Detection and Mitigation of Anti-Forensics

Emre Caglar Hosgor
Department of Computer Engineering
METU
Ankara, Turkey
e2320455@ceng.metu.edu.tr

***Abstract*—With the advances in IT, digital forensics became an important part of the juridical system. On the other hand, cyber-criminals have been developing counter tactics against digital forensics for fleeing from the justice. Those tactics are grouped under the term "anti-forensics". Anti-forensics includes data hiding, artifact wiping and trail obfuscation techniques which aim to subvert, hinder or make dysfunctional the digital forensic analysis. There are more than 300 anti-forensics related tools and methods. Categorization of, detection the use and mitigation against anti-forensics' related resources do improve digital forensic analysis processes. Therefore, this research aims to provide categorization of anti-forensics techniques by explaining how cyber-criminals use the tools and also aims to provide counter methods or mitigation techniques.**

Keywords-component; computer forensics; anti-forensics

I. INTRODUCTION

Although information technology has improved our living standards, it has also provided criminals new ways to commit crime. Digital crime includes identity theft, online piracy, hacking, and terrorism. For combating digital crime, new techniques and tools emerge frequently in digital forensics. On the other side, cyber-criminals develop counter-techniques called anti-forensics, which aim to disrupt or manipulate forensic analysis of a crime. This research investigates the effectiveness of popular anti-forensic tools for data hiding, artifact wiping, and trail obfuscation. We found they varied considerably in effectiveness and a variety of countermeasures can be used against them. Initially at the second section, we provided a background on the anti-forensics. Third section is about anti-forensic techniques, and their usage. Counter measures and mitigation methods are discussed at the fourth section. Paper concludes with recommendations for investigators and future work that can be pursued.

II. BACKGROUND

We are living in information era, thus digital devices play significant roles in our daily lives. Cyber criminals and attackers have also been using information technology and digital devices for malicious purposes since the beginning of 2000's [1]. This tendency has made DFIR (Digital Forensics and Incident Response) an important part of criminal investigation [1]. When a new forensic technique is proposed, related counter technique is developed by cyber-criminals [2]. This trend is a cycle, thus anti-forensics is an evolving area in DFIR. Defining and categorizing efforts for anti-forensics started at the beginning of 2000's [3].

A. Anti-Forensics

Anti-forensic techniques are aiming: Avoiding detection of malicious actions, disrupting the information collection, increasing investigation time required, decreasing the validity of forensic report, revealing the presence of forensic tools, and clearing the evidence of anti-forensic tool's existence [3, 4].

Understanding anti-forensics requires understanding the digital forensics, because it targets digital forensic examination [3]. Digital forensic examination has three main stages acquisition and preservation, analysis, and presentation [5, 6]. Figure 1 show these steps. Anti-forensics mainly target analysis step, because digital forensic investigator usually finds valuable evidence during this step [3, 4].

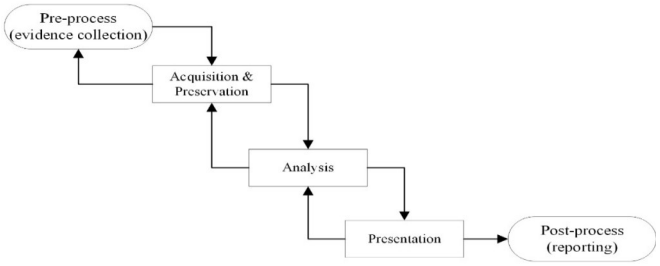

Figure 1. Digital forensic examination steps, adopted from [5].

B. Categorization of Anti-Forensic Techniques

Categorization of anti-forensics can be done either focusing on tools or techniques. A tool-specific categorization of anti-forensics was provided at [7]. This categorization includes: *wiping, stenography, rootkits, encryption, forensic tools vulnerabilities, trail obfuscation, exploits* and so forth. When a new digital forensic approach is defined, then cyber-criminals will eventually develop a related counter tool, therefore a tool-specific categorization is like keeping an exhaustive list of every tool at present. Moreover, a tool-specific categorization needs periodic updates, which is an extra burden to bear. We follow common categorization of anti-forensic techniques [3]. Main categories of anti-forensics are:

1. Data hiding,
2. Artifact wiping,
3. Trail obfuscation,
4. Attacks against digital forensic tools and processes.

Without losing the generality, after defining each category we categorized tools provided at [8]. The list at [8] is a well-

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

defined work of anti-forensic tools. In this work, we went over each tool and updated the list according to recent security trends[1].

