Leveraging File Hash Monitoring as a Proactive Early Warning System for Cybersecurity

Joshua A. Reyes
Master of Computer Science, Cybersecurity
Prof. Lisabel Rodríguez
Department of Computer Science
Polytechnic University of Puerto Rico

Abstract *– This study examines the behavior of Hidden Tear, an open-source ransomware, through a controlled attack conducted in a virtual environment. A Windows 11 virtual machine was utilized for the execution of the ransomware following several configuration adjustments and troubleshooting steps. A key aspect of the study involved the use of Autopsy to track and verify file hashes before, during, and after the ransomware attack. The findings indicate that although Hidden Tear alters the file hashes during the encryption process, it restores them to their original state upon decryption, thereby preserving file integrity. These results highlight the efficacy of file hash monitoring as a crucial technique for security analysts to detect and analyze ransomware attacks. The study advocates for further research into the development of automated hashing tools, which could significantly enhance the capabilities for rapid identification and prevention of ransomware threats by facilitating real-time monitoring of changes in file properties.*

Key Terms *– file hashes, file integrity, hidden tear, ransomware*

INTRODUCTION

Ransomware has emerged as one of the most disruptive threats in today's digital landscape. Imagine turning on your computer, ready to start your day, only to find that all your files are locked, and a demand for payment is staring you in the face. This nightmare scenario has become increasingly common, affecting individuals, businesses, and even critical infrastructure. Ransomware is a type of malicious software that encrypts a victim's files, rendering them inaccessible until a ransom is paid to the attacker. Over the past decade, we have witnessed a dramatic surge in ransomware attacks, both in frequency and severity, causing widespread concern and significant financial losses [1]. Ransomware attacks have been relentlessly increasing, creating major challenges for businesses worldwide. In 2022, these incidents rose by 13% compared to 2021, according to Verizon's Data Breach Investigation Report [2]. Statista projects that around 70% of businesses faced at least one ransomware attack in 2022, marking the highest annual rate on record. Cybercriminals are using more aggressive tactics, including data infiltrations and the threat of data leaks, to pressure companies into paying ransoms [2].

This rising tide of ransomware is fueled by the emergence of ransomware as a service (RaaS), a model that allows even those with limited technical skills to launch sophisticated attacks. The consequences are dire: hospitals unable to access patient records, companies crippled by locked data, and even local governments brought to a standstill. It is within this context that the study on Hidden Tear, an open-source ransomware, was conducted. Hidden Tear was initially developed for educational purposes, providing a glimpse into the mechanics of ransomware without malicious intent [3]. However, its availability has made it a popular subject of analysis in the cybersecurity community.

BACKGROUND AND RATIONALE

Given this growing threat, it is crucial to develop effective methods to detect and mitigate ransomware attacks. This study focuses on Hidden Tear, an open-source ransomware initially created for educational purposes. By analyzing Hidden Tear in a controlled environment, the study aims to gain a deeper understanding of ransomware operations without any malicious intent. During the

experiments, it was discovered that while Hidden Tear changes file hashes during encryption, it restores the original hashes after decryption. This finding suggests that monitoring file hash changes could be a powerful tool for detecting ransomware activity.

The insights from this study could lead to the development of automated hashing tools, aiding security analysts in quickly identifying and responding to ransomware attacks. By advancing detection capabilities, the goal is to enhance cybersecurity measures, protect data integrity, and minimize the devastating impact of ransomware on society.

OBJECTIVES

This study set out with a clear objective: to dive into the behavior of Hidden Tear and understand its impact on file integrity. We aimed to execute a controlled ransomware attack within a virtualized environment, meticulously analyze the changes in file hashes, and explore the implications for detecting and mitigating such attacks. The insights gained from this study could potentially inform the development of more robust defenses against ransomware. For this experiment, Hidden Tear, an open-source ransomware initially developed for educational purposes by Utku Sen in 2015, was employed [3]. Hidden Tear is known for its AES-256-CBC encryption method and for being the “father” of numerous ransomware variants that have since been modified by malicious actors for nefarious purposes [4]. This project involved deploying Hidden Tear on a Windows 11 OS system to monitor changes in file hashes of affected files, simulating a real-world ransomware attack scenario. The relevance of this study is accentuated by the significant rise in ransomware attacks in recent years, particularly those targeting critical infrastructures such as healthcare facilities, energy sectors, and defense systems [5]. These sectors have experienced escalated threats since the onset of the COVID-19 pandemic, illustrating the strategic targeting by cybercriminals during periods of global vulnerability.

