Automated Construction of a False Digital Alibi

Alfredo De Santis[1], Aniello Castiglione[1]*, Giuseppe Cattaneo[1]
Giancarlo De Maio[1], and Mario Ianulardo[2]

[1] Dipartimento di Informatica *"R. M. Capocelli"*
Università degli Studi di Salerno, I-84084, Fisciano (SA), Italy
ads@dia.unisa.it, castiglione@acm.org, cattaneo@dia.unisa.it,
demaio@dia.unisa.it
[2] IISFA Italian Chapter & Lawyer in Naples, Italy
marioianulardo@codicieleggi.it

Abstract. Recent legal cases have shown that the *digital evidences* are increasingly used in proceedings (by defense, accusation, public prosecutor, etc.). Digital tracks can be left on computers, phones, digital cameras and also on third party servers belonging to Internet Service Providers (ISPs), telephone providers, companies that provide services via Internet such as YouTube, Facebook, Gmail and so on.

This work points out that it is possible to set up a false digital alibi without any collaborator. The key idea is that it is possible to produce digital evidences in a fully automatic way without any human intervention. The so obtained digital traces are indistinguishable ex-post from digital evidences produced by an individual in the same place and in the same time.

A forensic investigation on the digital evidences cannot establish if such traces have been produced by a human activity or by an automated tool. These considerations emphasize the difference between digital and physical - namely traditional - evidences: in substance, digital evidences should be considered relevant only if supported by evidences collected using "traditional" investigation techniques.

The results of this work should be considered by any entity involved in a Digital Forensics activity, as it demonstrates that the sentences of Courts should not be based only on digital evidences, but these should be always correlated with additional information provided by the various disciplines of the Forensics Sciences.

Keywords: Digital Evidence; Digital Investigation; Digital Forensics; Anti-Forensics; Counter-Forensics; False Digital Evidence; Automated Alibi; False Alibi; Digital Alibi.

* Corresponding author: Aniello Castiglione, castiglione@ieee.org, Phone: +39089969594, FAX: +39089969821

1 Introduction

1.1 The Digital Evidences

The use of digital technology is rapidly growing. The number of users in the world that is using the Internet is almost 2 billions, with a penetration of 28.7% of the world population [1]. As a consequence, more and more crimes are performed on the Internet or have something to do with digital equipments. For these reasons, there is a growing increase in digital evidences being brought in Courts in the United States and elsewhere in the world. Consequently, Courts are becoming concerned about the admissibility and probative value of digital evidence. Although digital devices have not been directly used by an individual that have been indicted to have committed a crime, they can be subject to forensic investigations in order to collect useful traces about the behavior of such person, either to be cleared of an accuse or to charged of an offense. Elements required to determine the liability for having committed a crime often consist of files stored in a PC memory, photos on a digital camera, information on a mobile phone, and on many other digital devices.

Digital traces are *ubiquitous*: they can be located anywhere in the world. In fact, digital traces can be retrieved on mobile devices (phones, PDAs, laptops, GPSs, etc.) but especially on servers that provide services via Internet, which often register the IP addresses and other information concerning the connected clients. These servers can be located in countries different than the one in which the crime has been committed, and different national laws can be an obstacle for the acquisition of the digital evidences during the investigation.

Digital traces are also *immaterial*: it is well known that all digital data present on a device is represented as sequences of one and zero. These data can be modified by an user having enough privileges on such device. For example, in modern multiuser Operating Systems (OSs), such as Microsoft Windows, Apple Mac OSX or Linux, there exists a *superuser* (e.g. the “root” in Linux) that can execute every operation, including access to any hardware/software resource, leading to the possibility of modifying all data stored on the device.

Devices running simpler OSs, such as GPSs, digital cameras, and, in part, mobile phones, often do not distinguish between access modes. In such cases, any person who has physical access to the device can modify its memory without the necessity of gaining *superuser* privileges.

Digital Forensics is constantly subject to change and evolution as it is mainly influenced by technological innovations. It makes necessary to constantly upgrade not only the tools for detecting and reporting digital traces, but also the analysis methodologies and the personnel that manages such tools. The digital forensic techniques have to meet the growing demand of scientific evidences in legal cases: this phenomenon is known as the “*CSI effect*” [14]. It is a phenomenon reported by prosecutors who claim that television shows based on scientific crime solving have made actual jurors reluctant to vote to convict when, as is typically true, forensic evidence is neither necessary nor available.

