump, tcpdum2csv, afterglow and neato. However, it would still be remarkably quicker than searching through the log files manually to detect any anomalies. Some of the tools above, although not specifically designed to visualize network logs or malware identification they can be manipulated to do so.

Visualization at work

The following section will demonstrate some of the techniques and representations that are possible with the applications and discus what can be learnt form the out put of each application. (Blasco, 2005) takes a Nepenthes log file, turns the log into a CSV file and generates a visualization of the malware found in the log file. Finding the hash value of the malware he manages to produce this image:

```
# cat datos.csv | perl afterglow/src/perl/graph/afterglow.pl -c color.properties -e 6 -p 1 > img.dot
# cat img.dot | neato -Tgif -o test.gif
```

Image generated:

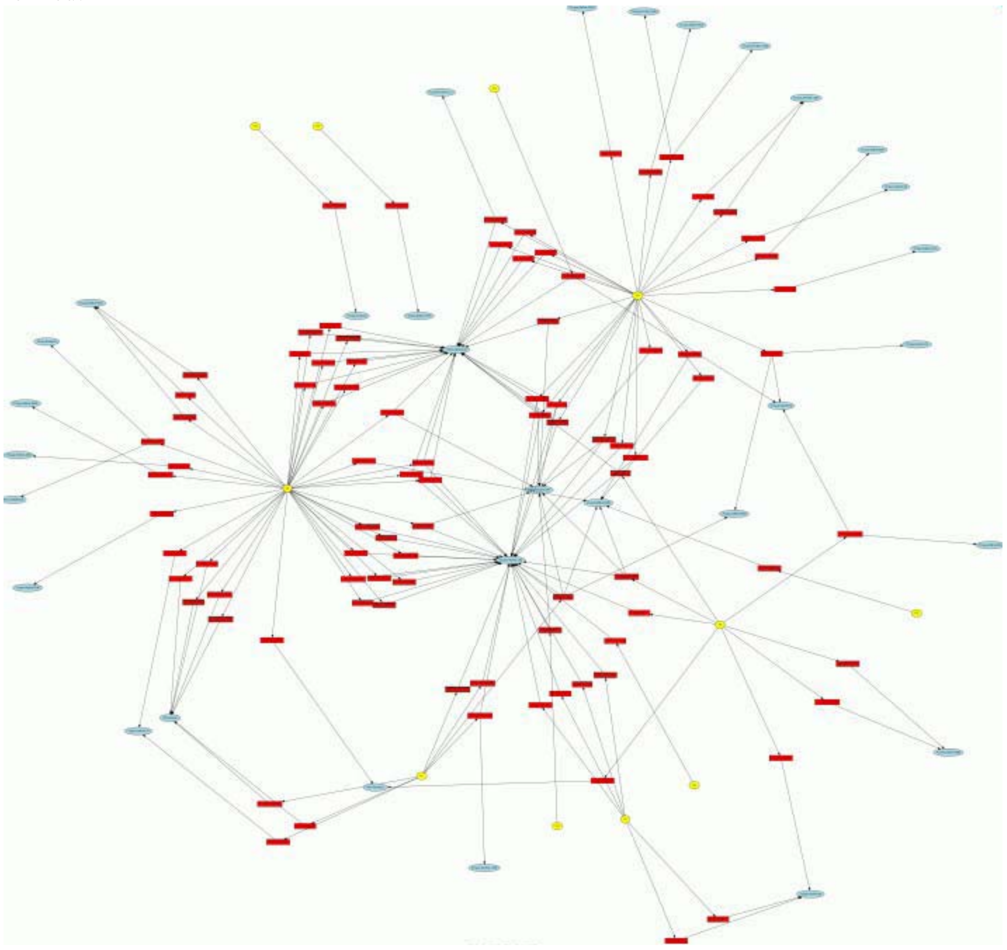
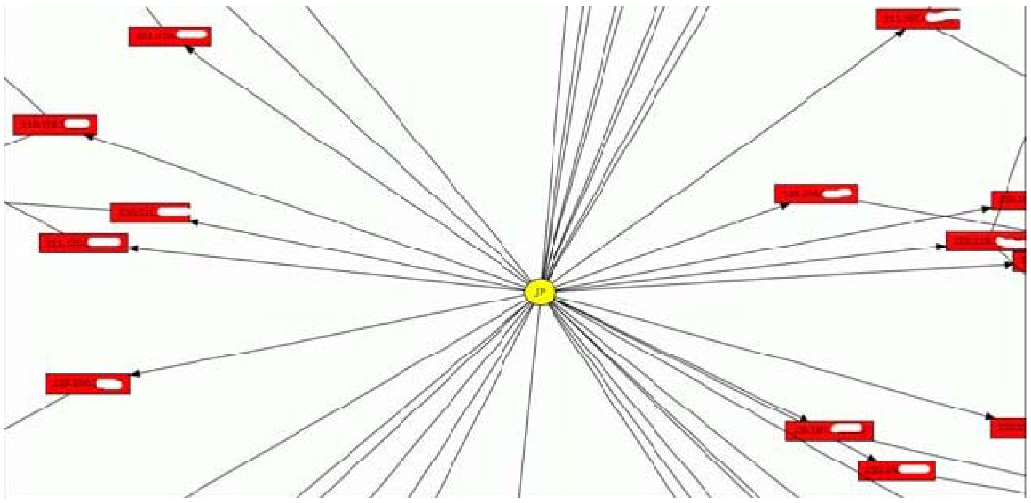


Figure 1 total image of malware spread (Blasco, 2005)



Figurev2 closer view of image(Blasco, 2005)

We can now see ip addresses appearing, giving more of an idea of where things are moving 'from' and where they are going 'to' generally countries but can be narrowed done to ISP ip ranges therefore identifying the culprit is possible.

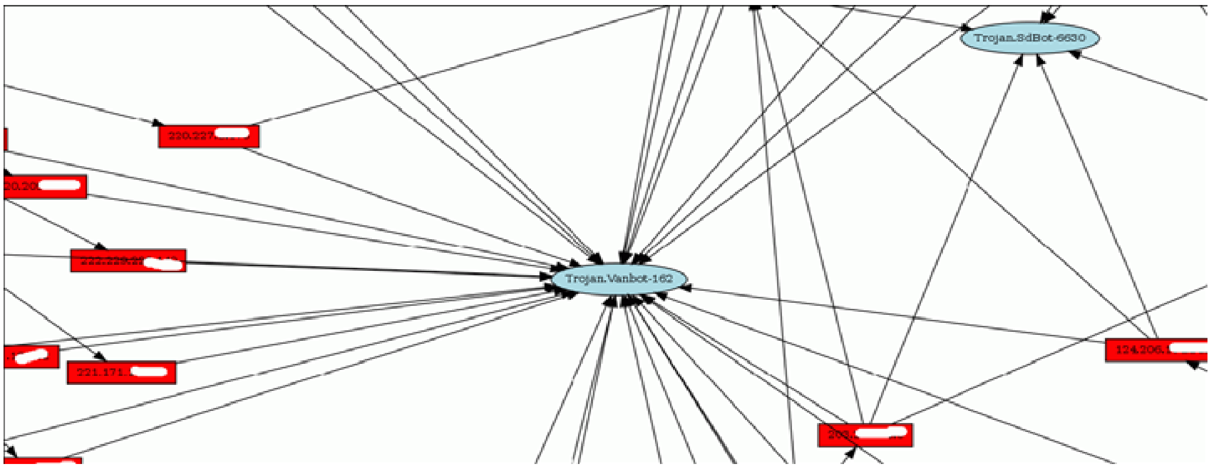
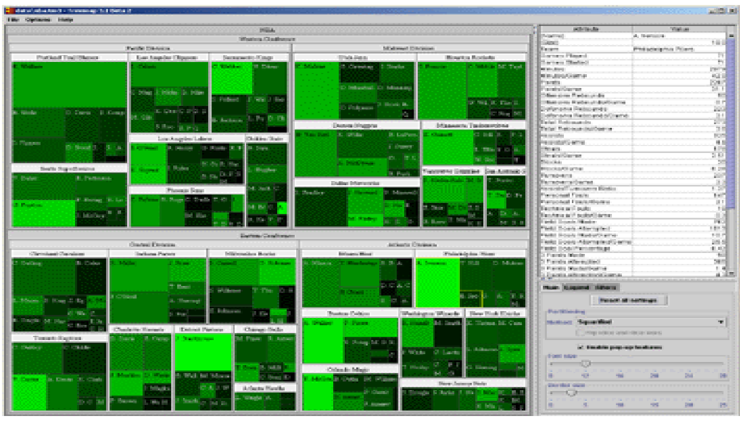


Figure 3 closer inspection (Blasco, 2005)

TreeMap (HCIL, 2008)

"Treemap is a space-constrained visualization of hierarchical structures"



SequoiaView(Bruls et al., 2002)

Similar to TreeMap but "Squarified treemaps". "The screen is subdivided such that rectangles approach squares as closely as possible"

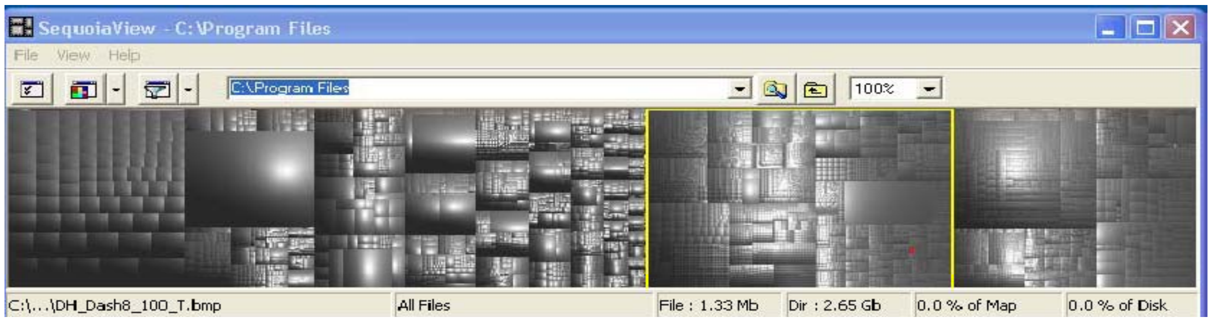


Figure 4 SequoiaView (Bruls et al., 2002)

InetVis(Van Riel, 2007)
InetVis is a 3-D scatter-plot visualization for network traffic. Taken for the "The Spinning Cube of Potential Doom". "The vast majority of coloured dots can be considered to be malicious traffic searching for vulnerable systems" (Paxson, 2003). Other network visualizations employ lines as a metaphor for connection, the 3-D scatter-plot of points proves to scale well with larger volumes of data.(Van Riel, 2007).

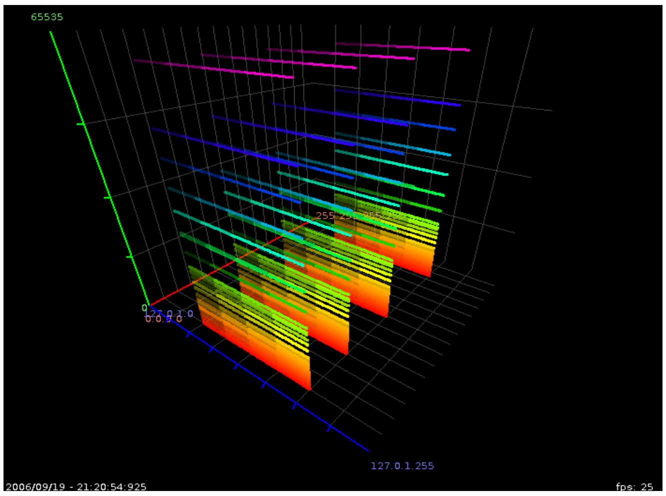


Figure 5 InetVis(Van Riel, 2007)

"The visualization below is from Tenable Network Security's, Security Center, which includes a 3D visualization tool that can derive network topology information from distributed Nessus vulnerability scanners. Each node in the center helix of the above graph is detected router"(Tenable Network Security, 2008)

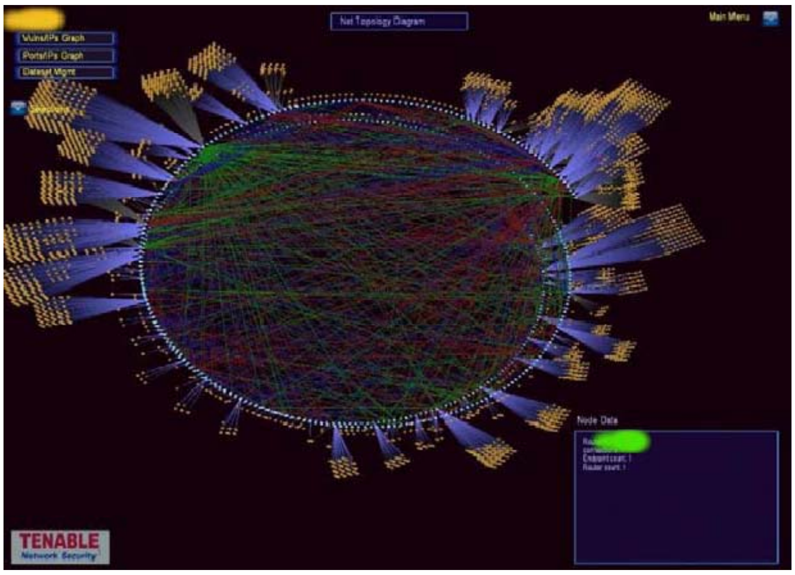


Figure 6 (Tenable Network Security, 2008)

Skyrails (Widjaja, 2007)
"Skyrails is a Social Network and Graph Visualization System with a built-in programming language" This can be programmed to input any log files and configured to render in different ways.