The reason the file hash was chosen as the baseline for the analysis is due to the industry’s best practice regarding the immutability of file hashes in computer forensics [6]. A hash function, such as MD5 or SHA-256, generates a unique string of characters (hash) from a file's contents, creating a digital fingerprint. Even minor changes in the file result in a completely different hash value. By comparing a file's current hash to a known good hash, one can detect unauthorized changes, ensuring the file's integrity [7]. This process, known as file integrity monitoring, involves periodically rehashing files and comparing the new hashes to stored values. Matching hashes confirm that the files remain unchanged, while differences indicate potential tampering [8].

Hashes can be identified and compared in many ways, however, because malware-related events require more sophisticated techniques of analysis, the open-source tool Autopsy was chosen to verify file hashes before, during, and after the ransomware attack.

As a comprehensive tool, Autopsy offers capabilities like hash filtering, keyword searching, and the organization of data into cases, which supports detailed forensic analysis. This platform is particularly valued for its extensibility, allowing users to tailor their forensic analysis through various add-on modules developed by the community. This makes it an ideal choice for sophisticated malware event analysis, ensuring both credibility and repeatability in forensic investigations [9]. Autopsy allows the user to create cases per project review to ensure that information is segregated and is well organized for computer forensic specialists. By using an industry-accepted tool for the analysis, the credibility of the project can be sustained, since it allows the user to create cases that can be replicated and shared with other experts while ensuring the same result each time.

With computer systems exponentially becoming targets of ransomware attacks, it is crucial to understand the impacts of these attacks,

their behaviors, and the potential for post-incident forensic analysis. Specifically, analyzing file hashes post-incident can reveal useful information about the type of ransomware involved. For instance, when a system is compromised, ransomware typically leaves a "fingerprint" in the file hash, which can be instrumental for recovery processes. This raises a question for security professionals: Should the implementation of an auto-hashing tool be recommended in all computer environments to facilitate recovery by comparing pre-incident and post-incident hashes?

Challenges

One of the main challenges in this study stemmed from the use of Hidden Tear, an active malware that, if mishandled, could significantly damage the Windows operating system. Fortunately, Autopsy, an open-source forensic tool, proved invaluable for forensic analysis. Utilizing free, open-source tools is often essential for students who need to complete projects with limited resources. Hidden Tear was sourced from GitHub, where an unmounted version was available [10]. For this project, it was necessary to use live malware, which was modified and compiled using Microsoft Visual Studio to create an executable file. This executable was then run in a controlled environment provided by Oracle VM VirtualBox on a Windows 11 ISO from Microsoft. To test the encryption capabilities of Hidden Tear, the three file types of Word, PowerPoint, and Excel were encrypted using the AES 256-bit encryption coded in the Hidden Tear source code, showcasing the malware's potent capabilities.

Additionally, the malware had to be run off a USB storage device to ensure that it did not live in the native operating system being targeted. This made the use of the malware even more dangerous, since it meant the storage of a live malware in a storage media that, if handled irresponsibly, could have significant adverse impact on a computer system. While the utilization of live malware in cybersecurity research poses significant risks, it also provides a deep understanding of malware behavior and its mitigation.

PROBLEM STATEMENT

As computer systems begin to be targets of increased ransomware attacks, it is important to understand the impact of these attacks, how they behave, and, in the event of being the victim of one, whether a post-incident analysis of the file hash may provide useful information on the type of ransomware. If a computer system is infected with ransomware, is a fingerprint always left behind in the file hash? Should security professionals recommend the implementation of an auto-hashing tool into the processes of any computer environment, to ensure that proper recovery is done by comparing previously logged hashes with post-incident hashes?

BACKGROUND AND APPLICATION

Hidden Tear, originally intended as an educational tool for security professionals, is an open-source malware that has been infamously co-opted by cybercriminals for malicious purposes. For this study, a version of Hidden Tear was obtained from GitHub, a platform that hosts a variety of software development projects, including potentially dangerous code like Hidden Tear. The version used was unmounted, meaning it was not actively running or installed, which reduces risk when handling or modifying the malware.

Modification and Execution in a Controlled Environment

To safely analyze the malware, modifications were necessary. This was achieved using Microsoft Visual Studio, a comprehensive development environment, to compile the source code into an executable file. This process not only familiarizes students with software development and debugging, but also with the intricacies of malware construction and deployment.

