An Introduction to Digital Forensics

Article

3 authors, including:

John P. Abraham
University of Texas Rio Grande Valley
15 PUBLICATIONS **82** CITATIONS

SEE PROFILE

An Introduction to Digital Forensics

Irma Resendez, Pablo Martinez, and John Abraham
University of Texas Pan American
jabraham@utpa.edu

Abstract

As new technologies develop criminals find ways to apply these technologies to commit crimes. With the explosion of web technologies almost all major businesses in the world have web presence thus exposing their data to legitimate and illegitimate users. Computers have become intrinsic part of our lives. Businesses have streamlined their operation saving millions of dollars because of the web technologies and services. Neither the businesses nor the consumers can live without these technologies. Because of the intricate involvement of computer technology in all aspects of our lives, it also has become legal evidence in both civil and criminal cases.

Computer evidences admitted in courts could be any file or fragment recovered from the storage devices such as email, browsing history, graphics, photographs, or application documents. These files may be undeleted or deleted. Deleted file recovery would require special techniques. Computer professionals trained in digital forensics preserve and retrieve evidence in a non-destructive manner. Evidence may be recovered from any storage medium installed in digital equipment such as computers, cameras, PDAs, or cell phones. All forensic work should be done with care including documenting clear chain of custody in order for the evidence to be admissive in a court of law.

Introduction

With the proliferation of computers in our everyday lives, the need to include computer contents or traces as part of formal evidence has become inevitable. Computerized devices are part of our world in the form of laptops, desktop computers, servers, etc., but there are also many other storage devices that may contain forensic evidence. Devices such as memory cards, personal digital assistants, and video gaming systems, are among a myriad of devices that have the ability to accept input, provide output, and also store data. It is this data or the usage of these devices that is at the center of computer forensics.

According to Marcella and Menendez, cyber forensics, e-discovery, digital, forensics, computer and computer forensics mean relatively the same thing yet none has emerged as a defacto standard (Marcella, Menendez 2008). They further present a working definition of computer forensics as "the science of locating, extracting, and analyzing types of data from different devices, which specialists then interpret to server as legal evidence (p5)." They further state that computer forensics can also be defined as "the discipline that combines elements of law and computer science to collect and analyze data from computer systems, networks, wireless communications, and storage devices in a way that is admissible as evidence in a court of law (Marcella, Menendez 2008)." Computer

forensics aims to attain as much information from electronic devices or media by utilizing sound forensic techniques that may be admissible in court. This includes concise and sound forensic techniques including a clear chain of custody and documentation. There are two different areas that must be considered when collecting digital forensic evidence. The first is the process of collecting the evidence without altering its contents and ensuring it is admissible in court. The second area is the actual use of law enforcement grade sound forensic practices that results in the collection being admissible in court. It is not the intent of this document to address the latter. As the authors are computer science professionals, it is the intent of this document to provide a concise overview of the process of collecting forensic evidence and review different methods, tools, and challenges during forensic analysis and collection.

