Automated Construction of a False Digital Alibi

Alfredo De Santis, Aniello Castiglione, Giuseppe Cattaneo, Giancarlo De Maio, Mario Ianulardo

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] Computer Crime Lawyer, Italy
marioianulardo@codicieleggi.it

Abstract. Recent legal cases have shown that *digital evidence* is becoming more widely used in court proceedings (by defense, accusation, public prosecutor, etc.). Digital tracks can be left on computers, phones, digital cameras as well as third party servers belonging to Internet Service Providers (ISPs), telephone providers and companies that provide services via Internet such as YouTube, Facebook, Gmail.
This work highlights the possibility to set up a false digital alibi in a fully automatic way without any human intervention. A forensic investigation on the digital evidence produced cannot establish whether such traces have been produced through either human activity or by an automated tool. These considerations stress the difference between digital and physical - namely traditional - evidence. Essentially, digital evidence should be considered relevant only if supported by evidence collected using "traditional" investigation techniques. The results of this work should be considered by anyone involved in a Digital Forensics, due to it demonstrating that court rulings should not be based only on digital evidence, with it always being correlated to additional information provided by the various disciplines of Forensics Sciences.

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

1 Introduction

1.1 The Digital Evidence

The use of digital technology is rapidly growing. The number of Internet users in the world is almost 2 billion, with a penetration of 28.7% of the world population [1]. As a consequence, more and more crimes are performed on the Internet

⋆ Corresponding author: Aniello Castiglione, ✉ Dipartimento di Informatica "*R.M. Capocelli*" - Università degli Studi di Salerno, Via Ponte don Melillo, I-84084 Fisciano (SA), Italy. ☎: +39089969594, 🖷: +39089969821, 🖃: castiglione@{ieee,acm}.org

or have something to do with digital equipment. For these reasons, there is an increase in the amount of digital evidence being used courtrooms around the world. Consequently, courts are now becoming concerned about the admissibility and probative value of digital evidence. Even if digital devices have not been directly used by an individual who has been indicted for a crime, they can be subject to forensic investigations in order to collect useful traces about the suspect activities, in order to be either cleared or charged with an offense. The 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, as well as 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 any other information concerning the connected clients. These servers can be located in remote countries, with different national laws being an obstacle for the acquisition of digital evidence during the investigation.

Digital traces are also *immaterial*. It is well known that all digital data present on a device are mere sequences of one and zero. These data can be modified by anyone who has enough privileges on that device.

1.2 The Digital Alibi

Computers cannot only be involved in as well as contain the proof of crimes, but they can also be an *alibi* for the defense of anyone who is under accusation. In the Latin the word "*alibi*" is an adverb meaning "*in or at another place*". According to the Merriam-Webster online dictionary [14], alibi is "the plea of having been at the time of the commission of an act elsewhere than at the place of commission".

There are several examples of legal proceedings in which digital evidence has been considered an alibi that contributed to exonerating the accused. These include the interesting case of Rodney Bradford ([2], [3]), accused of armed robbery and released thanks to his digital alibi, consisting of activities on his Facebook account. The Erb Law Firm, a corporation of lawyers in Philadelphia, emphasized that "Facebook Can Keep You Out of Jail" [21]. Another example is the Italian case of "Garlasco" ([4]), in which the proceedings of the first instance ended with the acquittal of Alberto Stasi, the main suspect in the murder of his girlfriend Chiara Poggi. Digital evidence of the work activity left on his laptop during the committing of the crime confirmed his digital alibi.

Identifying the true originator of digital evidence is a very hard task. In fact, it is possible to trace the owner of a digital device, but the digital evidence itself does not contain any information on *who* or *what* has produced it.

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 evidence could be claimed in a court to forge a valid alibi. The direct consequence of this result is that the forensic analysis in

legal cases should focus not only on the retrieval and analysis of digital evidence, but also on the identification of its author.

The paper is organized as follows: in Section 2 various approaches of forging a false digital alibi are discussed. In Section 3 the methodology of forging a false digital alibi creating a fully automated tool is presented and analyzed. In Section 4 a case study on Microsoft Windows systems is reported. Finally, this paper ends with the authors conclusions in Section 6.