C. Description of Anti-Forensic Categories

Anti-forensic can be divided into four major categories, we provided definitions to each category in this section. In each categorical section common-used tools are explained as well.

1) Data Hiding

Data-hiding techniques and tools use file-system, memory, or network capabilities of the operating system for hiding the data. Hiding tools, stenography, and encryption are closely related [8 – 10]. However, data hiding is a broader concept; moreover stenography, rootkits, and encryption are specialized hiding techniques.

a) File System Data Hiding Tools

Tools under this subsection make use of unused space in the file-system [11]. There are two common file-systems: UNIX-base (ext, jfs), and Windows-based (FAT32, NTFS). In NTFS, when a file is less than the cluster size, unused space occurs; if a file is larger than the cluster size it becomes fragmented. These empty spaces in a file-system are called slack space, and slack space data hiding is more effective in Windows environment, but it is an applicable technique for both UNIX and Windows [12]. Slacker, which was a part of Metasploit framework, is the prime example as a tool for this category.

b) Memory Data Hiding (Live Hiding) Tools

Main memory is volatile. Live hiding requires using file-less attack vectors and keeping data to hide in stack or heap allocations of a running program [13]. In the event of data hiding at the main memory, retrieval techniques are both hardware and software-based, and they are rather easy to implement, because anti-forensic tools in the main memory do not try to hide themselves very much [14].

c) Network-Based Data Hiding

These tools exploit structure or implementation of ISO OSI network layers [11]. Tools use covert channeling, protocol bending, and packet crafting for hiding data. Wrapping tools (UNIX Stunnel-https://www.stunnel.org/ index.html), terminal emulators (Absolute Telnet, Indigo Terminal Emulator, SecureCRT), and VPN protocol suit are used by cyber-criminals either data hiding or data exfiltration [8, 15].

d) Encryption

Encryption is transforming data into a secret code [16]. Modern encryption is an essential part of CIA (Confidentiality, Integrity, and Availability) triad. Encryption provides confidentiality and ways to hide data [17]. In order to keep our focus on implementation of anti-forensic technique and detection of that with forensic tools, disk encryption is the focus of this research. Predominant tool in disk encryption is VeraCrypt. Encryption disrupts the initial acquisition phase of the digital forensics examination so the examiner cannot complete the following phases. Encryption is a specialized data hiding technique in anti-forensics, because of these characteristics

e) Stenography

Stenography is techniques to hide secret information in image, video, audio, or text files so that the information cannot be detectable by a naked eye [18]. Distortion and spread-spectrum techniques are examples of audio stenography, and substitution techniques are examples of image stenography [19]. In all stenography methods, encryption can be used to provide extra protection against steganalysis.

f) Rootkits

Rootkits are specialized code sectors that hide in the operating system kernel [20]. Rootkits are a type of malicious software that runs at the inner levels of an operating system. Cyber criminals use rootkits not only hiding data, but also for logging the network activity, storing keystrokes, process hiding, and controlling registry entries.

2) Artifact Wiping

Artifact wiping is used for destructing the digital evidence. Common artifact wiping methods and tools are discussed at the following lines.

Disk wiping is erasing data from disk securely. There are many tools under this method (Blancoo, DBAN, and WipeDisk). Usage of these tools are fairly easy, however subsequent retrieval is quite hard [1]. *File wiping* is similar to disk wiping but focused on files. Sdelete is the most common tool for file wiping [8]. *Removable-disk wiping* uses similar techniques to that of disk wiping [1]. *Disk degaussing* is physical destruction of the data by applying magnetic energy [21]. *Physical destruction* includes melting, shredding and incarnating of the physical media [22]. *Generic data wiping tools* differ from file wiping tools by erasing artifacts like cookies, temporary data, and Internet browsing history. A well-known generic data-wiping tool is CCleaner [8]. The Windows registry is a database storing operating system and application-specific settings for the Microsoft Windows operating system. *Registry wiping tools* remove unused, broken, or wrong registry entries [8].