The executable was then tested within a highly controlled environment using Oracle VM

VirtualBox. This virtualization software creates a contained, isolated operating system on a single physical machine, allowing for the safe execution of potentially harmful software without risking the integrity of the host system. The choice of a Windows 11 ISO from Microsoft for the virtual machine ensured compatibility with the latest Windows security features and system architecture.

Forensic Analysis Using Autopsy

Autopsy is a digital forensics platform developed by Brian Carrier, a prominent expert in the field. Carrier, known for his influential book *File System Forensic Analysis*, began developing Autopsy in the early 2000s, with the first version released in 2001 [11]. Since then, it has evolved significantly, incorporating new features and improvements to aid in digital investigations.

Autopsy provides a graphical interface to The Sleuth Kit (TSK), making it easier for users to analyze hard drives and smartphones. Its capabilities include recovering deleted files, extracting web artifacts, and analyzing email messages, among other functions [12]. The tool is widely used by law enforcement, corporate investigators, and digital forensics professionals around the world.

Brian Carrier's motivation for creating Autopsy was to provide a powerful, cost-effective solution for digital forensic investigations. By offering it as an open-source tool, he ensured that high-quality forensic resources would be accessible to everyone, regardless of budget constraints. Today, Autopsy is a cornerstone in digital forensics, valued for its robust capabilities and user-friendly design [13].

Testing Encryption Capabilities

The primary functionality tested in this study was the AES 256-bit encryption used by Hidden Tear. The malware was directed to encrypt three common types of office files: Word documents, PowerPoint presentations, and an Excel spreadsheet. The choice of these file types was strategic, reflecting the common targets of ransomware attacks in real-world scenarios. Demonstrating the encryption on these files underscored the potent capabilities of Hidden Tear and illustrated the severe implications of ransomware attacks.

Educational Implications and Ethical Considerations

This project highlights the educational value of using real-world tools and scenarios to teach cybersecurity concepts. However, it also raises ethical considerations regarding the handling of dangerous software. Students must learn not only the technical skills associated with malware analysis, but also the legal and ethical frameworks that govern cybersecurity practices.

RESULTS

The results obtained from this study aims to aid cybersecurity professionals in understanding the behavior and impact of ransomware on computer systems and can move the discussions of automated file hashing of computer files for backup and restoration in the event of a cybersecurity incident.

Figure 1 shows the USB storage device with the files that will be used to execute the Hidden Tear ransomware attack. It includes a text file named “adobe” that will steal the computer’s information (name, device, OS) and store the decryption key the attacker could provide once payment is received.

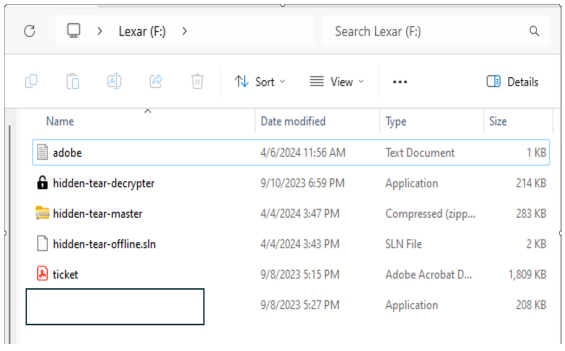


Figure 1
Hidden Tear initial files in USB storage device

With the ransomware ready to be executed, file hash validation was done using the forensic tool Autopsy before releasing the malware to the wild.

This allowed the study to maintain credibility by ensuring that an immutable foundation was created before the launch. Figure 2 demonstrates the steps taken in Autopsy to read each file hash and each file extension to confirm that all three file types chosen were included in the sample.