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 evidence. 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 evidence on behalf of another person (e.g. accessing his mailbox, leaving messages on Facebook, etc.). This technique does not require any particular skill. However, the presence of another person could produce unwanted non-digital (e.g. biological) evidence which can be revealed by traditional forensic investigation techniques.

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

- *Remotization.* In order to forge a digital alibi by themselves, it is necessary to produce evidence 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 evidence.
- *Automation.* The automation method 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 evidence.

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 intends to create an alibi can use a KVM over IP switch (iKVM) [15] to control his PC remotely. This technique does not require any suspicious software to be installed. However, the individual must take some precautions to limit the amount of unwanted traces. For example, he should configure the iKVM with a static IP address in order to avoid that requests to the local DHCP server

are recorded. While assuming that he could take all reasonable precautions to avoid suspicious evidence, 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 Someone looking for an alibi can use a Remote Control software. To limit suspicious traces, he can use a portable software from a USB flash drive (e.g. TeamViewer Portable for Windows), but traces of such softwares on the host computer may also be found. However, as in the previous case, the IP connection to the Remote Control software produces non-removable unwanted evidence at the ISP side as well as on the routers along the network path. In both cases, in order to try to fool a digital investigator, an unwary person should obfuscate the auxiliary hardware such as the iKVM switch and the USB flash drive in order to not raise any suspicion.

2.2 Manual vs Automation

The production of digital evidence for an alibi can be considered an Anti-Forensics activity. Following the "manual" approach, an individual can forge his alibi generating digital evidence a-priori or a-posteriori to the alibi timeline. For example, he can manually modify the access time of a file in order to pretend he was writing a document at the time of the crime. This can be considered the "classic" Anti-Forensic approach. However, this approach produces evidence that is "local" to the system of the suspected person and should not always be considered trusted by the Courts.

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

3 Undistinguishable Automated Production of Digital Evidence

In this paper the production of digital evidence by means of automated tools is discussed. It is also shown how this evidence is undistinguishable, upon a post-mortem forensic analysis, from that produced by the human behavior and therefore can be used in a legal case to claim a digital alibi. The typical actions performed by a human on a PC, which may be simulated by automated tools, are mouse clicks, the pressing of keyboard keys, the writing of texts, the use of specific software, which are all separated by random timings.

There are several automation tools used to avoid boring, manual, repetitive, and error-prone tasks. They speed up otherwise tedious, time-consuming

tasks, thus avoiding the possibility of errors while doing them. Applications of automation tools include data analysis, data munging, data extraction, data transformation as well as data integration.

In this paper, a new potential application of automation tools for the construction of a digital alibi is introduced. Some automation tools generally have the possibility to perform simple operations such as 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, but 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 URLs. Inserting username, password, text. Uploading or downloading files. These include interaction with social networks, and popular websites such as 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 a text message. For example, AT commands can be sent to a modem connected with a PC.

3.1 Digital Evidence of an Automation

An individual who intends to create an alibi should identify unwanted evidence 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 are 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 [19], and even the simplest OS provides mechanisms to trace the execution of each process it runs saving data such as executable name, the time it was started, the amount of CPU 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 such as AutoIt also leaves this kind of trace. For example, Windows stores accounting data in the Registry. In Linux, application logs are stored in the `/var/logs` directory and the memory map of the processes is maintained in `/proc`. Most of the more recent OSes implement techniques such as "Virtual Memory Allocation" and "Prefetch", which also store data about programs on the filesystem.

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 Evidence Handling

The use of an automation tool produces some unwanted traces that can be detected by digital forensics analysis. In order to forge an alibi all this evidence should be removed. There are basically two approaches that can be adopted to accomplish this task.

Avoid Evidence a-priori The individual can take several precautions in order to avoid as much unwanted evidence as possible. Sometimes, when it is not possible to completely delete some evidence, an *a-priori obfuscation* strategy could be used in order to avoid any logical connection between the evidence and the automation process, in a way that it could have been the result of "normal" operations within the system. For example, it is possible to disable some OS-specific mechanisms that record data about process execution. The fact that such mechanisms have been disabled could depend on either a direct user operation or an optimization software which is very common to speed-up the operating system.

