Implementation of Anti-Forensic Mechanisms and Testing with Forensic Methods

GRADUATE PROJECT REPORT

Submitted to the Faculty of
the Department of Computing Sciences
Texas A&M University-Corpus Christi
Corpus Christi, Texas

in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Computer Science

by

Srilakshmi Erasani
Fall 2010

Committee Members

Dr. Mario Garcia ______________________________
Committee Chairperson

Dr. Hongyu Guo ______________________________
Committee Member

Dr. Longzhuang Li ______________________________
Committee Member
ABSTRACT

The world’s dependence on digitally saved information has been significantly affecting the system in which every person stores data. Many criminal activities like theft, espionage along with the people who commit them have taken distinctive approaches. Anti-Forensics is the use of tools, methods, and processes that obstruct the forensic retrieval of evidence.

Being informed and understanding the goals and working models of anti-forensic tools, investigators can become better educated and more aware of the challenges and opportunities facing them, improving their success and continuing the advancement of the forensics industry. In this project, an investigation of the traditional anti forensic techniques is conducted such as erasing file systems, masking and disk wiping utilities by building an anti forensic prototype and also present a survey of few of the powerful anti forensic tools. In addition, a study on the technological and ethical challenges related to the mechanism of some of the many anti-forensic tools is conducted. Finally, a detailed analysis and evaluation is conducted on the effectiveness of anti forensic tools for defeating various computer forensic tools and present strategies for detection and countermeasures for future forensic investigation procedures.
# TABLE OF CONTENTS

Abstract........................................................................................................................................ii

Table of Contents ..........................................................................................................................iii

List of Figures...............................................................................................................................v

List of Tables.................................................................................................................................viii

1. Background and Rationale. ......................................................................................................1
   1.1 Computer Forensics and Anti-Forensics .................................................................1
   1.2 Traditional Anti-Forensics..........................................................................................2
      1.2.1 Overwriting Data and Metadata ...............................................................3
      1.2.2 Cryptography, Steganography & Other Data Hiding Approaches...4
      1.2.3 Attacks against Computer Forensics ..................................................6
   1.3 History............................................................................................................................7
      1.3.1 Background and Related Work .............................................................7
   1.4 Objective.........................................................................................................................8
   1.5 Rationale.........................................................................................................................9

2. Narrative..................................................................................................................................11
   2.1 Problems from Investigators Approach...............................................................11
      2.1.1 Scope...............................................................................................................12
   2.2 Functionalities of Anti-Forensic Tools.................................................................12
   2.3 Reducing the Methods of Anti-Forensic Methods........................................12

3. Proposed System Design .......................................................................................................15
   3.1 Framework.....................................................................................................................15
LIST OF FIGURES

Figure 3.1 Framework Supported for the Application and the Targets of Use ........16
Figure 3.2 Prototype of application developed ..............................................18
Figure 4.1 Interface of Discoff .................................................................24
Figure 4.2 User data cleaner for showing found files ....................................24
Figure 4.3 User data cleaner for viewing found files ....................................25
Figure 4.4 User data cleaner for clearing found files ....................................26
Figure 4.5 User data cleaner deleting process .............................................27
Figure 4.6 View of files that have not been deleted .....................................28
Figure 4.7 Recovery process using FTK .....................................................29
Figure 4.8 Results from the FTK after recovery process ...............................30
Figure 4.9: Data cleaning facilities by Evidence Blaster ...............................32
Figure 4.10: Data cleaning facilities by Evidence Blaster .............................33
Figure 4.11: Recovery Process using FTK ..................................................33
Figure 4.12: Error while recovering data by FTK .......................................34
Figure 4.13: Results displayed by FTK after recovery process ......................35
Figure 4.14: Snapshot of Tracks Eraser Pro ...............................................36
Figure 4.15: Successful deletion of Tracks Eraser Pro .................................37
Figure 4.16: Log consisting summary of deleted files .................................38
Figure 4.17: Recovery Process using FTK ..................................................38
Figure 4.18: FTK showing no traces of deleted data ..................................39
Figure 4.19: Snapshot of Discoff Application ..............................................41
Figure 4.20: Tree-list view of drives, folders and files .................................42
Figure 4.21: Deletion Process of Portable Drive by Discoff .........................43
Figure 4.22: Recovery Process using FTK.............................................44
Figure 4.23: Indexes in FTK..............................................................44
Figure 4.24: Results from the recovery process by FTK.............................45
Figure 4.25: Snapshot of Secure Clean..............................................46
Figure 4.26: Secure Clean interface after dropping the folders.....................46
Figure 4.27: Secure Zap Confirmation..................................................47
Figure 4.28: Deletion Process of Secure Clean......................................47
Figure 4.29: Error reports for unsuccessful deletion..................................48
Figure 4.30: FTK Recovery Process for Secure Clean Zapper......................49
Figure 4.31: Adding evidence to the case.............................................49
Figure 4.32: Results from FTK on Secure Clean Zapper...........................50
Figure 4.33: Eraser Interface............................................................51
Figure 4.34: Interface to select files and folders.....................................52
Figure 4.35: Task Properties pane for scheduling dynamically.....................52
Figure 4.36: Deletion Process by Eraser..............................................53
Figure 4.37: Processes Queue for deletion..........................................54
Figure 4.38: Indexes in FTK recovery process......................................54
Figure 4.39: Results by FTK on deleted data by Eraser.............................55
Figure 4.40: Discoff with File cloner functionality..................................56
Figure 4.41: Discoff illustrating cloning process of 4MB file.......................57
Figure 4.42: Space overwritten by discoff cloner..................................58
Figure 4.43: File cloner depicting 4000 files.......................................59
Figure 4.44: Self deletion process implemented by discoff…………………………..60
Figure 4.45: Quit process for self deletion in Discoff………………………………...60
Figure 4.46: Results after self-deletion process……………………………………….61
LIST OF TABLES

Table 1.1 Data Hiding File System Structure .........................................................5
Table 2.1 Various Exploitations under Anti-Forensic Methods ..............................13
Table 4.1 Test cases for HDD Analysis.................................................................22
Table 4.2 Test cases for Portable drive analysis...................................................40
Table 5.1 Results on Hard Disk Drive Testing......................................................62
Table 5.2 Results on Portable Drive Testing.......................................................63
1. BACKGROUND AND RATIONALE

1.1 Computer Forensics and Anti Forensics

Computer forensics is an active topic of research, with areas of study including wireless forensics, network security and cyber investigations. The goal of the computer forensics is to provide information about how the crime happened, why and who is involved in the crime in any legal proceeding by using the computer forensic tools. Much of computer forensics is focused on the tools and techniques used by investigators that deal with the preservation, identification, extraction and documentation of computer evidence. Attacks do occur mostly on the servers because of containing information that is not to be disclosed such as accounts, personal data and other transactions. According to the survey conducted by federal bureau of investigation, there is a huge loss related to computer crimes every year, which pre-dominantly engages data compromising, intrusions and other viruses.

Anti Forensics is a study of techniques and tools that confuse computer forensic tools (CFTs), investigators and any other forensic processes by hiding or destroying the data and meta data. Its primary goal would be to make the evidence acquiring process complex and difficult as possible. Looking into the past from six to nine years the research in anti forensics has been improved tremendously in the terms of both scope and popularity. Surrounding this field of research, anti-forensics can be defined in many ways. One of the most common and accurate definitions among those is said by Dr. Rogers from Purdue University where he makes use of a crime scene [Anti-Forensics
2010]. It is defined as an attempt to cause harm to the evidence that is obtained in the scene and also to make the investigation procedures more complicated to perform.

1.2 Traditional Anti-Forensics

Anti-forensics can be distinguished more as a technology due to its characteristics, procedures, applications and types of attacks. Discussion on the purpose and usage of anti-forensics is an on-going process. The positive side of it might result in advancing the future research of digital forensic procedures and its applications and also in generation stronger tools for investigations. The negative side of it is highly utilized by the criminal to perform malicious attacks on the computer being used or any other networks.