1.2 The Digital Alibi

Computers can be involved in the commission of crimes and can contain evidence of crimes, but can also be an *alibi* for the defense of an accused person. In the Latin language the word "*alibi*" is an adverb meaning "*in or at another place*". According to the Merriam-Webster online dictionary [16], alibi is "the plea of having been at the time of the commission of an act elsewhere than at the place of commission".

Here are discussed two examples of legal proceedings in which the digital evidence has been considered an alibi that contributed to exonerate the accused.

Rodney Bradford, a 19 years old resident of New York, was arrested on October 18, 2009 for suspicion of armed robbery at the Farragut Houses in Brooklyn, where he lives [2], [3], [4]. His defense lawyer, Robert Reuland, claimed the innocence of Mr. Bradford asserting that he was at his father's house, located in the Harlem quarter, at the time of the crime. The evidence offered in support of this thesis was a message posted by the suspected to his girlfriend - "*On the phone with this fat chick... where my IHOP.*" - on his Facebook page having timestamp "October 17 - 11:49 AM", exactly one minute before the robbery. The status update would take place from his father's PC. The subsequent investigation confirmed that the connection was established from an apartment located in the 71 West, 118th Street of Manhattan, i.e. the father's house, which was far more than thirteen miles from the scene of the crime. Rodney Bradford was released 12 days after his arrest. This is probably the first case in which a status update on Facebook has been used as an alibi. It is clear that anyone who knew the appropriate username and password could modify a Facebook profile. For example, these actions may have been made by a partner. However, according to defense attorney Reuland, this possibility was remote because it would imply a level of criminal genius unusual in a so young individual.

Another Court case, extremely interesting in terms of assessing the digital alibi, is the italian case named "Garlasco" from the small city located in North Italy where the facts happened ([5], [6]). The proceeding of first instance ended with the acquittal of Alberto Stasi, the main suspect in the murder of his girlfriend Chiara Poggi. The defendant proclaimed his innocence claiming a digital alibi: when his girlfriend was murdered he was writing the thesis on his computer. This court case is characterized by a close comparison between the results of analysis performed on each type of specimen, such as DNA traces and digital evidence on the PCs of the victim and the suspected. These findings were complemented by traditional techniques. However, the attention of the investigators still focus on verifying if the digital alibi claimed by Stasi was true or false. While noting the errors committed by the experts at the stage of retrieving and analyzing the digital evidence, the Court directed an acquittal of the accused person. This means that the digital alibi, although undermined by mistakes, has proved to the court that the suspect was working on his laptop during the time of the crime.

Identifying the true originator of digital evidences is a very hard task. In fact, it is possible to trace the owner of a digital device, but the digital evidences themselves do not contain any information on *who or what* has produced them. Also, in multiuser environments, where authentication is required to access the system, a malicious user could bypass the normal logon procedure and act on behalf of the owner. It is known as the *problem of user identification* and is raised by the *immaterial* nature of the digital data.

This work shows that it is possible to set up a series of automated actions in order to produce digital traces that are *post-mortem indistinguishable* from those left by a person, and how such evidences could be claimed in a Court to forge a valid alibi. Direct consequence of this result is that the forensic analysis in legal cases should focus not only on the retrieval and analysis of digital evidences, but also on the identification of their author.

The remainder of this work is organized as follows: in Section 2 various approaches of forging a false digital alibi have been discussed. In Section 3 the methodology of forging a false digital alibi creating a fully automated tool has been presented and analyzed. In Section 4 a case study on Microsoft Windows systems have been reported. Finally, this paper ends with the authors conclusions in Section 5.

2 Creation of a False Digital Alibi

In this work it is assumed that there is a particular device (e.g. PC, Smartphone, etc.) used to produce evidences. Moreover, there are some trusted companies providing services (e.g. social networks, mail boxes and so on) that record traces about their users, such as access date, session duration, which can be considered trusted in a legal case scenario. In order to forge a digital alibi based on these assumptions, it is possible to follow different strategies. A simple technique is to engage an accomplice which produces digital evidences on behalf of another person (e.g. accessing his mailbox, leaving messages on Facebook, etc.). This technique does not require that the involved people have particular skills. However, the presence of another person could produce unwanted non-digital (e.g. biological) evidences that can be revealed by traditional forensic investigation techniques.

In this work two new approaches are presented which do not require any human accomplice: Remotization and Automation.

- *Remotization.* In order to forge a digital alibi by themselves, it is necessary to produce evidences at some trusted entities during the same timeline of the alibi. To accomplish this task, it is possible to remotely control a device by means of an IP connection (e.g., over the Internet), using a KVM device or a Remote Control software. However, this technique requires the interaction with another device (the controller) while producing the evidences.
- *Automation.* This paper focuses on the automation method, which consists of forging a digital alibi using a fully automated software tool. This approach does not require any interaction with the device while producing the digital evidences.