Metadata wiping tools: Metadata is the data about the data. Metadata of a file stores times, ownership, size, etc. An example tool for metadata wiping is Timestomp. It is a part of the Metasploit framework [3]. Metadata wiping requires advanced knowledge and a successful exploit at the target system.

3) Trail Obfuscation

Similar to source code obfuscation, cyber-criminals try to obfuscate their trails on the system they exploit [23]. Cyber-criminals add misdirection as well. Misdirection includes timestamp modification, file defragmentation and manipulation of log files. Any inconsistencies on those suggest a trail-obfuscation activity.

[1] https://github.com/3mr3h/Common_Anti_Forensic_Tools_List

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

4) Attacks against Digital Forensic Tools and Processes

Attacks against digital forensics aims the analysis step in examination steps (Figure 1) [8]. A cyber-criminal by detecting either image creation or analysis of the logical partitions can alter the integrity of the evidence.

Denial of service is another attack type against forensic tools. By depleting resources like the RAM and CPU required by the tools, an attacker can impede analysis [24]. Anti-reverse engineering is another method against forensic tools.

One way is compression bombs. Current tools open compressed files like ”zip” files during the analysis of file system. Compression bombs are compressed files that when extracted gets bigger than the tool can handle, perhaps with recursively contained files.

III. Anti-Forensic Tools and Techniques

Detection of anti-forensics may seem trivial, however during the experiments we noticed that previous analysis of tools provide an insight to the forensic analyst. Methodology we followed includes four steps: Installation of the tool, configuration (using a tool for malicious purposes), usage, and analysis of the artifacts on the target system. At the preparation phase before installation, we choose tool(s) for each anti-forensic category. Having more than 300 tool to consider [8], the main criteria are effectiveness in circumventing forensics, availability, ease of usage, cross-platform capability, and resistance to cyber-attacks, novelty, community support and popularity among the cyber-criminals. Entire list is at GitHub page[1], Table 1 shows the best candidate tools list. Our research focused on those tools.

TABLE I. Chosen Anti-Forensic Tools

Technique	Sub Category	Specific item analyzed
Data Hiding	1. File System Data Hiding	BMAP, NTFS file streams
	2. Memory Data Hiding	Explanation of the techniques
	3. Network-based Hiding	Stunnel
	4. Encryption	VeraCrypt whole disk and file encryption
	5. Stenography	Audio
		Text using Hydan tool
		Image
		Protocol
	6. Rootkits	In general
Artifact Wiping	1. Disk Wiping	DBAN
	2. Disk Degaussing & Destruction	In general
	3. File Wiping	Sdelete and BitKiller
	4. Generic Data Wiping	CCleaner
	5. Metadata Wiping	Timestomp
	6. Registry Wiping	In general
	7. Removable Disk Wiping	In general
Trail Obfuscation		Log cleaners with the Metasploit framework
Attacks against Forensic Tools and Techniques		Packers (7zip)

A. Data Hiding Techniques

1) File System Data Hiding

Two separate hiding tools/methods were tested under this category. In UNIX environment BMAP tool, in Windows NTFS ADS (alternate data streams)/file streams were tested. For BMAP test; BMAP version 1.0.17 was used. For testing, two image files were created using “dd” command and mounted on the Ubuntu file system. Then a string (“secret”) was put into the slack space. For this test BMAP source code updated and compiled. For detecting slack space data hiding, “strings” terminal command used, and hidden text was revealed from EXT3, but not from FAT32 (Figure 2).