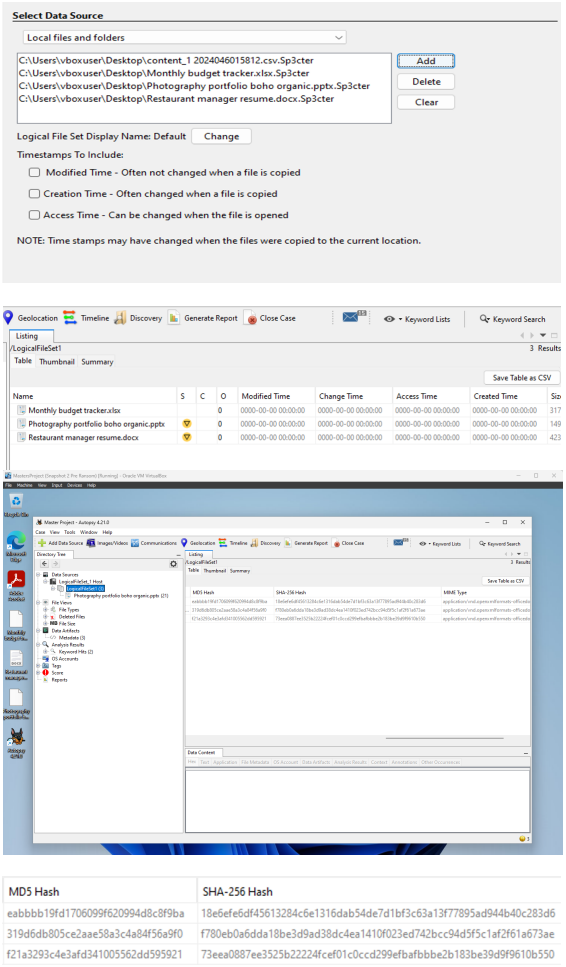


Figure 2
Results from Autopsy SHA-256 for all files prior to encryption

Because the import of the files was successful, we can extract the SHA-256 hash for each of the three files. Once this step was completed, the Hidden Tear malware was run in a virtual environment, and successfully encrypted the files. Figure 3 shows the text file that is created when the malware was successfully executed.

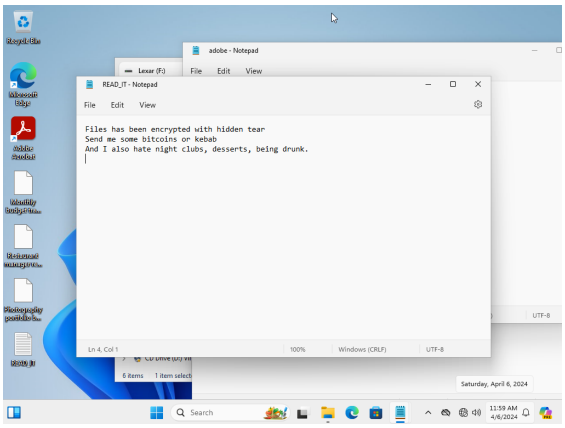


Figure 3
Hidden Tear Encryption notification text file

Figure 4 shows the change of the file extensions from .ppt, .xlsx, and .doc to .Sp3cter. The .Sp3cter extension is a custom extension that is coded in Visual Studio before mounting the malware program and serves as a visual indicator to the attacker that a victim has been encrypted by their custom malware.

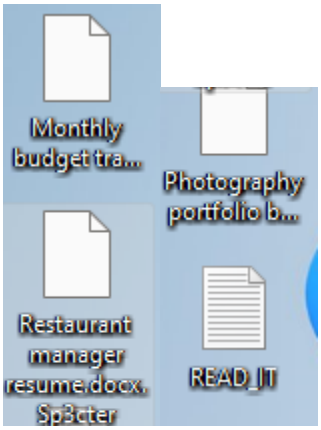
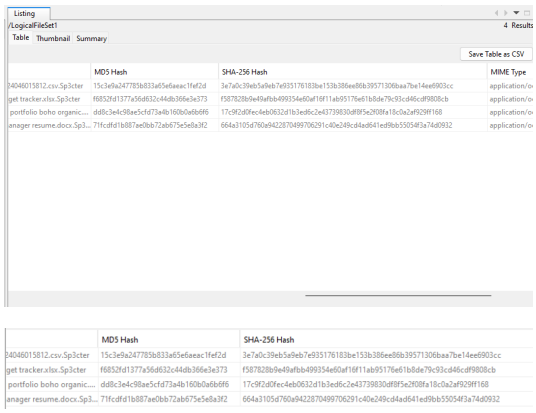


Figure 4
File types encrypted with the .Sp3cter extension from Hidden Tear

Once the files were encrypted, Autopsy was engaged again to import the encrypted files and observe the SHA-256 Hash values. Because there was a change to the file property, a change in the hash value was observed. Figure 5 shows the new hash value for each file.