2.1 Remotization

In this section two techniques to forge an alibi by using a personal computer to be remotely controlled are discussed.

Remote Connection by Means of KVM Over IP An individual who intend to create an alibi can use a KVM over IP switch (iKVM) [17] to control his PC remotely. This technique does not require any suspicious software to be installed. However, the acting individual must take some precautions to limit the amount of unwanted traces. For example, he should configure the iKVM with a static IP address to avoid that requests to the local DHCP server are recorded.

While assuming that such an individual could take all reasonable precautions to avoid suspicious evidences, an accurate investigation at the ISP side can reveal the unusual IP connection persisting for the overall duration of the alibi.

Remote Connection Through Remote Control Software An individual looking for an alibi can use a Remote Control software. To limit suspicious traces, he can use a portable software from a USB pendrive (e.g. TeamViewer Portable for Windows). However, as in the previous case, the IP connection to the Remote Control software produces non-removable unwanted evidences at the ISP side and on the routers along the network path.

In both case, it is important to obfuscate the auxiliary hardware such as the iKVM switch and the USB pendrive in order to not raise suspicions.

2.2 Manual vs Automation

The production of digital evidences in order to claim an alibi can be also considered an Anti-Forensics activity. Following the “manual approach”, an individual can forge his alibi generating digital evidences a-priori or a-posteriori to the alibi timeline. For example, he can manually modify the access time of a file in order to claim that he was writing a document at the time of the crime. This can be considered the “classic” Anti-Forensic approach. However, this approach produces evidences that are “local” to the criminal’s system and should not be always considered trusted by Courts.

With respect to manual techniques, the automation can act “at the same time” (or “during”) the crime is committed. It determines that the forged evidences can be *validated* by some trusted third parties. For example, the automation can activate the Internet connection and access the Facebook account of the criminal, so that both the ISP and Facebook will record its logon informations. These records can subsequently be claimed as evidences.

3 Undistinguishable Automated Production of Digital Evidence

In this paper it is discussed the approach of producing digital evidences by means of automated tools. It is also shown how such evidences are undistinguishable, upon a post-mortem forensic analysis, from those produced by the human behavior and therefore can be used in a legal case to claim a digital alibi. The typical actions performed by an human on his PC, which may be simulated by automated tools, are mouse clicks, pressure of keyboard keys, writing of texts, use of specific software, all interleaved with random timings.

There exists several automation tools which are useful to avoid boring, manual, repetitive, and error-prone tasks. That is, they speed up otherwise tedious, time-consuming tasks, avoiding the possibility of errors while doing those tasks. Applications of automation tools include data analysis, data munging, data extraction, data transformation, and data integration.

In this paper a new potential application of automation tools for the construction of a digital alibi is introduced. Some automation tools have generally the possibility to perform simple operations like:

- simulate keystrokes and mouse gestures;
- manage windows (e.g., activation, opening, closing, resizing);
- get information on and interact with edit boxes, check boxes, list boxes, combos, buttons, status bars;
- control time for operation (e.g., choose time to schedule each operation or choose time delay between consecutive tasks);

Automation tools usually provide much powerful functions. There are also libraries, modules and a high-level scripting language available for users and developers. However the basic and simple operations listed above are sufficient to automate tasks for the purpose of constructing a digital alibi. The list of tasks includes:

- *Web navigation.* Opening new tabs, new windows, new URL. Inserting username, password, text. Uploading or downloading files. These include interaction with social networks, and popular websites like Picasa, Dropbox, Gmail.
- *Files and folders.* Processing specific files, renaming them, working with folders.
- *Photos and images.* Processing photos, cropping images, creating thumbnails.
- *Music and audio files.* Play an audio file. Adjusting audio controls. Converting audio to text.
- *Compound files.* Create new text files, modifying (inserting and deleting) them, saving them. These include Office documents being processed by Word, Excel, and Powerpoint.
- *Computer applications.* Launching any application. For example, launching a browser or using email by opening unread messages and sending new messages with attachments.

– *Phone calls.* While it would be easy to simulate a phone call using IP telephony like Skype/VoIP, it is possible to make a phone call over the PSTN circuit or GSM mobile network by using additional hardware, as well as send SMS. For example, AT commands can be sent to a modem connected with the PC.

3.1 Digital Evidence of an Automation

An individual who intend to create an alibi should identify unwanted evidences that the deployed program leaves on the system, then implement a technique to avoid or remove such traces. The evidence of the automation strongly depends on the OS in which it is executed. As discussed later in this section, there exists two categories of unwanted traces that should be removed: execution traces and logon traces.