Techniques – Live Incident response

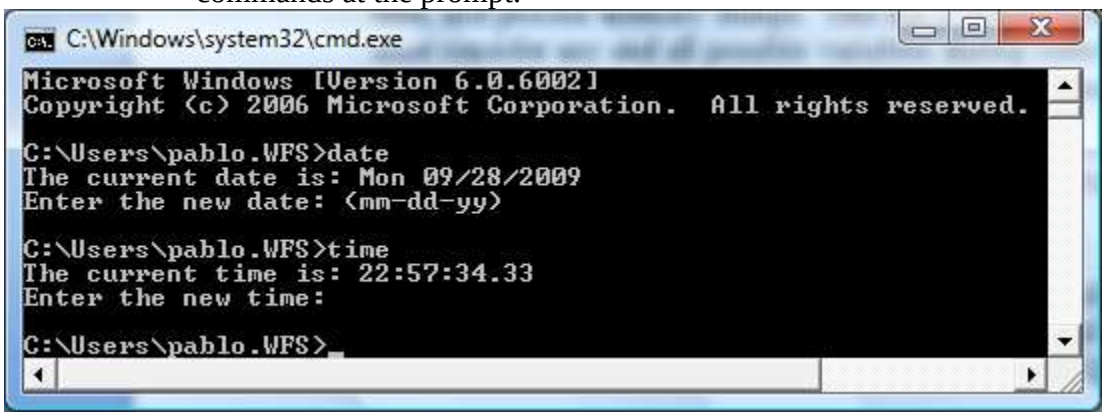
Live incident response collects all of the relevant data from the system that will be used to confirm whether an incident occurred. Live incident response includes collecting volatile and nonvolatile data. Volatile data is information we would lose if we walked up to a device and disconnected the power cord. Nonvolatile data includes data that would be very useful to collect during digital forensic collection such as system event logs, user logons, and patch levels, among many others. Live incident response further includes the collection of information such as current network connections, running processes, and information about open files. Live incident response is typically collected by running a series of commands that produces data that would normally be sent to the console but is sent to a different storage device or a forensic workstation. A forensic workstation is a machine that the forensic investigator considers trusted and is used to collect data from a suspected host computer. There are various ways to transmit the data to the forensic workstation during forensic collection. Some of the most common tools to collect data include commercial software such as EnCase and FTK (forensic toolkit from accessdata.com) but there are many other open source and commercial tools that can be part of our arsenal during collection.

Some of the most common tools include Netcat and Cryptcat. Netcat is the considered by many as the Swiss army knife of live incident response collections. Netcat crates TCP channels can be executed in listening mode like a telnet server, or in connection mode like a telnet client. Netcat also includes MD5 check sum capability which, as we will cover later, is an essential part of any sound digital collection order to prove the data was not altered during collection. Netcat works by booting to a shell and "listening" on a particular TCP port in verbose or listening mode. Once the forensic workstation is in "listening" mode via Netcat, the data is sent from the target host to the particular port the forensic workstation is listening to. Cryptcat is a variant of Netcat that is worth mentioning in this document as it encrypts all of the data across the TCP channel. It uses all of the same commands and command-line switches as Netcat but enhances Netcat by providing secrecy and authentication—which are now very important in the age of eavesdropping and wire tapping.