Like any other investigative procedures digital forensics is also restricted by rules of laws, human-effort, cost and time. Taking into consideration the validity of the digital evidence, the federal rules of evidence confirmed that it should be authentic and reliable if not, the evidence is measured as hearsay. These rules also state that any evidence produced from the computer is deemed to be genuine and dependable but monitoring any acts is not authentic if it’s not under any common business practices. This enforces an important constraint in the usage of digital evidence in any legal proceedings under anti forensics.

Depending on the conducted research and study, the techniques involved in anti forensic methods come as a challenge to destruct the evidence obtained either by hiding or eliminating it. Many forensic investigators have seen a considerable rise in the usage
of anti forensic techniques and this is not because of the technical slide from unix to windows rather it’s the ease of operation where a nontechnical person could master in it. Thus depending on the research conducted, the methods involved in this area can be superiorly classified into several categories to easily understand the anti-forensic practices being used currently. These sub-categories are summarized in the following section.

1.2.1 Overwriting information

One common Anti-Forensic technique is to overwrite or destroy pertinent information. The process of overwriting files can be accomplished by disk sanitizers and some of the most common techniques for properly sanitizing include physically destroying the drive, rendering it unusable, degaussing the drive, and overwriting the drive’s data so that recovery is fruitless. Drives can also be sanitized through erasing with the use of tools like Drive Scrubber, Active Kill Disk, and 123 Cleaner [Garfinkel 2007]. These over-writing procedures are usually performed in any of these three ways mentioned below.

1. The complete space of the media can be over-written by the application.
2. The application could only target specific files to be over-written.
3. The application could also specifically target deleted files on the media. Ex: the MFT in windows holds all the files deleted and un-deleted. Hence the application can target one file and overwrite on it.

Depending on the situation, some anti-forensic tools like timestomp can also be utilized by the forensic investigator to analyze the times accessed by the criminal on a specific
location. By using this application, the investigator could possibly know what were accessed at what time relying on the timestamp’s obtained. Even if the criminal erased any information from the disk media, the investigator could know that itself.

1.2.2 Cryptography, Steganography and Other Data Hiding Approaches

Encryption could be the other common system and plays a vital role in overcoming the digital forensic procedures. Current encryption standards make use of complex algorithms and other strong mathematical models that make it difficult to decrypt. Even if the investigator has the manual access to the machine, it makes him difficult to gain access to any encrypted files without the ownership of the key that is used to decrypt it.

Steganography is another Anti-Forensic tool that can be used to hide data. This technique involves hiding the information in any other source from the digital investigations and forensic processes. It is often relied on the abilities of the investigator. Normally Steganography is performed on those areas where an examiner would not perform the investigation procedures. The data or any file is usually embedded into another file such as an audio, video or 3D file without causing much changes to the nature and quality of the destination file. The criminal could also rename the destination files in order to be hidden from the view of the examiner [Garfinkel 2007].

According to Garfinkel there are several Data Hiding File System Structures some of these file systems are described in Table 1.1.
Table 1.1 Data Hiding File System Structure

<table>
<thead>
<tr>
<th>FS Type</th>
<th>Data Hiding Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slacker</td>
<td>Hides data in slack space</td>
</tr>
<tr>
<td>FragFS</td>
<td>Hides in NTFS Master File Table</td>
</tr>
<tr>
<td>RuneFS</td>
<td>Stores data in “bad blocks”</td>
</tr>
<tr>
<td>KY FS</td>
<td>Stores data in directories</td>
</tr>
<tr>
<td>Data Mule FS</td>
<td>Stores in inode reserved space</td>
</tr>
<tr>
<td>Host Protected Areas</td>
<td>Device Configuration Overlay</td>
</tr>
</tbody>
</table>

Some of the other data hiding approaches are described below [Scott 2007].

1. **Disk Cleaning Utilities**: This utility contains procedures to erase information from disks or any other medium. This feature is considered to be less effective because it leaves few operational signatures that some data was deleted. Hence these procedures are not widely utilized. Some of these tools include dban, bcwipe, killdisk, etc.

2. **File Wiping Utilities**: By the name itself, one can judge that these utilities concentrate on specific files. Due to its primary concentration, these operations can be performed in limited amounts of time thereby improving the performance. The timing factor can be considered as an advantage. Likewise, these utilities always need the presence of the user to perform the operation leading to a disadvantage. Some of these tools are aevita wipe and delete, bcwipe, etc [Guttman 1996].

3. **Trail Obfuscation**: The main goal of this functionality is to muddle up the digital forensic processes and sidetrack the investigator performing the actions. One of
those applications includes timestomp which is widely used among the area of anti-forensics. This tool pertains the capability to manipulate the dates and times of creation of files, which mainly confuses an investigator [Yasinac 2001]. The other such tool is transmogrify which pertains the capability to modify the data from the header of a file. By using this tool the user can change from a document file type to jpg file type and thereby this operation could sidetrack the investigator. A sound investigator who is specifically looking for doc file types

1.2.3 Attacks against Computer Forensics

Since many years anti-forensic processes primarily concentrated on destructing the information, hiding it and manipulating any evidence existed in the crime scene. Recently, the new era of anti-forensics started to mainly target the digital forensic tools itself. The tools, which are used in forensic procedures like forensic tool kit, encase and prodiscover are the new targets of anti-forensic applications. The anti-forensic tools that have recently developed took advantage of many aspects like accurate documentation of forensic analysis, its susceptibilities and dependence [Dixon 2005].

The common steps involved in any investigation entail acquiring the information and creating an image without any changes made to the source of the physical evidence. The next step involves finding out the hash value to maintain the integrity of the evidence captured. The generation of hash value is a critical step and usually depends on mathematical models. Hence some of the anti-forensic tools are developed to effectively target the hash value of the evidence such that it manipulates the evidence [Berghel 2007].
1.3 History

In the past, anti-forensics weren’t considered as much because most networks were internal and not open to the public. The processes used in the field of forensics were mainly reserved to such things like law enforcement, secret company information, or passwords [Anti-Forensics 2010]. Since then, the internet has become the basis for information sharing and business acumen. With this new direction and focus, anti-forensics has become a prominent issue in trying to protect networks and industries from outside forces.

Since their development computer operating systems and the resulting applications generate copious amounts of data about their user’s activity. These records increasingly have become valuable sources of evidence and, concomitantly, the focus of investigation and legal discovery [Rogers 2005]. At the same time user awareness has grown, for example users now know that deleting files does not mean obliterating the information they contain. This awareness has spawned demand for counter-forensic software, which developers market as guarding users' privacy and/or protecting them from being penalized for activity on the computer.

1.3.1 Background and Related Work

It becomes very critical to distinguish between commercial anti-forensic packages and other anti-forensic utilities, which are related to attackers. Most of the commercial anti-forensic applications have been developed for Microsoft windows operating system and very few of them are destined for unix/linux platforms [Harris 2006]. The main objective
and purpose of these anti forensic software’s developed could be classified into two groups which are mentioned below.

- Tracking down the position of specific files on a machine thereby involves the domain knowledge of specific operating systems and its file structure.
- The other group is concentrates on destructing or deleting the evidence/data found on a computer. Few of these applications not only focus on deletion but also implements masking functionality. That means the application can effectively overwrite the deleted space by any random values so that any forensic tool could not possibly recover the information.

Initially anti-forensic techniques included common practices of using traditional methods like encryption and data hiding commands. Later on as the worlds dependence on the digitally stored data has greatly affected many civil and criminal activities, the need for new techniques to combat the threats faced by forensic investigators were in high demand [Scott 2007]. The “Test Results” section of this paper discusses further challenges anti-forensic tools face in successfully locating and eliminating targeted data.

1.4 Objective

Digital forensic analysts may find their task complicated by any of more than a dozen commercial software packages designed to irretrievably erase files and records of computer activity. These counter-forensic tools have been used to eliminate evidence in criminal and civil legal proceedings and represent an area of continuing concern for forensic investigators.
The objective of the project is to build a prototype application that behaves like an anti forensic tool and also to validate the accuracy and efficacy by comparing and testing with the other commercial anti forensic tools within windows based environment.