Execution Traces For any OS, the *process* is considered as the basic execution unit [21], and even the simplest OS provides mechanisms to trace the execution of each process it runs saving data such as executable name, the time was started, the amount of CPU was allocated during the execution, maximum resident size for virtual memory and so on. These records are generally referred to as “accounting data”. Depending on the OS, the execution of an automation generated with tools like AutoIt also leaves this kind of traces. For example, Windows stores accounting data in the Registry. In Linux, application logs are stored in the /var/logs directory and memory map of processes is maintained in the /proc directory. Most of recent OSs implements techniques like “Virtual Memory Allocation” and “Prefetch”, which also store data about programs on the filesystem. In order to forge a strong alibi, the accounting data regarding the automation should be removed. In Section 4 it is discussed how this task can be performed on a Windows environment.

Logon Traces Besides the data related to the process execution, another specific OS module is in charge of storing each user access to the system *logon data.* Normally this is done during login-logout phases and the module is supposed to record data such as local login time, local logout time, source address of the connection (if the operation was performed through the net) or the tty (the serial line) the user used to connect to the terminal both for local or modem access. Although it is possible to modify the files containing such records, there are several Digital Forensics tools that can verify the integrity of such files and, in this case, they should be considered meaningful.

3.2 Different Approaches to Unwanted Evidences Handling

The use of an automation tool produce some unwanted traces that can be detected by Digital Forensic analysis. These evidences should be removed in order to forge an alibi. There are basically two approaches that can be adopted to accomplish this task:

- *Avoid evidences a-priori.* The criminal can take several precautions in order to avoid as much unwanted evidences as possible. For example, he can disable some OS-specific mechanisms that record data about processes execution.
- *Remove evidences a-posteriori.* It is possible to adopt wiping techniques in order to remove the unwanted traces left by the automation on the system drive(s).

The most productive approach to avoid that a Digital Forensics analysis reveals suspicious evidences about an automation is to design an automation that leaves as less unwanted traces as possible. However, even using this approach, a separate solution should be adopted to address the problem of removing (or obfuscating) the file(s) implementing the automation itself.

There are some OS-specific precautions that can be taken to avoid unwanted evidences. They mostly regards the OS configuration. For example, in Windows it is possible to disable the Virtual Memory technique and the Prefetch mechanism in order to avoid that data about processes is stored on the filesystem, as well as application logging can be disabled in Linux.

Some OS-independent tricks can be also adopted to avoid unwanted traces, for example running the automation executable from a removable device avoiding to copy it on the hard disk. This approach could address the problem of obfuscating the file(s) implementing the automation. However, an external drive can leave traces regarding its use which should be considered.

Generally, it is not possible to completely avoid the accounting data. For example, in Windows it is not possible to disable the recording of program execution paths in the Registry, as well as in Linux it is not possible to avoid that memory maps of processes are stored on the filesystem. Moreover, if the automation program is stored on the hard disk, itself is an unwanted evidence that must be deleted. There are two approaches for handling traces that cannot be avoided: they can be *obfuscated* or *wiped*.

3.3 Removing Unwanted Digital Evidence of an Automation

Evidence of automation can be removed employing three different approaches.

Manual deletion. The individual who intend to generate the alibi can manually remove the unwanted evidences from the system. In particular, he has to delete all the system information regarding the automation. For example, in Windows it includes Registry entries, while in Linux the memory map files. The file(s) constituting the automation itself must be removed using wiping techniques.

Semi-automatic method. It is possible to further minimize the unwanted data that will be left on the drive running the automation executable by using a removable device (e.g. an USB pendrive or a CD-ROM). Using this approach, the individual do not have to wipe the file(s) of the automation from the drive. However, he should also remove all suspicious evidences recorder by the OS about its execution. Moreover, the trace left by the use of the removable device should be considered.

Automatic method. The deletion process of unwanted evidences can be itself part of the automation. It requires that the criminal is skilled enough to create a kind of shell script that firstly runs the automation part, then deletes all unwanted traces about its execution “recorded” on the OS, and eventually wipes itself.

In this work it is just addressed the semi-automatic deletion method, which is considered the simplest. An analysis about the automatic method has been done in a companion work [20].