	MD5 Hash	SHA-256 Hash
24046015812.csv.Sp3cter	15c3e9a247785b833a65e6aeac1fef2d	3e7a0c39eb5a9eb7e935176183be153b386ee86b39571306baa7be14ee6903cc
get tracker.xlsx.Sp3cter	f6852fd1377a56d632c44db366e3e373	f587828b9e49afbb499354e60af16f11ab95176e61b8de79c93cd46cdf9808cb
portfolio boho organic....	dd8c3e4c98ae5cfd73a4b160b0a6b6f6	17c9f2d0fec4eb0632d1b3ed6c2e43739830df8f5e2f08fa18c0a2af929ff168
anager resume.docx.Sp3...	71fcdfd1b887ae0bb72ab675e5e8a3f2	664a3105d760a942287049970629 1c40e249cd4ad641ed9bb55054f3a74d0932

Figure 1

File hash change after malware execution

After confirming that the hash values had changed, the next step was to decrypt the files using the randomly generated decryption key and then studying the files one final time to determine if the hash value returned to its original value or if the malware left any lingering impact on the files. If the hash value returned to its original value that meant that the file itself had not been changed in any way and recovery of the file was 100% successful. Figure 6 shows the decryption key generated by Hidden Tear that was stored in the "adobe" text file.

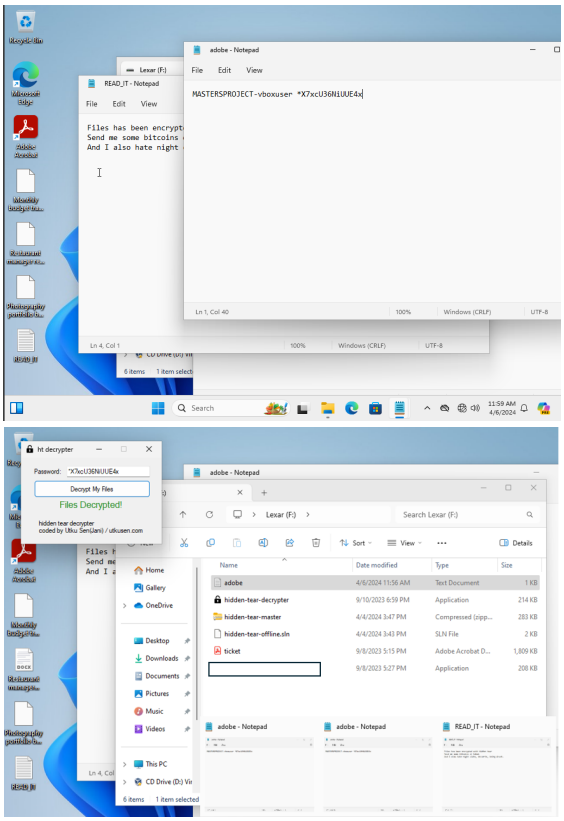
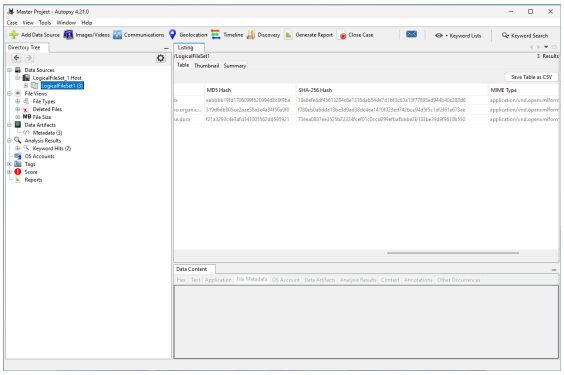


Figure 2

Hidden Tear decryption key successfully executed

Figure 7 shows the hash value post decryption and confirmed that the file hashes returned to their original value.



MD5 Hash	SHA-256 Hash
eabbbb19fd1706099f620994d8c8f9ba	18e6efe6df45613284c6e1316dab54de7d1bf3c63a13f77895ad944b40c283d6
319d6db805ce2aae58a3c4a84f56a9f0	f780eb0a6dda18be3d9ad38dc4ea1410f023ed742bcc94d5f5c1af2f61a673ae
f21a3293c4e3afd341005562dd595921	73eea0887ee3525b22224fcef01c0ccd299efbafbbbe2b183be39d9f9610b550

Figure 3

File Hash Values Post Decryption

CONCLUSION

This study of the Hidden Tear ransomware has illuminated critical aspects of ransomware behavior, particularly regarding its impact on file integrity. By executing Hidden Tear within a controlled virtual environment and meticulously analyzing file hash changes, we established that while the ransomware alters file hashes during the encryption process, these hashes revert to their original state upon decryption. This crucial finding emphasizes the importance of file hash monitoring as an effective method for detecting ransomware activity, providing a potential early warning system for security analysts. The ability to monitor file hashes in real time can serve as a frontline defense, alerting to unauthorized changes indicative of ransomware operations.

