Counterfeiting and Defending the Digital Forensic Process

Álvaro Botas and Ricardo J. Rodríguez
Research Institute of Applied Sciences in Cybersecurity
University of León, León, Spain
Email: {alvaro.botas, rj.rodriguez}@unileon.es

Teemu Väisänen and Patrycjusz Zdzichowski
NATO Cooperative Cyber Defence Centre of Excellence
Filtri tee 12, 10132 Tallinn, Estonia
Email: {teemu.vaisanen, patrycjusz.zdzichowski}@ccdcoe.org

***Abstract*—During the last years, criminals have become aware of how digital evidences that lead them to courts and jail are collected and analyzed. Hence, they have started to develop anti-forensic techniques to evade, hamper, or nullify their evidences. Nowadays, these techniques are broadly used by criminals, causing the forensic analysis to be in a state of decay. To defeat against these techniques, forensic analyst need to first identify them, and then to mitigate somehow their effects. In this paper, we review the anti-forensic techniques and propose a new taxonomy that relates them to the initial phase of a forensic process mainly affected by each technique. Furthermore, we introduce mitigation techniques for these anti-forensic techniques, considering the chance to overcome the anti-forensic techniques and the difficulty to apply them.**

***Keywords*—forensics, anti-forensics, categorization**

I. Introduction

Computer Forensics (CF) is the scientific process and methodology followed to identify, preserve, validate, analyze, interpret, and document any digital evidence derived from digital sources for the purpose of facilitating or understanding the sequence of events that led to any potentially criminal, unauthorized actions [1]. CF tools (CFT) allow investigators to reconstruct sequence of events of any criminal and unauthorized actions, recover deleted files, and gain intelligence about a computer's user.

The digital forensic process is divided into the following phases [1]: (i) *identification* of an incident (and its type) from indicators; (ii) *preparation* of tools, techniques, search warrants, and monitoring authorizations; (iii) *approach strategy* to dynamically assess the potential impact on bystanders and the specific technology in question; (iv) *preservation, isolation, and securization* of the state of any physical and digital evidence; (v) *evidence collection*, by recording the physical scene and duplicating digital evidence using standardized and accepted procedures [2], [3]; (vi) *in-depth systematic evidence examination* regarding to the suspected crime; (vii) *evidence analysis* to determine significance, reconstruct fragments of data, and draw preliminary conclusions; (viii) *report summary* to provide explanation of conclusions (normally written using terminology understandable by attorneys); (ix) *to give any collected evidence back*, ensuring any physical and digital property is returned to its legitimate owner as well as determining how and what criminal evidence must be removed.

The aforementioned phases can be reduced to four steps [2]: (i) *collection* of the possible sources of relevant data; (ii) *examination* of any evidence, assessing and extracting the relevant pieces of information from the collected data; (iii) *evidence analysis*, to obtain information about the activities and actions performed with the device or media; (iv) and *reporting*, where the results are presented describing any evidence found and how information was retrieved.

Since CF process is mainly used to collect enough digital evidence to the courts, criminals are aware of and emerged with techniques and tools to defeat from them [4]–[6]. Anti-Forensics (AF) has emerged as a set of techniques and methods to hinder, complicate, or lengthen a forensic process. Nowadays, criminals know how to minimize their digital traces or the origins of their attacks. Hence, the forensic analysts must improve their skills and knowledge trying to anticipate or minimize the damage inflicted by the criminals using AF techniques. Several techniques are developed to mitigate the increasing use of AF techniques by criminals, such as reviewing all sectors in a hard drive (to find any hidden data), using more intelligent decompression libraries (to avoid compression bombs), or helping in the development of current forensic tools (to minimize vulnerabilities or extend funcionalities), to name a few.

In this paper, we provide a taxonomy of AF techniques that relates the phases of a digital forensic affected by, as well as how these techniques can be mitigated by forensic analysts. We identify each AF technique with the first affected phase of a forensic process, the complexity to continue with the forensic process, and the existence of techniques to mitigate them. The difficulty to apply these mitigation techniques is also discussed. To the best of our knowledge, our taxonomy is the first introducing both concepts (anti-forensics and its mitigation) in a single picture.

This paper is structured as follows. Section II reviews the literature and discusses related work. Our taxonomy of anti-forensic techniques is introduced in Section III. Then, Section IV introduces how these AF techniques can be mitigated. Lastly, Section V states the conclusions and the future work.