3.4 Automation Development and Testing

The construction of an automation consists of two iterative phases: the development of the automation and the testing on the system. Together with the implementation of the automation, it is necessary to identify the unwanted evidences that the automation leaves on the system. It is possible to forge a digital alibi only if all (or at least the most suspicious) unwanted traces are detected and removed/obfuscated. First of all, the documentation about the OS and the used filesystem should be consulted and considered. However, the lack of documentation makes sometime necessary the use of software tools to identify unwanted evidences. For example, useful tools for this purpose are:

- *Process monitoring tools.* Some utilities to monitor the activities of the automation at execution time can be used. For example Process Monitor [15], which is an advanced monitoring tool for Windows that shows real-time filesystem, Registry and process/thread activity.
- *Digital forensic tools.* Digital forensic tools can be used in a post-mortem fashion in order to to analyze the system’s drive(s) and detect traces leaved by the execution of the automation.

Design of the Automation The automation itself must be developed and tested to verify if it acts correctly and does not leaves suspicious traces on the target system. In most cases the automation must be extensively tested before used for a so sensible task which is the creation of a false digital alibi. In fact, an automation created using software tools is strictly connected to the running environment. For example, when using AutoIt under Microsoft Windows, the mouse movements and clicks must be specified using absolute coordinates (x,y), therefore the different positions of an element on the screen result in a different behavior of the automation. Due to these considerations, the automation must be tested on a system that has the same appearance as the target system (screen resolution, windows position, desktop theme, icon size, etc.).

The automation must be also extensively tested in order to identify (and consequently minimize) all the unwanted traces leaved on the system by its execution, using the methodologies discussed above. Moreover, it is necessary to verify the effectiveness of the deletion method used to remove the automation from the system after its execution. It means that the effective construction of the automation is an iterative-incremental process constituted by two phases: the implementation of the actions that should be automatically executed and its testing.

Unwanted Evidences of the Automation Development The preparation of the automation can leave some unwanted traces: the OS, in fact, typically records recently opened files and applications. For example, Microsoft Windows stores these information in the Registry, which can be only modified by the Administrator user and the modifications take effect only after a system reboot.

It is possible to employ some workarounds to avoid most of suspicious traces about the development phase.

- *Virtual machine.* A virtual machine running an identical copy of the OS of the target system can be used in order to test the automation. This technique does not leave any unwanted trace on the target system except the file containing the virtual machine image and traces that the virtual machine itself has been powered on.
- *Live OS.* A live CD or live USB version of the target OS can be used in order to develop and test the automation. This technique does not leave any unwanted trace on the hard disk because the live OS only uses the central memory for all his operations.
- *Another system.* The automation can be simply developed and tested on another PC running the same OS with a similar configuration. Subsequently, the program responsible of the automation can be copied on a removable media and launched directly from there. In this case, the entire secondary PC must be obfuscated in order to avoid any forensic analysis on it.
- *External device.* It is possible to use portable software in order to implement and test the automation from an external (local or remote) device. In this case, it is possible to configure the OS in order to avoid that it records meaningful unwanted evidences, such as accounting data of the used programs. Following this approach, the development of the automation takes place on the same system where it will be deployed.

3.5 Additional Cautions

A recent paper [10] explain how it is possible to recognize the persons who have used a computer analyzing the bacteria left by their fingertips on the keyboard and mouse. The imprint left by the bacteria on the keys and mouse persists for more than two weeks. This is potentially a new tool for forensic investigation. Obviously, investigators should use gloves before examining the computer. This kind of analysis can be exploited by a criminal to validate his digital alibi. If the suspected made sure of being the only one to use the computer, the defender Advocate can request a forensic analysis within two weeks, which will confirm that bacterial traces on the keyboard and mouse are those of the suspect.

People have their habits and then follow a predictable pattern. For example, it may be usual for the suspect to connect to the Internet during the morning, access his mailbox, browse some websites and work on his thesis. In practice, the behavior of the suspected inferred from his digital alibi must be not very different from his typical behavior. That is, suspicious traces must not be discovered by an hypothetical Anomaly Detection analysis. The connection time, the amount of transmitted and received bytes, the amount of access to social networks, and other actions must be similar to those of the previous days according to the habits of the accused. The same behavior inferred from the digital evidences may be repeated on other days with some randomization. The testing phase of the automation can already give regularity to the behavioral pattern of the suspect and therefore may be useful in order to guard against eventual anomaly detection analysis [13].

4 Case Study

In this section a case study is analyzed: the development of an automation for producing a digital alibi in Microsoft Windows XP with Service Pack 3 and Microsoft Windows Vista. The script language chosen to implement the automation is AutoIt v3 for Windows [11]. AutoIt has been chosen for this experiment as it is a powerful and easy-to-use tool which does not require a deep knowledge of programming languages, and so can be used by non-skilled users.