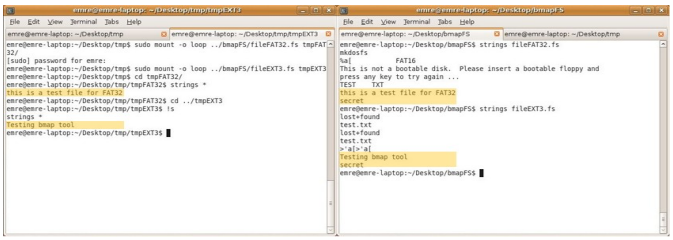

Figure 2. Detection of BMP tool on FAT32 and EXT3

Detection of hidden data using BMAP tool. Secondly, we tried forensic tools to extract hidden string from slack space, except the Autopsy tool, FTK Imager, Bulk Extractor and TSK could extract hidden string from both of the file systems. NTFS ADS uses metadata section of a file for hiding data. For experiment an executable (evil.exe) was created and a pointer in $DATA attribute of a regular file was used for hiding. System Internals “streams64.exe” can detect this data hiding (Figure 3). The pointer to malicious file can also be in other attributes (author or title) of an NTFS file so it may not attract attention during the forensic analysis.

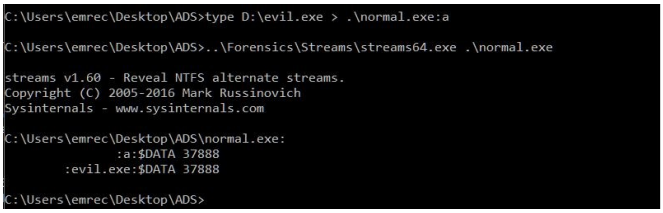

Figure 3. Detection of hidden data using strings64.exe.

2) Network Level Data Hiding

Network protocols provide means to hide data in packets. For testing network level data hiding Stunnel was tested. As shown at Figure 4, client and server Stunnel instances were run and netcat was used for capturing hidden messages. Single hindrance is symmetric key exchange. Cyber-criminals usually share keys offline or exchange using another encrypted protocol/application, such as OpenVPN. Experiments showed that stunnel communication and usage can be detectable but retrieval of the hidden data is challenging.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

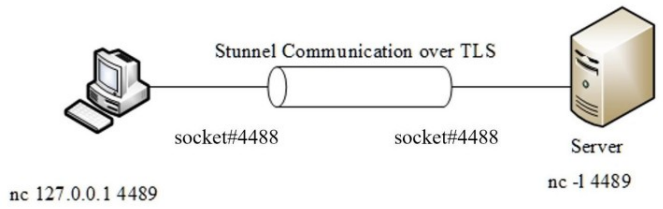

Figure 4. Testing Topology for Network-level Data Hiding

3) Data Hiding with Encryption

Encryption is commonly used for hiding data. There are many tools available. We used VeraCrypt for trying to hide data in an encrypted volume. In experiments, we stored three text files in the standard and hidden volumes. Our experimental environment was Windows 10 OS VM. To examine the VeraCrypt tool and volumes, a VMDK-file to binary-file conversion was done. Hexadecimal value analysis of the two volume files showed that neither contained successive zero bytes. Filling out empty parts of the volume is a feature of VeraCrypt. However, empty parts in a file must be filled with zero bytes in Windows 10; seeing no successive zero bytes is a clue to use of VeraCrypt. Figure 5 shows that TSK-Autopsy (forensic analysis tool) tags volume files as possible encrypted files.

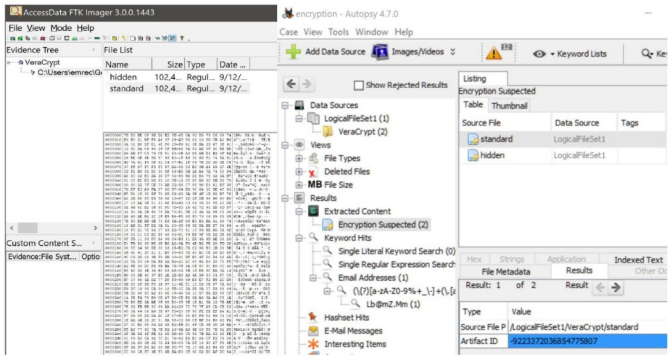

Figure 5. Forensic Analysis of Encrypted Volumes with DFIR Tools.