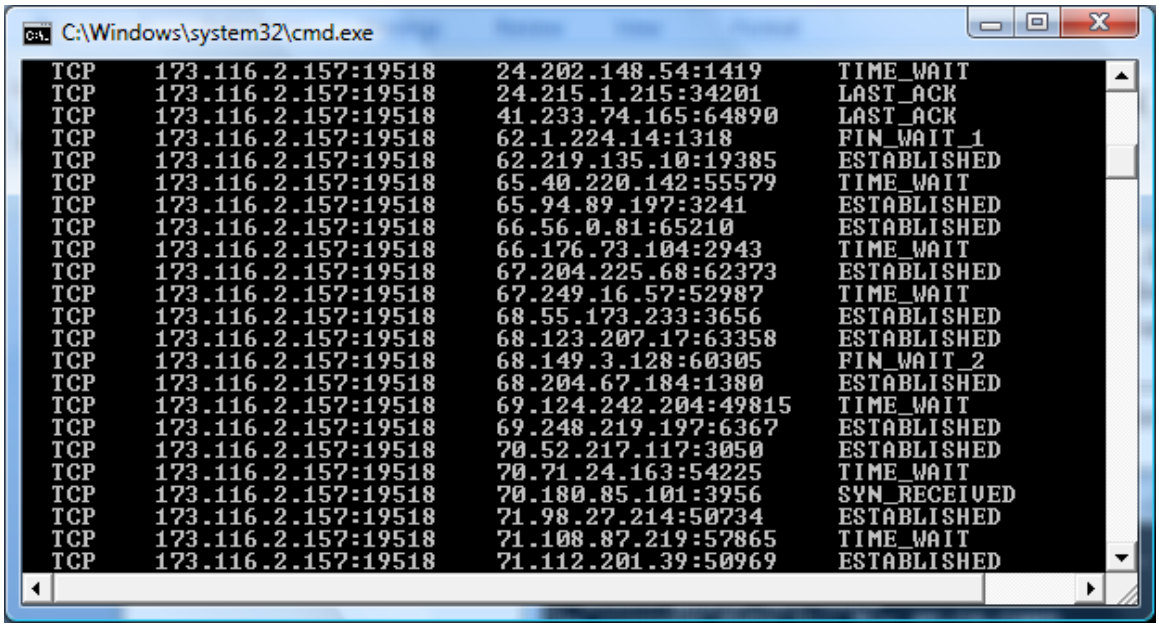
Volatile vs. Nonvolatile data

Some of the volatile data that should be collected includes system date and time, current network connections, open TCP and UDP ports, which executables are opening UDP and TCP ports, cached NETBIOS name table, users currently logged on, the internal routing table, running processes, scheduled jobs, open files, and process memory dumps. This list is not all inclusive as a forensic investigator must consider any and all possible variables during collection. However, one thing that all these have in common is that they would be lost if the power were removed from your target machine. While there are various methods of collecting volatile data, some of the most common and non-commercial methods will be presented:

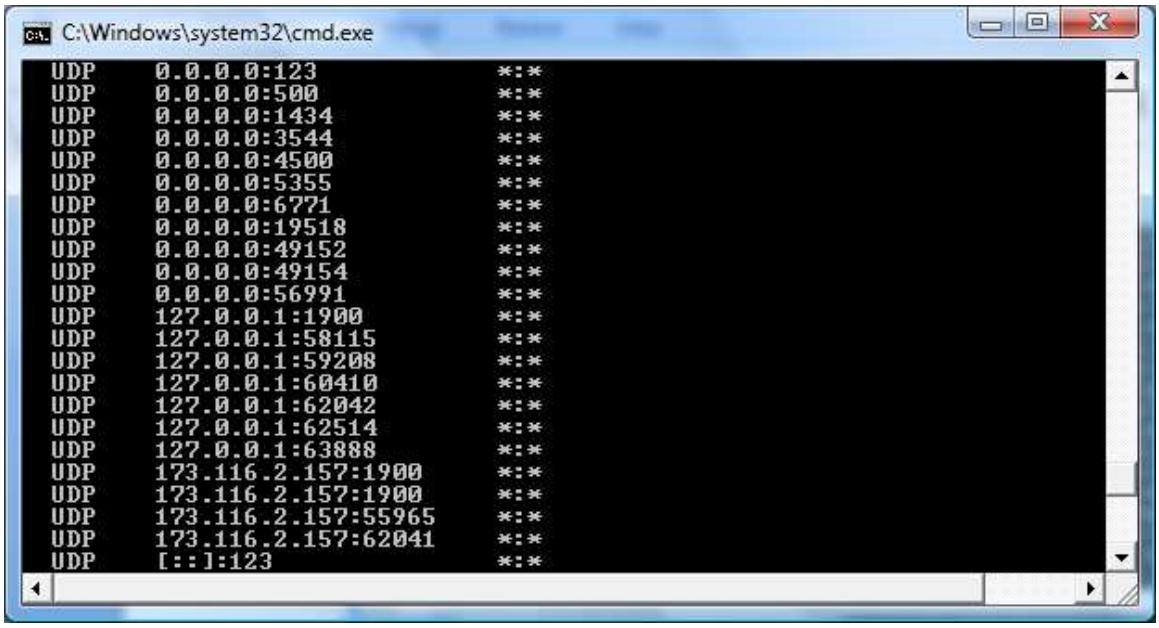
a. System date and time – can be collected by using the *time* and *date* commands at the prompt.



b. Current network connections – can be collected by using the *netstat* command with the –an flags to retrieve all of the network connections and see the *raw* IP addresses instead of the FQDN.