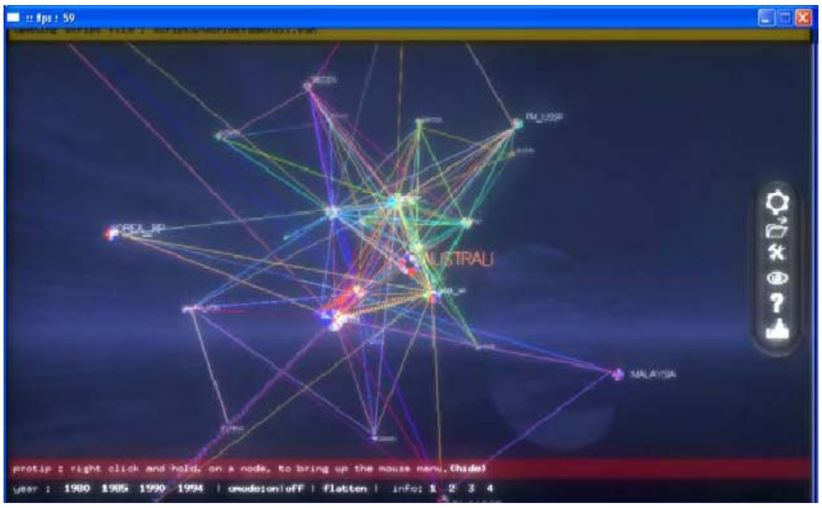

KOREA_IP
AUSTRALI
MALAYSIA
protip : right click and hold, on a node, to bring up the mouse menu.(hide)
year : 1980 1985 1990 1994 | omode:on|off | flatten | info: 1 2 3 4

Figure 7 Skyrails (Widjaja, 2007)

As can be seen for the images and software previously mentioned, the techniques, inputs an rendered outputs are vastly different from each other but all have the same aim. That aim is to see the coded world as a graphic representation. All the programs have a level of complexity that needs to be rectified if the tools are to become more useful. Raffael Marty and Jan P Monsch have brought to the arena a tool that encompass many of the current tool sets in one easy to use (relatively speaking) LiveCD. The DAVIX (Data Analysis and Visualization Linux), tools "provide an integrated out-of-the-box environment for data and visualization analysis"(Marty & Monsch, 2008). As noted by Dr Greg Conti "For InfoSec practitioners just starting out in visualization, the best place to start is by experimenting with existing tools. Unfortunately, finding and correctly installing these tools can be a tricky process. That is why I'm excited by the DAVIX project. It integrates a wide range of tools into one easy to use distribution."(Conti, 2007). The DAVIX collection contains about 25 different and complementing visualizing tools which are outside the scope of this paper to discuss all the tools in any great detail but some are mentioned in this paper for example, 'afterglow' suffice to say that having all visualizations tools in one environment makes the task much easier.

Conceptual Proposal

Background in Antivirus detection:

There are several ways a detector can detect malware or viruses these are scanning, integrity checking, interception, and heuristic detection. The pros and cons are listed below, taken form (Roberts, 2001)

- Scanning: "Scans all files and uses a signature to detect the malware, There are two major disadvantages to scanning-based techniques. Firstly if the software is using a signature string to detect the virus, all a virus writer would have to do is modify the signature string to develop a new virus. This is seen in polymorphic viruses. The other, and far greater disadvantage is the limitation that a scanner can only scan for something it has the signature of."

- Integrity Checking: "Uses Check summing, problem is with integrity checking is that not enough companies offer comprehensive integrity checking software."

- Heuristic Detection: "This is a generic method of virus detection. Anti-virus software makers develop a set of rules to distinguish viruses from non-viruses. Should a program or code segment follow these rules, then it is marked a virus and dealt with accordingly." The problems with this are "the technology today is not sufficient, and virus writers can easily write viruses that do not obey the rules, making the current set of virus detection rules obsolete."

- Interception: "Interception software detects virus-like behaviour? Interceptors are not very good at detecting anything else. Interceptors have all the drawbacks of heuristic systems – difficulty differentiating virus from non-virus, and easy to program around."

Based on a project by Alex Dragulescut of visualization of worms, viruses, trojans and spyware code. This paper proposes a theoretical solution to both network logs and malware verification in a timely manner.

Although Dragulescuts images are art, "each piece of disassembled code, API calls, memory addresses and subroutines are tracked and analyzed. Their frequency, density and grouping are mapped to the inputs of an algorithm that grows a virtual 3D entity. Therefore the patterns and rhythms found in the data drive the configuration of the artificial organism."(Dragulescut, 2008). Below are the theoretical structures of malware and viruses.

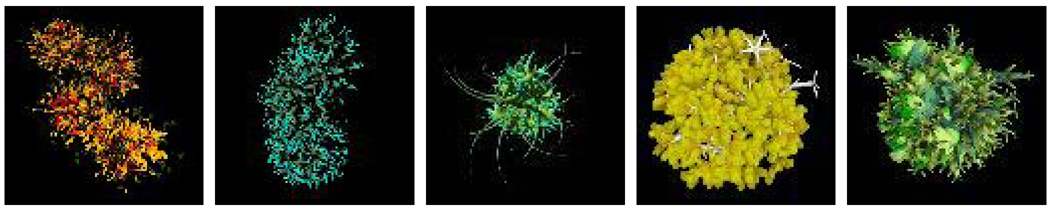
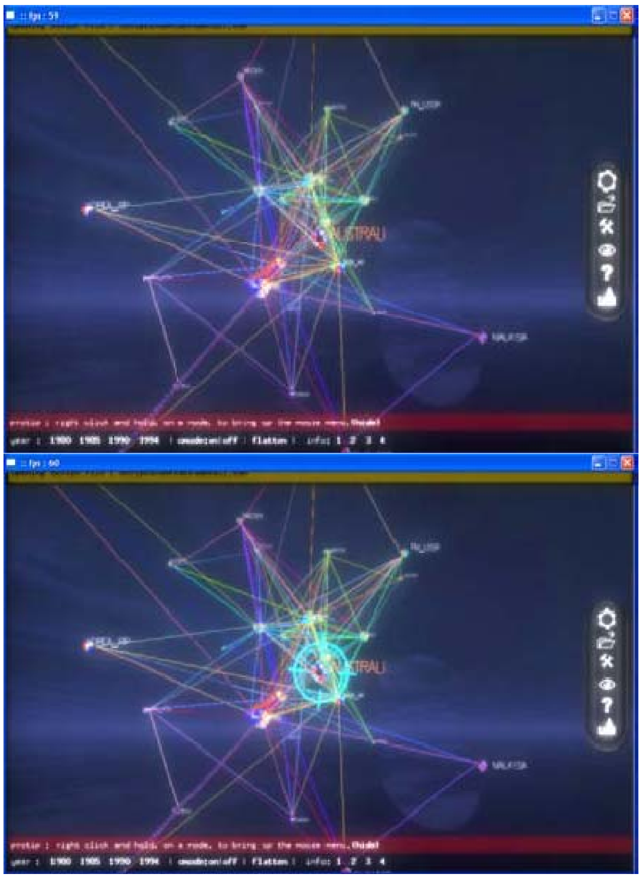


Figure 8 (Dragulescut, 2008)

from left to right: PWSLineage, Stormy, MyDoom, IRCbot, Virutmytob

Skyrails is a highly developed and visual tool that can render in 3D a DeepWorld© type of visualisation, showing what traffic is on the network, where it is going and pinpointing suspected anomalies in the traffic. The framework can be programmed to automatically search the traffic and discover suspect packets, extract the data and pass it to Dragulescuts organic algorithm. This will create the image, show it on screen or pass it to a fingerprint pattern classification application. This applications will map the pre determined minutiae points and derive the results then alert the administrator, all data collected would be stored in a large database for future analysis.

This could also be on screen via large LCD displayed in the NOC (network operations centre) monitored by staff.



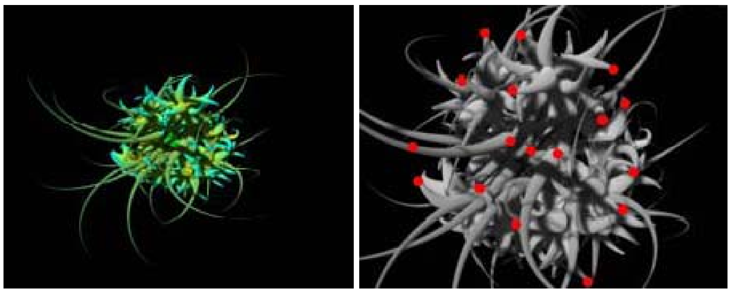


Figure 9 example of identification

The Scenario above see's Skyrails scanner watching the traffic on the network and either automatically or manually the anomaly is located, the code is extracted and passed to Dragulescuts organic algorithm which renders the code as an image. Which can then be displayed on a screen, staff can immediately identify the image on the screen as a virus or malware by it unique structure viewed as a recognisable image, steps can then be taken to deal with the anomaly, or it can be passed for verification to the pattern classification application for analysis and quarantined. This system could be rebuilt into one application and installed onto computer as part or instead of the traditional antivirus protection currently out in the market place.

CONCLUSION

The use of visualization can clearly be seen as a better solution to the endless task searching through logs by visualising the activity on the network. Is it better to use 3D or 2D? The belief is that 2D images using graph base representations although very useful do not scale well and have mapping and layout problems. It can be seen from the research in this paper that all applications can visualize data, some better than others in speed and accuracy but most can be reconfigured to perform different tasks in the identification of malware, where it came from and what patterns it forms if any.

This paper was intended to be an introduction and theriacal paper on visualization and has not drilled down into specific detail of application API's, storage backend or looked at tiering models, such as Application tiering, Model tiering and Scan tiering discuses in the paper "A Framework for Unified Network Security Management: Identifying and Tracking Security Threats on Converged Networks" (Dawkins, Clark, Manes, & Papa, 2005). More research needs to be done into how it would be possible to visualise malware more accurately and not rely on signatures, this paper believes this to be possible as all signs point to malware code being unique much like DNA Genome mapping. InSeon Yoo has done research on this and has come up with, Self-Organizing Map (SOM) "this is an unsupervised neural network method which has properties of both vector quantization and vector projection algorithms" (Yoo, 2004). This can detect viruses inside windows executable files without the aid of signatures by visualizing the virus code inside the executable. The future of visualization of either log file analysis or malware payloads within a network lie with 3D graphic images being displayed as interactive DeepWorld© visualization which can be drilled deep to identify anything that is happening, not just within suspect code or packet but the whole network for congestion or forensic analysis.

As mentioned in Greg Conti's book, Security Data Visualization: Graphical Techniques for Network Analysis, on the back cover, "a picture is worth a thousand packets"(Conti, 2007). In this researchers opinion it would probably be more like "a picture is worth million packets".

ACKNOWLEDGEMENT

This paper is loosely based on an idea from a lecture give at University of Glamorgan, UK and presented at the 7th Annual IEEE Information Assurance Workshop about "Visualization Framework for Intrusion Detection using three tiers, Visualisation, Middleware, and Database". (Read & Blyth, 2006)

REFERENCES

Blasco, J. (2005). An approach to malware collection log visualization. *Journal*. Retrieved from http://www.secviz.org/

Bruls, M., Geerlings, J., & Van Ham, F. (2002). *SequoiaView*, from http://w3.win.tue.nl/nl/onderzoek/onderzoek_informatica/visualization/sequoiaview//

Christian, & Raffy. (2007). *AfterGlow* 1.5.9. from http://afterglow.sourceforge.net/

Conti (2007). Security Data Visualization, Network Security Available from http://nostarch.com/securityvisualization.htm

Conti, G. (2007). *Rumint* 2.14. from http://www.rumint.org/

Dawkins, J., Clark, K., Manes, G., & Papa, M. (2005). A Framework for Unified Network Security Management: Identifying and Tracking Security Threats on Converged Networks. *Journal of Network and Systems Management,.*

Dragulescut, A. (2008). *Malwarez, a series of visualization of worms*, from http://www.sq.ro/index.php

Frie, A., & Rennhard, M. (2008). *Histogram Matrix: Log File Visualization for Anomaly Detection*. Paper presented at the International Conference on Availability, Reliability and Security. from http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4529398

Goodall, J. (2007). *TNV (The Network Visualizer or Time-based Network Visualizer)* 0.3.8. from http://tnv.sourceforge.net/

HCIL. (2008). *TreeMap* 4.1.1. from http://www.cs.umd.edu/hcil/treemap/

Honeytrap. (2008). *Honeytrap*, from http://honeytrap.sf.net/

Low, G. (2004). *GraphViz* 0.99. from http://www.graphviz.org/About.php

Marty, R., & Monsch, J. (2008). *DAVIX* 1. from http://www.secviz.org/content/the-davix-live-cd

Mwcollect. (2008). *Mwcollect*, from http://www.mwcollect.org/

NCSA. (2004). *NVisionIP*, from http://security.ncsa.uiuc.edu/distribution/NVisionIPDownLoad.html#NVisIP

Nepenthesdev. (2008). *Nepenthes* 0.2.2. from http://nepenthes.mwcollect.org/

Paxson, V. (2003). *The Spinning Cube of Potential Doom*. Paper presented at the SC03. from http://www.nersc.gov/nusers/security/TheSpinningCube.php

Read, H., & Blyth, A. (2006). An Integrated Visualisation Framework for Intrusion Detection. *Journal*. Retrieved from http://www.itoc.usma.edu/Workshop/2006/Program/Presentations/IAW2006-13-2.pdf

Roberts, E. (2001). Computers, Ethics, And Social Responsibility. *Journal*. Retrieved from http://cse.stanford.edu/class/cs201/Projects/viruses/anti-virus.html

Roth, F. (2008). *Nazar* 1. from http://nazar-interface.blogspot.com/

Tenable Network Security (2008). Security Center Security Center Available from http://www.tenablesecurity.com/solutions/

UCSD. (2008). *Cytoscape* 2.6.0. from http://www.cytoscape.org

Van Riel, J. (2007). *InteVis* 0.9.5. from http://www.cs.ru.ac.za/research/g02v2468/inetvis/0.9.3/doc/inetvisdoc.html#1.