4) Stenography

Stenography is the most common technique for text hiding [18]. We tested a home developed tool which based on substitution of LSB.

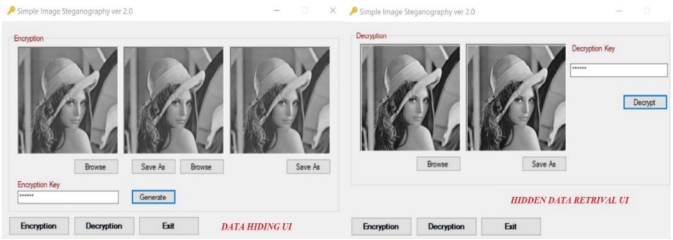

Figure 6. LSB Substitution-based Stenography Tool

The tool can hide BW and half grayscale secret image into a base image so that a naked eye cannot detect there is a hidden image. Figure 6 shows the tool GUI. The tool provides substitution-based LSB image hiding capability.

Entropy analysis can help to understand the uncertainty of the information source, so unusual values for entropy of an image file can indicate stenography. In our tests we analyzed PNG images with a HEX editor and a python program which creates a histogram of entropy of the bytes. For entropy analyze we need original image and image with stenography. Figure 7 shows entropy analysis of both images. Unexpected spikes in entropy shows that unusual bit changes (randomness), therefore this can be an indicator of stenography.

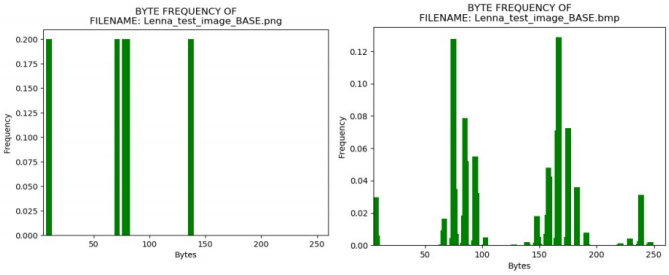

Figure 7. Bit Change Entropy Analysis of Stenography and Base Images

B. *Artifact Wiping Techniques*

Data hiding techniques leverages to keep data out of sight and provides ways to flee the data by hiding at the first place. Artifact wiping provides techniques to delete data for good in order to hide evidence from forensic analysis. Disk wiping and degaussing are close to physical destruction, therefore they are out of scope of this work.

1) File Wiping

Initially a carving on a file marked as deleted was tested. We used Ubuntu VM for this test and at first a file was deleted, then after working on the image of the OS, Foremost, Scaple, Autopsy, FTK and FTK Imager tools successfully retrieved the deleted file. When a user deletes a file using shred with following options "*shred -n1 -f -v -z*" then shred tool writes bunch of zeros and deletes the file, therefore forensic analysis yield no result after carving the file.

Similar to shred tool in UNIX, Sdelete and BitKiller deletes and overwrites the files passed them on Windows. As a result to our tests, after wiping artifacts are gone, however memory analysis yielded presence of the tool while it was being used by cyber-criminal.

2) Generic Data Wiping and Registry Wiping

Current forensic analysis shows that not only disks and files contain artifacts, but also web browsers, applications, and third-party tools. Generic wiping tools target those to cover cyber-criminal's tracks. One of the most common generic-artifact wiping tools is Piriform CCleaner, a commercial Windows tool. According to Cnet.com, it has been downloaded nearly 161 million times. CCleaner overwrites a file at least three times with random data using the "rand ()" function in Windows to make the file data random. However, CCleaner is easy to see in a disk image since it creates an ".INI" file for storing configuration data under the directory

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

"C:\Program Files\CCleaner." Another indicator is the prefetch data mentioning "ccleaner" and "piriform."