II. Related Work

Several works in the literature propose Computer AF (CAF) classifications [4]–[10]. In [9], the most common AF classifications are reviewed and categorized based on different criteria: on the attacked target, on the orientation of the attack, on the novelty of the techniques, or on its functionality. Traditional anti-forensic techniques, such as data and metadata

overwriting, steganography, or other data hidden techniques are explained in [4]. AF techniques have been also divided into categories considering privacy aspects [5], identifying the following categories: data hiding, artefact wiping, trail obfuscation, or attacks against CFT. Regarding privacy and from a non-criminal user's perspective, the work in [6] introduces the limitations of CF (mainly, it cannot be determined who put the data on a digital device, and it cannot find what doesn't exist in the first place – i.e., traceable data is not generated while a computer is used), as well as proposes solutions to defend against these attacks, thus preventing an attacker to recover useful data from a victim's computer. In [8], AF techniques are divided into source wiping, data hiding, and direct attacks against CFT. Similarly, data obfuscation, data hiding, and zero foot-printing are proposed in [7]. R. Harris focused more on evidences, and proposed categories such as destroying evidences, hiding evidences, eliminating evidence sources, or counterfeiting evidences [11]. A recent analytical review in AF techniques and tools is given in [10]. Other works focus on the description of concrete AF methods. For instance, a set of anti-forensic techniques capable of erasing compression fingerprints in digital images are proposed in [12]. Similarly, three AF methods are proposed to defeat digital forensics on mobile devices in [13]: obfuscation, string encryption, and environment verification. Our taxonomy, unlike these works, proposes a categorization of AF techniques based on the different components of one computer that handle data and the data itself, and identifies which phases of the computer forensic process are affected by them.

To the best of our knowledge, there are few works regarding the mitigation of anti-forensic techniques. In [11], some ways to reduce the effectiveness of anti-forensic methods are described. Mainly, Harris proposed to educate forensic analysts to face any AF measurement, thus overcoming the dependency on CF tools that can be targeted by an attacker. Furthermore, he proposed also to learn them to deal with logical/physical limitations when working with digital evidences. Forensic proactive approximations are proposed in [14] as a way to mitigate AF techniques. Proactively collecting and preserving evidences before an incident occurs ensures integrity of any evidence, preventing any method that an attacker could use to alter or destroy the digital evidence. Dahbur and Mohammad proposed in [9] to focus on better understanding of AF techniques and defending against them using any vulnerability in the AF tools, as well as on hardening CF tools. Some countermeasures are described in [4], such as to use improved monitoring systems, to place data where the attacker is unable to overwrite it, or to use more intelligent decompression libraries, to name a few. Our taxonomy also points out how the AF techniques that we identify can be mitigated.

III. Taxonomy of Anti-Forensic Techniques

Recall that a CF process aims at collecting digital evidences to be later examined and analyzed [2]. These digital evidences rely on data generated when using a computer. Hence, the own digital data and where they were handled are two key aspects when performing a CF process. Based on this idea, we have developed a taxonomy of AF techniques, sketched in Figure 1. Our taxonomy considers any component of a computer that handle data (i.e., *memory*, *software*, and *network*) and the own