4.1 AutoIt

AutoIt is a freeware automation language for Microsoft Windows. The syntax of AutoIt is like the BASIC family of languages. An AutoIt automation script can be compiled into a compressed, stand-alone executable which can be run on computers that do not have the AutoIt interpreter installed.

A very basic knowledge of the AutoIt scripting language is required in order to create a fully fledged automation program. The main functions used in the experiment are listed below:

- *Run("path/to/external/program")*
  Runs the external program pointed by the path;
- *Send("sequence_of_keys")*
  Sends simulated keystrokes to the active window;
- *MouseClick("mouse_button", x_coordinate, y_coordinate, number_of_clicks)*
  Performs a mouse click operation, simulating the pressure of a specific mouse button on the position specified by the coordinates;
- *WinWaitActive("title")*
  Pauses the script execution until the requested window is active;
- *Sleep(delay)*
  Pauses the script execution for a number of milliseconds equal to the delay.

4.2 AutoIt Script Example

Several AutoIt scripts have been created as proof of concept, which implement different number of actions and alibi timelines. The scripts have been compiled into standalone executables and do not require that the AutoIt interpreter is installed on the target system. Generally, for a sample source script of 300 lines (as those listed below) the resulting executable file is about 200Kb.

In order to show how simple is the construction of an automation using the AutoIt scripting language, a script excerpt is presented which simulates the actions of interacting with the webpages of BBC and Facebook. The automation opens the Firefox web browser and inserts the URL `http://www.bbc.co.uk/` in the location bar, then simulates the pressure of the ENTER key which lets the browser load the website. After the web page has been loaded, it clicks on a link and simulates the human activity of reading page contents waiting some minutes. Subsequently, the script simulates an access to Facebook loading the `http://www.facebook.com/` website and inserting the access credentials. The main part of the relative source code is listed below.

```
...
Run ("C:\Program files\Mozilla Firefox\
      firefox.exe")
Send ("^t" )
Send ("http://www.bbc.co.uk/")
Send ("{ENTER} )
WinWaitActive ("BBC")
MouseClick ("left","706","160","1")
WinWaitActive ("BBC SPORT")
Sleep (6500)
...
```

```
...
Send ("^t")
Send ("http://www.facebook.com/")
Send ("{ENTER}")
WinWaitActive("Facebook")
Send ("{TAB}")
Send ( "castiglione@acm.org")
Send ("{TAB}")
Send ("p4$$w0rd")
Send ("{ENTER}")
...
```

4.3 Unwanted Traces

In the presented case study the approach of avoiding as much unwanted evidences as possible has been followed (see Section 3.2). In this Subsection there are described the unwanted traces detected in the experiment and some simple tricks to avoid them. The only trace that remains on the filesystem is the automation executable file, which has to be deleted manually. For a more complete discussion about the deletion, see Subsection 4.4.

Windows Registry Microsoft Windows contains significant amounts of digital evidence that enables an investigator to reconstruct events that took place on the machine before it was seized. The Windows Registry, in particular, contains a wealth of information about the configuration and use of a computer [12].

In details, Windows records in the Registry data relative to programs executed on the system. If an executable is launched using the File Explorer mechanism, its complete pathname is recorded in the following Registry keys:

```
HKEY_CURRENT_USER\Software\Microsoft\Windows\ShellNoRoam\MUICache
HKEY_USERS\S-1-5-21-2025429265-688789844-854245398-1003\Software\Microsoft\Windows\
 \ShellNoRoam\MUICache
```

Otherwise, if an executable is launched using the DOS command prompt, only the value `x:\windows\system32\cmd.exe` (`x:` is the drive where Windows is installed) is recorded in the following Registry key:

```
HKEY_CURRENT_USER\Software\Microsoft\Windows\ShellNoRoam\MUICache
```

As it is not possible to completely avoid the recording of such evidence, in this experiment the execution of the automation has been *obfuscated* running it from a command prompt. In this case the string recorded in the Registry (`x:\windows\system32\cmd.exe`) does not reveal any information regarding the automation. In fact, the shell may have been used for launching any other command (e.g., a `ping`). According to authors experience, a further Digital Forensic analysis does not reveal any other meaningful information about the automation in the Registry.

Filesystem Windows XP and latter versions implement the Prefetch mechanism [18]. The *prefetcher* is a component of the *memory manager* that speeds up the Windows boot process, and shortens the amount of time it takes to start up programs. It attempts to accelerate application and boot launch times respectively by monitoring and adapting to usage patterns over periods of time and loading the majority of the files and data needed by them into memory so that they can be accessed very quickly when needed.