3) Metadata Wiping

Metadata wiping requires super user or administrator privileges on the target system. For testing metadata wiping, an exploit scenario was created using a Kali Linux and Windows machines. After a successful exploit on the Windows machine, timestamps were altered. Timestamp change was obvious anomaly for the forensic tools that this type of attacks create a big inconsistencies on the timeline analysis. An analyst can understand that cyber-criminal deliberately altered the timestamps for covering their tracks.

C. Trail Obfuscation

Cyber test network implementation for metadata wiping was used for testing trail obfuscation. After a successful exploit cyber-criminal used "clearev" tool for clearing events on the Windows machine during the post-exploit phase. A full log deletion can be detected from Security event logs on a Windows system. But there were no ways to recover those deleted logs if the system is not forwarding them. Even if systems are scheduled to forward logs, cyber-criminals usually try to change logs or create extra ones to cover their tracks.

D. Attack Against Forensic Tools and Processes

Packers are file-compression tools and they can be used against forensic tools to provide code and data obfuscation. To analyze packers, we created a simple C code segment to print a sentence to the terminal window, then packed it with PECompact, 7-zip and UPX. Our results showed these packers compressed up to 70%. We ran the forensic tools FTKImager, Autopsy, TSK, AXIOM, and BelkaSoft. They all successfully detected that the compressed file contained an executable. When details of the executable were examined with IDA PRO, it was seen that the "magic number" identifying the type of file remained untouched. So packing alone is not a useful anti-forensics technique.

IV. Counter Measures, Mitigation Against Anti-Forensics

We analyzed four kinds of anti-forensics. Table 1 summarized our analysis methodology. Our analysis suggested that common anti-forensics techniques and tools can be detectable by their leaving important evidence material in various places. In this chapter tool-specific detection methods and mitigation techniques are presented.

A. Data Hiding

1) Detection of File System Data Hiding

We installed and used the BMAP tool in a Linux environment. Detection of the tool can be done by the "strings" tool for the EXT3 file system. If the file-system format is FAT32, then detection can be done by the forensic tools FTKImager, Bulk-Extractor, and TSK. A second file system data-hiding method is using ADS on an NTFS environment. Detection of ADS can be done using Microsoft System Internals "stream64.exe" tool. Tests on ADS showed that it successfully detected a hidden stream in an executable that was pointing another malicious executable where the stream value was stored in $DATA attribute of the executable. However, a forensic analyst can easily miss that pointer. A more novel method for detection of ADS is to use PowerShell with following steps:

1. Collect user-created files (potential NTFS ADS files).
2. Run "Get-item -Path [file_path] -Stream * | Export-Csv " PowerShell cmdlet.
3. Analyze the CSV files to detect uncommon ADS values.
4. Run,
   "Get-Content -Path -Stream [uncommon_stream_name] >> Evidence_Streams.txt"
   PowerShell cmdlet. Common stream names such as $DATA, $AUTHOR, and $FILE_NAME are stored in the metadata of the file. Focus on those metadata values.

Current state-of-art forensic techniques do not provide a mitigation technique for ADS, because it has many valid usages. However, our PowerShell detection method can be turned into a script, which runs on the client and sends stream contents to a server for further analysis.

2) Detection of Network Communication Data Hiding with Stunnel

We tested Stunnel for data hiding. Stunnel communication is quite secure because it uses the SSL protocol. Network-traffic analysis did not reveal its hidden data. Communication initialization messages and socket communications (IP and port-tuple communications) were analyzed using Wireshark for network-traffic analysis, but there were no major indicators of the data hiding using Stunnel. There were clues in Stunnel configuration files, server-client TLS communication for high-end (>1024) ports, use of the OpenSSL library certificate, string search revelations of Stunnel keywords like stunnel, stunnel4, stunnel.conf, etc.), and the "var/run/stunnel.pid" file. Stunnel runs on the Linux. Stunnel requires OpenSSL library, a designated "uid" and "gid" pairs for Stunnel, and /var/run/stunnel.pid file, containing the "pid#" for Stunnel. A good mitigation against Stunnel is restricting users from creating or changing the Stunnel installation and configuration requirements.