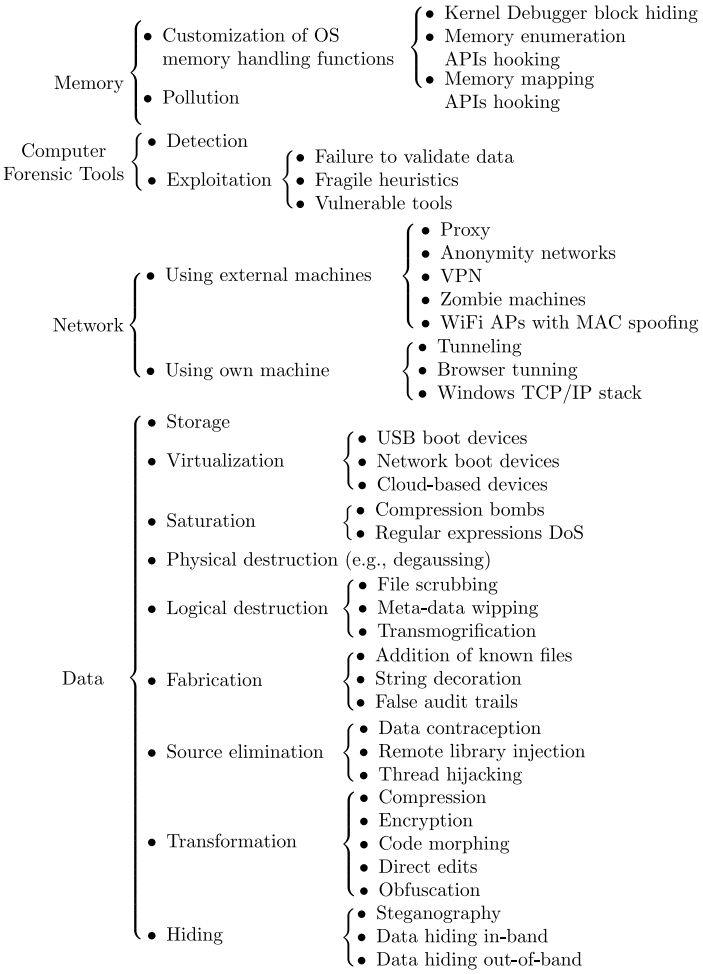

Figure 1. Taxonomy of anti-forensic techniques.

operation of data handling. Some categories are also extended showing a non-exhaustive list of the techniques involved.

The AF techniques aim at making the CF process more difficult at some spot. We have also identified what phase of a CF process is the first one affected by a AF techniques. Furthermore, we have distinguished whether an AF technique makes impossible to keep going through the CF process (with $\times$ symbol) or it can hardly continue, either because of it is almost impossible or because of evidences are altered (with $\sim$ symbol). Table I summarizes these relationships. In the following, we describe these categories in detail.

A. Memory

Memory is a great source of digital evidences since any piece of software uses it to handle their data. This category involves any anti-forensic technique focused on thwarting data acquisition from volatile memory. Other techniques to retrieve data from physical memory, such as cold boot attacks, are left out of scope in this paper. We have distinguished two main techniques:

- **Customization of OS memory handling functions**, where the functions (also termed as Application Programming Interfaces, APIs) regarding memory handling within an operating system (OS) are modified to

prevent the acquisition of the memory content. This prevention can be done through several techniques such as *Kernel Debugger Block Hiding*, *Hooking of memory enumeration APIs*, or *Hooking of memory mapping APIs*, to name a few. For instance, note that memory acquisition drivers of some CF tools need to enumerate the physical address space prior to acquisition. On Windows OS, an undocumented function named *MmGetPhysicalMemoryRanges()* allows to obtain a map of physical address space. An attacker can modify her Windows OS to return a null map, thus preventing volatile memory acquisition. These techniques attempt to collection of evidences.

- **Pollution**, where the memory is polluted (i.e., filled in with junk or irrelevant data) to confuse a forensic auditor. Since the memory is full of data and it is unknown whether these data are useful, to examine and determine the relevant information from the memory becomes hardly possible.

B. Computer Forensic Tools

The software tools developed to retrieve evidences for a computer forensic process is normally targeted by criminals to prevent these tools from what they are supposed to do. In this category, we distinguished the following:

- **Detection of CFTs**, which includes any AF technique used to detect the presence of a CFT and thus, either modify data to difficult posterior analysis or either block their acquisition directly. These techniques can affect any phase of a CF process, since once the data are altered, subsequent analysis and reporting can be incorrect, making evidences invalid to the courts.
- **Exploitation of CFTs**, which includes any AF technique that attempts to exploit any vulnerability or weakness of CFTs. This category can be refined in *failure to validate data*, *fragile heuristics*, and *vulnerable tools*. For instance, the latter category refers to traditional software vulnerabilities such as buffer overflow, format string, or denial-of-service; some of them already found on well-known forensic tools as EnCase [15] and FTK [16]. As previous techniques, these techniques can affect also any phase of a CF process, starting from collection.

C. Network

This category refers to network connections and the difficulty to establish the authorship of performed actions throughout the network. It can be split into two categories:

- **Using external machines**. Divided into *proxy*, *anonymity networks*, *VPN*, *zombie machines*, and *WiFi access points with MAC spoofing*, these techniques involve the use of external resources for performing the criminal actions, thus preventing the identification of the origin. Hence, examination is affected by this category. For instance, a forensic auditor can find an IP address of a legitimate user while analyzing any evidence and report it as suspicious.
- **Using own machines**, when criminals use their own machines to perform their activity. This category can be refined in *tunneling*, *browser tunning*, and *Windows TCP/IP stack*. For instance, browser plugins can be used to navigate on the Internet sending always a specific header, thus avoiding to fingerprint the browser based on HTTP headers. In this case, although evidences can be collected and examined, analysis of traces and reporting are difficult to achieve.