Auxiliary files (with .pf extension) containing information about used programs are stored on the filesystem in the directory `x:\WINDOWS\Prefetch`. In the conducted experiment this mechanism has been disabled in order to avoid that unwanted evidences of the automation program being stored on the hard disk by the *prefetcher*. This has been accomplished by setting the following Registry key value equal to zero:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SessionManager\MemoryManagement\
 PrefetchParameters
```

Disabling the *prefetch* mechanism could not be considered a suspicious action. In fact, this configuration can sometimes reduce the hard disk utilization and is quite used among the Windows users. Moreover, there exists many tweaking tools for optimizing the performance of a PC that, among other tasks, disable the *prefetch* feature.

Virtual Memory Another mechanism implemented by Microsoft Windows, which must be disabled in order to avoid unwanted evidences on the filesystem, is the Virtual Memory [21]. In order to free up space in memory, an operating system with a virtual memory capability transfers data that is not immediately needed from memory to the hard disk; when that data is needed again, it is copied back into memory. In Microsoft Windows there is a specific file on the filesystem used for swapping such data, namely `pagefile.sys`, which could also memorize information relative to the automation.

In the conducted case study, the Virtual Memory mechanism has been disabled setting the virtual memory size equal to zero in the System Properties of Windows (`Control Panel->Advanced->Performance->Settings->Advanced->Virtual Memory`).

Disabling the virtual memory can sometimes improve the system performance as well as increase the hard disk available space. Several Windows users use this customization, therefore could not be considered suspicious by investigators.

4.4 Wiping

In the case study, some Windows-specific settings have been modified in order to avoid that the OS would record meaningful evidences about the execution of the automation script. The only potential unwanted evidence that remains available is the compiled AutoIt script implementing the automation.

It is important to note that deleting a file using the OS-specific functions does not completely remove the file from the drive. In fact, the sectors that were occupied by a file become available for a new writing operation, but the previous data remains on the disk until it gets overwritten. Also a rewriting of these sectors does not guarantees the complete data deletion. It mostly happens on magnetic devices such as hard disks, where electromagnetic traces may remain even after several rewritings.

The amount of rewritings necessary to perform the secure wiping of data on a drive is a controversial theme [7], [8], [9], [19]. Considering the NIST Special Publication 800-88 [22], which asserts that "Studies have shown that most of today's media can be effectively cleared by one overwrite", the approach adopted in this experiment consists of a single rewrite. However, the replacement of this technique with a more paranoid one, consisting of multi-rewritings, can be straightforward implemented.

In the presented study, a *semi-automatic approach* for deleting the automation data has been adopted as it is more simple to perform for non-skilled people. In practice, an USB pendrive has been formatted and almost completely filled with audio and video files, then the automation script has been also copied on it. The USB pendrive has been plugged into the PC two days before executing the automation. After the automation execution, the script has been deleted (using the "classic" Windows `del` command from the shell) and the pendrive has been completely filled by copying on it additional multimedia files. These actions should guarantee that the traces left by the pendrive in the Registry are not suspicious as it was plugged in two days before the alibi timeline. Moreover, filling the pendrive after the deletion of the script should overwrite all sectors previously occupied by the automation script.

5 Conclusions

Computers are becoming more and more important in our society. People use PCs to accomplish a large set of activities, related to their work or personal purposes. A PC may contain lot of information about the people which use it, such as logon data, used applications, visited websites and so on. As a result, the number of Court cases involving digital evidence is increasing.

In this paper it has been shown how simple could be the set up of digital evidences in order to provide a criminal with a false digital alibi. In particular, an automated method of generating digital evidences has been discussed. Using this approach, it is possible to get a digital alibi involving some trusted third parties. In fact, the automation could, for example, activate the Internet connection by means of an ISP, access a Facebook account, send an email and so on, leaving traces on the respective servers. The problem of wiping unwanted evidences left by the automation has been addressed. Finally, a real case study has been presented in order to demonstrate that the implementation of such methodologies is not an hard task and can be performed even by non-skilled people.