Wachowski, A., & Wachowski, L. (Writer) (1999). *The Matrix*.

Wechsler, L., & Wright, A. (2000). Digital Wilt: Art from a Computer Virus. *Journal*. Retrieved from http://www.teigig.net/papers/visFinal.pdf

Widjaja, Y. (2007). *SkyRails* 0.1. from http://cgi.cse.unsw.edu.au/~wyos/skyrails/index.php

Yoo, I. (2004). Visualizing Windows Executable Viruses Using Self-Organizing Maps. *Journal*. Retrieved from http://vx.netlux.org/lib/aiy00.html

COPYRIGHT

Malware Detection and Removal: An examination of personal anti-virus software

Patryk Szewczyk & Murray Brand
SECAU Security Research Centre
Edith Cowan University
Perth, Western Australia

Abstract

SoHo users are increasingly faced with the dilemma of applying appropriate security mechanisms to their computer with little or no knowledge of which countermeasure will deal with which potential threat. As problematic as it may seem for individuals to apply appropriate safeguards, individuals with malicious intent are advancing methods by which malicious software may operate undetected on a target host. Previous research has identified that there are numerous ways in which malware may go undetected on a target workstation. This paper examines the quality of malware removal programs currently available on the market, which consumers may use whilst utilising the Internet. The research suggests that current anti-virus products, whilst able to detect most recently released malware, still fall short of eliminating the malware and returning the system to its original state. The paper does not compare or disclose potential flaws within each product; rather it depicts the current state of anti-virus products.

Keywords

Malware, malware detection, malware removal, anti-forensics, anti-virus software, SoHo

INTRODUCTION

It has been argued on numerous occasions that access to the Internet can be beneficial from a financial, social, recreational and educational perspective. As a result, the number of individuals acquiring Internet access in Australia alone is steadily increasing. The population in Australia as of August 2008, was just over 21 million, and of those, just under seven million households had Internet access (ABS, 2008). This is equivalent to approximately 33 percent of the national population. Unfortunately, individuals may be unaware of the threats that household computers may be susceptible to whilst using the Internet. Evidence suggests that at any given moment, approximately 90 percent of households worldwide may have some form of malware on their computer (Schmidt & Arnett, 2005, p.68).

A recent malware infection affected the International Epilepsy Foundation server which was subjected to a malware attack (Anonymous, 2006, p.2). The resultant outcome saw visitors machines infected with malware which would display fast-paced flashing images causing seizures. More recent malware infections have resulted in the target machine joining a collective botnet and used for malicious purposes, including identity theft and Distributed Denial of Service (DDoS) attacks. A new malware trend, namely 'ransomware' is encrypting documents, emails, pictures and music on the target host (Bridges, 2008). The aim of this class of malware is to extort money from the victim by offering to decrypt the contents of the victim's computer once the lump sum is paid. Malware affecting consumers is reaching all realms of the online world. Malware is increasingly targeting users of games such as World of Warcraft in an attempt to capture keystrokes during the login process, with the intent to steal in-game currency, inventory or accounts (Bridges, 2008, p.18).

Whilst each anti-virus vendor is continually promoting their product and releasing updates (signatures) on an almost daily basis, consumers are still falling victim to malware attacks. One of the threats facing consumers is the constant evolution and revolution of malware. Secure Computing (2008) reported a constant increase in malware propagation over the Internet towards the end of 2007. The analyst's team stated that "… Microsoft Windows continued to see a steady increase in exploits while decreases were seen on other platforms…" (Secure Computing, 2008). This quote emphasises that SoHo Microsoft Windows based hosts are still the predominant targets for attacks, whilst sabotage and attempted control of the infected host appears to be the main focus of each attack. This creates a dilemma because users must possess the knowledge and skills to safeguard and protect their computers from potential online threats.

The various penalties associated with malware development could include financial penalties and/or employment termination. As a result, there are various methods by which malware may go undetected by employing anti-forensic techniques to avoid detection and analysis (Brand, 2007). Detection avoidance may be achieved by data destruction, data hiding, and data contraception. Alternatively, malware may also go through a set of analysis avoidance methods including: exploiting flaws in analysis tools, system destruction when undergoing analysis, production of false disassembly listings, and subversion of dissembler heuristics. These anti-forensic methods are discussed in detail in Brand (2007, pp.1-8). Whilst numerous vendors discuss the advantages of their product in relation to malware removal, the anti-forensic and anti-avoidance techniques present a new challenge and hence as a result, not all malware may be detected during a system scan. A concept often overlooked by performance testing of anti-virus software is examined in this paper, by attempting to determine if recently released malware is detected and successfully removed from a Microsoft Windows based computer system.

Network based malware takes advantage of vulnerabilities on computers to install malicious software directly, or can initiate its download from an external, malicious site. The malware is generally packed, and the installation process usually involves a sequence of installing multiple files in the system directory, together with the creation and modification of registry keys, resulting in the creation of new malicious processes that are initiated each time the computer is started. Packing malware provides not only a method of compressing multiple files and installation instructions into a single file but also an opportunity to avoid signature detection. Most anti-virus software will simply detect that a file is packed and warn the user that the file is suspicious. The packed malware generally consists of multiple files which are then copied to various locations in the Windows System directory with the hidden attribute set and provided with file names that very closely resemble legitimate file names to provide additional camouflage. Registry changes can include disabling anti-virus and firewall software and the Microsoft Windows update mechanism. Auto start capabilities are generally changes made to the registry to ensure that the malware is activated each time the computer is restarted. Security settings are often changed in Internet Explorer so that more malware can be downloaded from sites without warnings displayed to the user.

METHODOLOGY

The paper depicts the current state of popular anti-virus products in relation to detection statistics provided by a third party malware analysis group.

Preparation

The test system was comprised of an IBM Pentium 4, 3.0GHz standalone workstation with 1GB of RAM. The chosen host operating system was Microsoft Windows XP Professional running Virtual PC 2007 as the virtual machine environment. In turn, the virtual image consisted of Microsoft Windows XP Professional with Service Pack 3 and all recommend updates from Microsoft up to and inclusive of August 19th 2008. All network connectivity was removed, resulting in the test machine being isolated from other workstations on the network.

Anti-virus Products

The top ten anti-virus products were downloaded from the Internet according to recommendations made by third parties (CNET, 2008; Top Ten Reviews, 2008). Locating a list of top products was based on a process that a SoHo user may utilise to compare and contrast available anti-virus products. As a result the top ten products were located by using the search terms "anti-virus review" and "top 10 antivirus" in Google and are presented in Table 1 below.

Table 7 Anti-virus products tested

Anti-Virus Product	Version
Avast	4.8
AVG	8.0.169
Bitdefender	2008 build 11.0.8
Eset Nod32	3.0.672

F-Secure Anti-Virus 2008	8.00 build 103
Kaspersky	8.0.0.454
Norman	-
Norton	2008 build 10.0.0.359
Panda	2009
Trend Micro	8.910

Product Testing
Each anti-virus product was installed independently on a separate baseline virtual machine image with all updates being applied to the product up to and inclusive of August 26th 2008. Utilising Regshot (Regshot, 2008), a *snapshot* was taken of the registry, the %\Windows and %\System directory of the operating system prior to the malware executing, after the malware had been installed (to monitor changes to the system files), and after the malware had been *supposedly* removed by the anti-virus product. Each malware specimen was executed on the baseline image via two methods:

1. Whilst the anti-virus product was disabled and all subsequent monitoring agents terminated.
2. Whilst the anti-virus product was enabled and hence providing consistent system monitoring.

Results were gathered as to whether each anti-virus product was successfully able to detect the malware specimen, eliminate the malware and all associated malicious files from the system, and return the system and associated system files to their original state.

Malware Specimens
Seven malware specimens were used in the course of this research. The following results were extracted from reports generated by the online malware analysis service Anubis (International Secure Systems Lab, Vienna University of Technology, Eurecom France, & UC Santa Barbara, 2008) and from Virus Total (2008). Table 2 below lists the malware signatures of the specimens as determined by the Ikarus anti-virus software. Names assigned to malware can vary widely between products. This can also include different detection results for the same malware specimen by numerous anti-virus products.

Table 8 Virus signature of specimens assigned by Ikarus

Sample	Virus Signature (Ikarus)
A	Net-Worm.Win32.Allaple.a
B	Trojan-Downloader.Win32.Small.hib
C	Trojan-Downloader.Win32.Small.hib
D	Trojan-PWS.Win32.OnLineGames.kil
E	Backdoor.Win32.Rbot
F	Backdoor.Win32.Rbot.cgu
G	Virus.Win32.Rbot.DCY

Specimen Justification

The malware specimens selected for the research have varying detection rates amongst numerous anti-virus products. Table 3 displays the detection rates for the specimens after submission through Virus Total to thirty-six different anti-virus products. Whilst an anti-virus product may make it to the top ten by third party reviewers it may not actually detect and/or eliminate the malware from an infected host. As depicted by Virus Total, malware specimen F and G appear to have a 100 percent detection rate amongst all current anti-virus products, whilst specimen C has the lowest detection rate.

Table 9 Virus Total Anti-virus detection rates

Malware Specimen Detection Rates						
A	B	C	D	E	F	G
94.40%	86.11%	69.44%	97.22%	94.40%	100%	100%

The malware chosen for testing encompasses varying degrees of sophistication and anti-avoidance techniques which any reputable anti-virus product should be capable of removing. Table 4 displays the varying, high level, functionality of the malware specimens which could include creation of files in the system directory, changing security settings in Internet Explorer, download of executable code, configure auto-start capabilities and joining of an IRC network.

Table 10 Sophistication of malware specimens

	Malware Specimen						
	A	B	C	D	E	F	G
Creates files in the Windows system directory	Yes	Yes		Yes	Yes	Yes	Yes
Changes security settings in Internet Explorer		Yes	Yes	Yes	Yes	Yes	Yes
Downloads executable code		Yes					
Autostart capabilities		Yes		Yes	Yes	Yes	Yes
Joins an IRC network		Yes			Yes	Yes	Yes

The number of malicious files and registry entries created is dependent upon the intention of malware. Table 5 below depicts the number of changes made by the malware specimens used in this research to the registry and to the file system as well as the number of services changed and processes created. A significant number of changes are made by each malware specimen. A malware removal process could be expected to remove the malicious programs, associated files and registry entries by detecting the presence of the malware through some combination of heuristics and signature matching. However, this technique is dependent upon an analyst having first analysed the malicious software to extract its signature and determination of the changes to the file system and the registry the malware makes.

Table 11 Malware characteristics

		Malware Specimen						
		A	B	C	D	E	F	G
Registry Keys	Created	5	0	2	0	8	4	2
	Modified	10	21	0	10	41	53	11
	Deleted	0	0	0	0	0	1	0
Files	Created	2	7	4	2	0	1	4
	Modified	12	13	24	8	3	5	11
	Deleted	0	2	2	2	1	1	1
Services Changed		1	0	0	0	0	0	0
Processes Created		1	0	1	1	0	1	3

Testing

The experimentation was aimed at mimicking the process that a SoHo user would use when installing and using an anti-virus product. As a result, the default settings were used for each and every product as recommended by the vendor during the install process. During the detection and elimination of malware stage, the steps taken by the vendor were also used to remove the malware from the system. If a system scan needed to be conducted, the depth of the scan was dependent on the recommend scan type outlined by the vendor. Whilst a "recommended" scan may not detect or eliminate malware as would a "deep scan", it is the purpose of this paper to mimic the actions taken by a SoHo user in dealing with a malware infected computer system.

RESULTS

In order to protect the vendor of each anti-virus product, names of the product are not used when describing the quality of a particular anti-virus product. Rather, each product is referred to as Product 1 through to Product 10 and does not relate to the order of the products presented in Table 1.

Table 6 depicts the malware specimens presented in Table 2 and whether or not they were detected by the anti-virus product. In this instance, the anti-virus product had been disabled and any monitoring capabilities of the product were terminated also, hence preventing the malware from being detected on install. The malware had been executed and the anti-virus product had been resumed. Product 6 and 7 were able to detect the malware immediately upon activation with a warning box appearing, notifying the user that a virus had been detected.

The research suggests that although the products tested have made it to the top ten list, they are still not capable of detecting malware (specimen C) which has been propagating throughout the Internet for a long period of time.