1. Evaluate the vulnerabilities of anti-forensic techniques and reduce the effectiveness of these methods by studying anti-anti-forensic procedures.
2. Address the issues of anti-forensic problems by exploiting wide variety of tools on various platforms like Windows, Linux, etc.
3. Help forensic tool developers build better products and to assist forensic investigators in understanding what they may be up against.

Additionally, this project would also mainly focus on the performance of the application and masking of data so that the tools used in forensically driven procedures could not recover any data deleted earlier.

1.5 Rationale

This field of study is comparatively considered to be a recent one since the mainstream research was mostly done in 2004 to 2005. In 2004-2005 the exploration into anti-forensics was primarily subversive. Later from 2006 the studies appear to layout the foundation of definitions, terms and other procedures involved in anti-forensics. Successively most of the work was then carried out starting with the description of methods and techniques involved and development of many applications that destruct the forensic vulnerabilities [Scott 2007]. Based on these ideas, in this project a testing
framework is developed to analyze few anti-forensic tools so that it could enhance new
directions in developing better forensic tools for investigation procedures.

Currently government funded agencies are conducting their experiments on these
anti-forensic tools on how they will work and exploit the digital forensic investigation
tools. This project also tries to develop a prototype version of this in the direction to
enhance further development in producing better forensic tools and traits to overcome the
traces left by the anti-forensic tools. The current approach to forensic examination during
search and seizure has predominantly been to disconnect the computer or turn the power
off the machine and subsequently perform a post mortem examination on the storage
medium [Harris 2006].

The description of these anti forensics tools, and how they affect this data is
emphasized. The constant threat of Anti- Forensics tools must be dealt with a serious
counter technique approach [Harris 2006]. Within this project there is an ability to point
most, or if not all of the Anti Forensics capabilities, how they work, and what can be
done to overcome the pain of tampered evidence using Anti-Anti-Forensic approach.
2. NARRATIVE

2.1 Problems from Investigator’s approach

The forensic investigation process becomes more and more tougher for every attack made by an imposter. Day-to-day technologies developed in encryption, steganography and viruses enforce the digital forensic examiners to re-think about their actions before they execute them. The current forensic processes may or may not possibly educate from the various anti-forensic applications but they will definitely incorporate intelligent modifications. That means the current forensic processes will not totally depend on the forensic tools but will depend on the knowledge and experience that they gain from each investigation.

Hence the need for assessing and exploiting the Anti-Forensic tools is needed. Most of the tools which are available for commercial purposes strive to erase data without leaving any operational signatures on the machine [Technet 2010]. To do this, a rigorous analysis on a wide range of Anti-Forensic tools that endangers investigators activities by revealing certain vulnerabilities is conducted. This approach would also enhance the improvements on the side of investigator by educating on how to overcome the traces left by every Anti-Forensic tool using Anti-Anti-Forensic methods [Rogers 2005].
2.1.1 Scope

The goal of the project is to provide a means to examine the performance of anti-forensic tools, evaluating the tools in the real world computer use under various platforms. This would eventually lead to document the loop holes observed in the conventional software forensic tools by producing different test cases. Furthermore, a detailed testing is done to solve the problems occurred by using anti-forensic tools using anti-anti-forensic commercial tools such as FTK, Encase, etc.

2.2 Functionalities of Anti-Forensic Tools

The main purpose of this project would be to help forensic tool developers build better products and to assist forensic investigators in understanding what they may be up against. The range of the project covers primarily on building the prototype and following tools to be evaluated, tested and assessed. Furthermore, it gives the ability to point out the operational signatures each tool left so that forensic analysis department can educate themselves in producing better tools for capturing digital evidence. They are designed to foil computer forensic technologists [Anti-Forensics 2010] [Scott 2007] [Technet 2010].

2.3 Reducing the effectiveness of Anti-Forensic Methods

In order for Anti-Forensic methods to work, they must rely on inherent problems with forensic methods. Anti-Forensics often makes use of attacks on the investigators and may also take advantage of the dependency on specific tools or processes [Grugq 2005]. Unfortunately, no one can completely control these issues and no one will ever be able to
completely prevent the corruption of evidence [Rogers 2005]. However, if the targets to the problems are faced one by one, there might be a chance to be able to minimize the susceptibility of Anti-Forensics.

Table 2.1 explains the various exploitations of methods which depends on three factors; human element, tool dependence, physical/logical limitations.

**Table 2.1: Various Exploitations under Anti-Forensic Methods [Garfinkel 2007]**

<table>
<thead>
<tr>
<th>Name</th>
<th>Human Element</th>
<th>Tool Dependence</th>
<th>Physical/Logical Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>MACE alteration</td>
<td>Investigator may assume accuracy of dates and times</td>
<td>Tools may not function with invalid or missing dates and times</td>
<td>Invalid times and dates make collating information from multiple evidentiary sources difficult or impossible</td>
</tr>
<tr>
<td>Removing/wiping files</td>
<td>Investigator may fail to examine deleted files</td>
<td>Methods of restoring deleted files are specific to the tool so effectiveness may vary</td>
<td>Time required to restore wiped file contents may outweigh the evidentiary value of the data it contained</td>
</tr>
<tr>
<td>Account hijacking</td>
<td>May fail to consider whether the owner of the account was actually the person at keyboard</td>
<td>Tools may not be capable of extracting information that would aid investigator</td>
<td>Zombied computers may produce indirection</td>
</tr>
<tr>
<td>Archive/Image bombs</td>
<td></td>
<td>Improperly designed software may crash</td>
<td>Useful data might be located in the bomb itself</td>
</tr>
<tr>
<td>Disabling logs</td>
<td>May not notice missing logs</td>
<td>Software may not flag events</td>
<td>Missing data might be impossible to reconstruct</td>
</tr>
</tbody>
</table>

Solving Anti-Forensics issues will require any user to understand the actual problem. This project will try to describe the definition of Anti-Forensics the growing problems forensic investigators face due the increased use of Anti-Forensics tools, and
how hackers use Anti-Forensics to combat and manipulate the forensic investigation. Even if the investigator gets hold of the evidence acquired from the machine, it might not be useful for him to conduct experiments in the future because the anti forensic tools possess the capability to destruct the integrity of the evidence, and exploit the nature of the data to forward the guilt on to the examiner. Hence a forensic investigator must stay up to date on the every changing development of new anti-forensics tools that are being introduced on the Internet today [TWG 2001].
3. PROPOSED SYSTEM DESIGN

The design of this project follows four basic goals.

**Simplicity:** The goal is to design a prototype based on a minimal set of powerful abstractions, which could be easily tested and adopted to the user's needs. So that there is a possibility to define the anti-forensic techniques and methodologies in a declarative way by analyzing the results obtained in the project.

**Adaptivity:** Adding new modules of testing procedures and dynamic (re-) configuration of data sources has to be supported during run-time without having to interrupt ongoing system operation.

**Scalability:** Targeting a very large number of forensic tool vendors with a variety of new test cases induced into them, it’s the ability of the product to function properly leaving any digital fingerprints.

**Light-weight implementation:** This project is planned to be easily implementable in standard computing environments (no excessive hardware requirements, only standard network connectivity, etc.), portable (Java-based implementation), require minimal initial configuration, and provide easy-to-use, Web-based management tools.

3.1 Framework

This project mainly focuses on the effectiveness of the anti-forensic methods, analyze various anti-forensic tools tested and categorize general guidelines to protect forensic integrity which breaks in through anti-anti forensic techniques.
This generic ideology thereby seeks to discover the weaknesses in the current forensic processes or tools used in the investigation procedures that could provide high quality, flexibility and ability to enforce today’s interest towards forensics tools and techniques. This project embodies the testing of various tools to educate the research under anti-forensics and techniques to combat against them. Figure 3.1 explains about the testing framework which depicts the various kind of tools used in the investigative procedures with respective to two types of users the investigator and the intruder.