Objective of this work is to highlight the need of an evolution in approaching legal cases that involve digital evidences. In particular, it is necessary to define accurate rules for the legal investigations that include the following keypoints:

- verdicts should not be based only on digital evidences;
- digital evidences should always be part of a larger pattern of behavior reconstructed by means of traditional forensics investigations;
- criminal investigation divisions should include digital forensics experts which constantly upgrade their knowledge in order to face the evolution of anti-forensic techniques.

Experiments on various OSs have been and are being conducted in order to prove that the techniques described in this paper really produce digital evidences undistinguishable from those produced by a human, which could be used to forge a digital alibi. Moreover, a fully automated approach of deleting evidences from a drive is analyzed in a companion work [20].

Acknowledgements

The authors would like to thank friends from IISFA (International Information System Forensics Association) for their support, their valuable suggestions and useful discussions during the research phase. In particular to Dr. Gerardo Costabile (President of IISFA Italian Chapter), Dr. Francesco Cajani (Deputy Public Prosecutor High Tech Crime Unit Court of Law in Milano, Italy), Dr. Mattia Epifani and Dr. Litiano Piccin (Head of Education Area within IISFA Italian Chapter). A warm thank goes to Paolo Iorio for the many discussions during the preparation of his thesis.

References

1. *Internet world stats*, June 30, 2010, http://www.internetworldstats.com/stats.htm
2. Msnbc News, *Facebook message frees NYC robbery suspect*, November 12, 2009. http://www.msnbc.msn.com/id/33883605/ns/technology_and_science-tech_and_gadgets/

3. The New York Times, *I'm Innocent. Just Check My Status on Facebook*, November 12, 2009. http://www.nytimes.com/2009/11/12/nyregion/12facebook.html?_r=1
4. CNN, *Facebook status update provides alibi*, November 12, 2009. http://www.cnn.com/2009/CRIME/11/12/facebook.alibi/index.html
5. S. Vitelli, GUP presso il Tribunale di Vigevano, *Sentenza del processo Stasi*, http://static.repubblica.it/laprovinciapavese/pdf/SENTENZA_STASI.pdf, 17 Dicembre 2009 (in Italian language)
6. F. Bravo, *La computer forensics nelle motivazioni della sentenza sull'omicidio di Garlasco*, http://internetsociety.wordpress.com/2010/03/16/la-computer-forensics-nelle-motivazioni-della-sentenza-sullomicidio-di-garlasco/, 16 Marzo 2010 (in Italian language)
7. U.S. Department of Defense, *DoD Directive 5220.22, National Industrial Security Program (NISP)*, http://www.dtic.mil/whs/directives/corres/html/522022m.htm, 28 February, 2010
8. P. Gutmann, *Secure Deletion of Data from Magnetic and Solid-State Memory*, Sixth USENIX Security Symposium Proceedings, San Jose, California, http://www.cs.auckland.ac.nz/∼pgut001/pubs/secure_del.html, July 22-25, 1996.
9. P. Gutmann, *Data Remanence in Semiconductor Devices*, 2001 Usenix Security Symposium, Washington DC, http://www.cypherpunks.to/∼peter/usenix01.pdf, August 13-17, 2001.
10. N. Fierer, C.L. Lauber, N. Zhou, D. McDonald, E.K. Costello and R. Knight, *Forensic identification using skin bacterial communities*, Proceedings of the National Academy of Sciences, Abstract, http://www.pnas.org/content/early/2010/03/01/1000162107.abstract, March, 2010.
11. J. Bennett, *AutoIt v3.3.6.0*, http://www.autoitscript.com/autoit3/, March 7, 2010.
12. V. Mee, T. Tryfonas and I. Sutherland, *The Windows Registry as a forensic artefact: Illustrating evidence collection for Internet usage*, Digital Investigation, vol. 3, pp. 166-173, 2006
13. V. Chandola, A. Banerjee and V. Kumar, *Anomaly detection: A survey*, ACM Computing Surveys, vol. 41, n. 3, July 2009
14. Honorable D.E. Shelton; *The 'CSI Effect': Does It Really Exist?*, National Institute of Justice Journal No. 259, March 17, 2008
15. *Microsoft Sysinternals Process Monitor*, http://technet.microsoft.com/en-us/sysinternals/bb896645
16. *Merriam-Webster online dictionary*, http://www.merriam-webster.com/
17. Wikipedia, *KVM switch*, http://en.wikipedia.org/wiki/KVM_switch
18. H. Carvey, *Windows Forensics Analysis, Second Edition*, Syngress, 2009
19. W, Craig, K. Dave and S.R.S. Shyaam, *Overwriting Hard Drive Data: The Great Wiping Controversy*, Vol. 5352 of Lecture Notes in Computer Science (Springer Berlin / Heidelberg), pp. 243-257, December 2008
20. A. Castiglione, G. Cattaneo, A. De Santis and G. De Maio, *Automatic and Selective Deletion Resistant Against Live Forensics Analysis*, Submitted, April 2011
21. A. Silberschatz, P. B. Galvin and G. Gagne, *Operating System Concepts, 7th Edition*, Wiley, 2004
22. *NIST Special Publication 800-88: Guidelines for Media Sanitization*, p. 7, 2006