Table 12 Malware Specimens Detected

Product	**Malware Specimen**						
	A	B	C	D	E	F	G
1	□	□	□	□	□	□	□
2	□	□	□	□	□	□	□
3	□	□	□	□	□	□	□
4	□	□	□	□	□	□	□
5	□	□	□	□	□	□	□
6	□	□	□	□	□	□	□
7	□	□	□	□	□	□	□
8	□	□	□	□	□	□	□
9	□	□	□	□	□	□	□
10	□	□	□	□	□	□	□

Each anti-virus product provides constant system monitoring to potentially prevent malware from installing itself, whilst the anti-virus product is active. Whilst numerous vendors promote their product with the added feature of "real time monitoring", the research suggests that these monitoring capabilities are far from effective. As presented in Table 7 the detection capabilities of the malware have improved in contrast to the previous detection rate. However, the top 10 anti-virus products have clearly not detected malware which has been available for a long period of time. In most instances when the malware was transferred from a CD to the virtual machine image, most anti-virus products immediately detected the binary instantly and confirmed whether it should be quarantined or deleted. From the products which detected the malware, there was no method by which to make an exception and hence allow the malware specimen to be installed, hence providing good reaction and detection capabilities.

Table 13 Malware specimens detected during install

Product	**Malware Specimen**						
	A	B	C	D	E	F	G
1							
2							
3							

4							
5							
6							
7							
8							
9							
10							

Table 8 depicts whether the malware had been completely removed from the computer system utilising the didactic instructions provided by the vendor. The blank spaces depict those products which could not detect the malware and hence were not able to remove it either. It was discovered that whilst a product may detect and potentially eliminate a piece of malware from a computer system – that there are always remnants which remain on the system after the malware has been eliminated.

Table 14 Malware elimination

Product	**Malware Specimen**						
	A	B	C	D	E	F	G
1	✗	✓		✗	✓	✓	✓
2	✗	✓		✓	✓	✓	✓
3		✓	✓	✓	✓	✓	✓
4	✓	✓	✓		✓	✓	✓
5	✗			✓	✓	✓	✓
6	✓	✓	✓	✓	✓	✓	✓
7	✓	✓	✗	✓	✓	✓	✓
8	✓			✓	✓	✓	✓
9	✓	✓	✓			✓	✓
10		✓	✗	✓		✓	✓

As is demonstrated by Table 8, those products which were able to detect the malware were not necessarily able to eliminate the malware from the host. It was soon discovered that malware specimen C was significantly problematic for most anti-virus products. Whilst the product claimed to have removed the product, and in some instances needed the host to be restarted – following this didactic process discovered that the malware would re-appear as soon as the system was restarted.

CONCLUSION

The research in progress effectively examined the state of the top ten claimed anti-virus products available to consumers. From the research conducted, the most effective anti-virus product included Kaspersky Anti-Virus and BitDefender Anti-Virus. This strongly conforms to the results and reviews provided by the third party websites which detailed their most highly recommend anti-virus product. Whilst numerous factors may be important when choosing an anti-virus product including ease of update, support and the user interface – it is however the opinion of the authors that the effectiveness of a product in detecting and removing the malware is the most prevailing factor. From a SoHo perspective, the research suggests that naïve or uneducated users will continue to fall victim to the numerous, predominant threats found on the Internet. Whilst many products do provide instructions by which the user should follow to eliminate the malware, it appears that in many instances that these instructions are not clear. This research re-iterates the issue of how difficult it is for a SoHo user to protect themselves when not only are the product instructions unclear in the steps needed to remove malware, but furthermore when the product itself is not able to remove the malicious content in the first place.

REFERENCES

ABS. (2008). 8153.0 - Internet Activity. Retrieved August 1, 2008, from http://www.abs.gov.au/ausstats/abs@.nsf/mf/8153.0/

Anonymous. (2006). Attackers target epilepsy site. Network Security, 2006(4), 2.

Brand, M. (2007). Forensic Analysis Avoidance Techniques of Malware. Paper presented at the 5th Australian Digital Forensics Conference, Edith Cowan University, Mount Lawley Campus, Western Australia.

Bridges, L. (2008). The changing face of malware. Network Security, 2008(1), 17-20.

CNET. (2008). Antivirus - Reviews. Retrieved August 20, 2008, from http://www.cnet.com.au/tag/reviews/antivirus.htm

International Secure Systems Lab, Vienna University of Technology, Eurecom France, & UC Santa Barbara, U.S. (2008). Anubis: Analyzing Unknown Binaries. Retrieved October 4, 2008, from http://anubis.iseclab.org/

Regshot. (2008). Regshot. Retrieved August 25, 2008, from http://sourceforge.net/projects/regshot

Schmidt, M. B., & Arnett, K. P. (2005). SPYWARE: A Little Knowledge is a Wonderful Thing. Communications of the ACM, 48(8), 67-70.

Secure Computing. (2008). Secure Computing's Trends in Email, Web, and Malware Threats Retrieved August 3, 2008, from http://www.securecomputing.com/index.cfm?skey=1739

Top Ten Reviews. (2008). AntiVirus Software Review 2008. Retrieved August 20, 2008, from http://anti-virus-
software-review.toptenreviews.com/

Virus Total. (2008). Virus Total. Retrieved October 4, 2008, from http://www.virustotal.com/en/virustotalf.html

COPYRIGHT

The Impact of U3 Devices on Forensic Analysis

R. Tank and P.A.H Williams

School of Computer and Information Science
Edith Cowan University
Perth, Western Australia

Abstract

Flash and USB portable drives are now in common place use in computing environments. The U3 smart drive is one emerging type of enhanced flash drive. It is believed that U3 smart drive devices do not leave any record or evidence on a host PC after use. Therefore, it is conceivable that it could be used in a digital crime or attack on a computer or networked system. In circumstances where a portable device such as a U3 has been used, it is more complex for a forensic analyst to find evidence of its use. This paper discusses the impact of U3 smart drive devices on a forensic investigation. Further, it describes the forensic investigation undertaken of a computer in which U3 was used.

Keywords: U3, U3 smart technology, computer forensics, forensic investigation, forensic tools.

INTRODUCTION

Computer forensics broadly consists of identification, acquisition, analysis and reporting (Williams, 2008). It focuses on gathering digital evidence from devices such as computers and networks. Evidence may include files, image or the traces of a user's activities. Data on activity is often left in activity logs of operating systems, browsers, databases, web proxies, or network firewalls and so on. The discipline of digital forensics requires a detailed technical knowledge of the relationship between a computer's operating system and the supporting hardware, and between the operating system and system/application programs and the network. Finally, all evidence gathering must precede in a manner that ensures that the evidence is admissible in a court of law, and can be documented and presented in an intelligible manner (Carrier, 2005). It is against this background that the following research was undertaken to find what digital evidence could be retrieved from the new U3 smart drive technology.

The U3 smart drive is an enhanced type of flash drive. The U3 was developed by SanDisk and M-Systems and using an open-platform, allows data and programs to be transferred easily. Essentially, it is a method to auto-launch applications using a portable, removable drive (Everythingusb, 2005). The U3 smart drive is different from the normal flash drive because it comes with preinstalled programs, system files and a U3 launch pad. The U3 launch pad is what makes the U3 device unique, as it is a pre-installed auto-run program manager similar in look to the XP start menu. This automation is achievable because a small partition of the U3 drive mimics a CD-ROM drive whilst the second partition is a normal USB data partition. When attached to a Windows system it is immediately recognised as CD and thus the auto-play feature is enabled on the Windows platform. In addition, U3 devices can be password protected so when plugged in, the users is prompted for a password and then only can access the applications or data from the U3 smart drive (U3, 2008).

When a U3 smart drive is plugged into a host PC it uses the host to record information about the U3 and the programs it runs. However, when the U3 device is ejected from the host computer it runs a program called 'Cleanup.exe' to clean the evidence of the U3 device usage on the host computer. It is promoted by the manufacturer that the U3 smart drive does not leave any residual information or records on to the host PC after it is ejected and the cleaning program has run: "nothing is installed from the U3 smart drive to the PC you are using" and when you remove the U3 smart drive "will put the PC you are using back into the state it was in before you plugged in the U3 smart drive. No personal data or files related the applications you were running are left behind. Everything is cleaned up for you automatically" (Demo clip) (U3, 2008). This misconception has been propagated further by discussion of the technology in the media (Lasky, 2007; Wighting and Christmann, 2006) This facility can be seen as an advantage of the U3 smart drive and because of this advantage, may be favoured for use in digital crimes. For instance a U3 could be used to install software such as malware into a system or network to compromise a host PC. In this situation it is very difficult for a forensic investigator to find evidence of potential crime (Everythingusb, 2005). This research investigates these claims.

U3 TECHNOLOGY

U3 technology is a technology which allows a user to install and remove their own applications on the device as well as keeping data on the device. U3 technology provides the portability for a user to take data and applications with them and place them on any computer system without copyright issues. It provide the facility to utilise a PC and mimic the users' own PC by plugging in the U3 enabled device and using the applications and data installed on it. The U3 technology allows applications to write files or registry information to the host computer whilst also having the facility to removes evidence once it is ejected and removed from the computer. Essentially, leaving the computer as it was prior to the U3 being attached.

A U3 smart drive has two detectable parts to it, as shown in Figure 1.

- The first part is as a CDRom. The U3 file system which contain the necessary files to run U3 applications and system files that are read only are shown as part of the CDRom. Actually, the CDRom part of a U3 device is an ISO image, with the extension .ISO which is a standard CDROM file types, and it contains the auto run file for U3. When U3 is plugged in, the auto-run file runs and launch the U3 launch pad which is similar to Windows start menu.
- The second part of U3 device is a storage device where user can keep all the data as well as the all installed program in U3. This is shown as a removable disk.

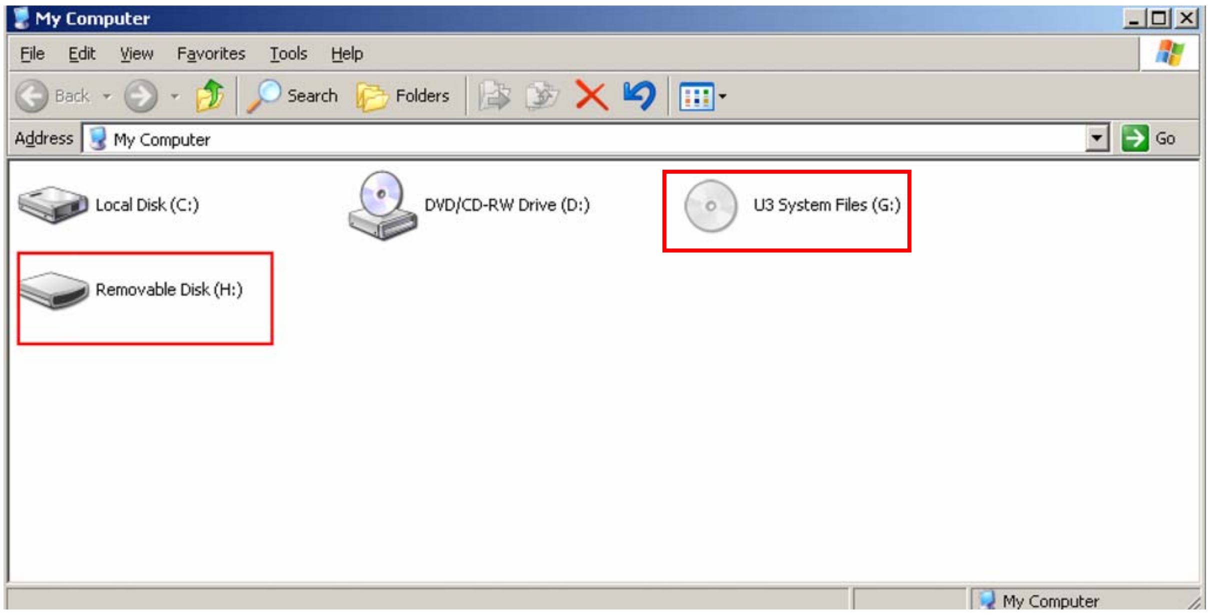


Figure 1: U3 devices detected by an operating system.

When the U3 device is plugged into a PC, the resulting entries can be found in the Windows registry consistent with the apparent detection of two devices by the operating system.

- The hardware id of the CDRom (G:) which can be located in registry is:
  - HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\USBSTOR\CdRom&Ven_Memorex&Prod_Mini_TravelDrive&Rev_6.50\0CF0C86102B0990B&1
- The hardware id of the storage device (H:) which can be located in the registry is:
  - HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\USBSTOR\Disk&Ven_Memorex&Prod_Mini_TravelDrive&Rev_6.50\0CF0C86102B0990B&0

PROCEDURE

This research used the hardware, software and procedures described below to investigate a U3 device. This investigation is similar in nature to that proposed by Spruil and Pavan (2007).

Preparation

The hardware and software used in this investigation were:

- PC with Intel P4 2.4 GHZ
- Gigabyte GA 8IG1000MK mother board
- Sandisk cruzer BI0805KCIB U3 smart drive 4GB
- WD 140 GB portable hard drive
- Windows XP SP2
- Access Data FTK Imager Version 2.1a
- Autopsy Forensic Browser
- Helix Linux 1.9

Prior to analysing the residual information the following steps were undertaken to prepare the U3 device and create known application usage from the U3 device.