<table>
<thead>
<tr>
<th>Client Side</th>
<th>Investigator Side</th>
</tr>
</thead>
<tbody>
<tr>
<td>Applications, Browsers, Files, Logs, Signature Files &amp; other Storage Media</td>
<td>Forensic Examiner Tools</td>
</tr>
<tr>
<td>Anti-Forensic Tools</td>
<td>Equipment</td>
</tr>
<tr>
<td></td>
<td>&amp; Techniques</td>
</tr>
<tr>
<td><strong>Various Platforms and File Systems</strong></td>
<td></td>
</tr>
<tr>
<td>Windows, Unix/Linux, FAT and NTFS</td>
<td></td>
</tr>
<tr>
<td><strong>Analysis Tools</strong></td>
<td></td>
</tr>
<tr>
<td>FTK, ProDiscover, EnCase etc</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 3.1: Framework Supported for the Application and the Targets of Use**

On the client side, the one who committed crime will have several applications, browsers and other files on the disk at the crime scene to be evaluated along with anti-
forensic tools to damage the digital investigation process. Whereas, the investigator side will possess forensic tool techniques used to acquire evidence. Both types of users work on various platforms and file systems in which the functionality of forensic tools should be well known in accordingly.

3.2 Proposed Mechanism

The project is mainly divided in two phases. The first phase concentrates on developing a prototype that behaves like an anti forensic application. This application is inspired by examining various commercial anti forensic packages developed recently from the modern versions of Windows environment [Joachim 2008]. The application developed promises to be more user friendly, simple and implemented with greater performance [Technet 2010]. The prototype tool that is developed in this project is based on the requirements mentioned below.

1. The application should be able to delete all of the information from the hard disk and any other portable drives connected to the machine specifically cookies, temporary internet files, recent history, my documents etc.

2. The application should be able to erase individual files or folders specified by the user dynamically.

3. The application should also be able to clone a program numerous times so that it fills up the space in the disk.

4. Additionally it should also implement masking functionality of anti forensics so that any computer forensic tool does not recover the erased data.
5. It should be able to show better measures in terms of performance, usability and implementation.

Figure 3.2 illustrates the user-friendly interface of the prototype developed in this project named as Discoff which mainly focuses on above requirements mentioned.

![Figure 3.2: Snapshot of the prototype application developed](image)

The second phase of the project includes testing several of the tools mentioned in the Section 2-2. Tools such as Evidence blaster, Tracks eraser Pro, Eraser and Secure clean zapper will be installed on the windows based environment corresponding to various file formats like FAT16, FAT32 and NTFS depending on flexible scenarios. In each round of testing each tool will be installed on an identical operating system environment created from the baseline file system image; this will allow the performance of each tool to be
tested on the same system with the developed application and against identical data and activity records [Joachim 2008]. Prior to the operating system’s installation, the hard disk will be overwritten with random values to help ensure that previous artifacts on the media will not be mistaken for data on the test system.

Although configuration details will vary from tool to tool, the set-up and use of the counter forensic software will tend to follow a consistent approach. Each tool will be configured to overwrite data targeted for deletion depending on the pass the user will induce. A single overwriting pass will be chosen, sufficient to obstruct recovery with standard software-based forensic applications [TWG 2001]. According to these results, the test cases will be documented assessing the performance of each tool and the operational fingerprints it leaves. The main platform for analyzing the performance of the tools will be the Forensic Tool Kit (FTK) version 1.50a-1.51 from Access Data. Additionally, using the forensic tools recovering of the data, which is erased, by the Anti-Forensic tools is performed and each phase of the procedure will be documented [Berghel 2007].

Finally, experiments will be conducted by using forensic techniques to combat against the effects left by Anti-Forensic tools and hopefully educate forensic tool developers.
4. TESTING AND EVALUATION

In this project, implementation has its own take of approach; the testing procedures described in this project would be deliberately chosen to be similar in function. The testing platform is either a desktop or a laptop machine with sufficient RAM. Some of the tools mentioned in the Section 2-2 like evidence blaster, tracks eraser pro, secure clean zapper and eraser are appropriately installed on the windows based environment to analyze operational fingerprints left by that respective tool. A principle user account will be created with all the administrative privileges. This account will be used for the subsequent activity on the system. As the main spotlight is on the testing procedures the working model of the project runs on test cases and will be further included in the final report.

This allows the performance of each tool to be tested in the same environment with identical data and activity records. Although the configuration details will vary somewhat from tool to tool, setting up and using the privacy software will follow a consistent approach. Inspite of the difficulties like meeting the hardware requirements such as installation problems of each tool on various platforms and respective functionality issues, the main project will try to focus on the effectiveness of the anti-forensic methods.

4.1 Forensic Tools Used: FTK

Forensic Toolkit is recognized all around the world as the standard in computer forensics software. This court validated digital investigations platform delivers cutting-edge computer forensic analysis with intuitive and customizable user interface. Sophisticated
search approaches are implemented by FTK to retrieve images, documents and deleted documents. It is designed to acquire crystal clear information reporting and with effective search indexes [Garfinkel 2007].

4.2 Testing Methodology

The testing is categorized into two phases depending upon the disk volumes, their characteristics and appropriate tools found. The first part is done on the Hard Disk Drive with Windows XP environment dealing with the low-level operating system files but not the running files, as that would crash the operating system. The second part is done on a portable drive with FAT file system. The prototype application developed in this project is named Discoff and is evaluated with other commercial anti forensic tools.

This project testing is based on four factors.

A) Interface (Excellent, Good, Satisfactory, Unsatisfactory)

B) Time for deletion (Quick, Fast, Moderate, Slow)

C) Percentage of deletion (100%)

D) Impact of the anti forensic tool on FTK (Excellent, Good, Satisfactory, Unsatisfactory)
4.2.1 HDD Analysis

Table 4.1: Test cases for HDD Analysis

<table>
<thead>
<tr>
<th>Tool</th>
<th>Interface</th>
<th>Time for Deletion</th>
<th>% of Deletion</th>
<th>Impact on FTK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diskoff</td>
<td>Expected User friendly interface</td>
<td>Expected quick deletion</td>
<td>Expected high percent of deletion</td>
<td>Files do not supposed to be retrieved by FTK</td>
</tr>
<tr>
<td>Evidence Blaster</td>
<td>Expected User friendly interface</td>
<td>Expected quick deletion</td>
<td>Expected high percent of deletion</td>
<td>Files do not supposed to be retrieved by FTK</td>
</tr>
<tr>
<td>Tracks Eraser Pro</td>
<td>Expected User friendly Interface</td>
<td>Expected quick deletion</td>
<td>Expected high percent of deletion</td>
<td>Files do not supposed to be retrieved by FTK</td>
</tr>
</tbody>
</table>

This project uses a hard drive with windows XP operating system and specifically concentrates on the documents and settings folder of the C drive. Following are the steps required for the test cases explained in Table 4.1.

1. Install the tools Evidence blaster, Tracks eraser Pro and the Discoff itself.
2. Test the cases explained in Table 4.1.
3. Use the Forensic Tool Kit (FTK) to examine the file system by implementing the recovery process.
4. Evaluate the operational fingerprints left by the anti forensic tools.
5. Repeat the steps from 2 to 4 for significant and consistent evaluations.
6. Observe the results and report them significantly.
4.2.2 HDD Testing Process

Tool 1: Discoff

Characteristics: Uses data wiping and masking functionalities within Windows based environment. Based on Windows MFC Programming (Microsoft Foundations Class) for higher performance. This application mainly focuses on the low level system oriented files within windows based operating system i.e., temporary windows files, temporary user files, recently used files, Internet cache, Cookies, My documents folder and any other volatile data. Also allows user to search and delete for any particular files or folders and their extensions depending upon user’s choice to delete when there is a limited time. This is based on the tree list feature of indexing file systems [Joachim 2008]. The other unique feature of discoff allows user to choose files and produces clones of any particular file from 10 to 9999 times such that it overwrites all of the disk space randomly. Additionally this application also tries to implement self-deletion by executing the batch-processed files [Technet 2010].