Remove Evidence 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). Sometimes it is not possible to wipe all unwanted data, which makes an

a-posteriori obfuscation strategy necessary in order to avoid logical connections between these data and the automation tool.

The most productive approach to avoid that a digital forensics analysis reveals suspicious evidence about an automation is to design it in a way 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 in order to avoid unwanted evidence. They mostly regard OS configuration. For example, in Windows it is possible to disable the Virtual Memory and the Prefetch mechanisms in order to avoid that data about processes is stored on the filesystem, as well as application logging being possible to disable 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 onto 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. 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. It is not possible to avoid that memory maps of processes are stored on the filesystem in Linux. In such cases, traces that cannot be avoided should be wiped or obfuscated. Moreover, if the automation program is stored on the hard disk, it is unwanted evidence that must be deleted.

3.3 Removing Unwanted Digital Evidence of an Automation

Evidence of automation can be removed employing three different approaches.

Manual Deletion. The individual who intends to generate the alibi can manually remove the unwanted evidence from the system. In particular, he/she 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 flash drive or a CD-ROM). Using this approach, the person does not have to wipe the file(s) of the automation from the drive. However, he/she should also remove all suspicious evidence "recorded" 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 evidence can itself be part of the automation. It requires that the individual who prepares the automation is skilled enough to create a shell script that firstly runs the automation part, then deletes all unwanted traces about its execution "recorded" on the OS, and eventually wipes itself. This work deals with the semi-automatic deletion method, due to it being considered the simplest. An analysis of the automatic method has been carried out in another study [18].

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. Along with the implementation of the automation, it is necessary to identify the unwanted evidence 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 the use of software tools to identify unwanted evidence sometime necessary. 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 [13], 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 drive(s) and detect traces left 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 being used for such a 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 also be extensively tested in order to identify (and consequently minimize) all the unwanted traces left 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.

Unwanted Evidence 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 this information in the Registry, which can only be modified by the Administrator, with the modifications taking effect only after a system reboot.

It is possible to employ some workarounds to avoid most of the 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 traces on the target system except for the files 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 traces on the hard disk because the live OS only uses the central memory for all its 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 for the automation can be copied onto 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 evidence, 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 [7] explains how it is possible to recognize who has 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 device. This kind of analysis can be exploited by an individual to validate his digital alibi. If the suspect made sure of being the only one to use the computer, the defending lawyer 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 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 suspect inferred from his digital alibi must be not very different from his typical behavior. Suspicious traces must not be discovered by an hypothetical Anomaly Detection analysis. 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 [11] [9].

4 Case Study

In this section a case study is analyzed with it being the development of an automation to produce 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 [8]. AutoIt has been chosen for this experiment due to it being a powerful and easy-to-use tool which does not require a detailed knowledge of programming languages, and therefore can be used by unskilled users.

4.1 AutoIt

AutoIt is a freeware automation language for Microsoft Windows. The syntax of AutoIt is similar to BASIC language. 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 an external program;
- *Send("sequence_of_keys")* Sends simulated keystrokes to the active window;
- *MouseClick("mouse_button", x_coordinate, y_coordinate, number_of_clicks)* Performs a mouse operation, simulating the pressure of a mouse button;
- *WinWaitActive("title")* Pauses until the requested window is active;
- *Sleep(delay)* Pauses the script for *delay* milliseconds.

4.2 AutoIt Script Example

Several AutoIt scripts have been created as proof of concept, which implement a 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 the resulting executable file is about 200Kb.

In order to show how simple is the construction of an automation is using the AutoIt scripting language, a script excerpt is presented which simulates the actions of interacting with the webpages of the 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 pressing 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","295","355","1")
WinWaitActive ("Sport")
Sleep (12940)
...
```