D. Data

This category relates any AF technique that affects directly to the data, such as the metadata, the location of the data, or the own data content. Several categories can be distinguished here, depending on the intention of the action performed:

- **Storage**. This category includes any technique that hampers to obtain data from devices under study in a CF process. These techniques target to collection phase, since it hampers to collect the data to be further analyzed and reported.
- **Virtualization**, related to the execution of virtual systems from external/remote disk storage and thus no evidence traces are left on the machine. Several mechanisms can be used to achieve data virtualization, such as USB boot devices, network boot devices, or even cloud-based systems. Similarly to previous category, collection phase is also affected by these techniques. Note that to seizure a virtualized system may be impossible, and hence, to collect data becomes infeasible.
- **Saturation**. Since the first phase of a CF process is to collect all data to be analyzed, these techniques aim at polluting the system with a lot of unusable content, making the data collection phase a higher time consuming task and thus almost impossible to be achieved. Denial of service attacks such as compression bombs (compressed content that largely expands its size when uncompressed) or regular expression Denial of Service (they use specific expressions to exploit some regular expression implementation [17]) are examples of this kind of AF techniques.
- **Physical destruction** of data aims at destroying the physical evidence. A common technique to this goal is *degaussing*, a physical process such that the magnetic field is reduced or eliminated, thus randomly altering data stored in magnetic devices (such as hard drives, floppy drives, or magnetic tapes). As before, these techniques target at collection phase, and makes impossible to continue with the forensic process.
- **Logical destruction** is focused on destructing any digital evidence, leaving as little useful evidence as possible. Examples of AF techniques belonging to this category are *file scrubbing*, *meta-data wiping*, or *transmogrification*, to name a few. For instance, meta-data wiping randomizes (or deletes) the time-stamp of every file so an investigator cannot build the timeline of events. Similarly, transmogrification modifies the header of a file such that it cannot be longer associated

with any known file type. These techniques hamper the collection of evidences.

- **Fabrication** consists to create data into devices to confuse a forensic analyst with false positives and bogus. Examples of these techniques are *addition of known files*, *string decoration*, and *false audit trails*. False data are discovered during the analysis phase.
- **Source elimination**, based on the principle of "it is better to prevent than to treat", i.e., they try to avoid leaving any trace that would need to be cleared. Examples of these techniques are *data contraception*, *remote library injection*, or *thread hijacking*, to name a few: data contraception allows to execute a binary on a remote system without creating a file on the disk [18], while remote library injection allows to insert a external library into the memory of a running process that loads and executes its code. Similarly, thread hijacking consists of inserting code into a process's memory to be later executed. These techniques complicate the collection of evidences.
- **Transformation** involves taking and repackaging data to disguise their meaning. This category splits in the following: *compression*, *encryption*, *code morphing*, *direct edits*, and *obfuscation*. Compression consists in creating a new file that contains the original (compressed) data, plus decompression code within a single file. Similarly, different encryption schemes can be used in individual files, folders or even entire disks, such as AES, Triple-DES, or RSA. Code morphing, as a variation of obfuscation, transforms the data into an intermediate representation. These techniques sum complexity up to the analysis phase, since the intermediate representation must be transformed before it can be further analyzed.
- **Hiding** refers to the techniques of storing data in such a manner that they are not likely to be found [19]. Techniques such as *steganography*, *data hiding in-band*, or *data hiding out-of-band* fit into this category. Steganography algorithms hide data in some media format (e.g., video or picture) or text files, thus hidden data are unnoticed for any person [20]. These techniques target to collection phase, since unseen data are not collected and thus not examined nor analyzed.