Goal: The disk wiping functionality implemented by discoff must not be recoverable by the computer forensic tool FTK. Should also be able to achieve higher performance with user-friendly interface.

Time taken to delete: 1 minute 7 seconds for 392 Files

Figure 4.1 illustrates the user interface of the prototype application developed in this project, which exhibits three functions: data cleaner, file cloner and file searcher.
Figure 4.1: User Interface of DiscOff

Figure 4.2: User data cleaner for showing found files
The Figure 4.2 shows all the options that a user can select to be deleted. The user has the ability to view all the files found for deleting. When the do not delete just show button is unchecked and the user clicks the show found files, the user can view all the files listed from the path C:\Documents and Settings\....\Local Settings as shown below in Figure 4.3.

![User data cleaner for viewing found files](image)

Figure 4.3: User data cleaner for viewing found files
After viewing the files detected by the Discoff, the user can start deleting process by clicking on the clear button as shown in Figure 4.4.

![User data cleaner for clearing found files](image)

**Figure 4.4: User data cleaner for clearing found files**
Figure 4.5: User data cleaner deleting process

Figure 4.5 depicts the completion of the disk wiping process. A few temporary files will not be deleted as it is shown in Figure 4.6 because of their security level.
Figure 4.6: View of files that have not been deleted

The files listed in Figure 4.6 are those that have not been deleted by the Discoff. These files have high-level security and they are also temporarily been used. Some of those files will be erased after restarting the machine.
Figure 4.7: Recovery Process using Forensic Tool Kit (FTK)

Figure 4.7 illustrates the data recovery process by the forensic tool kit FTK starting with designing a case for the investigation procedures.
Figure 4.8: Results from the FTK after recovery process

Figure 4.8 shows the results obtained from the acquiring process by adding the evidence of the local hard disk drive. The added evidence location is specifically the documents and settings from the C folder i.e., C:\Documents and Settings\Administrator... According to the experiment conducted, the forensic tool kit did not recover any data that was deleted by discoff as the erased data was overwritten by random values implementing the masking functionality.
**Tool 2: Evidence Blaster**

Characteristics: Evidence Blaster is a Windows based anti-forensic tool that cleans the computer’s history and the browser history. This tool mainly focuses on the Internet cache and its history [Privacy 2010]. Comparison between Discoff and Evidence blaster has been difficult since its features do not comply exactly. The best feature of evidence blaster is it can also be run on stealth mode that means the operation of this tool can be hidden. Evidence blaster does not delete volatile data from the computer. The search feature of individual files makes Discoff unique.

Goal: Evidence blaster should be able to delete the information from hard disk drive and its Internet related history. The deleted data must not be recoverable by using the forensic tool kit (FTK).

Time taken to delete: 1 minute 58 seconds for 306 files.
Figure 4.9: Data cleaning facilities by Evidence Blaster

This application allows a user to select the preferred tasks and to perform as it is illustrated in Figure 4.9. It mainly focuses on the Internet related information but not on temporary files, my documents folder, favorites and recent history from the C drive [Privacy 2010]. It does not allow users to dynamically search and select individual files for deletion. Figure 4.10 illustrates the data that is wiped out, the space it saved and number of files it deleted by the evidence blaster.
Figure 4.10: Data cleaning facilities by Evidence Blaster

Figure 4.11: Recovery Process using FTK
Figure 4.11 illustrates the recovery process by the forensic tool kit (FTK) by examining on the evidence items deleted by the evidence blaster. It also shows the time elapsed and number of items it examined.

![Add Evidence]

```
Errors were encountered while adding evidence. 16 of the files were not added. Please view the case log for the details.
```

**Figure 4.12: Error while recovering data by FTK**

Figure 4.12 shows an error during the recovery process by the FTK. This error describes about recovering few files but was not added to the FTK evidence items. This error has been displayed as many times as the experiment was conducted. Figure 4.13 illustrates the results recovered by the FTK basing on the evidence items deleted by the evidence blaster.
Figure 4.13: Results displayed by FTK after recovery process

By observing the log obtained from the FTK and the error displayed, the experiment was successful but the application could not delete the high level system files same as Discoff. Hence it left few operational signatures behind the system where forensic investigators can work to build powerful computer forensic tools.
**Tool 3: Tracks Eraser Pro**

Characteristics: Tracks Eraser Pro is the most powerful tool that can erase any information from the computer. Best features include deleting information by scheduling the time, running in stealth mode and giving the ability to user to select and choose particular files. It cleans the hard disk by overwriting every bytes of free space with random values as implemented in Discoff. This feature makes the deleted files unrecoverable [Acesoft 2010].

Goal: The goal of this tool is to delete low-level and high-level security in a way that they are irrecoverable by any computer forensic tool.

Time taken to delete: 2 minutes 4 seconds for 382 files.

![Figure 4.14: Snapshot of Tracks Eraser Pro](image)

Figure 4.14 shows the customizable interface of Tracks Eraser Pro, which shows various locations within the operating system to be deleted by the user.
Figure 4.15 illustrates the completion process by the tracks eraser pro by displaying the number of files it deleted and space it saved in the whole process. At the initial phase of testing this application, the user has the ability to choose what to delete and also has the potential to conduct the experiment in stealth mode [Acesoft 2010]. After the deleting process, the tool also maintains a log of deleted files as it is shown in Figure 4.16.
Figure 4.16: Log consisting summary of deleted files

Figure 4.17: Recovery Process using FTK
Figure 4.17 explains about the data recovery process being done by the FTK on the evidence items deleted by the tracks eraser pro.

![Image of FTK interface](image)

**Figure 4.18: FTK showing no traces of deleted data**

Hence by observing the Figure 4.18 and the experiments conducted above prove that Forensic Tool Kit is not powerful enough to acquire the deleted data. It is important to note that Hard Disk Drive analysis testing phase proves that 2 out of 3 anti-forensic applications are more efficient than current forensic tools used nowadays.
4.2.3 Portable drive Analysis

This phase of testing involves a 2 GB Thumb drive with FAT file system. Commercial anti forensic tools are appropriately selected depending on the specific characteristics that could work on the portable drive.

**Table 4.2: Test cases for Portable drive analysis**

<table>
<thead>
<tr>
<th>Tool</th>
<th>Interface</th>
<th>Time for Deletion</th>
<th>% of Deletion</th>
<th>Impact on FTK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diskoff</td>
<td>Expected User friendly interface</td>
<td>Expected quick deletion</td>
<td>Expected high percent of deletion</td>
<td>Files do not supposed to be retrieved by FTK</td>
</tr>
<tr>
<td>Eraser</td>
<td>Expected User friendly interface</td>
<td>Expected quick deletion</td>
<td>Expected high percent of deletion</td>
<td>Files do not supposed to be retrieved by FTK</td>
</tr>
<tr>
<td>Secure Clean Zapper</td>
<td>Expected User friendly Interface</td>
<td>Expected quick deletion</td>
<td>Expected high percent of deletion</td>
<td>Files do not supposed to be retrieved by FTK</td>
</tr>
</tbody>
</table>

Following are the steps involved based on the test cases explained in the Table 4.2.

1. Load the portable drive to the testing machine.

2. Involves proper installation of the tools mentioned in Section 2-2 particularly for portable drive devices, which are Secure Clean, Eraser and Discoff itself.

3. Repeat steps 2,3,4,5 and 6 from Section 4.2.1.

4.2.4 Portable drive Testing Process

**Tool 1: Discoff**

Characteristics: This application gives the ability to select the particular drive by implementing the tree-list option [Joachim 2008].
Goal: This application should be able to delete all of the information from the portable drive attached to the computer. Additionally, the deleted data wiped by this application must not be recoverable by any computer forensic tool.

Time taken for deletion: 3 minutes for 270MB of data.