```
...
Send ("^t")
Send ("http://www.facebook.com/")
Send ("{ENTER}")
WinWaitActive("Facebook")
Send ("{TAB}")
Send ("castiglione@ieee.org")
Send ("{TAB}")
Send ("password")
Send ("{ENTER}")
...
```

4.3 Unwanted Traces

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

Windows Registry Microsoft Windows contains significant amounts of digital evidence that enables an investigator to reconstruct the 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 [10].
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:

1) `HKEY_CURRENT_USER\Software\Microsoft\Windows\ShellNoRoam\MUICache`
2) `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` is recorded in the above Registry key number `1)`.

Due to it not being 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 to launch any other command (e.g., a `ping`). According to the authors' experience, a further digital forensics analysis does not reveal any other meaningful information about the automation in the Registry.

Filesystem Windows XP and subsequent versions implement the Prefetch mechanism [16]. The *prefetcher* is a component of the *memory manager* that

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 the 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 experiment, this mechanism has been disabled in order to avoid that unwanted evidence of the automation program was stored on the hard disk by the *prefetcher*. This has been accomplished by setting to zero the following Registry key value:
`3) 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 hard disk utilization and is often used among the Windows users. Moreover, there are 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 evidence on the filesystem, is the Virtual Memory [19]. In order to free up space in the memory, an operating system with a virtual memory capability transfers data that is not immediately needed from the memory to the hard disk. When that data is needed again, it is copied back into the 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 this case study, the Virtual Memory mechanism has been disabled by setting the virtual memory size equal to zero in the system properties of Windows using the following navigation: `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 space available. Several Windows users employ this customization, with it therefore not being 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 evidence 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 is overwritten.

The amount of rewritings necessary to perform secure wiping of data on a drive is a controversial issue [5], [6], [17]. Considering the NIST Special Publication 800-88 [20], which claims 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 rewriting. However, the replacement of this technique with a more paranoid one, consisting of multi-rewritings, can be quite straightforwardly implemented.

In this study, a *semi-automatic* approach for deleting the automation data has been adopted, due to it being easier to carry out by unskilled users. In practice, an USB flash drive has been formatted and almost completely filled with audio and video files, then the automation script has been also copied onto it. The USB flash drive 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 `cmd.exe` shell) and the USB flash drive has been completely filled by copying additional multimedia files onto it. These actions should guarantee that the traces left by the USB flash drive in the Registry are not suspicious as it was plugged in two days before the alibi timeline. Moreover, filling the USB flash drive after the deletion of the script should overwrite all sectors previously occupied by the automation script.

5 The Digital Alibi in Court

In some countries, it is a common practice that, in legal proceedings, digital evidence are vetted by digital forensics experts, which assess its trustworthiness according to the *Five Ws Rule* (Who, What, When, Where, Why).

It is well known that a human accomplice could be engaged in order to forge an alibi, but this approach is hazardous since he could avow his actions or even blackmail the suspect. Consequently, if the individual interested in producing the alibi has enough technical skills, he may prefer to use an automation in order to forge a digital alibi. In this case, the absence of accomplices and the creation of ad-hoc digital evidence, undistinguishable post-mortem from those left by ordinary human behavior could produce a "perfect alibi".

In fact, the Court would be in a delicate situation if the digital alibi confirms that the suspect was using his PC while the crime was being committed:

- if on the *locus committi delicti* (i.e. the crime scene) there is no evidence related to the suspect (biological traces, witnesses, etc.), the Court could consider decisive the probative value of the digital alibi and acquit the suspect;
- on the contrary, if on the crime scene biological traces referable to the suspect have been detected (left, for example, during previous contact with the victim), the probative value of the digital alibi should be carefully weighed.