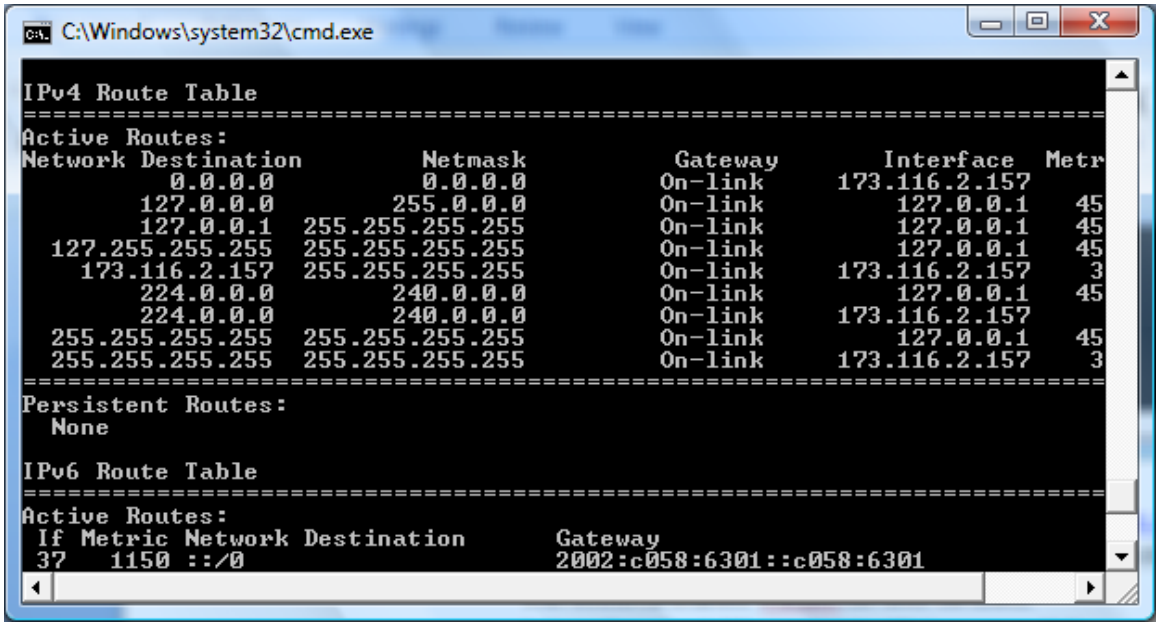


c. Open TCP and UDP ports – use netstat –an. Also use **FPort** tool (www.foundstone.com) in order to link the open ports to the executables that opened them.



d. Users currently logged on – run **PsLoggedOn** from www.sysinternals.com. This tool will return the users that are currently logged onto the system or accessing the resource shares. **PsExec** can also be used.

e. Internal routing table – netstat with the –rn switch.



f. Running processes – it is important to identify running processes as they could contain backdoors. Use **pslist** from PsTools suite.
g. Running Services – use *PsService* from PsTools suite.
h. Scheduled jobs – run *at* command at command prompt
i. Open files – run *Psfile* from pstools
j. Process Memory dumps – process memory dumps may provide critical investigative material of a volatile nature. These include Command line utilized by the intruder and remotely executed console commands and their resultant output. Memory dumps can be captured using *userdump.exe* provided by windows. It is available from Microsoft at: http://www.microsoft.com/downloads/details.aspx?FamilyID=E089CA41-6A87-40C8-BF69-28AC08570B7E&displaylang=en&displaylang=en

Nonvolatile data includes information we would like to acquire including system version and patch level, file system time and date stamps, registry data, the auditing policy, a history of logins, system event logs, user accounts, IIS logs, and any suspicion files. These can be acquired using the following methods:

a. System Version and Patch level – knowing which patches have been applied to a server or workstation will enable us to narrow our initial investigation to areas of high probability. Use *PSInfo* from *pstools* to query system information. Switches include –h to show installed hotfixes, -s to show installed software, and –d to show disk volume information.
b. File System Time and Date stamps (pg 31) – UnxUtils from http://sourceforge.net
c. Registry Data – Leads we want to capture include programs executed on bootup and entries created by intruder's tools. We use *RegDmp* to capture the complete registry. http://www.softpanorama.org/unixification/registry/microsoft_registry_tools.shtm#RegDmp
d. The Auditing Policy - Windows has auditing disabled by default. Use *auditpol* included in Microsoft resource kits
e. System event logs – Psloglist from sysinternals
f. User accounts – use *pwdump* utility to dump user accounts. http://foofus.net/fizzgig/pwdump
g. IIS Logs
    ii. Most attacks happen over port 80
    iii. You cannot block what you must allow in
    iv. C:\winnt\system32\logfiles – IIS logs