Figure 4.19: Snapshot of Discoff Application

Figure 4.19 shows the file searcher feature that allows the user to choose and select the particular drive for deletion. This property outputs a customizable interface for any type of user to efficiently use this application.
Figure 4.20: Tree-list view of drives, folders and files

The above Figure 4.20 depicts the tree-list view to search and select for particular drives, files and folders effectively.
Figure 4.21: Deletion Process of Portable Drive by Discoff

Figure 4.21 shows the status bar of deletion process done by Discoff. This feature from Discoff gives utmost flexibility to the user by selecting which drive, folders or files to be deleted. The user can also select on the type of extensions as shown in Figure 4.21. The next step in the testing phase is the recovery process, which is shown in Figure 4.22. Creating a case and inputting the details of the case initiates this step.
Figure 4.22: Recovery Process using FTK

Figure 4.23: Indexes in FTK
Figure 4.23 illustrates the refining procedure done by FTK before the recovery process. The user has the ability to select the file type and file status depending upon the evidence item.

![Image of FTK interface]

**Figure 4.24: Results from the recovery process by FTK**

From the above experiment conducted and from the Figure 4.24, it is proved that FTK could not recover any deleted data as masking functionality was implemented in the discoff application.
**Tool 2: Secure Clean**

Characteristics: This application could only delete a particular folder by dragging and dropping it into the desktop icon [WhiteCanyon 2010].

Goal: Should be able to delete the portable drive resulting in higher performance.

Time taken for deletion: 9 minutes 8 seconds for 270MB of data.

Figure 4.25: Snapshot of Secure Clean

Figure 4.25 illustrates the interface that explains a user how to delete files using secure clean zapper.

![Snapshot of Secure Clean](image)

Figure 4.26: Secure Clean interface after dropping the folders
Figure 4.27: Secure Zap Confirmation

Figure 4.28: Deletion Process of Secure Clean
Figure 4.26, 4.27 and 4.28 shows the deletion process by dragging the files onto the secure clean zapper desktop icon.

![Image of Secure Clean Zapper interface]

**Figure 4.29: Error reports for unsuccessful deletion**

From the Figure 4.29 the experiment shows that not all of the files from the portable device were deleted due to different file properties. Hence making it possess low potentials in terms of effectiveness and usability.
Figure 4.30: FTK Recovery Process for Secure Clean Zapper

Figure 4.31: Adding evidence to the case

Figure 4.30 and 4.31 demonstrates the recovery process of the data deleted by the secure clean zapper. This can be done by creating a case for investigation and adding the evidence location to the case.
From Figure 4.32, this experiment shows that the FTK was able to recover some of the free space files but could not acquire or analyze any data from it. It only tells the type of file system.
Tool 3: Eraser

Characteristics: This application can delete any file or folder by allowing the user to search, select and delete them appropriately. This application also gives the dynamic scheduling ability to user [Heidi 2010].

Goal: Should be able to delete any file or folder effectively with higher performance and should not be recoverable by any computer forensic tool.

Time taken to delete: 5 minutes for 270MB of data.

Figure 4.33: Interface of the Eraser

Figure 4.33 depicts the user-friendly interface of the tool eraser.
Figure 4.34: Interface to select files and folders

Figure 4.35: Task Properties pane for scheduling dynamically
Figure 4.34 and 4.35 gives the flexibility to the user to choose the destination of the file or folder to be deleted particularly. It also provides dynamic scheduling feature that means a user can choose when to trigger the deletion process.

![Erase Schedule](image)

**Figure 4.36: Deletion Process by Eraser**

Figure 4.36 shows the status of the deletion process. The deletion process by eraser involved lot of time for any file or folder. Hence dynamic scheduling played an important role so that the user can queue the processes of deletion. Figure 4.37 depicts the queue that is processed for deletion sequentially.
Figure 4.37: Processes Queue for deletion

Figure 4.38: Indexes in FTK recovery process
Figure 4.39: Results by FTK on deleted data by Eraser

Figure 4.38 and 4.39 shows the recovery process done by FTK on the data deleted by Eraser. The above experiment shows that FTK could not recover the information deleted by the eraser tool. Apparently, this application involves lot of time to perform deletion process but stands out in producing the dynamic scheduling functionality.

4.2.5 Additional functionalities of Discoff

Discoff also implemented cloning functionality i.e., a user can select an individual file and by using discoff application and create clones of that particular file so that it overwrites the entire disk space targeted depending upon the number of clones. This
application specifically targets to clone from 10-9999 files as shown in Figure 4.40 [Cloner 2010].

Figure 4.40: Discoff with File cloner functionality
Figure 4.41: Discoff illustrating cloning process of 4MB file

Figure 4.41 shows the cloning process started by the user by selecting an mp3 file and entering the number of clones. The progress of the file cloner is also displayed.
Figure 4.42: Space overwritten by discoff cloner

Figure 4.42 and 4.43 demonstrates the cloning process being successful as the 4000 clones fill the entire space, which are generated by the file cloner.
Figure 4.43: File cloner depicting 4000 files

The file cloner is a unique feature implemented in discoff and is tested by taking a 4 MB mp3 file and by producing 4000 clones of that particular file. The time involved in this whole process was 10 minutes 3 seconds.

The second additional feature included in discoff is self-deleting functionality, which is shown in Figure 4.44. This is implemented based on the batch process. When user sets on it, it creates the batch file, which will again re-run the application to generate an executable file, after opening it the DiscOff.exe delete this batched DiscOff.exe and marks it to zero [Joachim 2008].
Figure 4.44: Self deletion process implemented by discoff

Figure 4.45: Quit process for self deletion in Discoff
Figure 4.46: Results after self-deletion process

Figure 4.45 and 4.46 demonstrate the self-deletion process illustrated by the Discoff application.
5. RESULTS

The results seen in Table 5.1 and 5.2 are obtained by testing the developed prototype of anti forensics with the other commercial packages obtained from Internet. The results obtained from this project are based on these factors.

A) Interface (Excellent, Good, Satisfactory, Unsatisfactory)
B) Time for deletion (Quick, Fast, Moderate, Slow)
C) Percentage of deletion (100%)
D) Impact of the anti forensic tool on FTK (Excellent, Good, Satisfactory, Unsatisfactory)

Results on HDD Analysis:

Table 5.1: Results on Hard Disk Drive Testing

<table>
<thead>
<tr>
<th>Tool</th>
<th>Interface</th>
<th>Time for Deletion</th>
<th>% of Deletion</th>
<th>Impact on FTK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discoff</td>
<td>Good</td>
<td>1min 7 secs</td>
<td>98%</td>
<td>Unable to delete files currently in use</td>
</tr>
<tr>
<td>Evidence Blaster</td>
<td>Good</td>
<td>1min 58 secs</td>
<td>98%</td>
<td>Good</td>
</tr>
<tr>
<td>Tracks Eraser Pro</td>
<td>Satisfactory</td>
<td>2mins 48 secs</td>
<td>99%</td>
<td>Excellent</td>
</tr>
</tbody>
</table>

The results in Table 5.1 are based on the testing conducted in Section 4.2.1 on each different tool i.e., Discoff, Evidence blaster and Tracks eraser pro. Every tool tested in this project had its own advantages and disadvantages. Primarily, Discoff was very efficient in performance and uniqueness. It also resulted in having a good user-friendly interface. The percentage of deletion was not 100% since it was unable to delete the files,
which were in use, but gives an option of rebooting the machine to delete all files completely. Even the forensic tool kit (FTK) could not recover the information deleted by Discoff since it implemented masking functionality which most of the commercial anti-forensic tools implement. This tool not only erases but also overwrites the deleted space by random values. Additionally, the prototype developed in this project, Discoff also implemented two other features called File cloner and Self-deleter. File cloner does confuse investigators and their approach by filling up the disk space. Self-deletion feature makes forensic investigation procedures complex as it deletes itself when executed.