IV. Classification of Mitigation Techniques

In this section, we review some mitigation techniques that can overcome the aforementioned anti-forensic techniques, reducing its impact into the forensic process. For each mitigation technique, we identify its capacity (as a value of *high*, *medium*, *low*, or *null*) defined as the chance of being able to overcome the anti-forensic technique. Furthermore, the difficulty imposed to fully defeat against the corresponding anti-forensic technique has been also identified. We have considered three different levels of difficulty: low, represented by $\odot$; high, represented by $\ominus$; and impossible, represented by $\otimes$. Table I (right-hand side) summarizes these relationships. Note that some categories of the AF techniques can have different mitigation capacities, since they can be further divided. Similarly, the difficulty to defeat against these techniques considers the highest difficulty level within each category.

A. Memory

- **Customization of OS memory handling functions**. There exist tools that inspect operating system services and programs, reporting whether APIs have been somehow modified. Examples of these tools are Microsoft Detours, WinAPIOverride, or KaKeeware Application Monitor. These anti-forensic techniques can be mitigated by accessing directly to the memory through the I/O manager, by circumventing the hooked function using other related functions, or by unhooking using user-mode system call injections. These techniques can be mitigated, but it requires a high time-consuming task, as well as deep knowledge of low-level operating system behaviour.
- **Pollution**. These techniques can be mitigated by white-listing the type of files of interest for an analyst, although this task can be very time consuming. As before, they are very difficult to overcome.

B. Computer Forensic Tools

To mitigate both techniques regarding CFTs (detection and exploitation), these tools need to be improved by deep code revisions and bug fixing [21]. Similarly, fragile heuristics can be replaced with stronger ones. Note that we could use also data transformation techniques, such as compression or obfuscation, to avoid the detection of a CFT by common detection techniques such as the looking for known window handlers, communication pipes, or running processes. Although these techniques can be highly mitigated, it becomes hard in some cases (e.g., when a CFT is not open source and its code is not available).

C. Network

- **Using external machines**. Since these techniques may be very hard to overcome, an investigator needs to review any network packet (i.e., of any protocol) to discard traffic origins. When possible, intermediate machines used by a criminal can contain traces of the original location of the criminal. The capacity to mitigate these techniques ranges from null to low, becoming almost an infeasible task when possible.
- **Using own machines**. In this case, the analyst must monitor Internet connections and analyze the incoming and outcoming network packets. When the device to be analyzed is running, the volatile memory can have indicators of origin connections and contents about the data being sent/received. As before, there will be little or no chances to mitigate them; but on the contrary, the difficulty imposed in this case becomes easy.

D. Data

- **Storage**. When difficulties are found to work with storage units taken as evidences, the own hardware where these units were found can be used. These AF

Anti-Forensic Technique	Initial CF phase affected	Difficulty imposed	Mitigation Capacity	Difficulty imposed
Memory				
Customization of OS memory handling functions	*Collection*	×	*High*	⊖
Pollution	*Examination*	∼	*Low*	⊖
Computer Forensic Tools				
Detection	*Collection*	∼	*High*	⊙
Exploitation	*Collection*	×	*High*	⊖
Network				
Using external machines	*Collection*	×	*Low-Null*	⊖
Using own machines	*Analysis*	∼	*Low-Null*	⊙
Data				
Storage	*Collection*	∼	*High*	⊙
Virtualization	*Collection*	×	*Low-Null*	⊖
Saturation	*Collection*	∼	*High*	⊙
Physical destruction	*Collection*	×	*Null*	⊗
Logical destruction	*Collection*	∼	*High*	⊖
Fabrication	*Analysis*	∼	*High*	⊙
Source elimination	*Collection*	∼	*Low-Null*	⊖
Transformation	*Analysis*	∼	*Low*	⊖
Hiding	*Collection*	×	*Medium*	⊖

Table I. PHASES OF A COMPUTER FORENSICS PROCESS AFFECTED BY EACH ANTI-FORENSIC TECHNIQUES AND MITIGATION CAPACITY FOR EACH ANTI-FORENSIC TECHNIQUE.

techniques can be high likely mitigated and easily performed.