UNIX Live Response

Any forensic investigator should be prepared to encounter non-windows operating systems such as DOS, Linux, and UNIX. This section will concentrate on UNIX live response. In order to collect volatile data, we can utilize the following commands during a UNIX live response:

a. System date and time – *date*
b. Current network connections – *netstat –an*
c. Open files – *Lsof* lists open files, displays the files that the process has open and using *lsof-n* lists raw IP addresses.
d. Running Processes – *ps-aux* will display the open processes
e. Internal routing table – *netstat –rn*
f. Mounted file systems – *mount, df*

Nonvolatile data during UNIX live response is geared at collecting similar information as previously listed for Microsoft Windows:

a. System version and patch level – *uname –a*. In Linux and UNIX the patch level is difficult to retrieve because it is dependent on the Linux distribution used to create the system.
b. Users Currently Logged On - Users currently logged on are saved on: **/var/run/utmp** log.
c. A history of Logins - Saved in the **var/log/wtmp** binary log.
d. System logs - In Unix, the syslog daemon listens for messages from either local programs or other servers on the Internet and logs them according to the **/etc/syslog.conf** configuration file.
    i. In this case, the only relevant logs are **/var/log messages** and **/var/log/secure**.
    ii. Using **Netcat** we transfer the logs to our forensic workstation.
    iii. Logs: contain five columns
        1. Date
        2. Time
        3. Machine name
        4. Process that initiated the event, along with the process number
        5. Message that was logged

Network Based Evidence (NBE)

Investigators collect 4 types of network based evidence including full content data, session data, alert data, and statistical data full content data involves collecting all data transmitted over a wire. It consists of the actual packets, typically including headers and

application information, seen on the wire or in the airwaves. It records every bit present in a data packet. When looking at headers, analysts can identify subtle aspects of packets. The presence of certain options or the values of certain fields can be used to identify operating systems. Full content data presents the greatest opportunity for analysis, but it also requires the greatest amount of work.

I. Session Data
    a. Includes data that was created or modified during a particular user's session.

II. Alert Data
    a. Alert data is created by analyzing NBE for predefined items of interest.
    b. Example: a client using IP address 192.168.0.42 has queried the remote procedure call (RPC) port mapper on a server, 192.168.0.40. This could be the precursor to an attack because an intruder could use the information to identify vulnerable RPC services on the server. The alert is a judgment made by the network IDS (intrusion detection) that the packet it saw is a query of the port mapper service.

III. Statistical data
    a. Provides a big picture perspective
    b. In the data world, statistical data has traditionally been used to measure the health and performance of a network.

Common Forensic Analysis Techniques

When you conduct forensic analysis, there are a few steps that must be executed in nearly every type of investigation to prepare the data for analysis. For instance, it is usually recommended to recover any deleted files and add them to the analysis. It is also advantageous to reduce the data set collected to the smallest number of files and add them to the analysis for efficient review. Another step that should be incorporated into the analysis is string searching to identify relevant files and fragments of relevant files.