1. A fresh copy of Windows XP SP2 with the default drivers was loaded onto a PC.
2. Using a separate PC, a new U3 smart drive was plugged in and software for the U3 was downloaded directly to the device from the U3 official website. The software downloaded included Fire Fox, Open Office etc. This software is free to download and use on the U3 device.
3. The U3 smart drive was plugged into the newly loaded computer installed with XP SP2 (from step 1).
4. Open Office was opened directly from U3 smart drive and a file created and saved onto U3 drive only.
5. Next, Fire Fox was opened and an image downloaded from internet directly to the U3 flash drive.
6. Next, an email account was opened using Fire Fox on the U3 and an email sent to another email account.
7. Lastly, the U3 flash drive was ejected.

Post Procedure and Analysis
Following the preparation, the following procedure was followed for data acquisition and analysis:

- Using a Helix CD, the PC hard disk was imaged and a hash value calculated for this image.
- A copy of the image was made (and an additional copy of the copy) and the hash values re-calculated and compared to ensure they were identical.
- Using a copy of the image, the residual information from the U3 usage was investigated, to see what data was left behind on the image.
  - Initially Autopsy was used to analyse the image, however a cursory search for the known .jpg file could not find any trace and it was decided not to pursue the investigation using the Autopsy software.
  - Next, FTK Imager Version 2 was used to analyse the image, with appropriate hash values calculated at each step.
  - The investigation looked for time line information, user account and other information.
  - A list of the findings was made and hash values re-calculated and compared.

RESULTS

Using the FTK Imager, the PC investigated and the following results obtained.

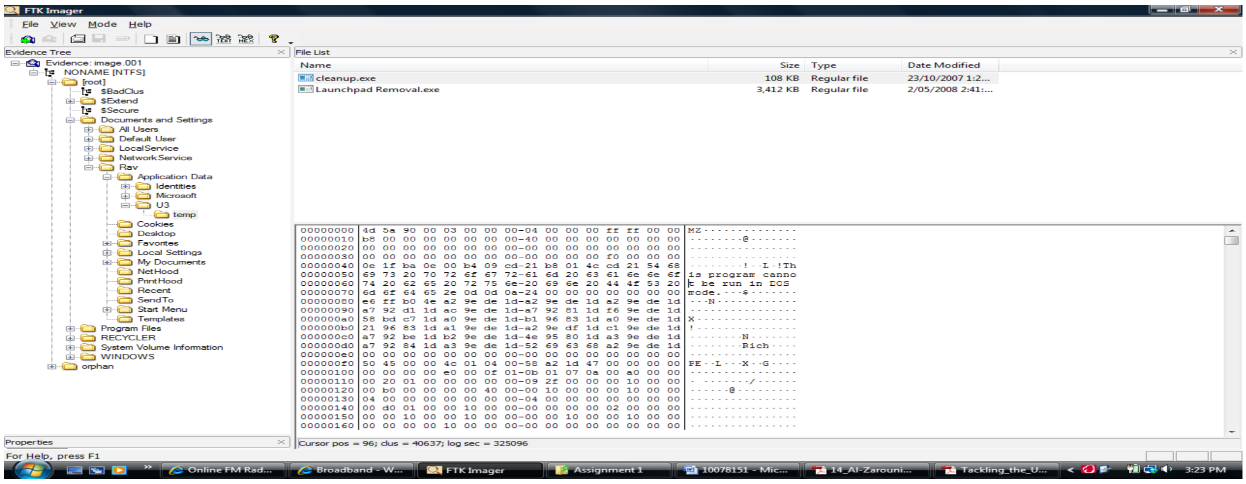
Post cleanup analysis
When the U3 drive was plugged into the PC it created a folder called U3 in the currently logged on users' directory. In this case the pathname was

Root/Document and settings/RAV (*User Name)/*Application data/U3

This folder holds an entry for each and every application run from the U3 device and thus logs user activity. Subsequently, when the U3 is ejected, the process automatically invokes a program called cleanup.exe to clean the entries from the host PC. However, after ejection of the U3 device, the files "cleanup.exe" and "Launch pad Removal.exe" remain on the host PC in a subfolder called 'temp' in the U3 folder, as shown in Figure 2. The path of the file and temp folder is

Root/document and setting/RAV (User Name)/Application data/U3/Temp/Cleanup.exe
Root/document and setting/RAV (User Name)/Application data/U3/Temp/Launch pad Removal.exe

It is therefore possible to ascertain that the U3 device has indeed been used on the host PC and the date and time the cleanup programs were executed.



Name	Size	Type	Date Modified
cleanup.exe	108 KB	Regular file	23/10/2007 1:2...
Launchpad Removal.exe	3,412 KB	Regular file	2/05/2008 2:41:...

```
00000000 4d 5a 90 00 03 00 00 00-04 00 00 00 ff ff 00 00 MZ··············
00000010 b8 00 00 00 00 00 00 00-40 00 00 00 00 00 00 00 ········@·······
00000020 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ················
00000030 00 00 00 00 00 00 00 00-00 00 00 00 f0 00 00 00 ················
00000040 0e 1f ba 0e 00 b4 09 cd-21 b8 01 4c cd 21 54 68 ········!··L·!Th
00000050 69 73 20 70 72 6f 67 72-61 6d 20 63 61 6e 6e 6f is program canno
00000060 74 20 62 65 20 72 75 6e-20 69 6e 20 44 4f 53 20 t be run in DOS 
00000070 6d 6f 64 65 2e 0d 0d 0a-24 00 00 00 00 00 00 00 mode.···$·······
00000080 e6 ff b0 4e a2 9e de 1d-a2 9e de 1d a2 9e de 1d ···N············
00000090 a7 92 d1 1d ac 9e de 1d-a7 92 81 1d f6 9e de 1d ················
000000a0 58 bd c7 1d a0 9e de 1d-b1 96 83 1d a0 9e de 1d X···············
000000b0 21 96 83 1d a1 9e de 1d-a2 9e df 1d c1 9e de 1d !···············
000000c0 a7 92 be 1d b2 9e de 1d-4e 95 80 1d a3 9e de 1d ········N·······
000000d0 a7 92 84 1d a3 9e de 1d-52 69 63 68 a2 9e de 1d ········Rich····
000000e0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ················
000000f0 50 45 00 00 4c 01 04 00-58 a2 1d 47 00 00 00 00 PE··L···X··G····
00000100 00 00 00 00 e0 00 0f 01-0b 01 07 0a 00 a0 00 00 ················
00000110 00 20 01 00 00 00 00 00-09 2f 00 00 00 10 00 00 · ·······/······
00000120 00 b0 00 00 00 00 40 00-00 10 00 00 00 10 00 00 ······@·········
00000130 04 00 00 00 00 00 00 00-04 00 00 00 00 00 00 00 ················
00000140 00 d0 01 00 00 10 00 00-00 00 00 00 02 00 00 00 ················
00000150 00 00 10 00 00 10 00 00-00 00 10 00 00 10 00 00 ················
00000160 00 00 00 00 10 00 00 00-00 00 00 00 00 00 00 00 ················
```

Cursor pos = 96; clus = 40637; log sec = 325096

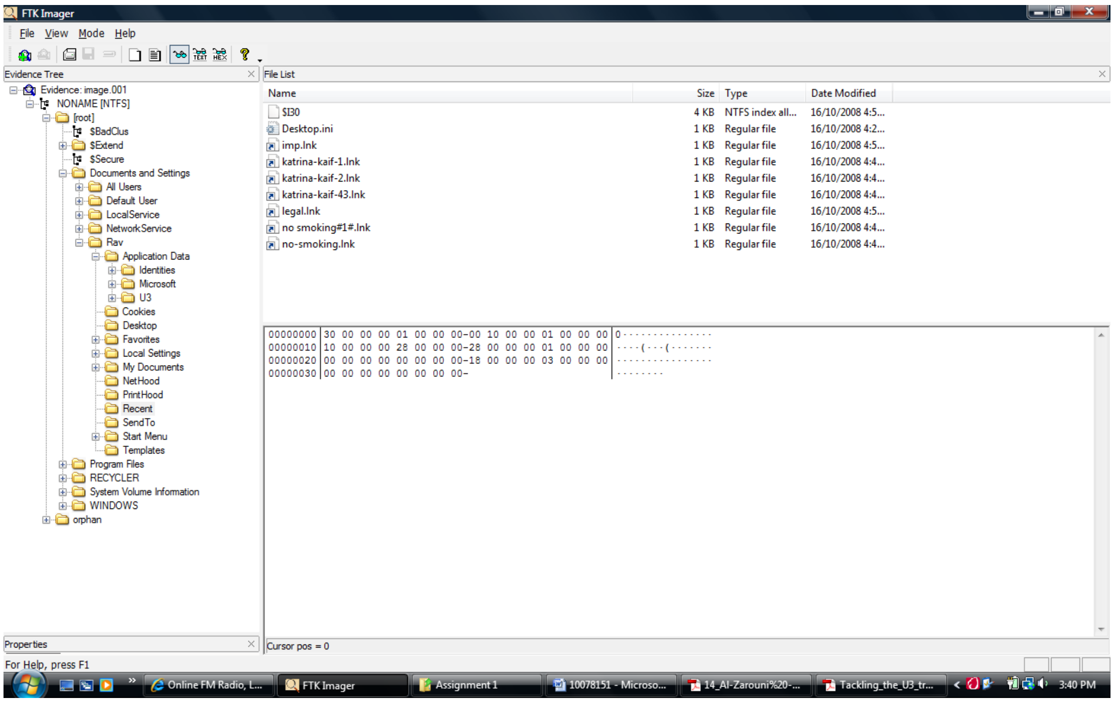
Figure 2: cleanup.exe and launch pad removal.exe

‘Recent’ folder data

Other information discovered which can be considered for evidence was found in the folder called ‘Recent’ under the current users’ folder. The path of that folder was:

Root/document and settings/RAV (user name)/Recent

As the U3 file system is linked with the Windows file system and therefore relies on the Windows built-in explorer option, it creates a link for each file accessed using the U3 in the ‘Recent’ folder, as shown in Figure 3. The ‘Recent’ folder has an entry of all files using the file extension .LNK. With the help of this information it is possible to build up a picture of what activities were carried out involving the U3 device. As previously, the date and time of execution or file modification is recorded.



Name	Size	Type	Date Modified
$I30	4 KB	NTFS index all...	16/10/2008 4:5...
Desktop.ini	1 KB	Regular file	16/10/2008 4:2...
imp.lnk	1 KB	Regular file	16/10/2008 4:5...
katrina-kaif-1.lnk	1 KB	Regular file	16/10/2008 4:4...
katrina-kaif-2.lnk	1 KB	Regular file	16/10/2008 4:4...
katrina-kaif-43.lnk	1 KB	Regular file	16/10/2008 4:4...
legal.lnk	1 KB	Regular file	16/10/2008 4:5...
no smoking#1#.lnk	1 KB	Regular file	16/10/2008 4:4...
no-smoking.lnk	1 KB	Regular file	16/10/2008 4:4...

```
00000000 30 00 00 00 01 00 00 00-00 10 00 00 01 00 00 00 0···············
00000010 10 00 00 00 28 00 00 00-28 00 00 00 01 00 00 00 ····(···(·······
00000020 00 00 00 00 00 00 00 00-18 00 00 00 03 00 00 00 ················
00000030 00 00 00 00 00 00 00 00-                         ········
```

Cursor pos = 0

Figure 3: Recent folder

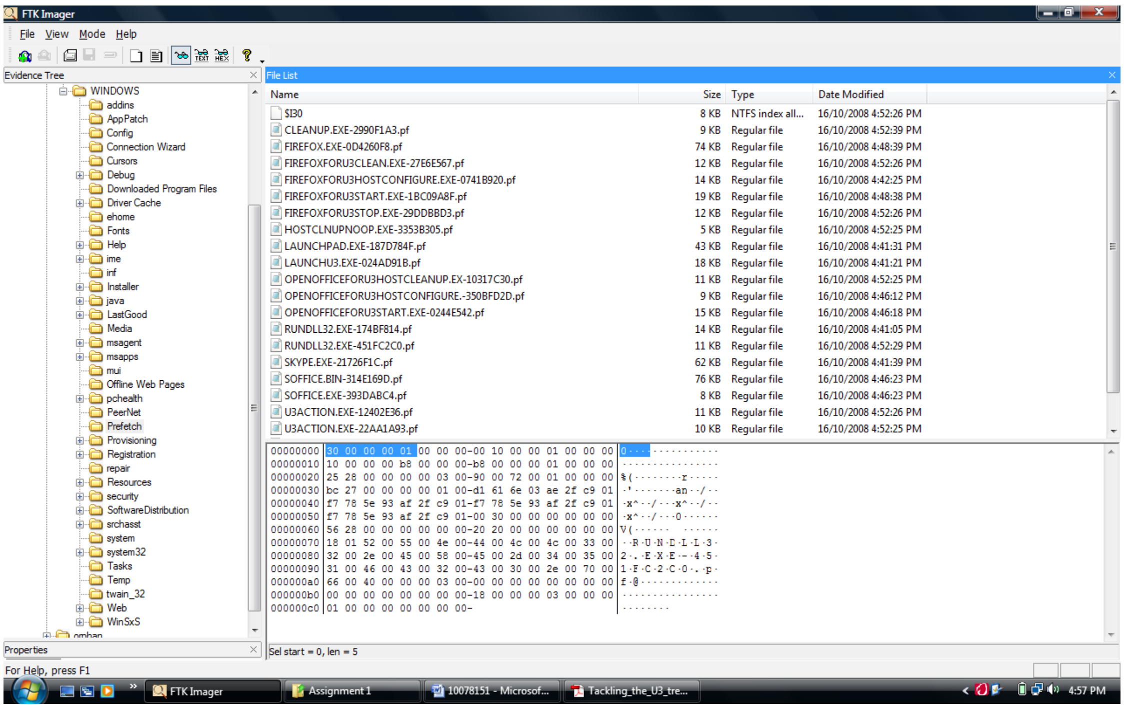
When analysing each link file dump, it can be seen that the information about the file modified. E.g. the file name Katrina-kaif-1.lnk contains information that the actual file accessed is Katrina-kaif-1.jpg and it also has been saved to the drive called G:, and which folder it was saved to, in this case under the folder ‘Legal’, together with the date and time. For each .LNK file it shows the particular information about the file. This information may be very useful to a forensic investigation.