- **Virtualization**. The mitigation techniques to defend against these AF techniques depend on the data virtualization used. For instance, the only way to extract evidences from volatile systems is to get them when the system is running. When the system is virtualized on the cloud, network connections must be analyzed to discover server locations [22]–[24]. Unfortunately, these AF techniques are really hard to mitigate, becoming rarely possible.
- **Saturation**. One way to mitigate these techniques consists of using wildcard selection of data to be collected and examined, and not examining any kind of type found as evidences. DoS attacks mainly produced by compression bombs can be overcome by using more intelligent decompression libraries. Data saturation is easy to mitigate, without almost any difficulty to be achieved.
- **Physical destruction**. Unfortunately, these techniques cannot be mitigated: when the media is physically destroyed, there is no turning back. For this reason, the mitigation capacity is null and the difficulty categorized as impossible.
- **Logical destruction**. Anti-forensic techniques in this category can be mitigated with specific tools or known methods. For instance, fuzzy hashing identifies potentially interesting files, thus avoiding transmogrification effect [25]. Similarly, broken log files can be reconstructed by manually recovering log pieces from related files. File scrubbing can be circumvented by seeking parts of traces left by tools to prove that the data were previously on the media under analysis [8]. This category involves a large number of techniques, being some of them high likely to be mitigated; however, the difficulty to achieve the original data is very high.
- **Fabrication**. False audit trails (concretely, dummy HDDs) can be mitigated by the investigator by checking physical USB drives attached in USB slots and USB connections on motherboards. Furthermore, the filepage on an USB drive can point out to network locations. In this case, it is easy to mitigate them, imposing also a low difficulty.
- **Source Elimination**. These technique are really difficult to mitigate, since there are not any trace to follow. Log files of connections, activities, or other registered events can be partial traces that may help to reconstruct data that were eliminated. Again, these techniques are really hard to mitigate, almost impossible in several cases.
- **Transformation**. Known attacks such as brute force attack can be used to defeat against an encryption scheme using a weak key. Other attacks, such as the use of keyloggers, overhead cameras, or side-channel attacks can also overcome an encryption scheme. Other AF techniques in this category, such as compression or obfuscation, can be defeated applying analysis methodologies from other security domains, such as static and dynamic data analysis commonly used in malicious software research. As before, these techniques are really complex to mitigate, being low the mitigation capacity.
- **Hiding**. Some techniques in this category can be really hard to defeat. For instance, to detect a message hidden with steganography is easily done using entropy tests [26], although the original message is almost impossible to be retrieved when the steganography algorithm used is unknown. Other techniques, such as data hidden in-band and out-of-band, are already handled by current implementations of commonly used CFTs. These techniques are unlikely to be mitigated, imposing a high difficulty when possible.

V. CONCLUSIONS

Nowadays, the forensic process is in state of decay. Criminals are aware of how to reduce the effectiveness of a forensic auditor doing a computer forensic process, thus becoming a completely ineffective and unusable process. Anti-forensic techniques used to this goal by criminals must be known by

the forensic auditors , as well as their impact into the forensic process. Hence, it is important to educate the forensic analyst and improve their knowledge in order to minimize the impact of the anti-forensic techniques. Furthermore, analysts must also be aware of techniques to mitigate these anti-forensic techniques.

In this work, we have reviewed the literature about anti-forensic techniques and mitigation forms to these techniques. We have introduced a new classification of anti-forensic techniques, relating them with the first phase of forensic process that is affected by, as well as the difficulty imposed by these techniques into the forensic process. We have also related these anti-forensic techniques with mitigation techniques that can be applied to minimize their impact.

As future work, we pursue at developing a new tool to look for already known vulnerabilities in computer forensic tools, giving also patches to fix them. At the same time, the tool also implements some mitigation techniques to overcome known anti-forensic methods. Furthermore, we aim at writing a step-by-step guide to specify how to apply the mitigation techniques to each anti-forensic technique, as well as how to behave and act when detecting anti-forensic techniques during the forensic process.

ACKNOWLEDGEMENTS

This work was partially supported by the Spanish National Institute of Cybersecurity (INCIBE) according to rule 19 of the Digital Confidence Plan (Digital Agency of Spain) and the University of León under contract X43.

REFERENCES

[1] DFRWS attendees, "A Road Map for Digital Forensic Research," Report for the First Digital Forensic Research Workshop (DFRWS), Tech. Rep., August 2001, technical Report DTR - T001-01.

[2] K. Kent, S. Chevalier, T. Grance, and H. Dang, "Guide to integrating forensic techniques into incident response," *NIST Special Publication*, pp. 800–86, 2006.

[3] E. C. S. Investigation, "A guide for first responders, second edition," *US Department of Justice, NCJ*, vol. 219941, 2008.

[4] S. Garfinkel, "Anti-Forensics: Techniques, Detection and Countermeasures," in *Proceedings of the 2nd International Conference on i-Warfare and Security (ICIW)*, 2007, pp. 77–84.