Evidence blaster resulted in having very good user interface. The files mentioned were not completely deleted; the current files, which have been used by the operating system, were left undeleted. The performance of this tool was moderate because it took longer times than other applications. Tracks eraser pro demonstrated very good results in terms of percentage of deletion, performance and recovery process by FTK. But there was some complexity using the interface of this application.

**Results on Portable Drive Analysis:**

<table>
<thead>
<tr>
<th>Tool</th>
<th>Interface</th>
<th>Time for Deletion</th>
<th>% of Deletion</th>
<th>Impact on FTK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discoff</td>
<td>Good</td>
<td>3 mins</td>
<td>100%</td>
<td>Excellent</td>
</tr>
<tr>
<td>Secure Clean Zapper</td>
<td>Satisfactory</td>
<td>9 mins 8 secs</td>
<td>98%</td>
<td>Good</td>
</tr>
<tr>
<td>Eraser</td>
<td>Unsatisfactory; Need to select files individually to delete</td>
<td>5 mins</td>
<td>95% Large files were unable to delete for the first time</td>
<td>Excellent</td>
</tr>
</tbody>
</table>
The results in the Table 5.2 are based on the testing conducted in Section 4.2.2 on each different tool i.e., Discoff, Secure clean zapper and Eraser on approximately 270 MB of data. Discoff demonstrated excellent results in Portable drive testing analysis as it gives the utmost flexibility to the user to choose a drive and search and select for files and folders particularly. It also gives the option to view the list of files before the deletion process. It also showed greater performance in terms to time it took to delete. Even FTK was unable to recover the deleted data. Secure clean zapper was good in terms of time it took to delete but the forensic tool kit did recover the deleted files from the slack space. The third tool eraser incorporated very poor user interface and took lot of time to delete a file or folder. Each individual file was selected individually and processed in queue. Best feature of eraser was dynamic scheduling. The user has the ability to schedule the deletion process at any particular time. Even forensic tool kit was unable to recover any data that was deleted by this application.

Comparisons were difficult among all these tools because each tool has its own features and concentrated on different areas to destruct particular portions of a machine. Hence it can be stated that every anti forensic tool has its own uniqueness and effectiveness in dealing with file systems. The experiments conducted show that there is a strong need of developing accurate and efficient computer forensic tools for future investigations.
6. CONCLUSION

Many of the anti-forensic techniques discussed in this project can be overcome through improved monitoring systems or by fixing bugs in the current generation of computer forensic tools. Examiners should be aware of the anti-forensic techniques and ways to overcome techniques. Forensic tools have to be updated to handle the anti-forensic techniques Out of many anti forensic tools available four of them were installed and tested with a prototype application that was developed as a main focus of this project. None of them were recoverable by FTK. This application ‘Discoff’ is unique in its own way considering the four important factors; performance, user-interface, percentage of file space deleted and impact on the FTK.

The main target and objective of anti-forensics is to sidetrack the investigator and the procedures conducted. In some of the business companies and associations, there is a strict rule of banning the usage and even the custody of any anti-forensic applications. In fact most of the current operating systems possess highly intelligent anti forensic tools, which exhibit no existence enforcing the privacy of data within the organization. Consequently these procedures make the rules ineffective. Forensic has always been dependent on the traces and operational signatures that are left out by the anti-forensic mechanisms. The government associations and other business companies are held responsible in building their own forensic standards in protecting the confidential information.
7. FUTURE WORK

The future of anti forensics is one of the most potential and vast areas to conduct a research and there is effective studies going on smoothly. The main focus of this project was to provide a means of education to digital forensic investigations to develop better and efficient tools to overcome the anti forensic packages. The future of this project could be extended to implement many other anti forensic techniques and features like encryption, steganography and even targeting computer forensic tool vulnerabilities. It would be a disaster if all these techniques are combined and implemented. The future developments could also make use of other digital forensic tools like Prodiscover, Encase and also new computer forensic tools being developed currently in the industry. This project could be specifically expanded to concentrate more on high-level security system files so as to cause massive destruction without any knowledge of the user. Research should be carried out with maximum expertise in this area to keep them update themselves to face all troubles and challenges that are arriving in Computer Forensics.


[Harris 2006] Harris, Arriving at an anti-forensics consensus: Examining how to define and control the anti-forensics problem, DFRWS 2006.


APPENDIX A. DEFINITIONS OF TOOLS

A-1: Tools which target Internet history, tracks of Internet activities and accounts

- Absolute Shield: Absolute Shield Internet Eraser protects privacy by cleaning up all the tracks of your Internet and computer activities.
- Evidence Blaster: It has the capability to clear all the browsers history, cache, system cookies and other temporary files.
- Secure Clean: It securely cleans up all unwanted files and internet clutters which thereby include the traces of passwords and other personal information.
- Tracks Eraser Pro: Tracks Eraser erases the cache, cookies, history, typed URLs, auto complete memory, index.dat from the browser and temp folder.

A-2: Tools which target computer related entities like logs, timestamps and hashes

- Clear Logs: Clear Logs clears the event log (Security, System or Application) that is specified.
- Timestomp: It can be used to modify date and time stamps thereby falsifying the validity of the document.

A-3: Tools which target forensic tool vulnerabilities

- Evidence Eliminator: Evidence Eliminator quickly and professionally deep cleans any computer that has sensitive material.
- Hash Tool: Hash (Hacker Shell) is a tool to enable people to evade detection while penetrating a system.
A-4: Tools which target the storage media in hard disk [Grugq 2005]

- DBan: DBAN will automatically and completely delete the contents of any hard disk that it can detect, which makes it an appropriate utility for bulk or emergency data destruction.
- Declasfy: The program is designed to "wipe" hard disks by writing the entire disk with O’s and 1’s.
- Diskzapper: Diskzapper Dangerous automatically begins erasing all the disks as soon as the booting process is completed.
- Eraser: Eraser is a Windows tool that allows you to securely remove files from computer’s hard drive and securely wipe free space.
- Overwrite: Overwrite is a UNIX utility that tries to make harder data recovering.
- Wipe: It is a tool that effectively degausses the surface of a hard disk, making it virtually impossible to retrieve the data.

A-5: Tools, which target on hiding of files using encryption and steganography techniques. [Bragg 2004]

- BestCrypt: It tries to disguise the data needed by using strong encryption techniques.
- Cryptomite: CryptoMite enables the user to encrypt, decrypt, and wipe files and folders of any type.
- Invisible Secrets: Not only encrypts but also hides in places which appear to be innocent.
APPENDIX B. PARTIAL CODE OF DISCOFF APPLICATION

// Discoff entry point of application:

#include "stdafx.h"
#include "MainDlg.h"
#include "Resource.h"
int APIENTRY WinMain(HINSTANCE hInstance,
                      HINSTANCE hPrevInstance,
                      LPSTR     lpCmdLine,
                      int       nCmdShow)
{
    DialogBox(hInstance, MAKEINTRESOURCE(IDD_MAIN), NULL,
              (DLGPROC)WndMainProc);
    return 0;
}