PREFETCH persistent data

In addition, the Windows PREFETCH folder could be analysed. This folder stores the PREFETCH files, which are created by Windows to enhance the speed of the system. PREFETCH file are with .PF format. Windows records every activity of the computer in the PREFETCH folder which is located at the path

Root/Windows/PREFETCH

Figure 4 indicates that the PREFETCH folder shows the activities/programs run on the computer, with the date and time modified. If any application runs from the computer it creates the file of the format '*application name*.pf', with file size, disk type and date and time the file was opened or the application executed. This information can be used to create a timeline and indicates what activities were undertaken or files accessed from the computer.



Name	Size	Type	Date Modified
$I30	8 KB	NTFS index all...	16/10/2008 4:52:26 PM
CLEANUP.EXE-2990F1A3.pf	9 KB	Regular file	16/10/2008 4:52:39 PM
FIREFOX.EXE-0D4260F8.pf	74 KB	Regular file	16/10/2008 4:48:39 PM
FIREFOXFORU3CLEAN.EXE-27E6E567.pf	12 KB	Regular file	16/10/2008 4:52:26 PM
FIREFOXFORU3HOSTCONFIGURE.EXE-0741B920.pf	14 KB	Regular file	16/10/2008 4:42:25 PM
FIREFOXFORU3START.EXE-1BC09A8F.pf	19 KB	Regular file	16/10/2008 4:48:38 PM
FIREFOXFORU3STOP.EXE-29DDBBD3.pf	12 KB	Regular file	16/10/2008 4:52:26 PM
HOSTCLNUPNOOP.EXE-3353B305.pf	5 KB	Regular file	16/10/2008 4:52:25 PM
LAUNCHPAD.EXE-187D784F.pf	43 KB	Regular file	16/10/2008 4:41:31 PM
LAUNCHU3.EXE-024AD91B.pf	18 KB	Regular file	16/10/2008 4:41:21 PM
OPENOFFICEFORU3HOSTCLEANUP.EX-10317C30.pf	11 KB	Regular file	16/10/2008 4:52:25 PM
OPENOFFICEFORU3HOSTCONFIGURE.-350BFD2D.pf	9 KB	Regular file	16/10/2008 4:46:12 PM
OPENOFFICEFORU3START.EXE-0244E542.pf	15 KB	Regular file	16/10/2008 4:46:18 PM
RUNDLL32.EXE-174BF814.pf	14 KB	Regular file	16/10/2008 4:41:05 PM
RUNDLL32.EXE-451FC2C0.pf	11 KB	Regular file	16/10/2008 4:52:29 PM
SKYPE.EXE-21726F1C.pf	62 KB	Regular file	16/10/2008 4:41:39 PM
SOFFICE.BIN-314E169D.pf	76 KB	Regular file	16/10/2008 4:46:23 PM
SOFFICE.EXE-393DABC4.pf	8 KB	Regular file	16/10/2008 4:46:23 PM
U3ACTION.EXE-12402E36.pf	11 KB	Regular file	16/10/2008 4:52:26 PM
U3ACTION.EXE-22AA1A93.pf	10 KB	Regular file	16/10/2008 4:52:25 PM

```
00000000 30 00 00 00 01 00 00 00-00 10 00 00 01 00 00 00 0···············
00000010 10 00 00 00 b8 00 00 00-b8 00 00 00 01 00 00 00 ················
00000020 25 28 00 00 00 00 03 00-90 00 72 00 01 00 00 00 %(········r·····
00000030 bc 27 00 00 00 00 01 00-d1 61 6e 03 ae 2f c9 01 ·'·······an··/··
00000040 f7 78 5e 93 af 2f c9 01-f7 78 5e 93 af 2f c9 01 ·x^··/···x^··/··
00000050 f7 78 5e 93 af 2f c9 01-00 30 00 00 00 00 00 00 ·x^··/···0······
00000060 56 28 00 00 00 00 00 00-20 20 00 00 00 00 00 00 V(······  ······
00000070 18 01 52 00 55 00 4e 00-44 00 4c 00 4c 00 33 00 ··R·U·N·D·L·L·3·
00000080 32 00 2e 00 45 00 58 00-45 00 2d 00 34 00 35 00 2·.·E·X·E·-·4·5·
00000090 31 00 46 00 43 00 32 00-43 00 30 00 2e 00 70 00 1·F·C·2·C·0·.·p·
000000a0 66 00 40 00 00 00 03 00-00 00 00 00 00 00 00 00 f·@·············
000000b0 00 00 00 00 00 00 00 00-18 00 00 00 03 00 00 00 ················
000000c0 01 00 00 00 00 00 00 00-                        ········
```

Sel start = 0, len = 5

Figure 4: Windows PREFETCH

Figure 4 shows that the program entries include (in most recent first order):

- Cleanup.exe-2990F1a3.pf
- Firefox.exe-0D4260F8.pf
- FirefoxforU3hostconfigure.exe-0741B920.pf
- FirefoxforU3stop.exe-29DDBBD3.pf
- Launchpad.exe-187D784F.pf
- OpenofficeforU3hostcleanup.ex-103117C30.pf
- Skype.exe-21726F1C.pf
- U3action.exe-12402E36

From these entries it is possible to ascertain that a U3 device has been plugged into the computer and which applications like Skype, Open Office, Firefox was accessed from the U3 drive.

In addition, other information was located in the 'textconv' file, located at:
Root/Program files/common files/Microsoft shared/TEXTCONV

DISCUSSION

The collection of evidence to prove organisational misconduct or criminal activity is crucial to successful prosecution. As the U3 smart drive becomes a more commonly used device the opportunity for digital crime increases with such devices since software can easily be installed on the device and it is highly portable. Tracking the use of portable devices is a security issue that organisations face day-to-day and therefore the use of such devices for malicious or criminal activity pose even larger problems in detection. Since it is believed that these devices are undetectable after usage is incorrect. Therefore, this finding has positive implications for organisational security where detection of portable devices is problematic.

From a forensic investigation perspective, another important aspect is that at present U3 is very new technology and therefore few tools and procedures for forensic investigation exist specifically for this device. This initial research indicates that there is some forensic trace of U3 usage left on the host computer. This provides a starting point for further forensic investigation of a computer if it is believed that a crime has been enacted involving a U3 smart drive.

The results obtained are in three basic categories:

- Cleanup files:
  - The two files Cleanup.exe and Launch pad removal.exe this both can be find , which proves that the U3 smart drive has been used in the computer.
- Recent Folder:
  - In the recent folder all the .lnk files can be find so this helps in to find that what files has been accessed in this computer.
- PREFETCH Folder:
  - From PREFETCH folder the list of all activity can be find.

From these investigations it may be possible for a forensic investigator to create a timeline and produce activity evidence which may be useful in a court of law. The three sections covering cleanup, the Recent folder and the Prefetch folder, indicate that a U3 device has been used and when. Further, it shows what software or application has been run from the U3 smart drive, and at what time and what files has been created or modified or saved to U3 drive. This information may assist in forensic investigations or U3 and other similar devices using the same or technology.

CONCLUSION

The U3 smart drive is a powerful transportable device for a user to keep applications and data. However, this test found that it is an erroneous assumption that it does not leave any trace on the host PC after it is used. There is potentially useful evidence which can be located on the host PC after use of a U3 smart drive, as this paper demonstrates. Record of U3 usage, if left uncleansed, can provide evidence of all applications run. Further, it provides links to the names of files accessed although not to the actually contents of these file.

Further research needs to be undertaken in regard to the cleanup.exe program itself, to identify if other useful evidence can be located. This would include the exacted files created or modified with the use of the U3 smart drive applications, or user installed applications. If security and system administrators are to protect their systems with the advent of new technologies such as smart drives, then finding methods to detect the use of such technology and collect forensic evidence to prove its use, is essential.

REFERENCES

Al-Zarouni, M., & Al-Hajri, H. (2007). A Proof of concept project for utilizing U3 technology in incicent response. In C. Valli and A. Woodward (Eds.), *Proceedings of the 5th Australian Digital Forensic Conference*, School of Computer and Information Science, Edith Cowan University, Perth, WA.

Carrier, B. (2005). *File system forensic analysis*. Upper Saddle River, NJ: Addison-Wesley

Everythingusb. (2005) *U3 - 'Official' Portable USB Apps Platform.* Retrieved October 3, 2008, from http://www.everythingusb.com/u3.html.

Forensicfocus. (n.d.). *U3 capable USB.* Retrieved October 7, 2008, from http://www.forensicfocus.com/index.php?name=Forums&file=viewtopic&t=1225.

ForensicsComputer. (n.d.). Retrieved September 26, 2008 from http://computer-forensics.safemode.org/index.php.

Lasky, M.S. (2007). Painless backup to USB drives. *PC World* 25(10), 140.

Mckinnon, M. (2007). *Computer Forensics/E-Discovery Tips/Tricks and Information.* Retrieved September 26, 2008, from http://cfed-ttf.blogspot.com/.

Microsoft. (2006). *U3 USB drives.* Retrieved October 8, 2008, from http://forums.microsoft.com/MSDN/ShowPost.aspx?PostID=942660&SiteID=1.

Nirsoft. (2003). *IEHistoryView v1.37 - View Visited Web Sites of Internet Explorer.* Retrieved October 14, 2008, from http://www.nirsoft.net/utils/iehv.html.

Searchenterprisedektop. (2005). *Empty the prefetch folder.* Retrieved September 26, 2008, from http://searchenterprisedesktop.techtarget.com/tip/0,289483,sid192_gci1088757,00.html.

Spruil, A. & Pavan, C. (2007). Tackling the U3 trend with computer forensics. Digital Investigation 4(1), 7-12.
U3. (2008). *Homepage.*. Retrieved September 19, 2008, from http://www.u3.com/default.aspx.
U3devforum. (2006). *U3 Deployment Kit.* Retrieved September 23, 2008, from www.u3devforum.com/claudio/U3_Deployment_Kit_081406.pdf.
Williams, P. A. H. (2008). Is there an ideal forensic process? In H.R. Arabnia, S. Aissi & M. Bedworth (Eds.) *Proceedings of the 2008 World Congress in Computer Science, (2nd ed)* Computer Engineering, and Applied Computing - SAM'08 - The 2008 International Conference on Security & Management, (pp. TBA). USA: CSREA Press.
Wighting , M. J. & Christmann, E.P. (2006). The Size of Things to Come.(LaCie Brick, portable hard drive). *Science Scope* (Dec), 72-73.

COPYRIGHT

Issues common to Australian critical infrastructure providers SCADA networks discovered through computer and network vulnerability analysis

Craig Valli
SECAU Security Research Centre
Edith Cowan University
Perth, Western Australia
c.valli@ecu.edu.au

Abstract

This paper reports on generic issues discovered as a result of conducting computer and network vulnerability assessments (CNVA) on Australian critical infrastructure providers. Generic issues discovered included policy, governance, IT specific such as segregation, patching and updating. Physical security was also lacking in some cases. Another issue was that previous security audits had failed to identify any of these issues. Of major concern is that despite education and awareness programs, and a body of knowledge referring to these issues, they are still occurring. It may be necessary for the federal government to force organisations to undergo computer and network vulnerability assessment from recognised experts on a regular basis.

Keywords

SCADA, security, computer and network vulnerability assessment, process control systems

INTRODUCTION

Traditionally SCADA systems were designed around reliability and safety, and if they were network connected they were connected on isolated internal networks for the purposes of control and management; essentially a closed system. Typically in these situations security was not a consideration due to the isolated nature of the systems and their closed nature. It should be remembered that these systems were implemented also in an era when computing and information technology will also largely conducted in isolated installations or laboratories around the globe (Stouffer *et al*, 2007). The world has now moved on and we are becoming increasingly interconnected and interdependent on these connections for the full functioning of modern society. One of the main conduits and enablers for this has been the rapid expansion of the Internet. Correspondingly, as a result of the growth of the Internet there has been a convergence on the TCP/IP protocol suite as the dominant network protocol for business and industry. This has seen many hardware and software vendors, including SCADA vendors, align their products with this kind of reality (Igure *et al*, 2006). This transition to a level of greater interconnectedness has impacted on the security posture of these systems. Threats and vectors that did not exist prior to this transition are now becoming realisable threats. There is an increasing reliance on public telecommunications networks to link previously separate SCADA systems making them more accessible to attacks. Then be increasing use of published open standards and protocols, in particular Internet technologies, expose SCADA systems to Internet or network borne attacks, that were not realisable for instance on proprietary protocols that were used previously to control such systems. The actual increased interconnection of SCADA systems to corporate networks is a significant threat in itself enabling and making them accessible to undesirable entities e.g. malfeasant insiders (Jackson-Higgins, 2007). Many systems that are in SCADA networks often lack mechanisms to provide confidentiality of communications for example the use of sufficiently strong encryption to protect data during transit. This type ofproblem coupled with a lack of appropriately strong or granular authentication mechanisms and controls in many SCADA systems and the problem becomes manifold even to a security novice. This paper reports on and discusses some generic issues discovered as a result of conducing computer and network vulnerability assessments on Australian critical infrastructure providers.