3) Detection and Mitigation Techniques against Encryption Usage for Data Hiding

Encrypted files are hard to analyze without the key. Sometimes in a criminal investigation the key can be retrieved from the suspect. On the other hand, if analyst cannot detect any encrypted files, then they need to put mitigation techniques beforehand. Our experiments on VeraCrypt revealed that TSK-Autopsy did flag VeraCrypt volumes as encrypted files. Mitigation techniques against this method for the enterprise level networks are:

1. A good corporate file-server (file storage) policy so users cannot map a file share, and only GPO or scripts can.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

2. Effective device control metrics. When a user plugs in a removable media, contents should be copied to a central location (an evidence folder) for examination.
3. Installing an executable must be disabled. If the user is a system administrator, installing the executable must be logged and monitored. It is important to prepare a master operating-system image containing all the required programs and executable. If a new executable is required, it must get approval from the ISM change-management board.
4. Users should access encryption libraries such as OpenSSL only with approval. And encryption must be done when required at the background and not by users.

4) Detection of Stenography

Detection of a stenography. (image/video) depends on statistical analysis of the image files. In this work detailed information about using and detecting substitution-based stenography. is at the third section. By entropy analysis, anomalies can be detected on the cover image. However, this does require having the both the base image and the final image.

B. Artifact Wiping

Detection of artifact wiping is easy by observing data patterns of 0s or 1s; however, retrieving the deleted data is cumbersome, and usually not possible. Mitigation methods against artifact wiping are thus important. Mitigation techniques against the artifact wiping are possible for enterprise networks where there is a central server that maintains and administers the network. The server can employ rules to control user activity. Some things they need to manage are:

1. Artifact-wiping tools are cleaning tools that delete registry, temp files, browser history, and so forth. System administrators must prevent users from downloading and installing such tools. The list of tools at the GitHub page[1] can serve as a guide.
2. User-activity logs are valuable for detecting artifact-wiping activity. They can be saved in a SIEM log-collection database for integrated analysis.
3. System administrators need to prevent users from accessing system root files and folders. This includes the Windows "C:" drive and in Linux all the directories except user's home director.
4. The ideal option is live forensic-artifact collection using an agent-based forensic application on the client systems.

C. Mitigation Techniques Against Trail Obfuscation and Attack Against Forensic Tools

Trail obfuscation tools misdirect the forensic analysis. Cyber-criminals use them in the post-exploit phase of an cyber-attack. The best mitigation against the trail obfuscation is protecting the systems against a cyber-attack. We tested zip bombs and packers as attack methods against the forensic tools FTK Imager, TSK-Autopsy, and Magnet Forensic AXIOM. They can detect zip bombs and recover themselves against this attack. Success of the packers depends on the technique used. If a packaging method like UPX is used against forensic tools, detection is possible because UPX cannot hide the contents of the executable. On the other hand, the 7-Zip tool encrypts both contents and file names, so this tool is effective against initial forensic analysis. However, forensic examiners can provide findings of encryption and legal authorities can request keys and passwords.

V. Conclusion And Future Work

Anti-forensics techniques and tools are ever-changing. Therefore, a solid set of forensic tools is required combating them. We first examined anti-forensic techniques and identified tools in a number of categories. We identified publicly available, popular, up-to-date, and easy-to-use tools. Our experiments on selected tools showed that anti-forensic techniques do complicated well-known forensic practices. To detect each tool, forensic examiner must look different parts of the operating system and must follow different methodologies. We suggested mitigation and detection techniques that can help forensic examiners to prevent anti-forensics or to detect its use. Important threats we have not examined are rootkits and malware activity. Future work should investigate mitigation and detection methods for them following our methodology

Acknowledgment

Prof. Neil Rowe and Mr. Glenn Cook provided expertise, guidance and help to this research. This research is a continuation of a thesis work which was done under their supervision.

References

[1] D. Lillis, B. Becker, T. O'Sullivan, and M. Scanlon, "Current challenges and future research areas for digital forensic investigation," arXiv:1604.03850, 2016.

[2] S. Raghavan, "Digital forensic research: current state of the art," CSIT 1, pp. 91-114. 2013.