After this paper, the technical consultants which carry out any form of Digital Forensics analysis should consider the hypothesis that the suspect might have used an automation to forge his digital alibi. A technical consultant, aware of such a possibility, has to carefully analyze the exhibits and look for eventual evidence left by an incorrect implementation of the automation process.

In general, criminal investigation divisions should include Digital Forensics experts who constantly update their knowledge and understanding in order to face the evolution of Anti-Forensics techniques. This is an additional argument on the importance of scientific knowledge for the expert testimony in a Court, according to the rule 702 of the "Federal Rules of Evidence" [23] and to the "Daubert Test" [22].

6 Conclusions

A PC may contain lot of information about the people who 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 the set up of digital evidence could be in order to provide an individual with a false digital alibi. In particular, an automated method of generating digital evidence has been discussed. Using this approach, it is possible to claim 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 their respective servers. The problem of avoiding unwanted evidence 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 a hard task and can even be carried out by unskilled users.

Experiments on various OSes have been and are being conducted in order to prove that the techniques described in this paper really do produce digital evidence that is undistinguishable from those produced by a human, which could be used to forge a digital alibi. Moreover, a fully automated approach of deleting evidence from a drive is analyzed in a companion work [18].

The main goal of this work is to stress the need of an evolution in approaching legal cases that involve digital evidence. Evidently, a legal investigation case should not only rely on digital evidence to pass judgement, but should also consider it to be part of a larger pattern of behavior reconstructed by means of traditional forensics investigations. In conclusion, the plausibility of a digital alibi should be verified *cum grano salis*.

Acknowledgements

The authors would like to thank their friends from IISFA (International Information System Forensics Association) for their support, their valuable suggestions and useful discussions during the research phase. In particular to Gerardo Costabile (President of IISFA Italian Chapter), Francesco Cajani (Deputy Public Prosecutor High Tech Crime Unit Court of Law in Milano, Italy), Mattia Epifani and Litiano Piccin of the 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. D. Beltrami, 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
3. V. Juarez, CNN, *Facebook status update provides alibi*, November 12, 2009. http://www.cnn.com/2009/CRIME/11/12/facebook.alibi/index.html
4. Xomba: A Writing Community, *Garlasco, Alberto Stasi acquitted*, http://www.xomba.com/garlasco_alberto_stasi_acquitted, December 2009
5. U.S. Department of Defense, *DoD Directive 5220.22, National Industrial Security Program (NISP)*, 28 February, 2010
6. P. Gutmann, *Secure Deletion of Data from Magnetic and Solid-State Memory*, Sixth USENIX Security Symposium Proceedings, San Jose, California, July 22-25, 1996.
7. 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, March 2010.
8. J. Bennett, *AutoIt v3.3.6.0*, http://www.autoitscript.com/autoit3/, March 7, 2010.
9. G. Di Crescenzo, A. Ghosh, A. Kampasi, R. Talpade, and Y. Zhang, Detecting anomalies in active insider stepping stone attacks, Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, vol. 2, no. 1, pp. 103-120, 2011
10. V. Mee, T. Tryfonas and I. Sutherland, *The Windows Registry as a forensic artefact: Illustrating evidence collection for Internet usage*, Journal of Digital Investigation, Elsevier, vol. 3, issue 3, pp. 166-173, September 2006
11. V. Chandola, A. Banerjee and V. Kumar, *Anomaly detection: A survey*, ACM Computing Surveys, vol. 41, n. 3, pp. 15:1–15:58, July 2009
12. D.E. Shelton; *The 'CSI Effect': Does It Really Exist?*, National Institute of Justice, journal No. 259, March 17, 2008
13. M. Russinovich and B. Cogswell, *Microsoft Sysinternals Process Monitor*, http://technet.microsoft.com/en-us/sysinternals/bb896645, April 13, 2011
14. *Merriam-Webster Dictionary*, http://www.merriam-webster.com/dictionary/alibi
15. Wikipedia, *KVM switch*, http://en.wikipedia.org/wiki/KVM_switch
16. H. Carvey, *Windows Forensics Analysis, Second Edition*, Syngress, 2009
17. 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
18. A. Castiglione, G. Cattaneo, G. De Maio and A. De Santis, *Automatic, Selective and Secure Deletion of Digital Evidence*, Submitted, April 2011
19. A. Silberschatz, P. B. Galvin and G. Gagne, *Operating System Concepts, 7th Edition*, Wiley, 2004
20. *NIST Special Publication 800-88: Guidelines for Media Sanitization*, p. 7, 2006
21. The Erb Law Firm, *Facebook Can Keep You Out of Jail*, November 2009, http://www.facebook.com/note.php?note_id=199139644051
22. Margaret A. Berger, *What Has a Decade of Daubert Wrought?*, in: American Journal of Public Health, Vol. 95 No. S1, pp. S59-S65, July 2005
23. U.S. House of Representative, *Federal Rules of Evidence*, December 2006, http://afcca.law.af.mil/content/afcca_data/cp/us_federal_rules_of_evidence_2006.pdf