THE CNVA PROGRAM AND WHY WE NEED IT FOR SCADA

The Computer Network Vulnerability Assessment (CNVA) program is an Australian Government grants scheme developed to help ensure the security of Australia's critical infrastructure (TISN 2008). The program is managed by GovCERT.au – the Australian Government computer emergency readiness team, which is under the umbrella of the Australian Government Attorney-General's Department. The program provides funding of up to 50 per cent of the cost of a vulnerability assessment to private critical infrastructure owners and/or operators, including Government Business Enterprises.

Although not many public stories, there are reports of attacks against critical infrastructure. Sources of attack include vectors both intentional and unintentional. Categories covered by the former include hacking, insider malfeasance, and malware. Unintentional vectors include malware and lack of due diligence. The most infamous of the intentional attacks is the case of Maroochy shire. Their SCADA system was hacked by a person who had been employed to install the system after a request for employment was turned down (Smith, 2001). An example of unintentional malware interfering with SCADA systems is the SQL Slammer worm released in 2003. This worm infected the Davis-Besse nuclear power plant in Ohio, USA, causing operators to lose a degree of control over the system. As a result, the plant safety display system and process control computers were shut down for nearly seven hours (Poulsen, 2003). A more recent incident and an example of lack of due diligence, orintentional consequences of a legitimate activity, was the shutdown of a nuclear power plant. The update of software on a computer on the plants business network caused reset of data on the control system. The safety systems interpreted this lack of data to mean that there was no water in the reservoirs that cool the rods, and shut down the reactor (Krebs, 2008). There has also been public commentary by a CIA analyst that utilities based in foreign countries had been attacked through the internet (Nakashima & Mufson, 2008). It was further reported that these attacks had resulted in power outages in a major city. The same article also indicated that there had been an increase in internet based attacks against utilities in the United States.

SPECIFIC ISSUES INDETIFIED THROUGH THE CNVA PROGRAM

This section discusses the issues that have been discovered as a result of the author's participation in the CNVA program. There were a range of specific issues discovered, with the generic, common issues highlighted in this paper. There are no specific cases discussed here as this would be in violation of various confidentiality agreements.

Connection of SCADA to corporate

This point is not really a surprise, as it is why the CNVA exists. However, despite available advice and documentation (Stoufer *et al*, 2007), and in some cases even in complete contradiction of corporate policy, corporate and SCADA networks are being connected with little, if any, segregation. As mentioned previously, SCADA has a soft underbelly as it was designed to work, with security coming second to functionality. However, when SCADA systems were initially designed, the internet didn't exist, and what elements may have existed was not a threat to systems installed at that time. The issue here is that despite evidence that connection of SCADA to corporate networks without appropriate barriers is an identified risk, and should not happen, it is still occurring.

Governance

All cases had significant issues with effective governance of the critical infrastructure assets. In this case we mean the assets to be both the corporate and SCADA networks of the organisations the authors examined. All organisations demonstrated poor delineation of corporate ownership and responsibility for the SCADA networks and in some cases the supporting information technology infrastructures. Several sessions during the assessments sometimes degenerated into a scene from Martin Scorsese film i.e. "are you talkin to me, are you talking to me?" with the general abrogation of responsibility ensuing shortly there afterwards. Sadly one of the common memes was "Engineers don't know much about security with the counter meme "IT nerds know less about SCADA" often being the first point both parties agreed on. Some fundamental governance questions had not been addressed by any of the organisations.
These were:

- Who was responsible for change management for each of the assets?
- Who is in charge of the IT network, and are they still in charge after it is connected to SCADA?
- What is the escalation process should there be a record of incident?

In some cases, there were even issues relating to the ownership and availability of documentation for corporate networks and computer hardware and software.

Policy

As a result of poor governance structures within the organisations the policy structures also suffered. There was little evidence of any policy review, policy enforcement, policy auditing or existence of any policyimplementation. Review of the policy documentation that was presented during the assessments revealed it was rarely current and was lacking in relevant details. The policy in some instances had not been reviewed or revised for several years. In some cases where there was the existence of a policy there existed approved procedures which were in direct contravention to the existing policy. These procedures are also often created, produced and implemented by third parties such as contractors. In one case the outsourced contract is for the network infrastructure eventually refused to produce policy documents. The basis for this was that they were not required under the terms of the contract to surrender such documents.

IT SPECIFIC ISSUES

Un-Patched Hardware And Software

All of the assessments conducted found exploitable vulnerability as a result of un-patched hardware and software issues. Of particular concern were perimeter firewalls that had multiple exploitable vulnerability as a result of dilettante patching regimes. These firewall exploits were well known and had been known in the security community for up to four to five years.

Due to legacy nature of SCADA systems all of the underlying supporting computer operating systems had significant vulnerability. Most of the operating systems examined in all assessments had long since become no longer supported by the vendor. Most of the supporting applications in the form of SQL servers, network operating systems or specific SCADA applications were likewise no longer supported by the vendor.

Lack Of Network Segregation And Segmentation

Assessments conducted so far have revealed networks that are severely compromised under the defence in-depth strategy. Several of the network architectures were based on flat 10.0.0.0/8 networks allowing for a reversal of the entire network including corporate and control systems. Penetration of perimeter controls would have allowed complete sight into the enterprises examined. This would allow the easy implantation of malcode such as packet sniffers or keystroke analysers into the network architectures. In one instance the organisation was unsure as to where the connections to a fibre ring they connected to which was used by other organisations actually terminated. Furthermore they were initially unsure and as to what controls and countermeasures were in place to prevent ingress into the network from this network.

Lack Of Sound Authentication Mechanisms

Several of the installations provided generic logins to staff members for example username equals staff password equals staff. All of the systems reviewed had no reliable or monitored audit trial for systems access. That is no coordinated logging of even simple statistics such as logon or log off was conducted. These logins were provided mainly for expediency and were handed out typically as a result of the work environment trusting individuals. Given that 60 to 80% of losses sustained by successful I T. enabled crime or malfeasance is committed by insiders (Richardson, 2008), one could postulate this is an extremely unwise move on the part of management. Furthermore, some of the generic accounts uncovered in the examinations had root or admin level access to the entire IT infrastructure. This would allow any malicious individual the power to corrupt absolutely.

The use of such simple and group based passwords also allows a substantial vector for the disgruntled employee to penetrate the system and wreak havoc with impunity.

Monitoring, logging, auditing

There was a complete lack of any substantive logging of network interactions with attendant monitoring and then subsequent auditing and review. This observation put all organisations in an invidious position with respect to the network based exploit and attack. This leads to the realisation that most of the organisations had no ability to answer fundamental questions such as have we been attacked? We got hacked – how did it happen? Inability to answer even these basic questions makes amelioration or reduction of any network borne threat a relatively impossible task.

In one organisation there was a logging of firewall connections but no actual review of these logs to determine any emergent threat, unauthorised connections or internal malfeasant activity. The net value of this type of logging is a completely negative proposition for the organisation. The organisation is wasting money loggingdata that was never intended to be examined.

It has been proven already due to the nature of SCADA systems that the use of intrusion detection systems and intrusion prevention systems can shutdown communications, which can have potentially catastrophic unintended consequences (Fink *et al*, 2006). If we cannot reliably implement and use intrusion detection systems or intrusion prevention systems to protect networks that run SCADA devices, the use of monitoring logging and auditing are fundamental tools and techniques in providing response against network based attack. In some cases the use of these is the first and last line of defence against attacks that would be perpetrated against networks.

Physical Security

Although not part of a computer system or hardware, physical security is still an important part of computer security, and also part of any properly conducted computer and network vulnerability assessment. For example, are server rooms protected by appropriate fire suppression systems? Are physical access control methods inplace for servers and other equipment?

There were a number of issues relating to physical controls which are of concern, as physical systems can be barriers to casual attacks. One organisation had no restriction on access into the main control room for the SCADA network. Another had servers and other communications equipment stored in a physically vulnerable location. Yet another had no fire suppression system for their SCADA master servers. A remote site visited as part of an assessment had no alarms on doors, unaccounted for keys, rooms that stayed open and unlocked that also had computer terminals. Locked doors were often the only barriers in some cases.

Previous reports and audits

At some of the organisations that were assessed, there were reports on the security that had been conducted by previous groups. It was unclear as to whether these had been conducted as part of the CNVA program, or if they had been commissioned by these organisations in a commendable attempt to increase their security posture. Unfortunately, the reports had missed most of the critical issues identified by the authors in these cases. That is not to say that the authors did not also miss some problems; nobody is perfect. The issue is that these were reports prepared by large organisations who specialise in IT security auditing. It would be of concern if these organisations were part of the CNVA panel, and had been conducting audits on this basis. Given that GovCERT has oversight on the reports produced by CNVA audits, it seems more likely they were not produced as part of the CNVA program. The issue, however, is that security audits are being performed by organisations that do not appear to have the necessary knowledge of specific SCADA IT security issues that must be addressed if security is to be improved.

CONCLUSION

The vulnerability assessments conducted by the authors have left little doubt that there is certainly a need for the CNVA program. It is understandable that safety and risk engineers and managers in the critical infrastructure sector are largely concerned with physical as they either don’t understand the nature or the risk from the information technology perspective. The authors have seen on more than one occasion the attitude that “It won’t happen to us”, or “I have far more to worry about with a faulty valve exploding”. There was little perceived risk from the IT vector. However, the reality is that cyber attacks against CIPs are happening around the world, as pointed out earlier in this article. The other upcoming issue is that these CIPs are looking to implement TCP/IP based communication protocols in place of largely unknown legacy protocols such as Modbus and DNP3. This will only introduce further vulnerability, as use of such protocols currently at least provides some sort ofsecurity through obscurity.

Despite the existence in Australia of SCADA communities of interest, despite numerous publications aimed at both management and technical audiences (TISN 2005), and even in spite of the CNVA program, there still appears to be serious issues with recognition of the security issues faced by connecting SCADA networks to the internet. Of concern is that most of the issues reported in this paper are not new, and have been discussed previously in other reports concerning SCADA security (Stamp *et al*, 2003; Fink *et al*, 2006; Igure *et al*, 2006; Stouffer *et al,* 2007). The authors would argue that not only does the CNVA program need to be continued, but that it needs to be expanded. Associated work by the authors has also revealed that hard drives and other computer hardware are being disposed of without being properly erased from CIPs. This adds further weight for a call for all critical infrastructure providers to undergo some form of external computer and network vulnerability assessment or audit on a regular basis.

REFERENCES

Fink, R.K., Spencer, D.F. & Wells, R.A. (2006). Lessons learned form cyber security assessments of SCADA and energy management systems. Retrieved 14th October 2008 from http://www.inl.gov/scada/publications/d/nstb_lessons_learned_from_cyber_security_assessments.pdf

Igure, V.M., Laughter, S.A. & Williams, R.D. (2006). Security issues in SCADA networks. *Computers and Security*. **25(7):** 498-506

Jackson-Higgins, K. (2007). SCADA state of denial. Retrieved 10th October from http://seclists.org/isn/2007/Apr/0068.html

Krebs, B. (2008). Cyber incident blamed for nuclear power plant shutdown. Retrieved 10th October from http://www.washingtonpost.com/wp-dyn/content/article/2008/06/05/AR2008060501958.html

Nakashima, E. & Mufson, S. (2008). Hackers Have Attacked Foreign Utilities, CIA Analyst Says. Retrieved 11th October 2008 from http://www.washingtonpost.com/wpdyn/ content/article/2008/01/18/AR2008011803277.html

Poulsen, K. (2003). Slammer worm crashed Ohio nuke plant network. Retrieved 10th October 2008 from http://www.securityfocus.com/news/6767

Rishardson, R. (2008). CSI Computer crime and security survey. Retrieved 14th October 2008 from http://i.cmpnet.com/v2.gocsi.com/pdf/CSIsurvey2008.pdf

Smith, T. (2001). Hacker jailed for revenge sewage attacks. Retrieved 10th October 2008 from http://www.theregister.co.uk/2001/10/31/hacker_jailed_for_revenge_sewage/

Stamp, J., Dillinger, J., Young, W. & DePoy, J. (2003). Common vulnerabilities in critical infrastructure control systems. Retrieved 14th October 2008 from http://www.sandia.gov/scada/documents/031172C.pdf

Stouffer, K., Falco, J. & Scarfone, K. (2007). *Guide to industrial control systems (ICS) security*. USA: NIST Trusted Information Sharing Network (2005). SCADA Security – Advice for CEOs. Retrieved 10th October 2008 from http://www.tisn.gov.au/www/tisn/rwpattach.nsf/VAP/(427A90835BD17F8C477D6585272A27DB)~Supervisory_Control+and+Data+Acquisition+(+SCADA+)+-+Security+Advice+for+CEOs.pdf/$file/Supervisory_Control+and+Data+Acquisition+(+SCADA+)+-+Security+Advice+for+CEOs.pdf

Trusted Information Sharing Network (2008). *Computer Network Vulnerability Assessment Program*. Retrieved 11th October from http://www.tisn.gov.au/www/tisn/tisn.nsf/Page/CIPPrograms_ComputerNetworkVulnerabilityAssessment(CNVA)Program

COPYRIGHT

The 2008 Australian study of remnant data contained on 2nd hand hard disks: the saga continues

Craig Valli
SECAU – Security Research Centre
Edith Cowan University
c.valli@ecu.edu.au

Andrew Woodward
SECAU – Security Research Centre
Edith Cowan University
a.woodward@ecu.edu.au

Abstract

This study looked for remnant data on enterprise level hard drives that were purchased through auctions. The drives were analysed for information, be it topical or formatted. In the event that drives were formatted, forensic tools were used to recover this data. This years study revealed a high level of not simply un-erased drives, but drives which contained information that related to critical infrastructure providers. That such a small sample size yielded such a high rate of un-erased drives is of considerable concern, and it may be necessary for the government to become involved.