Recovering deleted files is of crucial importance. Particularly in cases where the target computer in question has been utilized by a savvy computer user or a suspect who may have wanted to delete traces of their digital footprints. In order to avoid doing work twice, recovering deleted files should be done first. There are several different options and solutions that provide similar results which will be listed below. It is worth mentioning that at the time of this writing, the authors of this document tested all solutions presented here and discovered an interesting and extremely important fact that

should be of great importance during collection particularly when collection and recovered data is to be presented as legal evidence. Although commercial software was the easiest to use, it was unable to recover all deleted files. In some instances, utilizing open source tools produced the recovery of additional deleted files after a first pass with a commercial product. During our forensic tests, it was also found that in every instance, commercial software was able to recover most of the deleted files when compared to open source tools. It is then concluded that the best way to undelete the most data from a target host is to utilize a combination of commercial and open source tools. The tools tested at the time of this writing are the following:

I. EnCase & FDK– Windows-based commercial products that seems to be dominating the forensic market at the time of this writing. The main advantage to recover undeleted files using these tools is that they do not have to be installed on the target host. They can search and undelete files based on a forensic duplication (clone of the target host's hard drive).

II. Linux Kernel – it is an open source tool that enables us to take a forensic duplication and make it act like a real hard drive under Linux.
   a. ftp://ftp.hp.nasa.gov/pub/ig/ccd/enhanced_loopback
   b. Once the Linux kernel has been loaded and it recognizes our forensic duplication as a mounted volume, we need to recover the deleted files. The most used tool used to be *The Corner's Toolkit* t http://www.fish.com/tct but the limitation that it only recovers files from Windows made it less popular. The most popular at the time of this writing is *TASK* and *The Sleuth Kit* at: http://www.sleuthkit.org.

Production of Time Stamps and Other Metadata for Files

After undeleting files, the next step in forensic analysis is attaining metadata and time stamp of files. Metadata includes full title names, file sizes, MD5 hashes, among others. Commercial solutions such as EnCase and FDK produce metadata information by default. In lieu of commercial solutions, there are open source tools such as GNU find that produce the same information.

After attaining metadata, our next chore is attaining the MD5 checksum (**define**). As in the case of metadata, commercial solutions produce the MD5 checksum by default and there is little or no intervention from the forensic analyst. There are also open source solutions one of which is the *fls* command included in the *Sleuth kit*.

Removing known files

All of the aforementioned steps may include thousands of files that we may need to investigate. There are usually thousands of files associated with both operating systems and application programs that, upon an integrity check by the forensic analyst, can be removed in order to alleviate the load in the forensic analyst. Since during collection we don't know if the files we collect from alleged application programs and operating systems are of trusted origin, we must compare the MD5 hashes of every file against a known good installation (usually part of a forensic analyst's arsenal) and good set of hashes. If malicious code or an altered file exists masqueraded as a legitimate operating system file, comparing the suspect file against the legitimate file's MD5 checksum will produce a red flag.

In order to attain a reputable source of files, we can either install the operating system ourselves in a controlled environment or use somebody else's work. Installing the operating systems may present a great task due to the increasingly large number of operating systems. An alternative is using a list of MD5 hashes that may be available for this purpose. A well known source of such information is the National Software Reference Library (NSLR) provided by the National Institute of Standards and Technology (NIST) at http://www.nsrl.nist.gov. Comparing the MD5 hashes using opens source solutions involves tedious manual work including removing quotations from all file names, comparing lists, and writing a script (usually a Pearl script) to compare and identify the anomalies. Commercial solutions such as EnCase have the ability to import existing hash sets such as the ones provided by NSLR with very little user effort.

Web Browsing Activity Reconstruction

The last piece of forensic analysis that will be covered is the collection of web browsing activity and its reconstruction. In this area, nothing beats commercial solutions. EnCase, FTK and IE history at http://www.pillipsponder.com provide the necessary tools to collect and reconstruct Web browsing activity.

Conclusion

Although still considered in its infancy, computer forensics has been on the forefront of many recent legal investigations. As we digitize anything from video to live television, digital forensics will not only inevitably become increasingly important to legal investigations, but it may soon become a science with legislated methods. The most important conclusion from this document should be the fact that there are no tools that

are infallible. It is the job of a computer forensics scientist to utilize different tools in order to recover and capture digital evidence in a forensically sound manner.

References

Marcella, Albert J, Menendez, Doug (2008). *Cyber Forensics*. Boca Raton, FL: Auerbach Publications Taylor & Francis Group.

Gialanella, David. (2008) New Tech, Old Problem. *ABA Journal, 94(8),* 35.