The employment of Autopsy was crucial in validating our approach. Autopsy not only facilitated the verification of file hashes, but also demonstrated its practical application in real-world cybersecurity scenarios. The capability to verify and compare file hashes before and after ransomware attacks emphasizes the critical role that automated hashing tools can play in cybersecurity protocols. These tools can significantly enhance the speed and accuracy of ransomware detection, improving response times and mitigating potential damage. Incorporating such technologies into standard cybersecurity measures can fortify defenses against increasingly sophisticated ransomware attacks.

Additionally, this study highlights the educational value of engaging with real-world malware in a controlled environment. By providing hands-on experience with ransomware mechanics, future cybersecurity professionals can gain the knowledge and skills essential to combat these pervasive threats effectively. However, it also brings to the forefront the necessity of adhering to stringent ethical guidelines and legal frameworks when handling malicious software. Ensuring that these practices are followed is paramount to maintaining the integrity of cybersecurity research and practice.

The insights garnered from this research advocate for the development of advanced automated hashing tools. These tools are not only crucial for the rapid identification of ransomware attacks but also play a significant role in recovery efforts. Ensuring the integrity of restored files post-attack is essential for maintaining data reliability and trust. As ransomware continues to evolve and pose significant threats across various sectors, advancements in cybersecurity measures are imperative. Automated hashing tools could become indispensable in verifying the integrity of files, thus supporting comprehensive recovery strategies.

Future research should focus on refining these tools and exploring their integration into broader cybersecurity systems. By enhancing our detection and prevention capabilities, we can bolster our defenses against the ever-growing threat of ransomware. Integrating automated hashing tools into existing cybersecurity frameworks can provide a robust mechanism for real-time monitoring and response, ultimately safeguarding data and infrastructure from potential breaches. As we continue to advance in the field of cybersecurity, developing such proactive measures is crucial in staying ahead of malicious actors and ensuring the resilience of our digital environments. Should all files be monitored in this manner? No. However, having an immutable file that is known throughout an organization that is created with the intention to serve as an early warning system could be of great aid in the event of a malware attack.

REFERENCES

[1] R. Richardson and M. M. North, “Ransomware: Evolution, mitigation and prevention,” DigitalCommons@Kennesaw State University, Jan. 1, 2017. Available: https://core.ac.uk/download/pdf/231830935.pdf

[2] Fortinet, “50 ransomware statistics and latest ransomware trends for 2023.” Accessed May 28, 2024. Available: https://www.fortinet.com/resources/cyberglossary/ransomware-statistics

[3] WatchGuard, “Ransomware – Hidden Tear.” Accessed May 28, 2024. Available: https://www.watchguard.com/wgrd-ransomware/hidden-tear

[4] J. van der Wiel, “Hidden tear and its spin offs,” SecureList by Kapersky, Feb. 2, 2016. Available: https://securelist.com/hidden-tear-and-its-spin-offs/73565/

[5] Cybersecurity and Infrastructure Security Agency, “2021 trends show increased globalized threat of ransomware,” Feb. 10, 2022. Available: https://www.cisa.gov/news-events/cybersecurity-advisories/aa22-040a

[6] P. Callaghan, “Why hash values are crucial in evidence collection & digital forensics,” Pagefreezer, Accessed May 28, 2024. Available: https://blog.pagefreezer.com/importance-hash-values-evidence-collection-digital-forensics

[7] M. Bishop, *Introduction to Computer Security*, Addison-Wesley, Boston, MA, USA, 2005.

[8] J. Black, P. Rogaway, and T. Shrimpton, "Black-Box Analysis of the Block-Cipher-Based Hash-Function Constructions from PGV," in *Proceedings of the 26th Annual International Conference on Advances in Cryptology* (CRYPTO'02), Springer-Verlag, Berlin, Heidelberg, 2002, pp. 320-335.

[9] Autopsy Digital Forensics, “Is Autopsy® for you?” Accessed May 28, 2024. Available: https://www.autopsy.com/about/use-cases/

[10] GitHub, “Hidden Tear.” Accessed May 28, 2024. Available: https://github.com/goliate/hidden-tear

[11] B. Carrier, *File System Forensic Analysis*, Addison-Wesley Professional, 2005.

[12] Sleuth Kit, "Autopsy." Accessed May 28, 2024. Available: https://www.sleuthkit.org/autopsy/

[13] Sleuth Kit, "About." Accessed May 28, 2024. Available: https://www.sleuthkit.org/about.php