Keywords

Hard disks, forensics, erasure, enterprise drives

INTRODUCTION

The issue of remnant data on electronic storage devices, and hard drives in particular, is one that has been covered by a variety of researchers over a number of years. The authors first started examining hard drives for remnant data on second hand drives in 2004 (Valli 2004), with the study continued in subsequent years (Valli & Jones, 2005; Jones *et al*, 2006; Valli & Woodward, 2007). Whilst other storage media has the potential to contain remnant data, hard drives have the greatest potential to contain larger amounts of information. The reason for this statement is that with such large capacities available for a historically low price (approximately 0.015 cents / GB), there are many of large drives making their way on to the second hand market. The other issue is that to securely erase large drives takes a considerable amount of time (Valli & Patak, 2005).

This year saw the Australian study focused heavily on enterprise level small computer systems interface (SCSI) hard drives, with 2.5" notebook hard drives also examined. This bias was done with the intent of building a profile of large corporate and government sources, and this was again achieved as an outcome. The sources for the hard disk drives were a mixture of national on-line auctions and also traditional face to face auctions in the metropolitan area of Perth, Western Australia. This year 48 cases were examined. It should be noted that several of these were large RAID configurations.

The research on the disks was again undertaken using only tools that it was considered would be available to anyone who had obtained such disks. The series of actions were again an escalation of expertise in digital forensics. Should the hard disk not boot in suitable hardware, tools were used which carried out the same functions as the Windows Unformat and Undelete commands. In these cases a hex editor was also used to view any information that existed in the unallocated portion of the disk. For the extended analysis a customised Linux Debian install was used. In particular the scalpel file carver (Foremost 2007) and Autopsy the forensic browser (Sleuthkit 2006) were used in the analysis phase of disks that had been formatted.

As with previous years (Valli, 2004; Valli & Jones, 2005; Jones *et al,* 2006; Valli & Woodward, 2007), the first objective of the research was to determine whether there was any information on the disk that was readily recoverable with the techniques and tools identified above. The second stage of the research was to look for specific elements of information that would allow for the identification of the organisation or individual that had used the particular disk. Further, and, if possible, information such as the usernames, email addresses or documents, spreadsheets and data types of interest were examined. The purpose of this phase of the research was to determine the proportion of the disks that could be traced to an organisation or an individual and the level of exposure that data recovered would represent. The level of exposure looked not only at the privacy vector but also exposure that would allow the ready commissioning of criminal acts against the previous hard disk owner.

OVERALL TRENDS UNCOVERED IN THE RESEARCH

Overall in the study once again a lack of pornographic material was uncovered given the hyperbole and claims made by some parties with respect to the use of the Internet to access this material during work hours is cast further into doubt. Unlike the 2006 study, there were no detected cases of child pornography in any of the analysed hard disks from Australia.

In 2008, a high percentage of the examined hard disks that yielded data, again contained significant personal and corporate exposure of confidential and commercial information. This phenomenon is a disturbing consistent trend in the studies so far.

This year, however, did see a slight increase in the amount of competently erased hard disks. This occurrence combined with the same observation in last years study that many of the hard disks were acquired from organisations with the ability and resources to erase the hard disks is gaining veracity. Potentially also with many of the disks coming from servers, there may be a recognised need to erase these before disposal. Drives purchased from what seemed to be low trade individuals were more likely to have not formatted the drive, or to have formatted but not erased.

2008004-AU 1.8 Terabyte RAID from a chemical supply company. There was the entire catalogue and ordering systems for the company. It also included a complete website for the company. The dates on the drive were less than 3 weeks old. There was also business documents, spreadsheets and other data you would expect to find in a medium sized enterprise.

2008013-AU This case contained 2 x 36GB SCSI drives that were extracted from a Sun Server. The server was sold as is and was advertised as being unable to boot. The drives were extracted and imaged. In the meantime it was possible to resurrect the server and boot it with other hard disk media. The original disks although imaged had not been examined they were placed back in the server, the server booted to a Solaris login prompt. The subsequent forensic analysis revealed that the drives were from a superannuation board as per the asset stickers that were on the external case of the server. This server yielded administrator passwords, configuration files and also network details.

2008015-AU This case was a 4 x 36GB RAID. There was an identifiable World of Warcraft avatar on the machine. There was a large number of pornographic thumbnails but they were all 100 x 100 GIF files, no other substantive files of this type were found. The thumbnails were predominately from mature age porn sites. There were no documents able to be located on the drive.

2008020-023-AU These four 72GB 68 pin SCSI drives were all unformatted and appeared to have originally belonged to a mining company. Analysis of the drives indicated that they were separate SCADA masters. As such they contained information including, the Historian database, iFix and other SCADA components relevant to the companies process control. In addition, passwords and usernames, including administrator, were still present on the drives. There were also photos of the site and personnel present.

2008026-028-AU These three 72GB SCSI drives had belonged to an organisation that provided IT support for Australian and international banks, as well as network services for other critical infrastructure providers. The drives had been formatted, but not erased and forensic recovery yielded a large range of information. Remote access policies and procedures for staff to access the banks networks, and network diagrams were amongst the information uncovered. Names of staff and what type of equipment they had been issued were also present.

2008030-AU This 146 GB SCSI drive had been formatted, but the data was recoverable. The drive was a Windows Server, and appeared to belong to a crop protection company. The server appeared to have been used for storing voicemail, and may also have been used for VoIP. Registry hives and other documentation relating to the companies business were also recovered.

2008036-AU This 72GB SCSI drive had been formatted but data was recoverable. The drive appeared to be a Windows server domain controller from a community health centre. It contained minutes from meetings and some other documentation. It also contained illegal software keys.

2008037-AU This 72GB SCSI drive had been formatted but data was recoverable. The drive belonged to a Nursing home, and contained extensive information. Patient information, letters from medical doctors, drug information, pictures of patients and their wounds, menus, accounts, minutes of meetings were a sample of the information recoverable

POSSIBLE CAUSES

There is little chance that IT professionals could be unaware of the dangers of remnant data given the extensive coverage that this topic has now received. The availability of free and competent erasure tools such as DBAN also removes any financial impediment to acquisition of appropriate tools to accomplish erasure of media.

It is of considerable concern that so many enterprise level hard disks still contained recoverable data. Of greater concern is that some of this data was essential to the companies operations, and if made public could lead to significant civil and even criminal ramifications for the company. This year uncovered a set of drives that had a complete SCADA setup for a mine processing plant. This sort of exposure has several serious implications for security beyond the simple disclosure of the information or potential for theft. The intelligence found on the drive could make it a relatively trivial task to attack the plant that could result in a release of toxic chemicals or permanent destruction of the environment. From an economic perspective, destruction or partial destruction of the asset via deliberate malfeasance could be significant and widespread.

In the 2007 study, Valli and Woodward, postulated that the drives may have been purloined from organisations and on sold through the auction systems. The authors still contend that this maybe an option still being exercised by disgruntled employees this however does not explain the significant exposures uncovered. It may also be the case that organisations are relying, either implicitly or explicitly, on disposal companies / individuals to erase the drives for them. If this is the case, then it may be a failure in the processes of the disposer that has lead to this problem being so significant.

This year again it is clear that public and private organisations across a range of industry sectors are failing to discharge their responsibility to protect customer’s details and sensitive data adequately. There is still no official mandated law or statute that requires organisations in Australia to erase secondary memory devices such as hard disk drives and this problem is growing. Evidence of inadequate protection uncovered in this and previous studies surely should now see this being considered by governments to protect citizens from themselves. Failing a legislated approach to the problem the creation for government based organisations of a centralised clearance service for the destruction/safe erasure of secondary storage media is needed.

It could also be safely deduced that organisations are failing to conduct adequate risk analysis of the issue of remnant data on secondary memory devices. It could not be similarly reasonably argued that the problem of remnant data is not a new or unknown phenomenon (Anonymous, 2003; de Paula, 2004; Duvall, 2003; Garfinkel & Shelat, 2003; Jones, 2006; Rohan, 2002; Spring, 2003; Valli, 2004; Valli & Jones, 2005; Valli & Patak, 2005). There are now a wide range of devices that are standard artifices of modern business these include, organisational servers, desktops, laptops, mobile phones, USB memory sticks, USB hard drives, even iPOD and MP3 player devices.

The risk versus return equation is simply not making sense for any modern organisation or individual who storage technologies. Auctioneers and sellers of these hard disks are also unwittingly providing potential criminals with targeted options for purchase. Many of the sellers have advertising that clearly indicates that the devices are from financial institutions, superannuation boards, insurance companies, mining companies to name a few. One has to ask what the provenance of these devices has to do with their suitability or value for use as storage mechanisms. There is some argument that these types of organisations drives may have been looked after in clean, regulated environments but that is where it would logically end.

CONCLUSION

Similar to last years study, this year has again uncovered significant exposure of private, sensitive or fungible data on inadequately erased secondary memory devices. Organisations spend millions on protecting IT assets annually with firewalls, virus protection, intrusion prevention systems and other security countermeasures. They conduct regular audits of assets and procedures in an effort to protect and secure the information contained on their systems. We argue that these expenditures are largely symbolic and almost farcical when hard disks are disposed of without adequate protections.

The recovery of hard drives that contained complete process control system data and information is of considerable concern, as there are potentially lives at risk if the process control system is interfered with. The Australian Federal Government has the computer and network vulnerability assessment (CNVA) program and other initiatives and education programs in place to prevent such an event happening, which leads the authors to ponder as to whether the company in question had availed themselves of this program. The small office, home office (SOHO) users can be excused for disposing of drives that had been simply formatted, and not erased, but there is absolutely no excuse for multinationals to be disposing of unformatted hard drives with critical infrastructure information contained within. Possible legislation may include requiring critical infrastructure providers to undergo an external annual security audit to make sure that policies and standards are being adhered to.

This years study has revealed that not only is sensitive information being sold on the cheap, but so potentially are lives or the environment. The most expensive hard disk purchased in this years study was a 73GB SCSI hard drive that cost $85, with most costing less than that. The drives belonging to a mining company were purchased for just over $60, making them less than $1 per gigabyte. I don't see companies asking "Would you like private medical data, process control systems information, or network diagrams with that?", yet this is exactly what is still occurring today somewhere at an IT disposal sale or online auction in Australia.

In closing this study has been going in one form or another since 2004 it is now 2008 and still we are finding drives with significant exposures and realisable threats to individuals and the community at large. Some onus has to be placed on operating systems providers to enable a disk erasure routine for safe disposal of equipment. This study, has added further weight to the body of literature on the problem of remnant data. There is a real need for governments and in particular Australian governments to legislate and require organisations and custodians of electronic data to expunge it before disposal of the physical asset.

REFERENCES

Foremost (2007) Foremost, retrieved 25th October 2008 from http://foremost.sourceforge.net/

Jones, A., Valli, C., Sutherland, I. and Thomas, P (2006). The 2006 Analysis of Information Remaining on Disks Offered for Sale on the Second Hand Market. *Journal of Digital Forensics, Security and Law, 1*(3), 23-36.

Sleuthkit (2006). Autopsy overview, retrieved 25th October 2008 from http://www.sleuthkit.org/autopsy/

Valli, C. (2004). *Throwing the Enterprise out with the Hard Disk.* In Proceedings of the 2nd Australian Computer, Information and Network Forensics Conference, Fremantle, Western Australia.

Valli, C., & Jones, A. (2005). *A UK and Australian Study of Hard Disk Disposal.* In Proceedings of the 3rd Australian Computer, Information and Network Forensics Conference, Edith Cowan University, Perth, Western Australia.

Valli, C., & Patak, P. (2005). *An investigation into the efficiency of forensic erasure tools for hard disk mechanisms.* In Proceedings of the 3rd Australian Computer, Information and Network Forensics Conference, Edith Cowan University, Perth, Western Australia.

Valli, C. & Woodward, A. (2007). Oops they did it again: The 2007 Australian study of remnant data contained on 2nd hand hard disks. In Proceedings of the 5th Australian Digital Forensics Conference, Edith Cowan University, Perth, Western Australia.

COPYRIGHT