[3] S. Garfinkel, "Anti-forensics: Techniques, detection and countermeasure," 2nd International Conference on i-Warfare and Security, vol. 20087, pp. 77–84, 2007

[4] V. Liu, and F. Brown, "Bleeding-edge anti-forensics," Presentation at InfoSecWorld, 2006.

[5] Y. Yusoff, R. Ismail, and Z. Hassan, "Common phases of computer forensics investigation models," vol. 3, pp. 17–31, 2011.

[6] D. P. Joseph, and J, Norman, "An Analysis of Digital Forensics in Cyber Security," First International Conference on Artificial Intelligence and Cognitive Computing, vol. 815, 2019.

[7] H. Jahankhani, and E. Beqiri, "Digital evidence manipulation using anti-forensic tools and techniques," Handbook of Electronic Security and Digital Forensics, vol. 411, pp. 411–425, 2010

[8] K. Conlan, I. Baggili, and F. Breitinger, "Anti-forensics: Furthering digital forensic science through a new extended, granular taxonomy," Digital investigation, vol. 18, pp.566–575, Elsevier, 2016.

[9] N. Morimoto, A. Lu, W. Bender, and D. Gruhl, "Techniques for data hiding," IBM Systems Journal, vol. 35, pp.313–336, 1996

[10] Y. F. Abdullah, H. Nasereddin, "Proposed Data Hiding Technique-Text under Text," American Academic and Scholarly Research Journal, vol.5, pp.243, 2013

[11] H. Bergel, "Hiding data, forensics, and anti-forensics," Communications of the ACM, vol. 50, pp. 15–20, ACM New York, NY, USA, 2007.

[12] E. Huebner, D. Bern, and C.K. Wee, "Data hiding in the NTFS file system," Digital Investigation, vol. 3, pp. 211–226, 2006.

[13] M. Swanson, L. Stoller, and J. Carter, "Making distributed shared memory simple, yet efficient," Proceedings Third International Workshop on High-Level Parallel Programming Models and Supportive Environments, pp. 2–13, IEEE, 1998.

[14] A. Case, G. Golden, and III Richard, "Memory forensics: The path forward," vol.20, pp.23–33, Elsevier, 2017.

[15] E. C. Hosgor, "Detection and Mitigation of Anti-Forensics Using Forensic Tools," Thesis, Naval Postgraduate School Monterey United States, 2018.

[16] Webopedia.com, "Term Encryption", 2020. [Online]. Available: https://www.webopedia.com/TERM/E/encryption.htm. [Accessed: 30-Nov-2020].

[17] D. Boneh, A. Sahai, and B. Waters, "Functional encryption: Definitions and challenges," Theory of Cryptography Conference, pp. 253–273, Springer, 2011.

[18] M. Mishra, P. Mishra, and M.C. Adhikary, "Digital image data hiding techniques: A comparative study," arXiv, 2014.

[19] M. Sign, and J. Butler, "Shadow walker", 2005. [Online]. Available: https://blackhat.com/presentations/bh-usa-05/bh-us-05-sparks.pdf. [Accessed: 01-Dec-2020].

[20] G. Hoglund, and J. Butler, Rootkits: subverting the Windows kernel, Addison-Wesley Professional, 2006.

[21] S. Garfinkel, A. Shelat, "Remembrance of data passed: A study of disk sanitization practices," IEEE Security & Privacy vol. 1, pp.17–27, IEEE, 2003.

[22] R. Kissel, M. Scholl, S. Skolochenko, and X. Li, "Guidelines for Media Sanitization, Revision 1," National Institute of Standards and Technology (NIST), 2012.

[23] R. Harris, "Arriving at an anti-forensics consensus: Examining how to define and control the anti-forensics problem." digital investigation, vol. 3, pp.44-49, Elsevier, 2006.

[24] A. Jain, G. S. Chhabra, "Anti-forensics techniques: An analytical review," 2014 Seventh International Conference on Contemporary Computing (IC3), pp. 412-418, IEEE, 2014.