// File.cpp: contains main wiping function

#include "stdafx.h"
#pragma warning(disable : 4786)
#include "Search.h"
#include "File.h"

bool bShowFiles = true;
bool bNeedReboot = false;       // We need to reboot
std::vector<std::string> vShowFiles;
TCHAR szWipes[30][MAX_PATH*2] =
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
TCHAR szWipesDir[MAX_PATH*2];
//string (strWipePath)[30]; //= _T(""');
LRESULT WipeFile(string &strFileName)
{
    // if(strWipePath[0]!=_T('') && strWipePath[0].at(0) != strFileName[0].at(0))
    // strWipePath[0] = _T('');
    int iRes = 0;
    if(szWipesDir[0] != _T(0))
    {
        TCHAR szTmp[MAX_PATH*2];
    }
_tcscpy(szTmp, strFileName.c_str());
if(_tcsrchr(szTmp, _T(\\))!=NULL)
    _tcsrchr(szTmp, _T(\\))[0] = 0;
if(_tcscmp(szTmp, szWipesDir) != 0)
{
    _tcscpy(szWipesDir, szTmp);
szWipesDir[0] = _T(0);
    // else
    // if(_tcsstr(strFileName.c_str(), szWipesDir) == NULL)
}
// if(_T("") == strWipePath[0])
if(_T(0) == szWipesDir[0])
{
    TCHAR szWipePath[MAX_PATH*2];
    _tcscpy(szWipePath, strFileName.c_str());
    if(_tcsrchr(szWipePath, _T(\\))
        _tcsrchr(szWipePath, _T(\\))[0] = (TCHAR)0;
    }
    if(szWipePath[_tcslen(szWipePath)-1] != _T(\\))
    {
        _tcscat(szWipePath, _T("\"));
    }
    _tcscpy(szWipesDir, szWipePath);
    TCHAR szTmp[MAX_PATH*2];
    int i=0;
    for( i<30; i++)
    {
        //_tcscpy(szWipes[i], szWipePath);
        //strWipePath[i] = szWipePath;
        GetTempFileName(szWipePath, _T(""), 0, szTmp);
        //strWipePath[i] = szTmp;
        _tcscpy(szWipes[i], szTmp);
        DeleteFile(szWipes[i]);
    }
}
    vShowFiles.push_back(strFileName);
    if(bShowFiles)
        return (iRes = 1);
    SetFileAttributes(strFileName.c_str(), FILE_ATTRIBUTE_NORMAL);
    HANDLE hFile = CreateFile(strFileName.c_str(),
        GENERIC_READ|GENERIC_WRITE, 0, NULL, OPEN_EXISTING,
        FILE_ATTRIBUTE_NORMAL, NULL);
    if(INVALID_HANDLE_VALUE != hFile)
    {

DWORD dwSizeHi = 0;
int iSize = GetFileSize(hFile, &dwSizeHi);
SetFilePointer(hFile, 0, NULL, FILE_BEGIN);
if(dwSizeHi)
{
// BYTE *btPtr = new BYTE[0xffffffff];
// memset(btPtr, 0, 0xffffffff);
// WriteFile(hFile, tsData.m_psz, _tcslen(tsData.m_psz), &dwRead, NULL);
// delete[] btPtr;
}
DWORD dwRead;
BYTE *btPtr = new BYTE[iSize];
//memset(btPtr, 0, iSize);
WriteFile(hFile, btPtr, iSize, &dwRead, NULL);
FlushFileBuffers(hFile);
delete[] btPtr;
SetFilePointer(hFile, 0, NULL, FILE_BEGIN);
SetEndOfFile(hFile);
FlushFileBuffers(hFile);
CloseHandle(hFile);
if(MoveFile(strFileName.c_str(), szWipes[0]))
{
    int i = 1;
    for(; i<30; i++)
        MoveFile(szWipes[i-1], szWipes[i]);
    if(!DeleteFile(szWipes[i-1]))
    {
        MoveFileEx(szWipes[i-1], NULL, MOVEFILE_DELAY_UNTIL_REBOOT);
        bNeedReboot = true;
        iRes = 1;
    }
}
else
{
    MoveFileEx(strFileName.c_str(), NULL, MOVEFILE_DELAY_UNTIL_REBOOT);
    bNeedReboot = true;
    iRes = 1;
}
else
{
    MoveFileEx(strFileName.c_str(), NULL, MOVEFILE_DELAY_UNTIL_REBOOT);
74

bNeedReboot = true;
iRes = 1;
}
return iRes;

LRESULT WipeFile(CFileItem*pFile)
{
    string strFileName;
    //TCHAR szFileName[MAX_PATH*2];
    CFolder *pFolder = pFile->m_pFolder;
    strFileName = pFile->m_strName;
    while(pFolder)
    {
        if(pFolder->m_pFolder)
            strFileName = _T("\") + strFileName;
        strFileName = pFolder->m_strName + strFileName;
        pFolder = pFolder->m_pFolder;
    }
    return WipeFile(strFileName);
}

// Cloning Function: contains the controls of clone part dialog box
#include "stdafx.h"
#include <CommCtrl.h>
#include <Commdlg.h>
#include <stdio.h>
#include <stdlib.h>
#include "Clone.h"
HWND m_hCloneNumber;
HWND m_hClone;
HWND m_hCloneProgress;
HWND m_hClonePath;
bool m_bClonning;
LRESULT OnCloneBrowse(HWND hWnd, WPARAM wParam, LPARAM lParam)
{
    TCHAR szTmp[MAX_PATH*2];
    ::GetWindowText(m_hClonePath, szTmp, MAX_PATH*2);
    ofn.lpstrFile = szTmp;
    ofn.lpstrFileTitle = NULL;
    ofn.lpstrFilter = _T("all files(*.*)\0*.*\0\0");
    ofn.lpstrInitialDir = NULL;
    ofn.lpstrTitle = _T("select file");
    ofn.lStructSize = sizeof(OPENFILENAME);
ofn.nMaxFile = MAX_PATH*2;
if(!GetOpenFileName(&ofn))
{
    int iErr = CommDlgExtendedError();
    return 0;
}
::SetWindowText(m_hClonePath, szTmp);
return 0;
}

LRESULT OnClone(HWND hWnd, WPARAM wParam, LPARAM lParam)
{
    //::EnableWindow(m_hClone, FALSE);
    SetCursor(LoadCursor(NULL, IDC_APPSTARTING));
    // CHECK
    if(m_bClonnning)
    {
        m_bClonnning = false;
        return 0;
    }
    TCHAR szCaption[128];
    TCHAR szTmp[MAX_PATH*2];
    TCHAR szFolder[MAX_PATH*2];
    TCHAR szFile[MAX_PATH*2];
    ::GetWindowText(m_hClonePath, szTmp, MAX_PATH*2);
    if(!*szTmp)
        return 0;
    _tcscpy(szFile, szTmp);
    if(_tcsrchr(szFolder, _T(\\))
    { 
        _tcsrchr(szFolder, _T(\\))[0] = _T(0);
    }
    ::GetWindowText(m_hCloneNumber, szTmp, MAX_PATH*2);
    int iCount = _ttoi(szTmp);
    if(1 > iCount)
        iCount = 1;
    else if(9999 < iCount)
        iCount = 9999;
    int i = 0;
    ::SendMessage(m_hCloneProgress, PBM_SETRANGE32, 0, iCount);
    ::SendMessage(m_hCloneProgress, PBM_SETPOS, 0, 0);
    MSG msg;
    for(;i<iCount; i++)
    {
        if(!m_bClonnnning)
break;
::SendMessage(m_hCloneProgress, PBM_SETPOS, i, 0);
GetTempFileName(szFolder, _T(""), 0, szTmp);
CopyFile(szFile, szTmp, false);
//GetLastError();
while(PeekMessage(&msg, NULL, 0, 0, PM_REMOVE))
{
    TranslateMessage(&msg);
    DispatchMessage(&msg);
}

// EXIT
::SetWindowText(m_hClone, szCaption);
SetCursor(LoadCursor(NULL, IDC_ARROW));
::SendMessage(m_hCloneProgress, PBM_SETPOS, 0, 0);
m_bCloning = false;
//::EnableWindow(m_hClone, TRUE);
return 0;
}

LRESULT OnCloneNumber(HWND hWnd, WPARAM wParam, LPARAM lParam)
{
    int wmId, wmEvent;
    wmId = LOWORD(wParam);
    wmEvent = HIWORD(wParam);
    if(EN_KILLFOCUS == wmEvent)
    {
        int iCount;
        bool bUpdate = false;
        ::GetWindowText(m_hCloneNumber, szTmp, MAX_PATH*2);
        iCount = _ttoi(szTmp);
        if(1 > iCount)
            iCount = 1, bUpdate = true;
        else if(9999 < iCount)
            iCount = 9999, bUpdate = true;
        if(bUpdate)
        {
            _stprintf(szTmp, _T("%d"), iCount);
            ::SetWindowText(m_hCloneNumber, szTmp);
        }
    }
    return 0;
}