[5] G. C. Kessler, "Anti-Forensics and the Digital Investigator," in *Proceedings of the 5th Australian Digital Forensics Conference*, 2007, pp. 1–7.

[6] M. A. Caloyannides, "Forensics Is So "Yesterday"," *IEEE Security & Privacy*, vol. 7, no. 2, pp. 18–25, 2009.

[7] B. Sartin, "ANTI-Forensics – distorting the evidence," *Computer Fraud & Security*, vol. 2006, no. 5, pp. 4–6, 2006.

[8] P. Pajek and E. Pimenidis, "Computer Anti-forensics Methods and Their Impact on Computer Forensic Investigation," in *Proceedings of the 5th International Conference on Global Security, Safety, and Sustainability (ICGS3)*, ser. Communications in Computer and Information Science. Springer Berlin Heidelberg, 2009, vol. 45, pp. 145–155.

[9] K. Dahbur and B. Mohammad, "The Anti-forensics Challenge," in *Proceedings of the 2011 International Conference on Intelligent Semantic Web-Services and Applications (ISWSA)*. New York, NY, USA: ACM, 2011, pp. 14:1–14:7.

[10] A. Jain and G. Chhabra, "Anti-Forensics Techniques: An Analytical Review," in *Proceedings of the 2014 Seventh International Conference on Contemporary Computing (IC3)*, Aug 2014, pp. 412–418.

[11] R. Harris, "Arriving at an anti-forensics consensus: Examining how to define and control the anti-forensics problem," *Digital Investigation*, vol. 3, Supplement, no. 0, pp. 44–49, 2006, the Proceedings of the 6th Annual Digital Forensic Research Workshop (DFRWS '06).

[12] M. Stamm and K. Liu, "Anti-Forensics of Digital Image Compression," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 1050–1065, Sept 2011.

[13] J. Li, D. Gu, and Y. Luo, "Android Malware Forensics: Reconstruction of Malicious Events," in *Proceedings of the 32nd International Conference on Distributed Computing Systems Workshops (ICDCSW)*, June 2012, pp. 552–558.

[14] S. Alharbi, J. Weber-Jahnke, and I. Traore, "The Proactive and Reactive Digital Forensics Investigation Process: A Systematic Literature Review," *International Journal of Security and Its Applications*, vol. 5, no. 4, pp. 59–72, October 2011.

[15] CVE, [Online; accesed at July 09, 2015], 2007, http://www.cvedetails.com/vendor/3015/Guidance-Software.html.

[16] W. Dormann, "Forensics software and oracle outside in," [Online; accesed at July 09, 2015], August 2013, https://www.cert.org/blogs/certcc/post.cfm?EntryID=164.

[17] OWASP, "Regular expression denial of service - redos," [Online; accessed at July 07;2015], 9 2012, https://www.owasp.org/index.php/Regular_expression_Denial_of_Service_-_ReDoS.

[18] Grugq, "Remote exec," [Online; accesed at July 09, 2015], 07 2004, http://phrack.org/issues/62/8.html.

[19] B. Blunden, *Rootkit Arsenal: Escape and Evasion in the Dark Corners of the System*. Jones & Bartlett Publishers, 2013.

[20] N. F. Johnson and S. Jajodia, "Exploring steganography: Seeing the unseen," *Computer*, vol. 31, no. 2, pp. 26–34, 1998.

[21] B. Cusack and A. Homewood, "Identifying bugs in digital forensic tools," 2013.

[22] N. Jawale, "Locating and extracting digital evidence from hosted virtual desktop infrastructures: cloud context," Ph.D. dissertation, Auckland University of Technology, 2012.

[23] M. Hirwani, Y. Pan, B. Stackpole, D. Johnson *et al.*, "Forensic acquisition and analysis of vmware virtual hard disks." The 2012 International Conference on Security and Management, 2012.

[24] R. Poisel, E. Malzer, and S. Tjoa, "Evidence and cloud computing: The virtual machine introspection approach," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)*, vol. 4, no. 1, pp. 135–152, 2013.

[25] D. Hurlbut, "Fuzzy hashing for digital forensic investigators," *Access-Data, January*, 2009.

[26] M. Fontani, A. Bonchi, A. Piva, and M. Barni, "Countering anti-forensics by means of data fusion," in *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2014, pp. 90 280Z–90 280Z.